Energy, Mines and Resources Canada
Earth Physics Branch

Énergie, Mines et Ressources Canada
Direction de la physique du globe

1 Observatory Crescent
Ottawa Canada
K1A 0Y3

1 Place de l'Observatoire
Ottawa Canada
K1A 0Y3

Geodynamics Service of Canada

Service de la géodynamique du Canada

THREE-DIMENSIONAL MODELLING OF CRUSTAL DEFORMATION ALONG A FAULT

W. Scott Dunbar
5370 Keith Street
Burnaby, British Columbia

Earth Physics Branch Open File Number 85-3 Volume 2
Dossier public de la Direction de la Physique du Globe No. 85-3 Volume 2
Ottawa, Canada

NOT FOR REPRODUCTION

Department of Energy, Mines & Resources Canada
Earth Physics Branch
Division of Gravity, Geothermics and Geodynamics

REPRODUCTION INTERDITE

Ministere de l,Energie, des Mines et des Ressources du Canada
Direction de la Physique du Globe
Division de la gravite, geothermie et geodynamique

pp. 128
Price / Prix: $24.60

Preface

This open file comprises the programs and documentation submitted by Dr. W. Scott Dunbar in completion of contract work (DSS File #06SB.23227-4-0843) entitled: "Three-dimensional Modelling of Crustal Deformation Along a Fault".

The two main programs provided are:

1. DIS3D

   This program computes the elastic fields due to rectangular dislocation planes in an isotropic three-dimensional elastic halfspace with arbitrary elastic parameters. Combinations of several dislocation segments can be used to model complex fault systems. Common parameters calculated for given fault offsets include stress, strain, displacements, and angle and line-length changes.

2. SID3D

   Using linear least-squares methods, this program computes the slip on three-dimensional segmented dislocation models in an elastic halfspace based on measured changes in horizontal and vertical positions on the earth's surface. Several dislocation planes can be defined to derive complex slip distributions.

Both programs currently reside with the Geodynamics Section at the Pacific Geoscience Centre where the contract originated. At P.G.C., the programs have been compiled under a Sperry-Univac Fortran processor and tested with simple dislocation models. Inquiries concerning access to these programs should be directed to the Geodynamics Section at the Pacific Geoscience Centre, Sidney, B.C.

H. Dragert
Pacific Geoscience Centre
September, 1984

Résumé

Le présent dossier public comprend les programmes et la documentation présentés par M. W. Scott Dunbar en exécution du marché (dossier MAS no 05SB.23227-4-0843) intitulé "Three-dimensional Modelling of Crustal Deformation Along a Fault".

Les deux principaux programmes sont:

1. DIS3D

   Le programme calcule les champs élastiques dus aux plans de dislocation rectangulaires dans un demi-espace élastique tridimensionnel isotrope avec des paramètres élastiques arbitraires. On peut utiliser des combinaisons de plusieurs segments de dislocation pour la modélisation de systèmes de failles complexes. Les paramètres communs calculés pour les décalages des failles comprennent la contrainte, la déformation, les déplacements et les changements d'angle et de longueur.

2. SID3D

   À l'aide de méthodes linéaires des moindres carrés, le programme calcule le glissement pour des modèles tridimensionnels de dislocation segmentée dans un demi-espace élastique, en se fondant sur des mesures des variations dans les positions horizontales et verticales de la surface de la Terre. On peut définir plusieurs plans de dislocation afin de calculer des distributions complexes de glissement.

Les deux programmes se trouvent actuellement à la Section de la géodynamique du Centre géoscientifique du Pacifique où le marché avait été imparti. On y a compilé les programmes dans un processeur Fortran de Sperry-Univac et on les a mis à l'essai avec des modèles simples de dislocation. Adresser les demandes concernant l'accès aux programmes à la Section de la géodynamique du Centre géoscientifique du Pacifique à Sidney (C.-B.).

H. Dragert
Centre géoscientifique du Pacifique
Septembre 1984

SID3D

A PROGRAM TO ESTIMATE SLIP DISTRIBUTIONS

ON THREE DIMENSIONAL FAULT MODELS

W. SCOTT DUNBAR

APRIL 1984

S1D3D

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

SECTION 1 - THEORETICAL BACKGROUND

1.1 Introduction

Given changes in geodetic data on the earth's surface, SID3D employs linear least squares methods to compute slip distributions on three dimensional segmented dislocation models in an elastic halfspace. Several dislocation planes, each with a constant strike and/or dip-slip displacement discontinuity, can be used to model complex slip distributions.

This manual describes the basic theory and methods used in the program. A user's manual is also provided together with some examples.

1.2 Representation of Geodetic Data Changes by Dislocation Models

In order to represent changes in geodetic data on the earth's surface by means of dislocation models, a means of computing the displacements at the earth'surface due to such models is required. An equation for the displacement due to an individual dislocation segment may be derived by means of the Volterra integral. The geometry of the rectangular dislocation segment used in SID3D is shown in Figure 1.1.

For constant strike-slip, SS, on a rectangular dislocation segment, Mansinha and Smylie (1971) derived the following integral for the displacement component u_i:

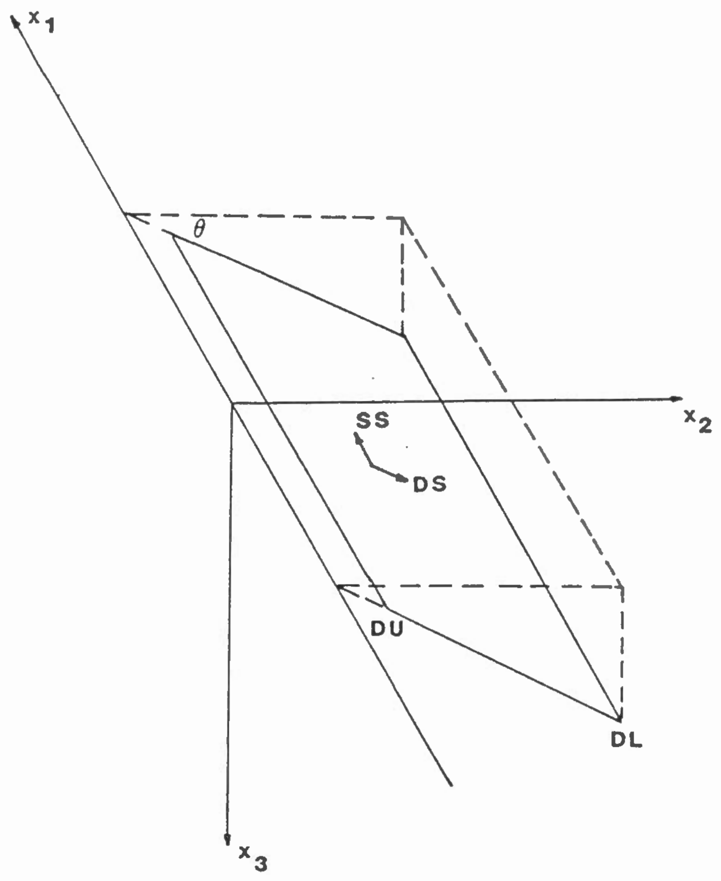
$$u_i = G \cdot SS \int_{DU}^{DL} \int_{-H}^{H} \left[(U_i^{1,2} + U_i^{2,1}) \sin\theta - (U_i^{1,3} + U_i^{3,1}) \cos\theta \right] dc_1 dc_0 \qquad (1.1)$$

For constant dip-slip, DS, on a rectangular segment, the following integral for the displacement component u_i was derived:

$$u_i = G \cdot DS \int_{DU}^{DL} \int_{-H}^{H} \left[(U_i^{2,2} - U_i^{3,3}) \sin 2\theta - (U_i^{2,3} + U_i^{3,2}) \cos 2\theta \right] dc_1 dc_0 \qquad (1.2)$$

The variables in these equations are defined below:

G	- shear modulus of medium
H	- half length of segment
DU	- upper depth of segment, measured in the direction c_0



RECTANGULAR DISLOCATION SEGMENT GEOMETRY FIGURE 1.1

DL - lower depth of segment, measured in the direction c_0

θ - dip of segment, measured positive as shown in Figure 1.1

$U_i^j(x,c)$ - the displacement in the x_i direction at $x = (x_1, x_2, x_3)$ due to a point force in the x_j direction at $c = (c_1, c_2, c_3)$. $U_i^{j,k}$ denotes the derivative of U_i^j with respect to c_k.

Mansinha and Smylie (1971) integrated Equations 1.1 and 1.2 to obtain analytical expressions for the displacements due to a rectangular dislocation segement in an elastic halfspace of Poisson's ratio 0.25. Converse (1973) derived similar expressions for a halfspace of Poisson's ratio in the range $0 < \nu < 0.5$. The latter expressions were rewritten for the case $x_3 = 0$ and are presented in Appendix A.1. Poisson's ratio is assumed to be 0.25 in SID3D.

A key element in the estimation of slip distributions on faults is an expression for the change in a geodetic datum due to a unit component of slip. Such expressions are sometimes called influence coefficients or partial derivatives. In the following sub-sections, the influence coefficients for elevation, angle, and line length changes are derived.

1.2.1 Elevation Changes

The influence coefficients for an elevation datum change are given by the expressions for vertical displacement due to strike or dip-slip with SS = DS = 1.

1.2.2 Angle Changes

Consider the geometry shown in Figure 1.2a. The original directions to stations B and C from station A are denoted ϕ_B and ϕ_C respectively. The final directions are denoted ϕ_B' and ϕ_C'. The clockwise angle change $\Delta\alpha$ is then given by

$$\Delta\alpha = \phi_C' - \phi_B' - (\phi_C - \phi_B) \tag{1.3}$$

If the coordinates of station j are denoted x_1^j and x_2^j ($x_3 = 0$), the original directions are given by

$$\phi_B = \tan^{-1}\left[\frac{\Delta x_2^B}{\Delta x_1^C}\right] \qquad \phi_C = \tan^{-1}\left[\frac{\Delta x_2^C}{\Delta x_1^C}\right]$$

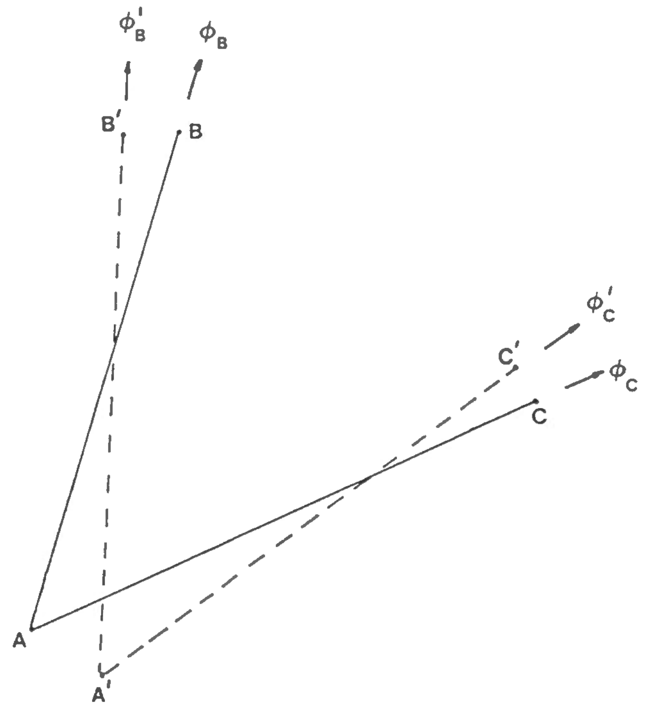
where $\Delta x_1^j = x_1^j - x_1^A$ and $\Delta x_2^j = x_2^j - x_2^A$. If the horizontal displacements at station j due to unit strike or dip-slip are denoted u_1^j and u_2^j the final directions are given by

$$\phi_B' = \tan^{-1}\left[\frac{\Delta x_2^B + \Delta u_2^B}{\Delta x_1^B + \Delta u_1^B}\right] \qquad \phi_C' = \tan^{-1}\left[\frac{\Delta x_2^C + \Delta u_2^C}{\Delta x_1^C + \Delta u_1^C}\right]$$

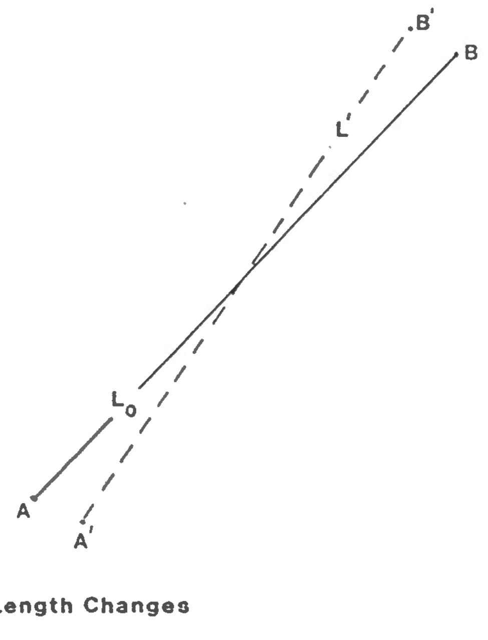
where $\Delta u_1^j = u_1^j - u_1^A$ and $\Delta u_2^j = u_2^j - u_2^A$. Substituting these expressions into Equation 1.3 and expanding in a first order Taylor series about zero displacement gives

$$\Delta\alpha = \frac{\Delta x_1^C \cdot \Delta u_2^C - \Delta x_2^C \cdot \Delta u_1^C}{D_{AC}^2} - \frac{(\Delta x_1^B \cdot \Delta u_2^B - \Delta x_2^B \cdot \Delta u_1^B)}{D_{AB}^2} \tag{1.4}$$

where $D_{Aj}^2 = (\Delta x_1^j)^2 + (\Delta x_2^j)^2$.



a. Angle Changes



b. Line Length Changes

GEOMETRY FOR ANGLE AND LINE LENGTH CHANGES **FIGURE 1.2**

1.2.3 Line Length Changes

Consider the geometry shown in Figure 1.2b. The original line length is denoted L_0. The final line length is denoted L'. Using the same notation as in Section 1.2.2, the line length change $\Delta L = L' - L_0$ is given by

$$\Delta L = \left[(\Delta x_1^B + \Delta u_1^B)^2 + (\Delta x_2^B + \Delta u_2^B)^2\right]^{\frac{1}{2}} - L_0$$

Expanding in a first order Taylor series about zero displacement gives

$$\Delta L = \frac{\Delta x_1^B \cdot \Delta u_1^B + \Delta x_2^B \cdot \Delta u_2^B}{D_{AB}} \qquad (1.5)$$

1.3 The Linear Least Squares Model

From the results in the preceding section, it is possible to represent a geodetic datum change, d_i, in terms of the strike-slip, SS_j, and/or dip-slip, DS_j, on n/2 dislocation segments as follows:

$$d_i = \sum_{j=1}^{n/2} F_{ij} \cdot SS_j + \sum_{j=1}^{n/2} G_{ij} \cdot DS_j + e_i \qquad (1.6)$$

where F_{ij} and G_{ij} are influence coefficients depending on the type of data d_i. F_{ij} is written with displacements due to unit strike-slip; G_{ij} is written with displacements due to unit dip-slip. (Note that each segment need not have both components of slip.) The e_i is an error term introduced to account for errors in the model. It is this error that is minimized in the least squares procedure.

Writing Equation 1.6 for m data, $1 \leq i \leq m$, the following matrix equation results:

$$d = Ax + e \qquad (1.7)$$

where $x = (SS_1, DS_1, SS_2, DS_2, \ldots, SS_{n/2}, DS_{n/2})^t$, $A = (F_{ij}, G_{ij})$ is a m by n matrix known as the model matrix, $d = (d_i)$, and $e = (e_i)$. The superscript t denotes the transpose.

Two types of least squares model may be defined. They are:

1) Overdetermined - where the number of data, m, exceeds the number of desired slip parameters, n.

2) Underdetermined - where the number of slip parameters, n, exceeds the number of data, m.

1.3.1 Maximum Likelihood Estimation

The probability that a given set of data is due to the model given by Equation 1.7 is assumed to be given by the multivariate Gaussian distribution (Lindgren, 1968):

$$f(d) = \frac{1}{(2\pi)^m |T|^{\frac{1}{2}}} \exp\left[-\tfrac{1}{2}(d-Ax)^t T^{-1}(d-Ax)\right] \qquad (1.8)$$

T is the covariance of the data, $E(dd^t)$, where E denotes the expectation operator. The superscript t denotes the transpose. Equation 1.8 is sometimes known as a likelihood function. To maximize it; i.e., to maximize the probability that d is due to the model, the quadratic form

$$Q = \tfrac{1}{2}(d-Ax)^t T^{-1}(d-Ax)$$

is minimized. The result of the minimization is

$$\hat{x} = (A^t T^{-1} A)^{-1} A^t T^{-1} d \qquad (1.9)$$

which is known as the maximum likelihood estimate of x. It may be seen that minimizing Q is equivalent to minimizing the weighted squares of the errors $e = d-Ax$.

1.3.2 Constrained Estimation

Often it is desireable to place constraints on the estimate of x. Mathematically, this problem is posed in the following manner:

$$\text{Minimize} \quad \tfrac{1}{2}(d-Ax)^{t}T^{-1}(d-Ax)$$

$$\text{subject to} \quad Bx = c \qquad (1.10a)$$

$$\text{or} \quad Bx \geq c \qquad (1.10b)$$

where B is a pxn matrix of rank p (i.e., a set of consistent and non-redundant constraints) and c is a vector of length p. The main purpose of such constraints is to place bounds on an otherwise unbounded solution. However, a more important application is in the realm of hypothesis testing to investigate the nature of possible solutions allowed by the data.

1.4 Data Covariance Matrices

The calculation of the m by m data covariance matrix $T = E(dd^t)$, required for both unconstrained and constrained estimation is somewhat involved for geodetic data changes. The reason for this is that the data changes depend on two independent surveys. In the case of levelling and trilateration, two measured numbers are required to compute a datum change; in the case of triangulation, four measured directions are required to compute an angle change. Thus the techniques of error propagation must be used to compute T (Davis et al, 1981).

1.4.1 Elevation Changes

The basic idea of error propagation may be demonstrated by means of the derivation of T for elevation changes. Consider N levelling stations. At the ith station, the elevation change Δz_i is given by

$$\Delta z_i = z_i' - z_i \qquad 1 \leq i \leq N$$

where z_i' is the elevation determined by adjustment of the more recent survey and z_i is the elevation determined by adjustment of the previous survey. For N levelling stations, the elevation changes may be written as

$$\begin{bmatrix} \Delta z_1 \\ \Delta z_2 \\ \cdot \\ \cdot \\ \Delta z_N \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 & & & \\ 0 & 0 & 1 & -1 & & & \\ & & & & \cdots & & \\ & & & & & 0 & 0 \\ & & & & & 1 & -1 \end{bmatrix} \begin{bmatrix} z_1' \\ z_1 \\ z_2' \\ z_2 \\ \cdot \\ \cdot \\ z_N' \\ z_N \end{bmatrix}$$

or d = Jz. Thus

$$T = E(dd^t) = JE(zz^t)J^t$$

$$= \begin{bmatrix} v_1'+v_1 & & & & 0 \\ & v_2'+v_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ 0 & & & & \cdot \; v_N'+v_N \end{bmatrix}$$

where v_i and v_i' are the variances of z_i and z_i' respectively. These variances are usually determined by distributing the closure error uniformly over the network of N levelling stations. The adjusted elevations of each survey are assumed to be statistically independent. Thus $E(zz^t)$ is a diagonal matrix i.e., zero covariance between all elevations.

1.4.2 Line Length Changes

Consider N observed line length changes observed at a given station. For each observed station the line length change is given by

$$\Delta L_i = L_i' - L_i \qquad 1 \leq i \leq N$$

where L_i' is the line length observed during the more recent survey and L_i is the line length observed during the previous survey. The derivation of T for line length changes is therefore exactly the same as that for elevation changes. The result is

$$T = 2 \begin{bmatrix} v_1 & & & & 0 \\ & v_2 & & & \\ & & \cdot & & \\ 0 & & & \cdot & \\ & & & & \cdot \; v_N \end{bmatrix}$$

where v_i is the variance of L_i. The variances of each observed line length are proportional to the distance over which the observation is made. Thus, unless there are other sources of error, the variances of L_i and L_i' are likely to be the same i.e., $v_i' = v_i$. Each observed line length is assumed to be statistically independent.

1.4.3 Angle Changes

Consider N observed directions at a given station. This results in N-1 independent angles which may be computed. Each (clockwise) angle change is given by

$$\Delta\alpha = \phi_{i+1}' - \phi_i' - (\phi_{i+1} - \phi_i) \qquad 1 \leq i \leq N-1$$

where ϕ_i' is the observed direction to station i during the more recent survey and ϕ_i is the observed direction to station i during the previous survey. For N observed directions, the clockwise angle changes may be written as

$$\begin{bmatrix} \Delta\alpha_1 \\ \Delta\alpha_2 \\ \cdot \\ \cdot \\ \Delta\alpha_{N-1} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & -1 & 0 & 0 & & & & \\ 0 & 0 & -1 & 1 & 1 & -1 & \cdot & & & \\ & & & & & \cdots\cdots & & & & \\ & & & & & & -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \phi_1' \\ \phi_1 \\ \phi_2' \\ \phi_2 \\ \cdot \\ \cdot \\ \phi_N' \\ \phi_N \end{bmatrix}$$

or $d = J\phi$. Thus

$$T = E(dd^t) = JE(\phi\phi^t)J^t$$

$$\begin{bmatrix} D_1 & E_1 & & & \\ E_1 & D_2 & & 0 & \\ & & \ddots & & \\ & 0 & & D_{N-2} & E_{N-2} \\ & & & E_{N-2} & D_{N-1} \end{bmatrix}$$

where $D_i = v_i + v_{i+1} + v_i' + v_{i+1}'$ and $E_i = -v_{i+1} - v_{i+1}'$. The quantities v_i and v_i' are the variances of ϕ_i and ϕ_i' respectively. These variances may be obtained by various methods. One method is to successively turn angles and note the error in re-observing a reference station. Another method distributes the error in triangle closures over the directions associated with each triangle in the network. Thus, in general, each direction would have a different variance. Each observed direction is assumed to be statistically independent.

1.5 Algorithms for Maximum Likelihood Estimation

In this section the numerical algorithms used to solve the unconstrained least squares problem discussed in Section 1.3.1 are outlined.

The first step in the solution of the unconstrained problem is to reduce it to a canonical form. Since the data covariance matrix is symmetric and positive definite, it may be written

$$T = R^t R$$

where R is an upper triangular matrix. The inverse of T is therefore

$$T^{-1} = R^{-1} R^{-t}$$

Substituting into Equation 1.9, one obtains

$$\begin{aligned} \hat{x} &= (A^t R^{-1} R^{-t} A)^{-1} A^t R^{-1} R^{-t} d \\ &= (Z^t Z)^{-1} Z^t w \end{aligned} \qquad (1.11)$$

where $Z = R^{-t}A$ and $w = R^{-t}d$. This is the desired canonical form.

Note that the units of A are (data units/slip units) and the units of R are (data units). Thus, the scaling represented by $R^{-t}A$ and $R^{-t}d$ renders the problem dimensionless and allows the simultaneous inversion of multiple types of geodetic data changes. The minimized quantity is then the dimensionless quadratic form $e^t T^{-1} e = (w-Zx)^t (w-Zx)$ where $e = d-Ax$.

1.5.1 Calculation of Z and w

If angle changes are used in the estimation, the data covariance matrix is symmetric, positive definite, and tridiagonal. Such matrices are extremely easy to factorize into the product R^tR where R would be upper bidiagonal. If the vectors d and e represent the diagonal and super-diagonal of T respectively, the factorization algorithm is

```
$d_1 := \mathrm{sqrt}(d_1)$
For i = 2 to m
  $e_{i-1} := e_{i-1}/d_{i-1}$
  $d_i := \mathrm{sqrt}(d_i - e_{i-1} \cdot e_{i-1})$
end i
```

The vectors d and e now contain the diagonal and super-diagonal of R, respectively.

The two matrix equations $R^tZ = A$ and $R^tw = d$ may be written $R^ty = b$, where y represents a column of Z or the vector w and b represents a column of A or the vector d. The solution algorithm for $R^ty = b$ is also extremely easy once the above factorization is performed:

```
$b_1 := b_1/d_1$
If n = 1
  then
    quit
  else
    For i = 2 to m
      $b_i := (b_i - e_{i-1} \cdot b_{i-1})/d_i$
    end i
```

The solution now resides in the vector b. Thus the columns of Z and the vector w can overwrite the columns of A and the vector d, respectively.

1.5.2 Singular Value Decomposition

A decomposition of the matrix Z, useful in subsequent analysis of the unconstrained least squares solution, is the singular value decomposition (SVD)

$$Z = USV^t \tag{1.12}$$

where U is a m by m orthogonal matrix ($U^tU = I_m$), V is a n by n orthogonal matrix ($V^tV = I_n$), and S is a m by n diagonal matrix whose elements are the singular values, S_i, of Z. The successive diagonal entries of S can be arranged to be non-increasing. The algorithm used to compute the SVD is described in Golub and Reinsch (1971). The program used to compute the SVD was adapted from that in Forsythe et. al. (1977).

1.5.3 Rank of Z

The rank, r, of Z is an important concept in the solution and analysis of the unconstrained problem. The rank of Z is defined as the number of linearly independent columns of Z. (Thus the rank of A equals the rank of Z.) Since an orthogonal transformation such as the SVD does not affect the rank, it may be seen that the rank of Z is the rank of S in its SVD. Thus a practical definition of rank is the number of non-zero singular values in S. However, owing to finite precision in any computer, it may be difficult to distinguish a small singular value from zero. Therefore, the effective rank is defined as the number of singular values greater than some prescribed tolerance which reflects the accuracy of the data.

Generally, for the inverse problems discussed herein, the rank of Z for an overdetermined problem is n and the rank of Z for an underdetermined problem is m.

1.5.4 Solution Algorithm

Substituting Equation 1.12 into 1.11 gives

$$\begin{aligned}\hat{x} &= (VSU^tUSV^t)^{-1}VSU^tw \\ &= V(S^\dagger)^2V^tVSU^tw \\ &= VS^\dagger U^tw = Z^\dagger w\end{aligned}$$

where $Z^\dagger$ is known as the generalized inverse of Z. The generalized inverse of S, $S^\dagger$, is a n by m diagonal matrix whose elements are

$$\begin{aligned}S^\dagger_{ii} &= 1/s_i \qquad & i \le r \\ &= 0 & i \ge r\end{aligned}$$

The solution algorithm proceeds as follows:

```
For $i = 1$ to $r$
For $j = 1$ to $m$
  $g_i := U_{ji}w_j$
end $j$
end $i$
For $i = 1$ to $r$
  $g_i := g_i/s_i$
end $i$
For $i = 1$ to $n$
For $j = 1$ to $r$
  $x_i := V_{ij}g_j$
end $j$
end $i$
```

1.6 Analysis of Unconstrained Least Squares Solutions

Using the SVD, numerous quantities may be computed which are useful in the interpretation of an unconstrained least squares solution. These quantities and the algorithms used to compute them are described below.

1.6.1 Errors

Given the estimate $\hat{x}$, estimates of the normalized data and the errors may be computed according to

$$\hat{w} = Z\hat{x} = USV^t\hat{x}$$

$$e = w - \hat{w}$$

The errors $e = (e_i)$, $1 \leq i \leq m$ are one representation of how well the model fits the data.

1.6.2 Correlation Coefficient

The total sum of squares, SST, is

$$SST = w^t w$$

The sum of squares of the errors e, SSE, is

$$SSE = e^t e$$

The difference SSR = SST - SSE represents the portion of SST attributed to having fitted the model to the data. It is often called the reduction in the sum of squares. The ratio

$$R^2 = SSR/SST$$

is known as a correlation coefficient. R^2 is a number between 0 and 1. The closer R^2 is to 1, the better the fit of the model to the data.

1.6.3 Variance of $\hat{x}$

Using Equation 1.11, the covariance matrix of the estimate $\hat{x}$ is given by

$$C = E(\hat{x}\hat{x}^t) = E\left|(Z^tZ)^{-1}Z^tww^tZ(Z^tZ)^{-1}\right|$$

$$= (Z^tZ)^{-1}Z^tE(ww^t)Z(Z^tZ)^{-1}$$

From Section 1.5

$$E(ww^t) = E(R^{-t}dd^tR^{-1})$$

$$= R^{-t}E(dd^t)R^{-1}$$

$$= R^{-t}R^tRR^{-1} = I$$

so that

$$C = (Z^tZ)^{-1}$$

Given the SVD $Z = USV^t$, an element C_{ij} of C is computed according to

$$C_{ij} = \sum_{k=1}^{r} V_{ik}V_{jk}/S_{kk}^2$$

where r is the effective rank of Z. The square root of the ith diagonal element of C is the standard deviation, σ_i, of $\hat{x}$. Assuming a normal distribution of $\hat{x}$, the following probability statement may be made:

$$P(\hat{x}_i-\sigma_i \le x_i \le \hat{x}_i+\sigma_i) = 0.6827$$

where P denotes the probability (Searle, 1971, pp. 107-108).

1.6.4 Resolution Matrix

The resolution matrix, R, is defined in terms of the SVD of Z according to

$$\begin{aligned}\hat{x} &= Z^{\dagger}w = VS^{\dagger}U^{t}w \\ &= VS^{\dagger}U^{t}USV^{t}x \\ &= V_rV_r^{t}x = Rx\end{aligned}$$

where V_r is the matrix composed of the columns of V corresponding to the non-zero singular values of Z. Two cases may be identified:

1) $\text{Rank}(Z) = n \qquad R = I_n$
2) $\text{Rank}(Z) \le n \qquad R = V_rV_r^{t}$

The resolution matrix is a 'window' or filter through which the true solution is seen. If $R \neq I_n$, the solution is a weighted sum of the true solution (Jackson, 1972).

1.6.5 Information Density Matrix

The information density matrix, H, is defined in terms of the SVD of Z according to

$$\begin{aligned}\hat{w} &= Z\hat{x} = ZVS^{\dagger}U^{t}w \\ &= USV^{t}VS^{\dagger}U^{t}w \\ &= U_rU_r^{t}w = Hw\end{aligned}$$

where U_r is the matrix composed of the columns of U corresponding to the non-zero singular values of Z. Two cases may be identified:

1) $\mathrm{Rank}(Z) = r < m \qquad H = U_r U_r^t$

2) $\mathrm{Rank}(Z) = m \qquad H = I_m$

The ith diagonal element of H is a number between 0 and 1 which represents the importance of the ith datum to the model (Wiggins, 1972).

1.7 Algorithm for Equality Constrained Estimation

The equality constrained problem defined by Equation 1.10a is solved by means of an algorithm described in Lawson and Hanson (1974, Chapter 21). The algorithm, as implemented in SID3D is described below. It is assumed that the matrix A and the data vector d have been normalized as described in Section 1.5 so that

$$Z = R^{-t}A \qquad w = R^{-t}d$$

where the covariance matrix of the data $T = R^tR$.

The matrices Z and B and vectors w and c are assumed to be partitioned as shown below:

$$Z = (Z_p, Z_{n-p}, w)$$
$$B = (B_p, B_{n-p}, c)$$

where the subscripts denote the number of columns in the sub-matrix. The constraint equation $Bx = c$ can be solved for the first p elements of x, x_p:

$$x_p = B_p^{-1}(c - B_{n-p}x_{n-p})$$

Substitution of this expression into w-Zx, the dimensionless error, gives

$$\begin{aligned} w - Zx &= w - Z_pB_p^{-1}(c - B_{n-p}x_{n-p}) - Z_{n-p}x_{n-p} \\ &= w - Z_pB_p^{-1}c - (Z_{n-p} - Z_pB_p^{-1}B_{n-p})x_{n-p} \\ &= \bar{w} - \bar{Z}_{n-p}x_{n-p} \end{aligned}$$

which is to be minimized to give x_{n-p}. Given x_{n-p}, x_p may be computed from

$x_p = B_p^{-1}(c - B_{n-p}x_{n-p})$.

The above is effected as follows:

1) Using Householder transformations, Q_p, the matrix B_p is triangularized. Q_p are also applied to B_{n-p} and c to give

$$Q_p(B_p, B_{n-p}, c) = (R_p, \bar{B}_{n-p}, \bar{c})$$

where R_p is a p by p upper triangular matrix.

2) Solve the triangular system $\bar{Z}_p R_p = Z_p$ for $\bar{Z}_p$.

3) Compute $\bar{Z}_{n-p} = Z_{n-p} - \bar{Z}_p \bar{B}_{n-p}$ and $\bar{w} = w - \bar{Z}_p \bar{c}$.

4) Compute Householder transformations, Q_{n-p}, to triangularize $\bar{Z}_{n-p}$ and also apply these transformations to $\bar{w}$:

$$Q_{n-p}(\bar{Z}_{n-p}, \bar{w}) = \begin{array}{|cc|l} R_{n-p} & \hat{w} & n-p \\ 0 & v & p+m-n \end{array}$$

where $v^t v$ = SSC, the sum of squares of the errors w-Zx.

5) Solve the triangular system

$$\begin{vmatrix} R_p & \bar{B}_{n-p} \\ 0 & R_{n-p} \end{vmatrix} \begin{vmatrix} x_p \\ x_{n-p} \end{vmatrix} = \begin{vmatrix} \bar{c} \\ \hat{w} \end{vmatrix}$$

for x.

In step 1, the columns of B and Z are permuted simultaneously so that the upper triangular matrix R_p is non-singular. This means that the absolute values of the diagonal elements of R_p will be greater than a user-supplied tolerance, TOL. This strategy enables any redundant or inconsistent constraints to be detected.

The algorithm requires no two dimensional arrays of storage other than those required to store Z and B. All computed quantities can overwrite the original data.

Steps 2 and 3 may be interpreted as Gaussian elimination and are accomplished by the following operations:

$$Z_{i1} = Z_{i1}/B_{11}$$

$$Z_{ij} = (Z_{ij} - \sum_{k=1}^{j-1} Z_{ik}B_{kj})/B_{jj} \qquad 2 \leq j \leq p$$

$$Z_{ij} = Z_{ij} - \sum_{k=1}^{p} Z_{ik}B_{kj} \qquad p+1 \leq j \leq n$$

$$w_i = w_i - \sum_{k=1}^{p} Z_{ik}c_k$$

all for $1 \leq i \leq m$.

1.8 Hypothesis Tests

The constraints Bx = c may be interpreted as a hypothesis concerning x. For example, if the kth constraint in B were that the ith and jth slip parameters be equal, then the non-zero elements of the kth row of B would be

$$B_{ki} = 1 \qquad B_{kj} = -1$$

with $c_k = 0$. A simple statistic may be used to test such an hypothesis.

Woonacott and Woonacott (1970) and Searle (1971, pp. 188-191) show that the F statistic can be used to test the hypothesis

$$H_0: \quad Bx = c$$

against the hypothesis

$$H_1: \quad Bx \neq c$$

If SSE is the sum of squares of the errors for the unconstrained model and SSC is that of the constrained model, then the ratio

$$F = \frac{(SSC - SSE)/p}{SSE/(m-r)}$$

has an F distribution with p and m-r degrees of freedom, where p is the rank of B, m is the number of data and r is the rank of A (or Z). When this ratio is computed, it is compared with tabulated values of F. H_0 is rejected if

$$F \geq \text{tabulated } F_{p,m-r}$$

where the probability $P(F \geq F_{p,m-r}) = 0.95$. Other probability levels (e.g., 0.99) are often used for comparison.

1.9 Inequality Constraints

The inequality constrained problem defined by Equation 1.10b is characterized in thefollowing manner:

$$q_i = 0 \qquad 1 \le i \le k$$
$$q_i > 0 \qquad k+1 \le i \le n$$

where $q = B\hat{x}-c$, $\hat{x}$ being the solution to the unconstrained problem. Thus, either the inequalities are satisfied ($q_i > 0$) or they are incorporated as equalities. The solution algorithm for the inequality constrained problem is therefore:

```
Solve unconstrained problem (Equation 1.9) to give $\hat{x}$
k:= 0
For i = 1 to p
For j = 1 to n
  $q_i := B_{ij}\hat{x}_j - c_i$
end j
  If $q_i < 0$
    then
      k:= k+1
      $B_{kj} := B_{ij}$
      $c_k := c_i$
    else
end i
If k = 0
  quit
else
  Solve constrained problem (Equation 1.10a)
```

At the end of the i loop, the unsatisfied inequalities are in the first k rows of B. The optimum solution is then given by the solution to Equation 1.10a using the first k rows of B. The interesting question is whether the data will allow such a solution. This is tested by the method described in Section 1.8.

In the inverse problems described herein, the number of constraints, p, is small. Therefore, given the unconstrained solution, constraint satisfaction may be quite easily determined by hand. The remainder of the problem may be solved by the algorithms available.

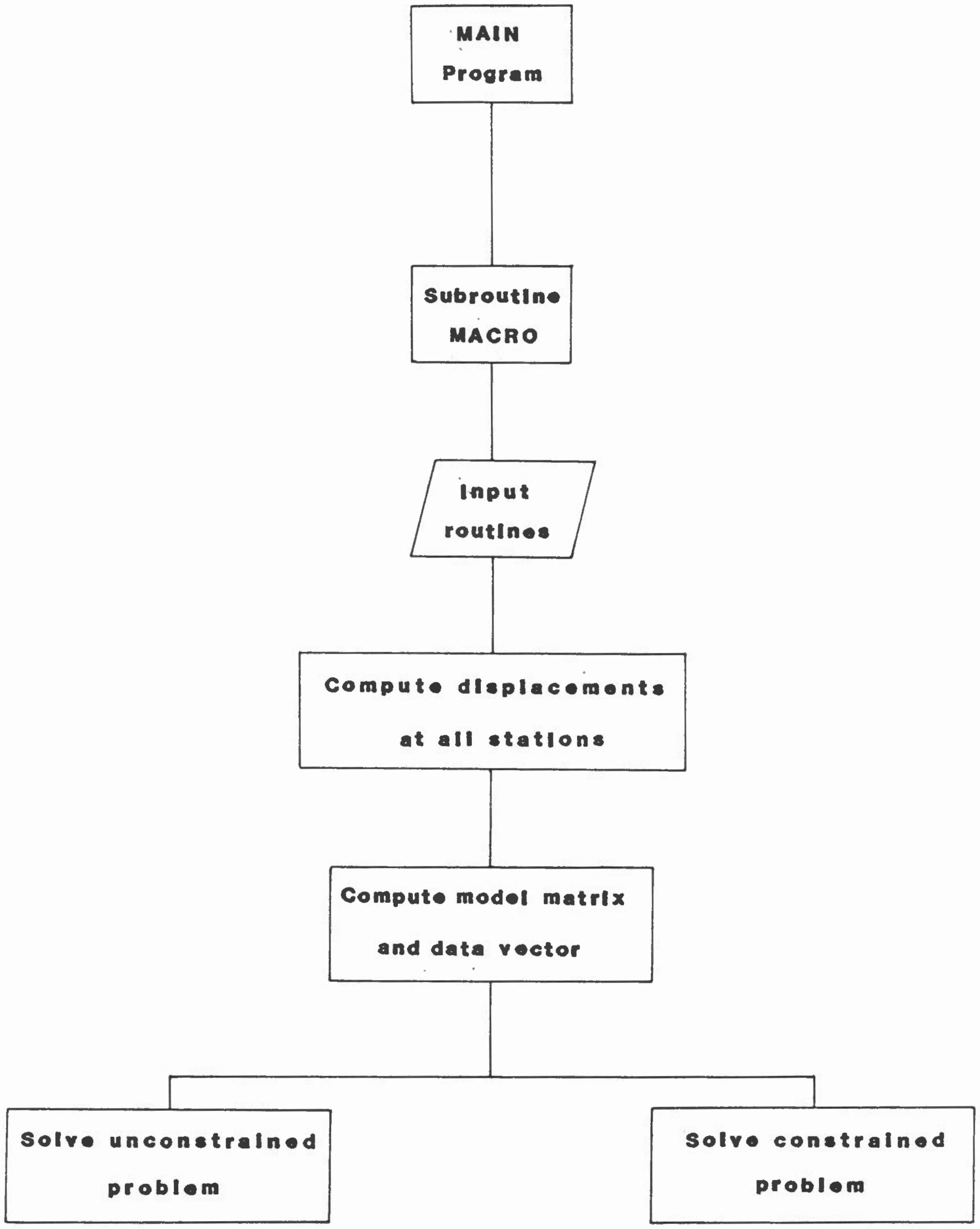
SECTION 2 - DESCRIPTION OF PROGRAM

2.1 Program Structure

SID3D consists of 37 subroutines located in a file named SID3D. A short MAIN program and two subroutines related to input operations are located in another file named SID3D.USER. SID3D is intended to be at least semi-permanent in that all of the routines in the file are used as is. SID3D.USER is more volatile; modifications to it might occur depending on the input needs of a particular problem. The entire program is listed in Appendix B.

The overall flow diagram of the program is shown in Figure 2.1. The MAIN program calls subroutine MACRO which controls program flow and storage allocation depending on commands in the input file. For every run a title and control parameters are read from the input file and printed. Following this, dislocation segment parameters and station coordinates are read and the displacements at each station due to unit strike and/or dip-slip on each dislocation segment are computed. Given these displacements, the model matrix, A, is then computed (see Section 1.3). The data and associated variances are read at the same time A is computed. The desired least squares solution is then calculated.

The entire program is written in a subset of ANSI FORTRAN 77. No special data structures or routines are required.


MAIN Program
Subroutine MACRO
Input routines
Compute displacements at all stations
Compute model matrix and data vector
Solve unconstrained problem
Solve constrained problem

PROGRAM STRUCTURE

FIGURE 2.1

2.2 Array Storage

The amount of array storage in the program is controlled by the size of a master array, A, located in blank COMMON in the MAIN program and in subroutine MACRO. Array storage in A is allocated in subroutine MACRO when the control parameters and the desired type of least squares solution are known. For an unconstrained least squares problem, the required size of A must be greater than or equal to

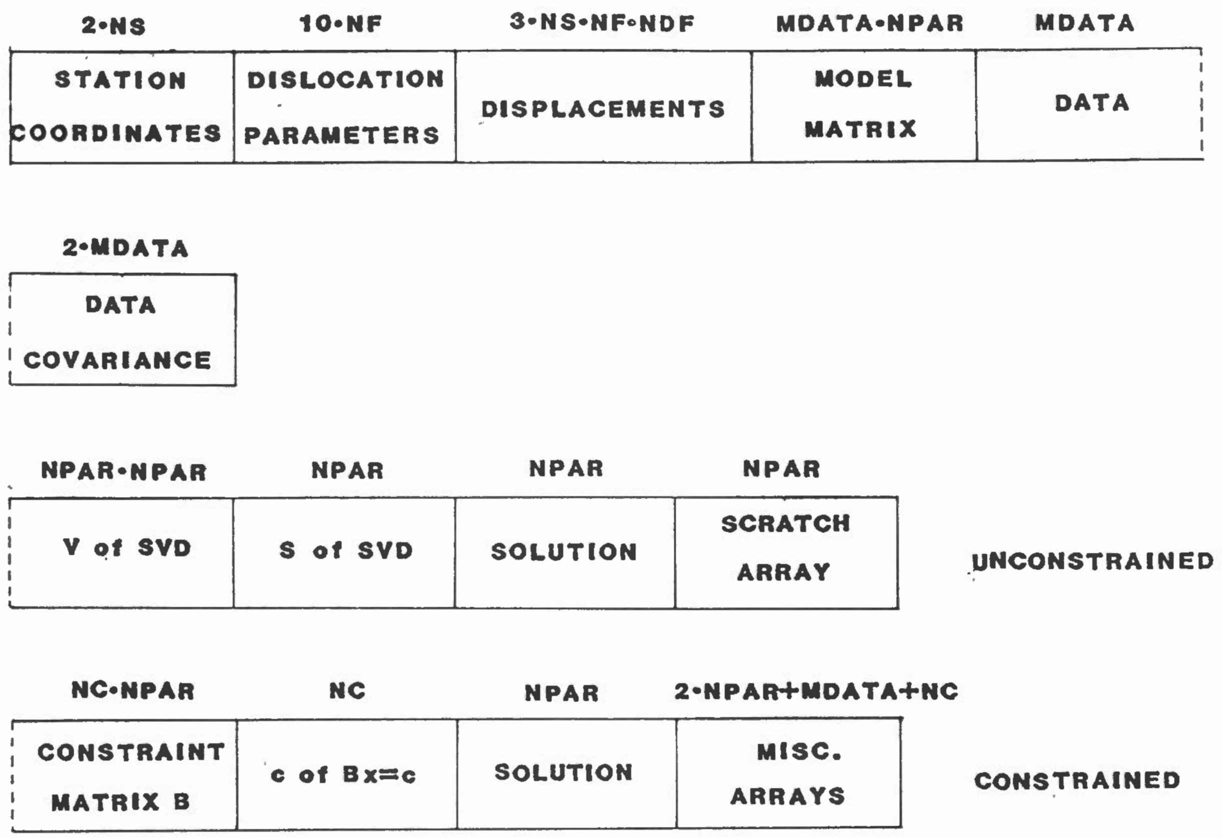
$$2\cdot NS + 10\cdot NF + 3\cdot NS\cdot NF\cdot NDF + MDATA\cdot(NF\cdot NDF + 1) + \max[2\cdot MDATA,\ NF\cdot NDF\cdot(NF\cdot NDF + 3)]$$

For a constrained least squares problem, the required size of A must be greater than or equal to

$$2\cdot NS + 10\cdot NF + 3\cdot NS\cdot NF\cdot NDF + MDATA\cdot(NF\cdot NDF + 1) + \max[2\cdot MDATA,\ 2\cdot NC + NF\cdot NDF\cdot(NC + 3) + MDATA]$$

where NS is the number of stations, NF is the number of dislocation segments, NDF = 2 if both strike and dip-slip estimates are required on all segments; NDF = 1 otherwise, MDATA is the total number of data, and NC is the number of constraints.

If the declared size of A in the MAIN program is too small, an error message will be printed and the program will stop. The current size of A is 5000 words. The configuration of storage in A is shown in Figure 2.2.



NS - Number of stations

NF - Number of dislocation segments

NDF = 1 for strike or dip-slip only

= 2 for both strike and dip-slip

MDATA - Number of data

NPAR = NF•NDF

NC - Number of constraints

CONFIGURATION OF MASTER ARRAY A

FIGURE 2.2

2.3 SID3D.USER Subroutines

In the file SID3D.USER there are the MAIN program and two user-defined subroutines, UCOORD and UFPARM. The latter two subroutines define the observation coordinates and the dislocation segment parameters, respectively, according to the user's specifications.

Note that the MAIN program, which calls subroutine MACRO, may also be rewritten so that the user can call the routines in SID3D file in any desired order. The memory allocations within the master array A (see Section 2.2) would then have to be included in the MAIN program.

2.4 SID3D Subroutines

The principal subroutines in the file SID3D are TRIANG, TRILAT, and LEVEL which compute the model matrix, A, for angle change, line length change, and elevation change data, respectively. The unconstrained least squares solution procedure is performed by subroutines ULS and SVD. The constrained least squares solution procedure is performed by subroutines CLS, HTRAN, HTAPP, and SWAP. Output for ULS and CLS is performed by subroutines ULSOUT and CLSOUT, respectively.

The other routines in SID3D are used to compute the necessary displacements (see Appendix A.2) and transform matrices into a form suitable for the least squares routines.

SECTION 3 - USER'S MANUAL

3.1 Program Operation

The operation of the program is controlled by means of macro commands located in the input file. Each macro command denotes a certain sequence of operations to be performed. For example, the command ULS initiates the sequence of calculations required to compute the unconstrained least squares estimate. The advantage of this approach is that operation of the program can be controlled from the input file and not by means of rewriting and compiling a driver program.

A complete list of the 15 existing macro commands and their purpose is given in Table 3.1. Thirty-five macro commands may be specified in subroutine MACRO; 20 commands are deliberately left blank for possible future modifications.

The only restrictions on the sequence of macro commands in the input file are the following:

1) START must be the first command.

2) CPARM must be the second command.

3) The station coordinates, dislocation segment parameters, displacements, and the command STRIKE or DIP (if necessary) must be specified or calculated before the model matrix is computed.

4) To avoid an ungraceful exit from the program, the STOP command must be given at the end of all calculations.

All macro commands are read as REAL variables in FORMAT (A4). The commands must be left-justified in the first four columns of the line. Each command may be abbreviated by its first four characters.

TABLE 3.1

MACRO COMMANDS	
START	- Read title of model
CPARM	- Read control parameters
STRIKE	- Strike-slip model only
DIP	- Dip-slip model only
UCOORD	- Read or generate station coordinates by means of user-defined subroutine UCOORD located in file SID3D.USER
CO1	- Read station coordinates individually by means of subroutine CO1 located in file SID3D
UFPARM	- Read or generate dislocation segment parameters by means of user-defined subroutine UFPARM located in file SID3D.USER
FP1	- Read dislocation segment parameters by means of subroutine FPARM1 located in file SID3D
DISPLACE	- Compute displacements at all stations due to unit strike and/or dip-slip on each dislocation segment
TRIANG	- Assemble model and data covariance matrices for angle change data
TRILAT	- Assemble model and data covariance matrices for line length change data
LEVEL	- Assemble model and data covariance matrices for elevation change data
ULS	- Compute and print unconstrained least squares estimate of slip
CLS	- Compute and print constrained least squares estimate of slip
STOP	- Stop all computations

All numerical input data is read in list-directed format. In this type of format each data item in a line of input data must be separated by a blank or comma. All data required by the READ statement in the input routine must be provided. However, a slash in the input record indicates that no more data is to be read during the current execution of the READ statement.

The data associated with the commands START, CPARM, UFPARM, FP1, UCOORD, C01, TRIANG, TRILAT, LEVEL, ULS, and CLS must immediately follow the command. The format of the data required by the commands START and CPARM is shown below.

START:

This command denotes the beginning of a run. The title of the run is read in FORMAT (20A4). The title may begin in any column of the 80 column field. The title is printed at the beginning of each major section of output.

CPARM:

The control parameters

NS NAC NLLC NEC NF

are read in the order given in list-directed format where

NS	- number of stations
NAC	- number of angle changes
NLLC	- number of line length changes
NEC	- number of elevation changes
NF	- number of dislocation segemnts

The control parameters are printed on the output file.

3.2 Units

The units of the input parameters are given below:

Coordinates, Fault dimensions	kilometres
Dip, Strike	degrees
Angle changes	seconds
Line length changes	metres
Elevation changes	metres
Variances	(data units)^2

The units of computed quantities are given below:

Displacements	dimensionless
Strike-slip, Dip-slip	metres
Errors, Predicted data, Statistics	dimensionless

3.3 Input Routines

There presently exist two input routines in the file SID3D. These are FPARM1 and COORD1. Each routine is invoked by macro commands given in Table 3.1. Data required by these routines must follow the macro command. The purpose of each of the two input routines and the data required by each of them are given below.

FPARM1 (macro command FP1):

This routine reads the parameters of individual dislocation segments. Each line of data input to this routine must contain the following eight data items in the order given:

$$K \quad H \quad DU \quad DL \quad \theta \quad \phi \quad x_1^c \quad x_2^c$$

where K is the number of the dislocation segment ($1 \leq K \leq NF$). See Figures 1.1 and A.2 for the definition of the other parameters.

COORD1 (macro command CO1):

This routine reads individual station coordinates one at a time. Each line of data input to this routine must contain the following data in the order given:

```
K   X1G   X2G
```

where K is the station number ($1 \leq K \leq NS$) and where XiG is the ith component of the global coordinates of the station (see Figure A.2).

3.4 Model Matrix Routines

The three subroutines TRIANG, TRILAT, and LEVEL assemble the model and data covariance matrices for angle change, line length change, and elevation change data, respectively. Each routine is invoked by a macro command of the same name. Each routine also reads the associated data and variances. The data required by each of these routines are given below.

TRIANG (macro command TRIANG):

For each station at which angle changes are given, the following data must be provided:

ISA ND	(line 1)
$IOS_1 \quad IOS_2 \ldots IOS_{ND}$	(line 2)
$DA_1 \quad DA_2 \ldots DA_{ND-1}$	(line 3)
$V_1 \quad V_2 \ldots V_{ND}$	(line 4)
$VP_1 \quad VP_2 \ldots VP_{ND}$	(line 5)

where ISA is the number of the occupied station ($1 \leq ISA \leq NS$), ND is the number of observed stations, IOS_i is the number of the ith observed station ($1 \leq IOS_i \leq NS$), and DA_i is the ith angle change datum. The observed stations and angle changes must be specified in clockwise order. V_i and VP_i are, respectively, the past and present variances of the ith observed direction. ND-1 independent angle changes can be specified at each occupied station. The total number of independent angle changes at all occupied stations must equal NAC (see Section 3.1).

TRILAT (macro command TRILAT):

For each station at which line length changes are given, the following data must be provided:

ISA NL	(line 1)
IOS_1 IOS_2 ... IOS_{NL}	(line 2)
DL_1 DL_2 ... DL_{NL}	(line 3)
V_1 V_2 ... V_{NL}	(line 4)

where ISA is the number of the occupied station ($1 \leq ISA \leq NS$), NL is the number of observed line lengths, IOS_i is the number of the ith observed station, ($1 \leq IOS_i \leq NS$), and DL_i is the ith line length change datum. V_i is the variance of the ith observed line length. The total number of line length changes must equal NLLC (see Section 3.1).

In both subroutines TRIANG and TRILAT up to 20 observed stations may be specified at each station. If more than 20 observed stations are specified, an error message is printed and the program stops.

LEVEL (macro command LEVEL):

For each station at which an elevation change is given, the following data must be provided:

```
IS   DE   V   VP
```

where IS is the number of the occupied station ($1 \leq IS \leq NS$) and DE is the elevation change. V and VP are, respectively, the past and present variances of the elevation at station IS. The total number of elevation changes must equal NEC (see Section 3.1).

To avoid the possibility of a singular data covariance matrix, the variances supplied to the routines TRIANG, TRILAT, and LEVEL should be greater than zero (i.e., all data have errors, regardless of their origin).

In all problems, two degrees of freedom (strike-slip and dip-slip) are assumed unless the command STRIKE or DIP is placed in the input stream.

3.5 Least Squares Routines

Subroutines ULS and CLS perform the unconstrained and constrained least squares estimation procedures, respectively. The data required for the two subroutines are given below.

ULS (macro command ULS):

Following this command the tolerance, TOL, to be used in determining the pseudorank of the model matrix must be provided.

CLS (macro command CLS):

The data required for this routine are given below.

TOL NC	(line 1)
B_{11} B_{12} ... B_{1N} C_1	(line 2)
B_{21} B_{22} ... B_{2N} C_2	(line 3)
...	
$B_{NC,1}$ $B_{NC,2}$ ... $N_{NC,N}$ C_{NC}	(line NC+1)

where NC is the number of rows in the constraint matrix $B = (B_{ij})$ and C_i is the right hand side of the constraint equation $Bx = c$. TOL is a tolerance used in determining the pseudorank of B.

Output for ULS and CLS is performed by subroutines ULSOUT and CLSOUT, respectively. These routines are called automatically after the estimation procedure is completed.

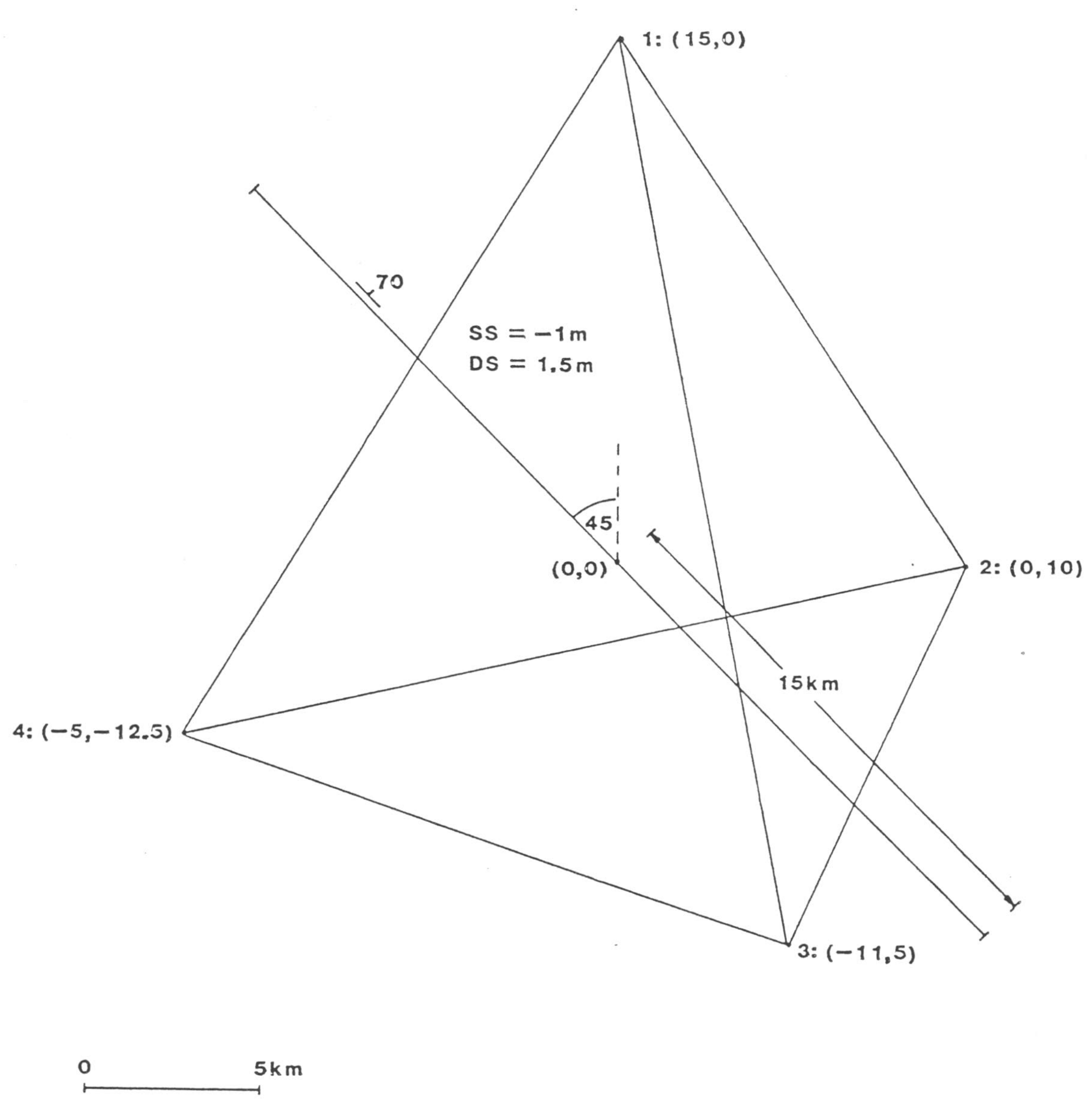
SECTION 4 - EXAMPLES

4.1 Test Case

The first example is a test case, the geometry of which is shown in Figure 4.1. Using program DIS3D (Dunbar, 1984), the angle, line length, and elevation changes due to the single dislocation segment were computed at stations 1 through 4. These data were then used as input to SID3D and an unconstrained (overdetermined) least squares slip estimate was made.

The macro commands and data input for this problem are shown below. Explanations of some data are given in parentheses.

```
START
EXAMPLE 1 - TEST CASE
CPARM
4 6 6 4 1      (NS=4, NAC=6, NLLC=6, NEC=4, NF=1)
FP1
1 15 0 5 70 -45 0 0
CO1
1 15 0
2 0 10
3 -11 5
4 -5 -12.5
DISP
TRIANG
1 3
2 3 4
2.937 -0.750
1.E-6 1.E-6 1.E-6
1.E-6 1.E-6 1.E-6
2 3
3 4 1
-4.831 -1.598
1.E-6 1.E-6 1.E-6
1.E-6 1.E-6 1.E-6
3 3
2 3 4
1.322 3.492
1.E-6 1.E-6 1.E-6
1.E-6 1.E-6 1.E-6
TRILAT
1 3
2 3 4
0.065 -0.024 0.081
1.E-6 1.E-6 1.E-6
2 2
3 4
```



EXAMPLE 1 — TEST CASE

FIGURE 4.1

```
0.164 0.258
1.E-6 1.E-6
3 1
4
-0.077
1.E-6
LEVEL
1 2.272E-3 1.E-6 1.E-6
2 5.087E-2 1.E-6 1.E-6
3 -1.148E-1 1.E-6 1.E-6
4 -5.277E-3 1.E-6 1.E-6
ULS
1.E-6
STOP
```

Notice that, even though the angle changes at station 4 are not included, there is some redundancy in the angle change data in that the sum of the angle changes in a triangle should equal zero. There is no redundancy in the line length or elevation change data.

The results are shown in the following pages. The interesting aspect of these results is the high importance of the angle change data (the first six data) relative to the other data, particularly the elevation change data (the last four data). The most important elevation change datum is that at station 3, which is reasonable in that it is closer to the fault.

```
EXAMPLE 1 - TEST CASE

     NUMBER OF STATIONS                      4
     NUMBER OF ANGLE CHANGES                 6
     NUMBER OF LINE LENGTH CHANGES           6
     NUMBER OF ELEVATION CHANGES             4
     NUMBER OF DISLOCATION SEGMENTS          1
```

```
EXAMPLE 1 - TEST CASE

STATION COORDINATES (KILOMETRES)

STATION #          X1           X2

         1   1.500E+01   0.0
         2   0.0          1.000E+01
         3  -1.100E+01   5.000E+00
         4  -5.000E+00  -1.250E+01
```

```
EXAMPLE 1 - TEST CASE

FAULT PARAMETERS

     SEGMENT   HALF LENGTH   UPPER DEPTH   LOWER DEPTH       DIP    STRIKE       X1C       X2C

           1        15.000           0.0         5.000    70.000   -45.000     0.0       0.0
```

```
EXAMPLE 1 - TEST CASE

MODEL MATRIX FOR TRIANGULATION DATA

STATION #      1

     OBSERVED STATIONS:      2     3     4

     ANGLE CHANGES (SECONDS):  2.937   -0.750

     PREVIOUS VARIANCES (SECS**2):  0.100E-05    0.100E-05    0.100E-05

     PRESENT VARIANCES (SECS**2):   0.100E-05    0.100E-05    0.100E-05

STATION #      2

     OBSERVED STATIONS:      3     4     1

     ANGLE CHANGES (SECONDS): -4.831   -1.598

     PREVIOUS VARIANCES (SECS**2):  0.100E-05    0.100E-05    0.100E-05

     PRESENT VARIANCES (SECS**2):   0.100E-05    0.100E-05    0.100E-05

STATION #      3

     OBSERVED STATIONS:      4     1     2

     ANGLE CHANGES (SECONDS):  1.322    3.492

     PREVIOUS VARIANCES (SECS**2):  0.100E-05    0.100E-05    0.100E-05

     PRESENT VARIANCES (SECS**2):   0.100E-05    0.100E-05    0.100E-05
```

```
EXAMPLE 1 - TEST CASE

MODEL MATRIX FOR TRILATERATION DATA

STATION #      1

     OBSERVED STATIONS:      2     3     4

     LINE LENGTH CHANGES (M):  0.065  -0.024   0.081

     DATA VARIANCES (M**2):  0.100E-05   0.100E-05   0.100E-05

STATION #      2

     OBSERVED STATIONS:      3     4

     LINE LENGTH CHANGES (M):  0.164   0.258

     DATA VARIANCES (M**2):  0.100E-05   0.100E-05

STATION #      3

     OBSERVED STATIONS:      4

     LINE LENGTH CHANGES (M): -0.077

     DATA VARIANCES (M**2):  0.100E-05
```

```
EXAMPLE 1 - TEST CASE

MODEL MATRIX FOR LEVELLING DATA


STATION #      1

     ELEVATION CHANGE =    0.227E-02
     PREVIOUS VARIANCE =   0.100E-05
     PRESENT VARIANCE =    0.100E-05


STATION #      2

     ELEVATION CHANGE =    0.509E-01
     PREVIOUS VARIANCE =   0.100E-05
     PRESENT VARIANCE =    0.100E-05


STATION #      3

     ELEVATION CHANGE =   -0.115E+00
     PREVIOUS VARIANCE =   0.100E-05
     PRESENT VARIANCE =    0.100E-05


STATION #      4

     ELEVATION CHANGE =   -0.528E-02
     PREVIOUS VARIANCE =   0.100E-05
     PRESENT VARIANCE =    0.100E-05
```

```
EXAMPLE 1 - TEST CASE

UNCONSTRAINED LEAST SQUARES ESTIMATION

SINGULAR VALUES

  3911.9      1288.4

TOLERANCE FOR SINGULAR VALUES =  0.10000E-05
PSEUDORANK =     2

SLIP ESTIMATE

      STRIKE-SLIP(M)          DIP-SLIP(M)

      -1.000+- 0.000        1.500+- 0.001

STATISTICS, IMPORTANCES

        DATA          DHAT          ERROR  IMPORTANCE

   0.147E+04   0.147E+04   0.339E-01   0.169E+00
   0.415E+03   0.415E+03   0.152E+00   0.140E+00
  -0.267E+04  -0.266E+04  -0.102E+00   0.484E+00
  -0.307E+04  -0.308E+04   0.874E-01   0.470E+00
  -0.166E+04  -0.166E+04  -0.496E-01   0.122E+00
   0.885E+03   0.885E+03  -0.188E-01   0.584E+00
   0.460E+02   0.461E+02  -0.104E+00   0.162E-03
  -0.170E+02  -0.168E+02  -0.199E+00   0.101E-01
   0.573E+02   0.571E+02   0.221E+00   0.322E-02
   0.116E+03   0.116E+03  -0.186E+00   0.138E-01
   0.182E+03   0.183E+03  -0.240E+00   0.172E-02
  -0.544E+02  -0.545E+02   0.623E-01   0.202E-03
   0.161E+01   0.161E+01   0.114E-03   0.157E-04
   0.360E+02   0.360E+02  -0.870E-03   0.289E-03
  -0.812E+02  -0.811E+02  -0.262E-01   0.143E-02
  -0.373E+01  -0.373E+01   0.118E-02   0.172E-04

SUM OF SQUARED ERRORS (SSE)        = 0.24135
TOTAL SUM OF SQUARES (SST)         = 0.22480E+08
REDUCTION IN SUM OF SQUARES (SSR) = 0.22480E+08
CORRELATION COEFFICIENT            =  1.0000

THE END
```

4.2 The Forbidden Plateau Triangulation Network

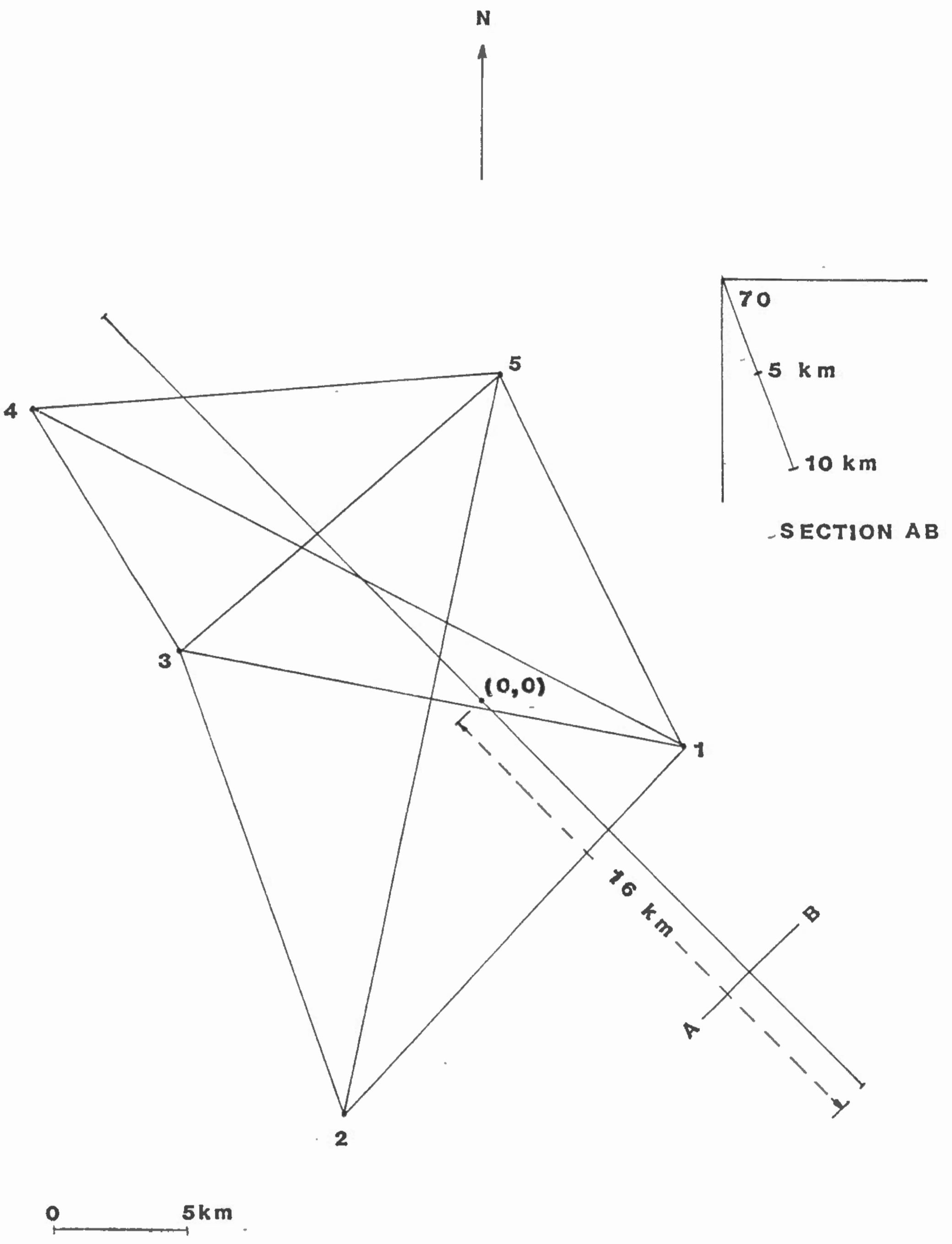
This network is located in central Vancouver Island and overlies the Beaufort Range fault. The locations of the triangulation stations and the proposed model of the Beaufort Range fault are shown in Figure 4.2. More complete descriptions of the network and the Beaufort Range fault may be found in Slawson (1978) and Slawson and Savage (1979).

The network was first surveyed in 1935-36 and resurveyed in 1978 by Slawson (1978). The unadjusted direction lists for each survey were provided by W. F. Slawson (pers. comm.). These were used to compute 10 angle changes within the network. Standard deviations of observed directions for each survey were found by adjusting both surveys for triangle closures (Slawson, 1978). For the 1935 survey, the average standard deviation of a direction was found to be 1.2"; for the 1978 survey, the standard deviation was found to be 1.3".

A two segment dislocation model was chosen in order to model possible variation of slip with depth. Each segment was 32 km. long, striking northwest, and dipping 70^0 northeast. The first segment extended from 0 to 5 km. depth (along dip); the second from 5 to 10 km.

The macro commands and data input for this problem are shown below.

```
START
EXAMPLE 2 - FORBIDDEN PLATEAU TRIANGULATION DATA
CPARM
5 10 0 0 2
C01
1 -1.25 6.0
2 -12.25 -4.13
3 1.5 -9.0
4 8.5 -13.38
5 9.75 0.5
```



EXAMPLE 2 — FORBIDDEN PLATEAU
TRIANGULATION NETWORK

FIGURE 4.2

```
FP1
1 16 0 5 70 -45 0 0
2 16 5 10 70 -45 0 0
DISP
TRIANG
1 4
2 3 4 5
-0.03 1.4 -2.4
1.44 1.44 1.44 1.44
1.69 1.69 1.69 1.69
2 2
5 1
1.6
1.44 1.44
1.69 1.69
3 4
5 1 2 4
0.8 -7.87 2.67
1.44 1.44 1.44 1.44
1.69 1.69 1.69 1.69
4 3
5 1 3
0.7 5.2
1.44 1.44 1.44
1.69 1.69 1.69
5 2
2 3
-1.4
1.44 1.44
1.69 1.69
ULS
1.E-4
STOP
```

The results for the unconstrained slip estimate are shown on the following pages. The large standard deviations of the slip on the deeper segment imply that the data does not constrain the solution at such a depth. The fit of the model is also poor in that the correlation coefficient is less than 0.5.

```
EXAMPLE 2 - FORBIDDEN PLATEAU TRIANGULATION DATA

     NUMBER OF STATIONS                    5
     NUMBER OF ANGLE CHANGES              10
     NUMBER OF LINE LENGTH CHANGES         0
     NUMBER OF ELEVATION CHANGES           0
     NUMBER OF DISLOCATION SEGMENTS        2
```

```
EXAMPLE 2 - FORBIDDEN PLATEAU TRIANGULATION DATA

STATION COORDINATES (KILOMETRES)

STATION #           X1          X2

        1 -1.250E+00  6.000E+00
        2 -1.225E+01 -4.130E+00
        3  1.500E+00 -9.000E+00
        4  8.500E+00 -1.338E+01
        5  9.750E+00  5.000E-01
```

```
EXAMPLE 2 - FORBIDDEN PLATEAU TRIANGULATION DATA

FAULT PARAMETERS

     SEGMENT   HALF LENGTH   UPPER DEPTH   LOWER DEPTH      DIP    STRIKE      X1C       X2C

           1        16.000          0.0          5.000   70.000   -45.000    0.0       0.0
           2        16.000          5.000       10.000   70.000   -45.000    0.0       0.0
```

```
EXAMPLE 2 - FORBIDDEN PLATEAU TRIANGULATION DATA

MODEL MATRIX FOR TRIANGULATION DATA


STATION #      1

     OBSERVED STATIONS:      2     3     4     5
     ANGLE CHANGES (SECONDS): -0.030    1.400  -2.400
     PREVIOUS VARIANCES (SECS**2):  0.144E+01   0.144E+01   0.144E+01   0.144E+01
     PRESENT VARIANCES (SECS**2):   0.169E+01   0.169E+01   0.169E+01   0.169E+01


STATION #      2

     OBSERVED STATIONS:      5     1
     ANGLE CHANGES (SECONDS):  1.600
     PREVIOUS VARIANCES (SECS**2):  0.144E+01   0.144E+01
     PRESENT VARIANCES (SECS**2):   0.169E+01   0.169E+01


STATION #      3

     OBSERVED STATIONS:      5     1     2     4
     ANGLE CHANGES (SECONDS):  0.800   -7.870    2.670
     PREVIOUS VARIANCES (SECS**2):  0.144E+01   0.144E+01   0.144E+01   0.144E+01
     PRESENT VARIANCES (SECS**2):   0.169E+01   0.169E+01   0.169E+01   0.169E+01


STATION #      4

     OBSERVED STATIONS:      5     1     3
     ANGLE CHANGES (SECONDS):  0.700    5.200
     PREVIOUS VARIANCES (SECS**2):  0.144E+01   0.144E+01   0.144E+01
     PRESENT VARIANCES (SECS**2):   0.169E+01   0.169E+01   0.169E+01


STATION #      5

     OBSERVED STATIONS:      2     3
     ANGLE CHANGES (SECONDS): -1.400
     PREVIOUS VARIANCES (SECS**2):  0.144E+01   0.144E+01
     PRESENT VARIANCES (SECS**2):   0.169E+01   0.169E+01
```

```
EXAMPLE 2 - FORBIDDEN PLATEAU TRIANGULATION DATA

UNCONSTRAINED LEAST SQUARES ESTIMATION

SINGULAR VALUES

  9.1733      1.7979      1.1135     0.54714

TOLERANCE FOR SINGULAR VALUES =  0.10000E-03
PSEUDORANK =      4

SLIP ESTIMATE

      STRIKE-SLIP(M)          DIP-SLIP(M)

      -0.763+- 0.623        1.208+- 0.680
       0.840+- 1.520        1.343+- 1.143

STATISTICS, IMPORTANCES

        DATA          DHAT          ERROR  IMPORTANCE

  -0.120E-01  -0.111E+01   0.110E+01   0.190E+00
   0.639E+00  -0.767E+00   0.141E+01   0.243E+00
  -0.723E+00  -0.500E+00  -0.223E+00   0.892E+00
   0.249E+00   0.534E+00  -0.285E+00   0.324E+00
   0.616E+00  -0.824E+00   0.144E+01   0.572E+00
  -0.360E+01  -0.202E+01  -0.158E+01   0.532E+00
  -0.170E+01   0.687E-01  -0.177E+01   0.197E+00
  -0.113E+01  -0.586E+00  -0.544E+00   0.324E+00
   0.178E+01   0.111E+00   0.167E+01   0.406E+00
   0.854E+00   0.905E+00  -0.509E-01   0.320E+00

SUM OF SQUARED ERRORS (SSE)        =  14.092
TOTAL SUM OF SQUARES (SST)         =  22.386
REDUCTION IN SUM OF SQUARES (SSR)  =  8.2940
CORRELATION COEFFICIENT            = 0.37050

THE END
```

The opposite sense of strike-slip on both segments is undesirable. Suppose it were desired to know whether the data would allow the strike-slip on segment 1 to be zero. This may be tested by means of the constraint equation Bx = c where

$$B = (1\ 0\ 0\ 0) \qquad c = (0)$$

The macro commands and data for this problem are the same as those of the unconstrained problem, except that the command ULS must be replaced by CLS. The data and commands following CLS are:

```
      .
      .
      .
CLS
1.E-4 1
1 0 0 0 0
STOP
```

The results of this are shown on the following page.

Using the values of SSE and SSC from the unconstrained and constrained solutions, respectively, the F statistic

$$F = \frac{(SSC-SSE)/p}{SSE/(m-r)}$$

$$= \frac{(15.591-14.092)/1}{14.092/6} = 0.64$$

From tables $F_{1,6} = 5.99$, so that the hypothesis $SS_1 = 0$ is acceptable with a probability of 0.95.

```
EXAMPLE 2 - FORBIDDEN PLATEAU TRIANGULATION DATA

CONSTRAINED LEAST SQUARES ESTIMATE

CONSTRAINT MATRIX

NUMBER OF CONSTRAINTS =      1

   0.100E+01   0.0         0.0         0.0         0.0

SLIP ESTIMATE

      STRIKE-SLIP(M)         DIP-SLIP(M)

               0.0                 0.919
              -0.918               0.601

SUM OF SQUARED ERRORS (SSC) =  15.591

THE END
```

REFERENCES

Converse, Glenn. Equations for the Displacements and Displacement Derivatives due to a Rectangular Dislocation in a Three Dimensional Elastic Halfspace. U.S. Geol. Survey, Menlo Park, Ca., 1973.

Davis, R. E., Foote, F. S., Anderson, J. M., and Mikhail, E. M. _Surveying Theory and Practice_. 6th edition, McGraw-Hill, 1981.

Dunbar, W. Scott. DIS3D, A Three Dimensional Fault Modelling Program. Report to Dep't Energy, Mines and Resources, Canada, Contract No. OSB83-00692, April 1984.

Forsythe, G.E., Malcolm, M. A. and Moler, C. B. _Computer Methods for Mathematical Computations_. Prentice-Hall, 1977.

Golub, G. H. and Reinch, C. Singular Value Decompostion and Least Squares Solutions. Numer. Math., 14, 403-420.

Jackson, D. D. Interpretation of Inaccurate, Insufficient and Incomplete Data. Geophys. J. Roy. Astr. Soc. 28, 97-109, 1972.

Lawson, C. L. and Hanson, R. J. _Solving Least Squares Problems_. Prentice-Hall Inc., 1974.

Lindgren, B. W. _Statistical Theory_. 2nd edition,Macmillan Co., 1968.

Mansinha, L. and Smylie, D. E. The Displacement Fields of Inclined Faults. Bull. Seis. Soc. Amer., 61, 1433-1440, 1971.

Searle, S. R. _Linear Models_. John Wiley and Sons Inc., 1971.

Slawson, W. F. and Savage, J. C. Geodetic Deformation Associated with the 1946 Vancouver Island, Canada Earthquake. Bull. Seis. Soc. Amer., 69, 1487-1496, 1979.

REFERENCES (Continued)

Slawson, W. F. Geodetic Deformation Associated with the 1946 Vancouver Island, Canada Earthquake. Report to Dep't Energy, Mines and Resources, Canada, Contract IST78-00072, 1978.

Wiggins, R. A. The general Linear Inverse Problem: Implications of Surface Waves and Free Oscillations for Earth Structure. Rev. Geophys., 10, 251-285, 1972.

Woonacott, R. J. and Woonacott, T. H. Econometrics. John Wiley and Sons, Inc., 1970.

APPENDIX A

A.1 Surface Displacements due to Rectangular Dislocations

In the following pages the equations for the surface displacements due to constant strike or dip-slip on a rectangular dislocation segment in a three dimensional elastic halfspace are presented. The geometry of the dislocation segment is shown in Figure 1.1. The displacement in the x_i direction is denoted u_i.

Each of the equations is the indefinite integral form of Equations 1.1 or 1.2 and must be evaluated at the four corners of the dislocation segment to obtain the desired displacement. Thus, if $F(c_0,c_1)$ represents a particular equation, the displacement is computed according to

$$F(DL,H) - F(DU,H) - F(DL,-H) + F(DU,-H)$$

Each of the equations has been programmed as a function subroutine in SID3D. The routines for the ith component of displacement due to strike and dip-slip are labelled SUiS and SUiD, respectively. Thus, SU3D evaluates the u_3 displacement due to unit dip-slip on a dislocation segment.

The notation used in these equations is given below.

ν = Poisson's ratio

$r_2 = x_2 \sin\theta \qquad r_3 = x_2 \cos\theta$

$c_2 = c_0 \cos\theta \qquad c_3 = c_0 \sin\theta$

$$R^2 = (x_1-c_1)^2 + (x_2-c_2)^2 + c_3^2$$

$$k^2 = (x_1-c_1)^2 + r_2^2$$

u_1 STRIKE-SLIP (Surface)

$$8\pi(1-\nu)\frac{u_1}{SS} = -4(1-\nu)\tan^{-1}\left[\frac{r_2 R}{(x_1-c_1)(r_3-c_0)}\right]$$

$$+\frac{r_2(x_1-c_1)}{R}\left[\frac{1}{R+r_3-c_0}-\frac{(3-4\nu)}{R-r_3+c_0}\right]$$

$$+4(1-\nu)(1-2\nu)\tan\theta\left\{2\tan\theta\tan^{-1}\left[\frac{(k-r_2\cos\theta)(r_3-c_0)-k(R+k)\sin\theta}{(x_1-c_1)(R+k)\cos\theta}\right]\right.$$

$$\left.-\frac{x_1-c_1}{R+c_3}\right\}$$

As $\theta \to \pi/2$, $\tan\theta\{\ \ \} \to \dfrac{x_2(x_1-c_1)}{2(R+c_3)^2}$

u_2 STRIKE-SLIP (Surface)

$$8\pi(1-\nu)\frac{u_2}{SS} = -(1-2\nu)\sin\theta\left[\ln(R+r_3-c_0) + \ln(R-r_3+c_0)\right]$$

$$+ \frac{4(1-\nu)r_2\cos\theta}{R} + \frac{r_2^2\sin\theta}{R}\left[\frac{1}{R+r_3-c_0} - \frac{(3-4\nu)}{R-r_3+c_0}\right]$$

$$+ 4(1-\nu)(1-2\nu)\tan\theta\left\{\tan\theta\left[\ln(R+c_3) - \sin\theta\ln(R-r_3+c_0)\right] - \frac{x_2-c_2}{R+c_3}\right\}$$

As $\theta \rightarrow \pi/2$, $\tan\theta\{\ \} \rightarrow \frac{1}{2}\left\{\frac{c_3}{R+c_3} + \ln(R+c_3) + \frac{x_2^2}{(R+c_3)^2}\right\}$

u_3 STRIKE-SLIP (Surface)

$$8\pi(1-\nu)\frac{u_3}{SS} = (1-2\nu)\cos\theta\left[\ln(R+r_3-c_0) + \ln(R-r_3+c_0)\right] + \frac{4(1-\nu)r_2\sin\theta}{R}$$

$$- \frac{r_2^2\cos\theta}{R}\left[\frac{1}{R+r_3-c_0} - \frac{(3-4\nu)}{R-r_3+c_0}\right]$$

$$- 4(1-\nu)(1-2\nu)\tan\theta\left[\ln(R+c_3) - \sin\theta\ln(R-r_3+c_0)\right]$$

As $\theta \to \pi/2$, $\tan\theta[\quad] \to \frac{x_2}{R+c_3}$

u_1 DIP-SLIP (Surface)

$$8\pi(1-\nu)\frac{u_3}{DS} = \frac{4(1-\nu)r_2}{R}$$

$$+ 4(1-\nu)(1-2\nu)\left\{\tan\theta\left[\sin\theta\ln(R+c_3) - \ln(R-r_3+c_0)\right] - \frac{(x_2-c_2)\sin\theta}{R+c_3}\right\}$$

As $\theta \to \pi/2$, $\{\ \} \to 0$

u_2 DIP-SLIP (Surface)

$$8\pi(1-\nu)\frac{u_2}{DS} = -4(1-\nu)\cos\theta\tan^{-1}\left[\frac{r_2R}{(x_1-c_1)(r_3-c_0)}\right] + \frac{4(1-\nu)r_2(x_2-c_2)}{R(R+x_1-c_1)}$$

$$- 4(1-\nu)(1-2\nu)\sin\theta\left\{2\tan\theta\tan^{-1}\left[\frac{(k-r_2\cos\theta)(r_3-c_0) - k(R+k)\sin\theta}{(x_1-c_1)(R+k)\cos\theta}\right] + \frac{x_1-c_1}{R+c_3}\right\}$$

As $\theta \to \pi/2$, $\{\ \} \to 0$

u_3 DIP-SLIP (Surface)

$$8\pi(1-\nu)\frac{u_3}{DS} = -4(1-\nu)\sin\theta\tan^{-1}\left[\frac{r_2 R}{(x_1-c_1)(r_3-c_0)}\right] - \frac{4(1-\nu)r_2 c_3}{R(R+x_1-c_1)}$$

$$+\ 8(1-\nu)(1-2\nu)\sin\theta\tan^{-1}\left[\frac{(k-r_2\cos\theta)(r_3-c_0) - k(R+k)\sin\theta}{(x_1-c_1)(R+k)\cos\theta}\right]$$

No singularity as $\theta \to \pi/2$

A.2 Computation of Displacements

The sequence of computations required to compute the displacement field is shown in Figure A.1.

The global and local (dislocation segment) coordinate systems in the $x_1^g - x_2^g$ plane are shown in Figure A.2. The angle ϕ between the x_1^g and x_1 axes is measured positive clockwise. Thus, either $-180 \leq \phi \leq 180$ or $0 \leq \phi \leq 360$.

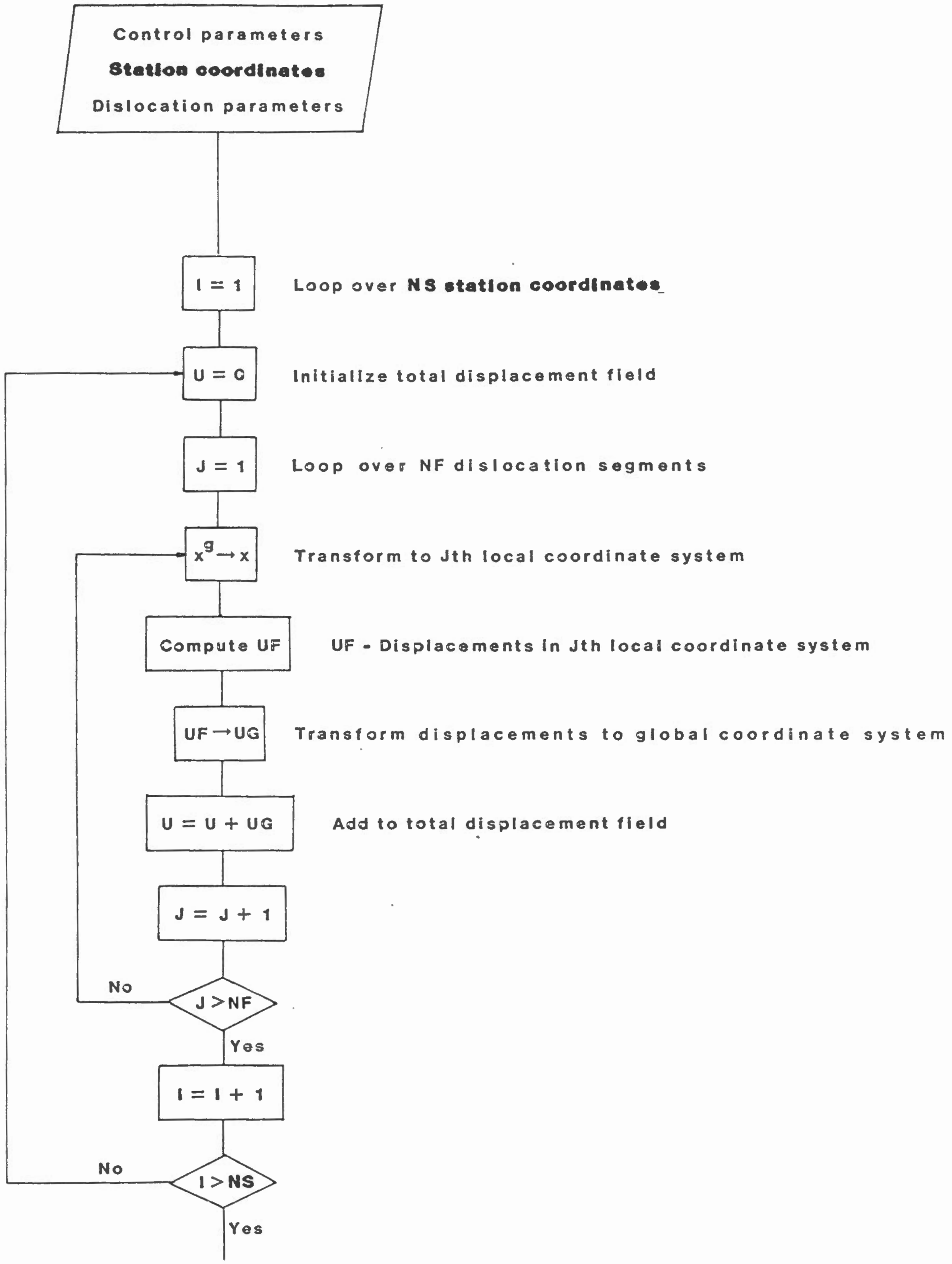
The relationship between a vector $g = (g_1, g_2, g_3)$ in the global coordinate system and a vector $f = (f_1, f_2, f_3)$ in the local coordinate system is given by

$$\begin{aligned} g_1 &= f_1\cos\phi - f_2\sin\phi \\ g_2 &= f_1\sin\phi + f_2\cos\phi \\ g_3 &= f_3 \end{aligned} \tag{A.1}$$

The inverse relationship is given by

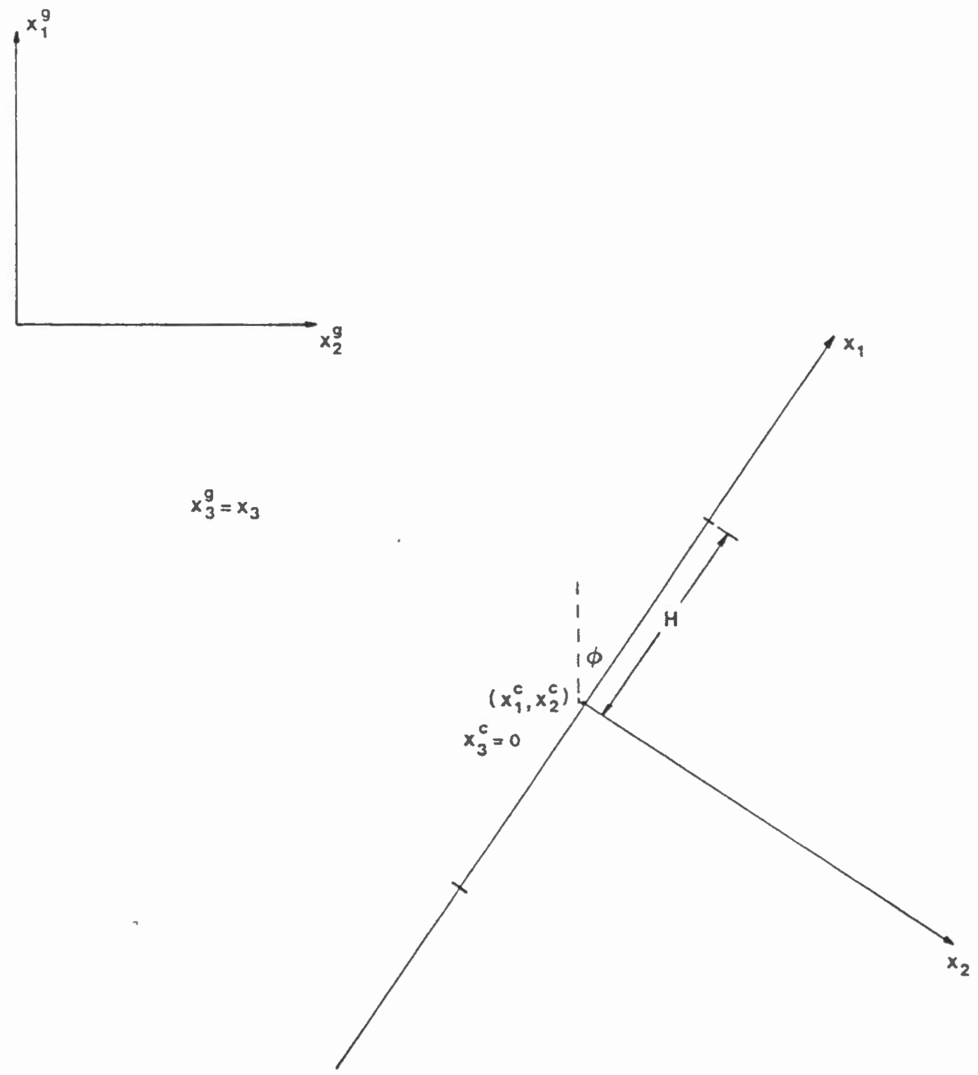
$$\begin{aligned} f_1 &= g_1\cos\phi + g_2\sin\phi \\ f_2 &= -g_1\sin\phi + g_2\cos\phi \\ f_3 &= g_3 \end{aligned} \tag{A.2}$$

The above two transformations are embodied in subroutine TRAN12 which is programmed as shown in Equation A.2. By changing the sign of ϕ and interchanging the vectors f and g, the transformation given by Equation A.1 may be effected.


Control parameters
Station coordinates
Dislocation parameters
I = 1
Loop over NS station coordinates
U = 0
Initialize total displacement field
J = 1
Loop over NF dislocation segments
$x^g \rightarrow x$
Transform to Jth local coordinate system
Compute UF
UF - Displacements in Jth local coordinate system
UF → UG
Transform displacements to global coordinate system
U = U + UG
Add to total displacement field
J = J + 1
No
J > NF
Yes
I = I + 1
No
I > NS
Yes

COMPUTATION SEQUENCE – DISPLACEMENT FIELD

FIGURE A.1



GLOBAL AND LOCAL COORDINATE SYSTEMS

FIGURE A.2

APPENDIX B

B.1 Notation

As much as possible, the notation used in the program is the same as that in this manual. The definitions of some key variables are given below (in alphabetical order):

A - model matrix (see Equation 1.7)

B - constraint matrix (see Equation 1.10)

C - right hand side of constraint equation $Bx = c$ (see Equation 1.10)

C0, C1, C2, C3 - c_0, c_1, c_2, c_3 (seeFigure 1.1 and Appendix A.1)

CP = $\cos\phi$ (see Figure A.2)

CT = $\cos\theta$ (see Figure 1.1)

D - data vector (see Equation 1.7)

D1, D2 - (x_1-c_1), (x_2-c_2)

DL - lower depth of dislocation segment (see Figure 1.1)

DU - upper depth of dislocation segment (see Figure 1.1)

FACT = $1/8\pi(1-\nu)$

FP(10,j) - the parameters H, DU, DL, $\sin\theta$, $\cos\theta$, $\tan\theta$, $\sin\phi$, $\cos\phi$, x_1^c, x_2^c of the jth dislocation segment, $1 \leq j \leq NF$ (see Figures 1.1 and A.2)

H - half length of dislocation segment (see Figure 1.1)

K - k (see Appendix A.1)

NAC - number of angle change data

NEC - number of elevation change data

NF - number of dislocation segments

NLLC - number of line length changes

NS - number of stations

PHI - ϕ (see Figure A.2)

R - R (see Appendix A.1)

R2, R3 - r_2, r_3 (see Appendix A.1)

SP = $\sin\phi$ (see Figure A.2)

ST = $\sin\theta$ (see Figure 1.1)

T - data covariance matrix (see Equation 1.8)

TT = $\tan\theta$ (see Figure 1.1)

V, V1, V2, V3 - constants involving Poisson's ratio

W - normalized data vector (see Section 1.5)

X - solution vector of least squares problem (see Equations 1.9 or 1.10)

X1, X2 - x_1, x_2 (see Figure A.2)

X1C, X2C - x_1^c, x_2^c (see Figure A.2)

X1G, X2G - x_1^g, x_2^g (see Figure A.2)

XG(2,i) - global coordinates of ith station, $1 \leq i \leq NS$

Z - normalized model matrix (see Section 1.5)

B.2 Program Listing

A complete listing of the program is given in the following pages. The first page contains the MAIN program and subroutines located in the file SID3D.USER. The remaining pages contain the subroutines located in the file SID3D.

```
      COMMON A(5000)
      COMMON /ASIZE/ MAX
C
C MAIN DRIVER FOR SID3D
C
C MAX MUST AGREE WITH DIMENSION OF A
C
      MAX = 5000
      CALL MACRO
      STOP
      END



      SUBROUTINE UCOORD (XG,NP)
      DIMENSION XG(2,NP)
      RETURN
      END



      SUBROUTINE UFPARM (FP,NF)
      DIMENSION FP(10,NF)
      RETURN
      END
      SUBROUTINE MACRO
      COMMON A(1)
      COMMON /ASIZE/ MAX
      COMMON /ELAST/ V,V1,V2,V3,FACT
      LOGICAL COMP,CPAR,FPAR,DISP
      DIMENSION CA(35),TITLE(20)
      DATA NC/35/
      INTEGER LIST(1)/1H*/
      INTEGER P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12,P13,P14,P15
      DATA CA/4HSTAR,4HCPAR,4HSTRI,4HDIP ,4H    ,
     1        4HUCOO,4HCO1 ,4H    ,4H    ,4H    ,
     2        4HUFPA,4HFP1 ,4H    ,4H    ,4H    ,
     3        4HDISP,4H    ,4H    ,4H    ,4H    ,
     4        4HTRIA,4HTRIL,4HLEVE,4H    ,4H    ,
     5        4HULS ,4HCLS ,4H    ,4H    ,4H    ,
     6        4H    ,4H    ,4H    ,4H    ,4HSTOP/
C
      CPAR = .FALSE.
      FPAR = .FALSE.
      DISP = .FALSE.
      CALL ZERO (A,MAX)
C
    1 CONTINUE
      READ (5,500) C
  500 FORMAT (A4)
      IF (COMP(C,CA(35))) GO TO 350
C
    2 CONTINUE
      DO 3 I = 1,NC
        IC = I
        IF (COMP(C,CA(I))) GO TO 4
    3 CONTINUE
      CALL ERROR (1,0,C)
C
    4 GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,
     1       140,150,160,170,180,190,200,210,220,230,240,
     2       250,260,270,280,290,300,310,320,330,340,350),IC
C
C READ JOB TITLE
C
   10 CONTINUE
      READ (5,501) TITLE
      WRITE (6,600) TITLE
  501 FORMAT (20A4)
  600 FORMAT (1H1,20A4/)
      GO TO 1
C
C READ CONTROL PARAMETERS
C
   20 CONTINUE
      READ (5,LIST) NS,NAC,NLLC,NEC,NF
      WRITE (6,605) NS,NAC,NLLC,NEC,NF
  605 FORMAT (1H ,5X,36HNUMBER OF STATIONS                    ,I5/
     1        1H ,5X,36HNUMBER OF ANGLE CHANGES               ,I5/
    2         1H ,5X,36HNUMBER OF LINE LENGTH CHANGES       ,I5/
    3         1H ,5X,36HNUMBER OF ELEVATION CHANGES         ,I5/
    4         1H ,5X,36HNUMBER OF DISLOCATION SEGMENTS      ,I5)
      MDATA = NAC+NLLC+NEC
      ISS = 1
      IDS = 1
      NDF = 2
      P2 = 2*NS+1
      NPAR = NF*NDF
      NNN = NS*NPAR
      NROW = 0
      CPAR = .TRUE.
C
C POISSON'S RATIO = 0.25 ASSUMED
C
      V = 0.25
      V1 = 0.75
      V2 = 0.50
      V3 = 2.0
      FACT = 1./(24.*ATAN(1.0))
      GO TO 1
C
C STRIKE-SLIP ONLY
C
   30 CONTINUE
      IDS = 0
      NDF = 1
      NPAR = NF
      WRITE (6,610) NPAR
  610 FORMAT (47HOSTRIKE-SLIP ONLY. NUMBER OF PARAMETERS (=NF) = ,I5)
      GO TO 1
C
C DIP-SLIP ONLY
C
   40 CONTINUE
      ISS = 0
      NDF = 1
      NPAR = NF
      WRITE (6,611) NPAR
  611 FORMAT (44HODIP-SLIP ONLY. NUMBER OF PARAMETERS (=NF) = ,I5)
      GO TO 1
C
   50 CONTINUE
      GO TO 1
C
C READ STATION COORDINATES - USER DEFINED ROUTINE (UCOO)
C
   60 CONTINUE
      IF (.NOT.CPAR) CALL ERROR (3,0,C)
      CALL UCOORD (A,NS)
      CALL PRINT (A,A(P2),TITLE,NS,NF,1)
      GO TO 1
C
C READ STATION COORDINATES - COORD1 ROUTINE (CO1)
```

```
C
   70 CONTINUE
      IF (.NOT.CPAR) CALL ERROR (3,0,C)
      CALL COORD1 (A,NS)
      CALL PRINT (A,A(P2),TITLE,NS,NF,1)
      GO TO 1
C
   80 CONTINUE
      GO TO 1
C
   90 CONTINUE
      GO TO 1
C
  100 CONTINUE
      GO TO 1
C
C READ FAULT PARAMETERS - USER DEFINED ROUTINE (UFPA)
C
  110 CONTINUE
      CALL UFPARM (A(P2),NF)
      FPAR = .FALSE.
      CALL PRINT (A,A(P2),TITLE,NS,NF,2)
      GO TO 1
C
C READ FAULT PARAMETERS - FPARM1 ROUTINE (FP1)
C
  120 CONTINUE
      CALL FPARM1 (A(P2),NF)
      FPAR = .FALSE.
      CALL PRINT (A,A(P2),TITLE,NS,NF,2)
      GO TO 1
C
  130 CONTINUE
      GO TO 1
C
  140 CONTINUE
      GO TO 1
C
  150 CONTINUE
      GO TO 1
C
C COMPUTE DISPLACEMENTS AT ALL STATIONS (DISP)
C
  160 CONTINUE
      IF (.NOT.FPAR) CALL FPMOD (A(P2),NF)
      FPAR = .TRUE.
      P3 = P2+10*NF
      P4 = P3+NNN
      P5 = P4+NNN
      LAST = P5+NNN-1
      IF (LAST.GT.MAX) CALL ERROR (2,LAST-MAX,C)
      CALL DCOEF (A,A(P2),A(P3),A(P4),A(P5),NS,NF,NDF,ISS,IDS)
      DISP = .TRUE.
      GO TO 1
```

```
C
  170 CONTINUE
      GO TO 1
C
  180 CONTINUE
      GO TO 1
C
  190 CONTINUE
      GO TO 1
C
  200 CONTINUE
      GO TO 1
C
C COMPUTE MODEL MATRIX FOR TRIANGULATION DATA
C
  210 CONTINUE
      IF (.NOT.DISP) CALL ERROR (4,0,C)
      P6 = P5+NNN
      P7 = P6+MDATA*NPAR
      P8 = P7+MDATA
      LAST = P8+MDATA*2-1
      IF (LAST.GT.MAX) CALL ERROR (2,LAST-MAX,C)
      CALL TRIANG (A,A(P3),A(P4),A(P6),A(P7),A(P8),TITLE,NAC,NS,
     1             MDATA,NF,NDF,ISS,IDS,NROW)
      GO TO 1
C
C COMPUTE MODEL MATRIX FOR TRILATERATION DATA
C
  220 CONTINUE
      IF (.NOT.DISP) CALL ERROR (4,0,C)
      P6 = P5+NNN
      P7 = P6+MDATA*NPAR
      P8 = P7+MDATA
      LAST = P8+MDATA*2-1
      IF (LAST.GT.MAX) CALL ERROR (2,LAST-MAX,C)
      CALL TRILAT (A,A(P3),A(P4),A(P6),A(P7),A(P8),TITLE,NLLC,NS,
     1             MDATA,NF,NDF,ISS,IDS,NROW)
      GO TO 1
C
C COMPUTE MODEL MATRIX FOR LEVELLING DATA
C
  230 CONTINUE
      IF (.NOT.DISP) CALL ERROR (4,0,C)
      P6 = P5+NNN
      P7 = P6+MDATA*NPAR
      P8 = P7+MDATA
      LAST = P8+MDATA*2-1
      IF (LAST.GT.MAX) CALL ERROR (2,LAST-MAX,C)
      CALL LEVEL (A(P5),A(P6),A(P7),A(P8),TITLE,NEC,NS,MDATA,NF,
     1            NDF,ISS,IDS,NROW)
      GO TO 1
C
  240 CONTINUE
      GO TO 1
```

```
C
 250 CONTINUE
     GO TO 1
C
C SOLVE UNCONSTRAINED LEAST SQUARES PROBLEM
C
 260 CONTINUE
     WRITE (6,626) TITLE
 626 FORMAT (1H1,20A4//39H UNCONSTRAINED LEAST SQUARES ESTIMATION)
     READ (5,LIST) TOL
     CALL ZWCOMP (A(P6),A(P7),A(P8),MDATA,MDATA,NPAR)
     P9 = P8+NPAR*NPAR
     P10 = P9+NPAR
     P11 = P10+NPAR
     LAST = P11+NPAR-1
     IF (LAST.GT.MAX) CALL ERROR (2,LAST-MAX,C)
     CALL ULS (A(P6),A(P7),A(P8),A(P9),A(P10),A(P11),KR,TOL,MDATA,
    1          MDATA,NPAR,NPAR)
     CALL ULSOUT (A(P6),A(P7),A(P8),A(P9),A(P10),A(P11),
    1             TOL,KR,MDATA,MDATA,NPAR,NPAR,ISS,IDS)
     GO TO 1
C
C SOLVE CONSTRAINED LEAST SQUARES PROBLEM
C
 270 CONTINUE
     WRITE (6,627) TITLE
 627 FORMAT (1H1,20A4//35H CONSTRAINED LEAST SQUARES ESTIMATE)
     READ (5,LIST) TOL,NC
     CALL ZWCOMP (A(P6),A(P7),A(P8),MDATA,MDATA,NPAR)
     P9 = P8+NC*NPAR
     P10 = P9+NC
     P11 = P10+NPAR
     P12 = P11+NPAR
     P13 = P12+NPAR
     P14 = P13+MDATA
     LAST = P14+NC-1
     IF (LAST.GT.MAX) CALL ERROR (2,LAST-MAX,C)
     CALL CONMAT (A(P8),A(P9),NC,NPAR)
     CALL CLS (A(P6),A(P7),A(P8),A(P9),A(P10),MDATA,MDATA,NPAR,
    1          NC,NC,A(P11),A(P12),A(P13),A(P14),TOL,SSC,INFO)
     IF (INFO.NE.0) CALL ERROR (9,INFO,C)
     CALL CLSOUT (A(P10),SSC,NPAR,ISS,IDS)
     GO TO 1
C
 280 CONTINUE
     GO TO 1
C
 290 CONTINUE
     GO TO 1
C
 300 CONTINUE
     GO TO 1
C
 310 CONTINUE
```

```
      GO TO 1
C
  320 CONTINUE
      GO TO 1
C
  330 CONTINUE
      GO TO 1
C
  340 CONTINUE
      GO TO 1
C
C THE END
C
  350 CONTINUE
      WRITE (6,699)
  699 FORMAT (/8H THE END)
      RETURN
C
      END



      LOGICAL FUNCTION COMP (A,B)
C
C TEST EQUALITY OF CHARACTER STRINGS A AND B
C
      COMP = .FALSE.
      IF (A.EQ.B) COMP = .TRUE.
      RETURN
      END
      SUBROUTINE ERROR (IE,N,C)
C
C PRINT FATAL ERROR MESSAGES
C
      GO TO (1,2,3,4,5,6,7,8,9),IE
C
    1 WRITE (6,601) C
  601 FORMAT (/15H MACRO COMMAND ,A4,16H  DOES NOT EXIST)
      STOP
C
    2 WRITE (6,602) C,N
  602 FORMAT (/1H ,A4,28H  ARRAY STORAGE EXCEEDED BY ,I5)
      STOP
C
    3 WRITE (6,603) C
  603 FORMAT (/1H ,A4,40H CANNOT DO THIS. THE CONTROL PARAMETERS,
     1 27H HAVE NOT YET BEEN DEFINED.)
      STOP
C
    4 WRITE (6,604) C
  604 FORMAT (/1H ,A4,39H CANNOT DO THIS. THE DISPLACEMENTS,
     1 40H AT EACH STATION HAVE NOT BEEN COMPUTED.)
      STOP
C
    5 WRITE (6,605) C,N
  605 FORMAT (/1H ,A4,27H MAXIMUM NUMBER OF OBSERVED,
     1 26H DIRECTIONS EXCEEDED,ND = ,I5)
      STOP
C
    6 WRITE (6,606) C,N
  606 FORMAT (/1H ,A4,27H MAXIMUM NUMBER OF OBSERVED,
     1 28H LINE LENGTHS EXCEEDED,NL = ,I5)
      STOP
C
    7 WRITE (6,607) N
  607 FORMAT (/33H ULS OR CLS COVARIANCE MATRIX NOT,
     1 25H POSITIVE DEFINITE,ROW # ,I5)
      STOP
C
    8 WRITE (6,608) C,N
  608 FORMAT (/1H ,A4,23H SVD FAILED TO CONVERGE,
     1 3H AT,I5,17HTH SINGULAR VALUE)
      STOP
C
    9 WRITE (6,609) C,N
  609 FORMAT (/1H ,A4,25H CONSTRAINTS INCONSISTENT,
     1 20H CHECK CONSTRAINT # ,I5)
      STOP
      END
      SUBROUTINE COORD1 (XG,NS)
      DIMENSION XG(2,NS)
      INTEGER LIST(1)/1H*/
C
C READ AND PRINT STATION COORDINATES
C
      DO 10 J = 1,NS
        READ (5,LIST) K,XG(1,K),XG(2,K)
   10 CONTINUE
      RETURN
C
      END



      SUBROUTINE FPARM1 (FP,NF)
      DIMENSION FP(10,NF)
      INTEGER LIST(1)/1H*/
C
C READ PARAMETERS OF INDIVIDUAL FAULT SEGMENTS
C
      DO 10 J = 1,NF
        READ (5,LIST) K,(FP(I,K),I = 1,7)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE PRINT (XG,FP,TITLE,NS,NF,ITYPE)
      DIMENSION XG(2,NS),FP(10,NF),TITLE(20)
C
C GENERAL PRINT ROUTINE
C
      NLINE = 0
      IF (ITYPE.EQ.0) RETURN
      IF (ITYPE.EQ.2) GO TO 20
C
   10 CONTINUE
      DO 19 J = 1,NS
        IF (NLINE.GT.0) GO TO 15
        WRITE (6,610) TITLE
        NLINE = 44
   15 CONTINUE
        WRITE (6,611) J,(XG(I,J),I = 1,2)
        NLINE = NLINE-1
   19 CONTINUE
      RETURN
C
   20 CONTINUE
      DO 29 J = 1,NF
        IF (NLINE.GT.0) GO TO 25
        WRITE (6,620) TITLE
        NLINE = 44
   25 CONTINUE
        WRITE (6,621) J,(FP(I,J),I = 1,7)
        NLINE = NLINE-1
   29 CONTINUE
      RETURN
C
  610 FORMAT (1H1,20A4//33H STATION COORDINATES (KILOMETRES)//
     1 11H STATION # ,9X,2HX1,9X,2HX2/)
  611 FORMAT (1H ,I10,1PE11.3,E11.3)
  620 FORMAT (1H1,20A4//17H FAULT PARAMETERS//
     1 1H ,5X,7HSEGMENT,3X,11HHALF LENGTH,3X,11HUPPER DEPTH,3X,
     2 11HLOWER DEPTH,7X,3HDIP,4X,6HSTRIKE,7X,3HX1C,7X,3HX2C/)
  621 FORMAT (1H ,7X,I5,3F14.3,6F10.3)
C
      END
      SUBROUTINE FPMOD (FP,NF)
      DIMENSION FP(10,NF)
C
C COMPUTE TRIG FUNCTIONS OF DIP AND STRIKE
C STORE RESULTS IN FP ARRRAY
C
      RPD = ATAN(1.0)/45.0
      DO 10 J = 1,NF
        FP(10,J) = FP(7,J)
        FP(9,J) = FP(6,J)
        THETA = FP(4,J)*RPD
        PHI =   FP(5,J)*RPD
        SP = SIN(PHI)
        CP = COS(PHI)
        FP(8,J) = CP
        FP(7,J) = SP
        ST = SIN(THETA)
        CT = COS(THETA)
        FP(6,J) = 90.0
        IF (CT.NE.0.0) FP(6,J) = ST/CT
        FP(4,J) = ST
        FP(5,J) = CT
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE DCOEF (XG,FP,U1G,U2G,U3G,NS,NF,NDF,ISS,IDS)
      DIMENSION FP(10,NF),XG(2,NS),U1G(NS,NF,NDF),U2G(NS,NF,NDF)
      DIMENSION U3G(NS,NF,NDF),UF(3,2)
C
C COMPUTE DISPLACEMENT COEFFICIENTS AT NS STATIONS FOR NF SEGMENTS
C
      DO 30 I = 1,NS
        X1G = XG(1,I)
        X2G = XG(2,I)
      DO 20 J = 1,NF
        CALL SETUP (FP(1,J),X1G,X2G,SP,CP)
        CALL DISP1 (FP(1,J),FP(2,J),FP(3,J),ISS,IDS,UF)
      DO 10 K = 1,NDF
        CALL TRAN12 (UF(1,K),UF(2,K),-SP,CP,U1G(I,J,K),U2G(I,J,K))
        U3G(I,J,K) = UF(3,K)
   10 CONTINUE
   20 CONTINUE
   30 CONTINUE
      RETURN
C
      END
      SUBROUTINE DISP1 (H,DU,DL,ISS,IDS,U)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      COMMON /ELAST/ V,V1,V2,V3,FACT
      DIMENSION U(3,2)
C
C COMPUTE DISPLACEMENT COEFFICIENTS FOR SINGLE SEGMENT
C
      CALL ZERO (U,6)
      NDF = ISS+IDS
      S = 1.0
      C1 = H
      DO 40 I = 1,2
        CO = DL
      DO 30 J = 1,2
        CALL RS (CO,C1)
        IF (ISS.EQ.0) GO TO 20
        U(1,1) = U(1,1)+S*SU1S(CO,C1)
        U(2,1) = U(2,1)+S*SU2S(CO,C1)
        U(3,1) = U(3,1)+S*SU3S(CO,C1)
   20 CONTINUE
        IF (IDS.EQ.0) GO TO 25
        U(1,NDF) = U(1,NDF)+S*SU1D(CO,C1)
        U(2,NDF) = U(2,NDF)+S*SU2D(CO,C1)
        U(3,NDF) = U(3,NDF)+S*SU3D(CO,C1)
   25 CONTINUE
        S = -S
        CO = DU
   30 CONTINUE
        S = -S
        C1 = -H
   40 CONTINUE
      CALL SCALE (U,FACT,6)
      RETURN
C
      END
      SUBROUTINE SETUP (FP,X1G,X2G,SP,CP)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      DIMENSION FP(10)
C
C SET UP COMMON BLOCK GEOM1
C
      ST = FP(4)
      CT = FP(5)
      TT = FP(6)
      IVERT = 0
      IF (TT.EQ.90.0) IVERT = 1
      SP = FP(7)
      CP = FP(8)
      X1C = FP(9)
      X2C = FP(10)
      CALL TRAN12 (X1G-X1C,X2G-X2C,SP,CP,X1,X2)
      IF (ABS(X1).LE.1.E-6).X1 = 0.0
      IF (ABS(X2).LE.1.E-6) X2 = 0.0
      R2 = X2*ST
      R3 = X2*CT
C
      RETURN
      END
      SUBROUTINE ZERO (A,N)
      DIMENSION A(N)
C
C ZERO ARRAY A
C
      DO 10 I = 1,N
        A(I) = 0.0
   10 CONTINUE
      RETURN
C
      END



      SUBROUTINE SCALE (A,S,N)
      DIMENSION A(N)
C
C SCALE ARRAY A BY S
C
      DO 10 I = 1,N
        A(I) = A(I)*S
   10 CONTINUE
      RETURN
C
      END



      SUBROUTINE TRAN12 (X1,X2,SP,CP,Y1,Y2)
C
C COORDINATE TRANSFORMATION IN X1G-X2G PLANE
C
      Y1 = X1*CP + X2*SP
      Y2 = -X1*SP + X2*CP
      RETURN
      END



      SUBROUTINE RS (CO,C1)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      COMMON /GEOM2/ D1,D2,C2,C3,R
C
C COMPUTE DISTANCE R
C
      C2 = CO*CT
      C3 = CO*ST
      D1 = X1-C1
      D2 = X2-C2
      R = SQRT(D1*D1+D2*D2+C3*C3)
      RETURN
      END
      FUNCTION SU1S (CO,C1)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      COMMON /GEOM2/ D1,D2,C2,C3,R
      COMMON /ELAST/ V,V1,V2,V3,FACT
      REAL K
C
C U1 DISPLACEMENT ON X3 = 0 STRIKE-SLIP
C
      R3M = R3-CO
      IF (ABS(R3M).LE.1.E-6) R3M = 0.0
      FRP = R+R3M
      FRM = R-R3M
C
      T1 = -4.*V1*ATAN2(R2*R,D1*R3M)
      T2 = R2*D1*(1./FRP-V3/FRM)/R
      RPC3 = R+C3
      IF (IVERT.EQ.0) GO TO 1
      T3 = 2.*V1*V2*X2*D1/RPC3**2
      GO TO 2
    1 K = SQRT(D1*D1+R2*R2)
      RPK = R+K
      TOP = (K-R2*CT)*R3M-K*RPK*ST
      BOT = D1*RPK*CT
      T3 = 4.*V1*V2*TT*(2.*TT*ATAN2(TOP,BOT)-D1/RPC3)
C
    2 SU1S = T1+T2+T3
C
      RETURN
      END
      FUNCTION SU2S (CO,C1)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      COMMON /GEOM2/ D1,D2,C2,C3,R
      COMMON /ELAST/ V,V1,V2,V3,FACT
C
C U2 DISPLACEMENT ON X3 = O STRIKE-SLIP
C
      R3M = R3-CO
      IF (ABS(R3M).LE.1.E-6) R3M = O.O
      FRP = R+R3M
      FRM = R-R3M
      ALM = ALOG(FRM)
C
      T1 = -V2*ST*(ALOG(FRP)+ALM)
      T2 = (4.*V1*R2*CT+R2**2*ST*(1./FRP-V3/FRM))/R
      RPC3 = R+C3
      IF (IVERT.EQ.O) GO TO 1
      T3 = 2.*V1*V2*(C3/RPC3+ALOG(RPC3)+(X2/RPC3)**2)
      GO TO 2
    1 T3 = 4.*V1*V2*TT*(TT*(ALOG(RPC3)-ST*ALM)-D2/RPC3)
C
    2 SU2S = T1+T2+T3
C
      RETURN
      END
      FUNCTION SU3S (CO,C1)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      COMMON /GEOM2/ D1,D2,C2,C3,R
      COMMON /ELAST/ V,V1,V2,V3,FACT
C
C U3 DISPLACEMENT ON X3 = O STRIKE-SLIP
C
      R3M = R3-CO
      IF (ABS(R3M).LE.1.E-6) R3M = 0.0
      FRP = R+R3M
      FRM = R-R3M
      ALM = ALOG(FRM)
C
      T1 = V2*CT*(ALOG(FRP)+ALM)
      T2 = (4.*V1*R2*ST-R2**2*CT*(1./FRP-V3/FRM))/R
      RPC3 = R+C3
      IF (IVERT.EQ.O) GO TO 1
      T3 = -4.*V1*V2*X2/RPC3
      GO TO 2
    1 T3 = -4.*V1*V2*TT*(ALOG(RPC3)-ST*ALM)
C
    2 SU3S = T1+T2+T3
C
      RETURN
      END
      FUNCTION SU1D (CO,C1)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      COMMON /GEOM2/ D1,D2,C2,C3,R
      COMMON /ELAST/ V,V1,V2,V3,FACT
C
C U1 DISPLACEMENT ON X3 = O DIP-SLIP
C
      T1 = 4.*V1*R2/R
      T2 = 0.0
      IF (IVERT.NE.O) GO TO 2
      RPC3 = R+C3
      T2 = 4.*V1*V2*(TT*(ST*ALOG(RPC3)-ALOG(R-R3+CO))-D2*ST/RPC3)
C
    2 SU1D = T1+T2
C
      RETURN
      END
      FUNCTION SU2D (CO,C1)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      COMMON /GEOM2/ D1,D2,C2,C3,R
      COMMON /ELAST/ V,V1,V2,V3,FACT
      REAL K
C
C U2 DISPLACEMENT ON X3 = O DIP-SLIP
C
      R3M = R3-CO
      IF (ABS(R3M).LE.1.E-6) R3M = 0.0
C
      T1 = -4.*V1*(CT*ATAN2(R2*R,D1*R3M)-R2*D2/(R*(R+D1)))
      T2 = 0.0
      IF (IVERT.NE.O) GO TO 2
      K = SQRT(D1*D1+R2*R2)
      RPK = R+K
      TOP = (K-R2*CT)*R3M-K*RPK*ST
      BOT = D1*RPK*CT
      T2 = -4.*V1*V2*ST*(2.*TT*ATAN2(TOP,BOT)-D1/(R+C3))
C
    2 SU2D = T1+T2
C
      RETURN
      END
      FUNCTION SU3D (CO,C1)
      COMMON /GEOM1/ X1,X2,ST,CT,TT,R2,R3,IVERT
      COMMON /GEOM2/ D1,D2,C2,C3,R
      COMMON /ELAST/ V,V1,V2,V3,FACT
      REAL K
C
C U3 DISPLACEMENT ON X3 = O DIP-SLIP
C
      R3M = R3-CO
      IF (ABS(R3M).LE.1.E-6) R3M = 0.0
C
      T1 = 4.*V1*(ST*ATAN2(D1*R3M,R2*R)-R2*C3/(R*(R+D1)))
      K = SQRT(D1*D1+R2*R2)
      RPK = R+K
      TOP = (K-R2*CT)*R3M-K*RPK*ST
      BOT = -D1*RPK*CT
      T2 = -8.*V1*V2*ST*ATAN2(TOP,BOT)
C
      SU3D = T1+T2
C
      RETURN
      END
      SUBROUTINE TRIANG (XG,U1G,U2G,A,D,T,TITLE,NAC,NS,MDATA,NF,NDF,
     1                   ISS,IDS,NR)
      DIMENSION XG(2,1),U1G(NS,NF,NDF),U2G(NS,NF,NDF),A(MDATA,1)
      DIMENSION D(1),T(MDATA,2),TITLE(20)
      DIMENSION IOS(20),AC(20),V(20),VP(20)
      INTEGER LIST(1)/1H*/
C
C ASSEMBLE MODEL AND DATA COVARIANCE MATRICES FOR ANGLE CHANGE DATA
C
      SPR = 162.0/ATAN(1.0)
      NDAT = 0
      WRITE (6,600) TITLE
   10 CONTINUE
        READ (5,LIST) ISA,ND
        WRITE (6,601) ISA
        IF (ND.GT.20) CALL ERROR (5,ND,4HTRIA)
        READ (5,LIST) (IOS(K),K = 1,ND)
        WRITE (6,602) (IOS(K),K = 1,ND)
        NDM1 = ND-1
        NDAT = NDAT+NDM1
        READ (5,LIST) (AC(K),K = 1,NDM1)
        WRITE (6,603) (AC(K),K = 1,NDM1)
        READ (5,LIST) (V(K),K = 1,ND)
        WRITE (6,604) (V(K),K = 1,ND)
        READ (5,LIST) (VP(K),K = 1,ND)
        WRITE (6,605) (VP(K),K = 1,ND)
        X1A = XG(1,ISA)
        X2A = XG(2,ISA)
      DO 40 J = 2,ND
        IOSJM1 = IOS(J-1)
        IOSJ   = IOS(J)
        DX1B = XG(1,IOSJM1) - X1A
        DX1C = XG(1,IOSJ  ) - X1A
        DX2B = XG(2,IOSJM1) - X2A
        DX2C = XG(2,IOSJ  ) - X2A
        NR = NR+1
        D(NR) = AC(J-1)
        T(NR,1) = V(J)+VP(J)+V(J-1)+VP(J-1)
        T(NR,2) = -V(J)-VP(J)
        NC = 0
      DO 30 K = 1,NF
        IF (ISS.EQ.0) GO TO 25
C
C STRIKE-SLIP
C
        DU1B = U1G(IOSJM1,K,1) - U1G(ISA,K,1)
        DU1C = U1G(IOSJ  ,K,1) - U1G(ISA,K,1)
        DU2B = U2G(IOSJM1,K,1) - U2G(ISA,K,1)
        DU2C = U2G(IOSJ  ,K,1) - U2G(ISA,K,1)
        NC = NC+1
        A(NR,NC) = DALPHA(DX1B,DX2B,DX1C,DX2C,DU1B,DU2B,DU1C,DU2C)*SPR
C
C DIP-SLIP
C
```

```
  25 CONTINUE
        IF (IDS.EQ.O) GO TO 30
        DU1B = U1G(IOSJM1,K,NDF) - U1G(ISA,K,NDF)
        DU1C = U1G(IOSJ  ,K,NDF) - U1G(ISA,K,NDF)
        DU2B = U2G(IOSJM1,K,NDF) - U2G(ISA,K,NDF)
        DU2C = U2G(IOSJ  ,K,NDF) - U2G(ISA,K,NDF)
        NC = NC+1
        A(NR,NC) = DALPHA(DX1B,DX2B,DX1C,DX2C,DU1B,DU2B,DU1C,DU2C)*SPR
C
  30 CONTINUE
C
  40 CONTINUE
      IF (NDAT.LT.NAC) GO TO 10
      T(NR,2) = 0.0
      RETURN
C
 600 FORMAT (1H1,20A4//36H MODEL MATRIX FOR TRIANGULATION DATA)
 601 FORMAT (//11H STATION # ,I5)
 602 FORMAT (1H0,5X,19HOBSERVED STATIONS: ,20I5)
 603 FORMAT (1H0,5X,25HANGLE CHANGES (SECONDS): ,10(F6.3,2X))
 604 FORMAT (1H0,5X,30HPREVIOUS VARIANCES (SECS**2): ,10(E10.3,2X))
 605 FORMAT (1H0,5X,30HPRESENT VARIANCES (SECS**2):  ,10(E10.3,2X))
C
      END




      FUNCTION DALPHA (DX1B,DX2B,DX1C,DX2C,DU1B,DU2B,DU1C,DU2C)
C
C COMPUTE ANGLE CHANGE D(<BAC) AT STATION A
C
      DAC2 = DX1C**2+DX2C**2
      DAB2 = DX1B**2+DX2B**2
      T1 = (DX1C*DU2C-DX2C*DU1C)/DAC2
      T2 = (DX1B*DU2B-DX2B*DU1B)/DAB2
      DALPHA = T1-T2
C
      RETURN
      END
      SUBROUTINE TRILAT (XG,U1G,U2G,A,D,T,TITLE,NLLC,NS,MDATA,NF,NDF,
     1                   ISS,IDS,NR)
      DIMENSION XG(2,1),U1G(NS,NF,NDF),U2G(NS,NF,NDF),A(MDATA,1)
      DIMENSION D(1),T(MDATA,2),TITLE(20)
      DIMENSION IOS(20),DL(20),V(20)
      INTEGER LIST(1)/1H*/
C
C COMPUTE MODEL MATRIX FOR TRILATERATION DATA
C
      NDAT = 0
      WRITE (6,600) TITLE
   10 CONTINUE
        READ (5,LIST) ISA,NL
        WRITE (6,601) ISA
        IF (NL.GT.20) CALL ERROR (6,NL,4HTRIL)
        READ (5,LIST) (IOS(K),K = 1,NL)
        WRITE (6,602) (IOS(K),K = 1,NL)
        READ (5,LIST) (DL(K),K = 1,NL)
        WRITE (6,603) (DL(K),K = 1,NL)
        READ (5,LIST) (V(K),K = 1,NL)
        WRITE (6,604) (V(K),K = 1,NL)
        X1A = XG(1,ISA)
        X2A = XG(2,ISA)
        NDAT = NDAT+NL
      DO 40 J = 1,NL
        IOSJ = IOS(J)
        DX1B = XG(1,IOSJ)-X1A
        DX2B = XG(2,IOSJ)-X2A
        NR = NR+1
        T(NR,1) = V(J)*2.
        T(NR,2) = 0.0
        D(NR) = DL(J)
        NC = 0
      DO 30 K = 1,NF
        IF (ISS.EQ.0) GO TO 25
        DU1B = U1G(IOSJ,K,1)-U1G(ISA,K,1)
        DU2B = U2G(IOSJ,K,1)-U2G(ISA,K,1)
        NC = NC+1
        A(NR,NC) = DLINE(DX1B,DX2B,DU1B,DU2B)
C
C DIP-SLIP
C
   25 CONTINUE
        IF (IDS.EQ.0) GO TO 30
        DU1B = U1G(IOSJ,K,NDF)-U1G(ISA,K,NDF)
        DU2B = U2G(IOSJ,K,NDF)-U2G(ISA,K,NDF)
        NC = NC+1
        A(NR,NC) = DLINE(DX1B,DX2B,DU1B,DU2B)
C
   30 CONTINUE
C
   40 CONTINUE
      IF (NDAT.LT.NLLC) GO TO 10
      RETURN
```

```
C
 600 FORMAT (1H1,20A4//36H MODEL MATRIX FOR TRILATERATION DATA)
 601 FORMAT (//11H STATION # ,I5)
 602 FORMAT (1H0,5X,19HOBSERVED STATIONS: ,20I5)
 603 FORMAT (1H0,5X,25HLINE LENGTH CHANGES (M): ,10(F6.3,2X))
 604 FORMAT (1H0,5X,23HDATA VARIANCES (M**2): ,10(E10.3,2X))
C
     END



     FUNCTION DLINE (DX1B,DX2B,DU1B,DU2B)
C
C COMPUTE LINE LENGTH CHANGE
C
     DAB = SQRT(DX1B*DX1B+DX2B*DX2B)
     DLINE = (DX1B*DU1B+DX2B*DU2B)/DAB
     RETURN
     END
```

```
      SUBROUTINE LEVEL(U3G,A,D,T,TITLE,NEC,NS,MDATA,NF,NDF,ISS,IDS,NR)
      DIMENSION U3G(NS,NF,NDF),A(MDATA,1),D(1),T(MDATA,2),TITLE(20)
      INTEGER LIST(1)/1H*/
C
C MODEL MATRIX FOR LEVELLING DATA
C
      NDAT = 0
      WRITE (6,600) TITLE
    5 CONTINUE
        READ (5,LIST) IS,DE,V,VP
        WRITE (6,601) IS,DE,V,VP
        NR = NR+1
        NDAT = NDAT+1
        T(NR,1) = V+VP
        T(NR,2) = 0.0
        D(NR) = DE
        NC = 0
      DO 20 J = 1,NF
        IF (ISS.EQ.0) GO TO 10
        NC = NC+1
        A(NR,NC) = U3G(IS,J,1)
C
   10 CONTINUE
        IF (IDS.EQ.0) GO TO 20
        NC = NC+1
        A(NR,NC) = U3G(IS,J,NDF)
C
   20 CONTINUE
      IF (NDAT.LT.NEC) GO TO 5
      RETURN
C
  600 FORMAT (1H1,20A4//32H MODEL MATRIX FOR LEVELLING DATA)
  601 FORMAT (//11H STATION # ,I5/
     1          1H0,5X,20HELEVATION CHANGE =  ,E10.3/
     2          1H ,5X,20HPREVIOUS VARIANCE = ,E10.3/
     3          1H ,5X,20HPRESENT VARIANCE =  ,E10.3)
C
      END
      SUBROUTINE ZWCOMP (A,D,T,MDIM,M,N)
      DIMENSION A(MDIM,1),D(1),T(MDIM,2)
C
      CALL TRIFA (T(1,1),T(1,2),M,INFO)
      IF (INFO.NE.O) CALL ERROR (6,INFO,O)
      DO 10 J = 1,N
        CALL TRISOL (T(1,1),T(1,2),A(1,J),M)
   10 CONTINUE
      CALL TRISOL (T(1,1),T(1,2),D,M)
      RETURN
      END



      SUBROUTINE TRIFA (D,E,N,INFO)
      DIMENSION D(1),E(1)
C
      INFO = 1
      IF (D(1).LT.O.O) RETURN
      D(1) = SQRT(D(1))
      INFO = O
      DO 10 I = 2,N
        IM1 = I-1
        EIM1 = E(IM1)/D(IM1)
        E(IM1) = EIM1
        S = D(I)-EIM1*EIM1
        INFO = I
        IF (S.LT.O.O) RETURN
        D(I) = SQRT(S)
        INFO = O
   10 CONTINUE
      RETURN
C
      END



      SUBROUTINE TRISOL (D,E,B,N)
      DIMENSION D(1),E(1),B(1)
C
      B(1) = B(1)/D(1)
      IF (N.EQ.1) RETURN
      DO 10 I = 2,N
        IM1 = I-1
        B(I) = (B(I)-E(IM1)*B(IM1))/D(I)
   10 CONTINUE
      RETURN
C
      END
      SUBROUTINE ULS (A,D,V,S,X,G,KR,TOL,MDIM,M,NDIM,N)
      DIMENSION A(MDIM,1),D(1),V(NDIM,1),G(1),X(1),S(1)
      LOGICAL MATU,MATV
C
C SOLVE LEAST SQUARES PROBLEM USING SVD
C
      MATU = .TRUE.
      MATV = .TRUE.
C
      CALL SVD (MDIM,M,NDIM,N,A,S,MATU,A,MATV,V,IERR,G)
      IF (IERR.GT.0) CALL ERROR (8,IERR,4HULS )
C
C DETERMINE EFFECTIVE RANK-
C
      DO 10 I = 1,N
        IF (S(I).LT.TOL) GO TO 15
        KR = I
   10 CONTINUE
C
   15 CONTINUE
      DO 30 K = 1,KR
        SUM = 0.0
      DO 20 I = 1,M
        SUM = SUM+A(I,K)*D(I)
   20 CONTINUE
        G(K) = SUM/S(K)
   30 CONTINUE
C
      DO 50 J = 1,N
        SUM = 0.0
      DO 40 K = 1,KR
        SUM = SUM+V(J,K)*G(K)
   40 CONTINUE
        X(J) = SUM
   50 CONTINUE
      RETURN
C
      END
      SUBROUTINE ULSOUT (U,D,V,S,X,G,TOL,KR,MDIM,M,NDIM,N,ISS,IDS).
      DIMENSION U(MDIM,1),S(N),V(NDIM,1),X(N),D(M),G(N)
C
C PRINT SINGULAR VALUES
C
      WRITE (6,601) (S(I),I = 1,N)
      WRITE (6,602) TOL,KR
C
C COMPUTE SOLUTION VARIANCE
C
      DO 20 I = 1,N
        SUM = 0.0
      DO 10 K = 1,KR
        SUM = SUM+(V(I,K)/S(K))**2
   10 CONTINUE
        G(I) = SUM
   20 CONTINUE
C
C PRINT SOLUTION
C
      IF (ISS.EQ.1 .AND. IDS.EQ.1) GO TO 30
      IF (ISS.EQ.1 .AND. IDS.EQ.0) GO TO 40
      IF (ISS.EQ.0 .AND. IDS.EQ.1) GO TO 50
C
   30 WRITE (6,630)
      DO 35 I = 1,N,2
        SI = SQRT(G(I))
        SIP1 = SQRT(G(I+1))
        WRITE (6,605) X(I),SI,X(I+1),SIP1
   35 CONTINUE
      GO TO 60
C
   40 WRITE (6,640)
      DO 45 I = 1,N
        SI = SQRT(G(I))
        WRITE (6,605) X(I),SI
   45 CONTINUE
      GO TO 60
C
   50 WRITE (6,650)
      DO 55 I = 1,N
        SI = SQRT(G(I))
        WRITE (6,605) X(I),SI
   55 CONTINUE
C
   60 CONTINUE
      WRITE (6,620)
      SST = 0.0
      SSE = 0.0
      DO 100 I = 1,M
        SUM = 0.0
      DO 70 K = 1,KR
        SUM = SUM+U(I,K)**2
        G(K) = U(I,K)*S(K)
   70 CONTINUE
        DIMP = SUM
C
        SUM = 0.0
      DO 80 J = 1,N
        AIJ = 0.0
      DO 75 K = 1,KR
        AIJ = AIJ+G(K)*V(J,K)
   75 CONTINUE
        SUM = SUM+AIJ*X(J)
   80 CONTINUE
        DHAT = SUM
        R = D(I)-DHAT
        WRITE (6,625) D(I),DHAT,R,DIMP
        SST = SST+D(I)**2
        SSE = SSE+R**2
  100 CONTINUE
      SSR = SST-SSE
      CC = SSR/SST
      WRITE (6,626) SSE,SST,SSR,CC
C
      IF (KR.EQ.N) GO TO 150
      WRITE (6,670)
      DO 130 I = 1,N
      DO 120 J = 1,N
        SUM = 0.0
      DO 110 K = 1,KR
        SUM = SUM+V(I,K)*V(J,K)
  110 CONTINUE
        G(J) = SUM
  120 CONTINUE
        WRITE (6,675) (G(J),J = 1,N)
  130 CONTINUE
C
  150 RETURN
C
  601 FORMAT (/16H SINGULAR VALUES//1H ,10G12.5)
  602 FORMAT (33HOTOLERANCE FOR SINGULAR VALUES = ,G12.5/
     1         14H PSEUDORANK = ,I5)
  620 FORMAT (/24H STATISTICS, IMPORTANCES//
     1 1H ,8X,4HDATA,8X,4HDHAT,7X,5HERROR,2X,10HIMPORTANCE/)
  625 FORMAT (1H ,4E12.3)
  626 FORMAT (/36H SUM OF SQUARED ERRORS (SSE)        = ,G12.5/
     1          36H TOTAL SUM OF SQUARES (SST)        = ,G12.5/
     2          36H REDUCTION IN SUM OF SQUARES (SSR) = ,G12.5/
     3          36H CORRELATION COEFFICIENT           = ,G12.5)
  630 FORMAT (/14H SLIP ESTIMATE//
     1 1H ,6X,14HSTRIKE-SLIP(M),9X,11HDIP-SLIP(M)/)
  640 FORMAT (/14H SLIP ESTIMATE//1H ,6X,14HSTRIKE-SLIP(M)/)
  650 FORMAT (/14H SLIP ESTIMATE//1H ,9X,11HDIP-SLIP(M)/)
  605 FORMAT (1H ,2(5X,F7.3,2H+-,F6.3))
  670 FORMAT (18HORESOLUTION MATRIX/)
  675 FORMAT (1H ,16G8.4)
C
      END
      SUBROUTINE SVD (MDIM,M,NDIM,N,A,W,MATU,U,MATV,V,IERR,RV1)
      REAL A(MDIM,N),W(N),U(MDIM,N),V(NDIM,N),RV1(N)
      LOGICAL MATU,MATV
C
C COMPUTE SINGULAR VALUE DECOMPOSITION OF MATRIX A
C
      IERR = 0
      DO 100 I = 1,M
      DO 100 J = 1,N
        U(I,J) = A(I,J)
  100 CONTINUE
C
      G = 0.0
      SCALE = 0.0
      ANORM = 0.0
C
      DO 300 I = 1,N
        L = I+1
        RV1(I) = SCALE*G
        G = 0.0
        SCALE = 0.0
        S = 0.0
        IF (I.GT.M) GO TO 210
C
      DO 120 K = I,M
  120 SCALE = SCALE+ABS(U(K,I))
C
      IF (SCALE.EQ.0.0) GO TO 210
C
      DO 130 K = I,M
        U(K,I) = U(K,I)/SCALE
        S = S+U(K,I)**2
  130 CONTINUE
C
      F = U(I,I)
      G = -SIGN(SQRT(S),F)
        H = F*G-S
        U(I,I) = F-G
        IF (I.EQ.N) GO TO 190
C
      DO 150 J = L,N
        S = 0.0
C
      DO 140 K = I,M
        S = S+U(K,I)*U(K,J)
  140 CONTINUE
C
        F = S/H
C
      DO 150 K = I,M
        U(K,J) = U(K,J)+F*U(K,I)
  150 CONTINUE
C
  190 DO 200 K = I,M
```

```
        U(K,I) = SCALE*U(K,I)
  200 CONTINUE
C
  210   W(I) = SCALE*G
        G = 0.0
        S = 0.0
        SCALE = 0.0
        IF (I.GT.M .OR. I.EQ.N) GO TO 290
C
      DO 220 K = L,N
  220 SCALE = SCALE+ABS(U(I,K))
C
      IF (SCALE.EQ.0.0) GO TO 290
C
      DO 230 K = L,N
        U(I,K) = U(I,K)/SCALE
        S = S+U(I,K)**2
  230 CONTINUE
C
        F = U(I,L)
        G = -SIGN(SQRT(S),F)
        H = F*G-S
        U(I,L) = F-G
C
      DO 240 K = L,N
  240 RV1(K) = U(I,K)/H
C
        IF (I.EQ.M) GO TO 270
C
      DO 260 J = L,M
        S = 0.0
C
      DO 250 K = L,N
        S = S+U(J,K)*U(I,K)
  250 CONTINUE
C
      DO 260 K = L,N
        U(J,K) = U(J,K)+S*RV1(K)
  260 CONTINUE
C
  270 DO 280 K = L,N
        U(I,K) = SCALE*U(I,K)
  280 CONTINUE
C
  290 ANORM = AMAX1(ANORM,ABS(W(I))+ABS(RV1(I)))
C
  300 CONTINUE
C
      IF (.NOT.MATV) GO TO 410
C
      DO 400 II = 1,N
        I = N+1-II
        IF (I.EQ.N) GO TO 390
        IF (G.EQ.0.0) GO TO 360
```

```
C
      DO 320 J = L,N
        V(J,I) = (U(I,J)/U(I,L))/G
  320 CONTINUE
C
      DO 350 J = L,N
        S = 0.0
C
      DO 340 K = L,N
        S = S+U(I,K)*V(K,J)
  340 CONTINUE
C
      DO 350 K = L,N
        V(K,J) = V(K,J)+S*V(K,I)
  350 CONTINUE
C
  360 DO 380 J = L,N
        V(I,J) = 0.0
        V(J,I) = 0.0
  380 CONTINUE
C
  390 V(I,I) = 1.0
        G = RV1(I)
        L = I
  400 CONTINUE
C
  410 IF (.NOT.MATU) GO TO 510
      MN = N
      IF (M.LT.N) MN = M
      DO 500 II = 1,MN
        I = MN+1-II
        L = I+1
        G = W(I)
        IF (I.EQ.N) GO TO 430
C
      DO 420 J = L,N
  420 U(I,J) = 0.0
C
  430 IF (G.EQ.0.0) GO TO 475
        IF (I.EQ.MN) GO TO 460
C
      DO 450 J = L,N
        S = 0.0
      DO 440 K = L,M
  440 S = S+U(K,I)*U(K,J)
C
        F = (S/U(I,I))/G
      DO 450 K = I,M
        U(K,J) = U(K,J)+F*U(K,I)
  450 CONTINUE
C
  460 DO 470 J = I,M
  470 U(J,I) = U(J,I)/G
C
```

```
        GO TO 490
C
  475 DO 480 J = I,M
  480 U(J,I) = 0.0
C
  490   U(I,I) = U(I,I)+1.0
  500 CONTINUE
C
  510 DO 700 KK = 1,N
        K1 = N-KK
        K = K1+1
        ITS = 0
C
  520 DO 530 LL = 1,K
        L1 = K-LL
        L = L1+1
        IF (ABS(RV1(L))+ANORM.EQ.ANORM) GO TO 565
C
        IF (ABS(W(L1))+ANORM.EQ.ANORM) GO TO 540
  530 CONTINUE
C
  540 C = 0.0
      S = 1.0
C
      DO 560 I = L,K
        F = S*RV1(I)
        RV1(I) = C*RV1(I)
        IF (ABS(F)+ANORM.EQ.ANORM) GO TO 565
        G = W(I)
        H = SQRT(F*F+G*G)
        W(I) = H
        C = G/H
        S = -F/H
        IF (.NOT.MATU) GO TO 560
C
      DO 550 J = 1,M
        Y = U(J,L1)
        Z = U(J,I)
        U(J,L1) = Y*C+Z*S
        U(J,I) = -Y*S+Z*C
  550 CONTINUE
C
  560 CONTINUE
C
  565 Z = W(K)
      IF (L.EQ.K) GO TO 650
      IF (ITS.EQ.30) GO TO 1000
      ITS = ITS+1
      X = W(L)
      Y = W(K1)
      G = RV1(K1)
      H = RV1(K)
      F = ((Y-Z)*(Y+Z) + (G-H)*(G+H))/(2.0*H*Y)
      G = SQRT(F*F+1.0)
      F = ((X-Z)*(X+Z)+H*(Y/(F+SIGN(G,F))-H))/X
      C = 1.0
      S = 1.0
C
      DO 600 I1 = L,K1
        I = I1+1
        G = RV1(I)
        Y = W(I)
        H = S*G
        G = C*G
        Z = SQRT(F*F+H*H)
        RV1(I1) = Z
        C = F/Z
        S = H/Z
        F = X*C+G*S
        G = -X*S+G*C
        H = Y*S
        Y = Y*C
        IF (.NOT.MATV) GO TO 575
C
      DO 570 J = 1,N
        X = V(J,I1)
        Z = V(J,I)
        V(J,I1) = X*C+Z*S
        V(J,I) = -X*S+Z*C
 570  CONTINUE
C
 575   Z = SQRT(F*F+H*H)
        W(I1) = Z
        IF (Z.EQ.0.0) GO TO 580
        C = F/Z
        S = H/Z
 580    F = C*G+S*Y
        X = -S*G+C*Y
        IF (.NOT.MATU) GO TO 600
C
      DO 590 J = 1,M
        Y = U(J,I1)
        Z = U(J,I)
        U(J,I1) = Y*C+Z*S
        U(J,I) = -Y*S+Z*C
 590  CONTINUE
C
 600  CONTINUE
C
      RV1(L) = 0.0
      RV1(K) = F
      W(K) = X
      GO TO 520
C
 650  IF (Z.GE.0.0) GO TO 700
      W(K) = -Z
      IF (.NOT.MATV) GO TO 700
C
```

```
      DO 690 J = 1,N
  690 V(J,K) = -V(J,K)
C
  700 CONTINUE
C
C SORT SINGULAR VALUES
C
      KR = MIN0(M,N)
      KRM1 = KR-1
      DO 740 K = 1,KRM1
        T = -1.0
      DO 710 I = K,KR
        IF (W(I).LT.T) GO TO 710
        T = W(I)
        L = I
  710 CONTINUE
        IF (L.EQ.K) GO TO 740
        W(L) = W(K)
        W(K) = T
        IF (.NOT.MATV .OR. K.GT.N) GO TO 725
      DO 720 I = 1,N
        T = V(I,L)
        V(I,L) = V(I,K)
        V(I,K) = T
  720 CONTINUE
  725 CONTINUE
      IF (.NOT.MATU .OR. K.GT.M) GO TO 740
      DO 730 I = 1,M
        T = U(I,L)
        U(I,L) = U(I,K)
        U(I,K) = T
  730 CONTINUE
  740 CONTINUE
C
      GO TO 1001
 1000 IERR = K
 1001 RETURN
      END
      SUBROUTINE CONMAT (B,C,NC,N)
      DIMENSION B(NC,N),C(NC)
      INTEGER LIST(1)/1H*/
C
C READ CONSTRAINTS FOR CONSTRAINED LEAST SQUARES ESTIMATE
C
      WRITE (6,600) NC
      DO 10 I = 1,NC
        READ (5,LIST) (B(I,J),J = 1,N),C(I)
        WRITE (6,605) (B(I,J),J = 1,N),C(I)
   10 CONTINUE
      RETURN
C
  600 FORMAT (/18H CONSTRAINT MATRIX//
     1 25H NUMBER OF CONSTRAINTS = ,I5/)
  605 FORMAT (1H ,11E12.3)
C
      END
      SUBROUTINE CLS (A,D,B,C,X,MDIM,M,N,PDIM,P,JP,RHO,UA,UB,
     1                TOL,SSC,INFO)
      INTEGER P,PDIM,PP1
      DIMENSION A(MDIM,N),B(PDIM,N),D(M),C(P),JP(N),RHO(N),UA(M),
     1          X(N),UB(P)
C
C SOLVE CONSTRAINED LEAST SQUARES PROBLEM
C
      PP1 = P+1
      DO 20 J = 1,N
        JP(J) =J
        SUM = 0.0
      DO 10 I = 1,P
        SUM = SUM+B(I,J)**2
   10 CONTINUE
        RHO(J) = SUM
   20 CONTINUE
C
      INFO = 0
      DO 60 K = 1,P
        RHOMAX = 0.0
        JMAX = K
      DO 30 J = K,N
        IF (RHO(J).LE.RHOMAX) GO TO 30
        RHOMAX = RHO(J)
        JMAX = J
   30 CONTINUE
        IF (JMAX.EQ.K) GO TO 40
        IF (RHOMAX.LE.TOL) GO TO 55
        CALL SWAP (P,B(1,K),B(1,JMAX))
        CALL SWAP (M,A(1,K),A(1,JMAX))
        H = RHO(K)
        RHO(K) = RHO(JMAX)
        RHO(JMAX) = H
        IH = JP(K)
        JP(K) = JP(JMAX)
        JP(JMAX) = IH
C
C APPLY HOUSEHOLDER TRANSFORMATIONS TO B AND C
C
   40 CONTINUE
        CALL HTRAN (B(1,K),UB,ALPHA,BETA,K,P)
        B(K,K) = -ALPHA
        IF (K.EQ.N) GO TO 60
        KP1 = K+1
      DO 50 J = KP1,N
        CALL HTAPP (B(1,J),UB,BETA,K,P)
        RHO(J) = RHO(J)-B(K,J)**2
   50 CONTINUE
        CALL HTAPP (C,UB,BETA,K,P)
        GO TO 60
C
C CONSTRAINTS INCONSISTENT OR REDUNDANT
C
```

```
   55 INFO = K
C
   60 CONTINUE
C
      IF (INFO.NE.O) RETURN
C
      DO 130 I = 1,M
C
        A(I,1) = A(I,1)/B(1,1)
        IF (P.LE.1) GO TO 95
      DO 90 J = 2,P
        SUM = A(I,J)
        JM1 = J-1
      DO 80 K = 1,JM1
        SUM = SUM-A(I,K)*B(K,J)
   80 CONTINUE
        A(I,J) = SUM/B(J,J)
   90 CONTINUE
C
   95 CONTINUE
      DO 110 J = PP1,N
        SUM = A(I,J)
      DO 100 K = 1,P
        SUM = SUM-A(I,K)*B(K,J)
  100 CONTINUE
        A(I,J) = SUM
  110 CONTINUE
C
        SUM = D(I)
      DO 120 K = 1,P
        SUM = SUM-A(I,K)*C(K)
  120 CONTINUE
        D(I) = SUM
C
  130 CONTINUE
C
C APPLY HOUSEHOLDER TRANSFORMATION TO A(N-P)
C
      DO 160 K = PP1,N
        CALL HTRAN (A(1,K),UA,ALPHA,BETA,K-P,M)
        A(K-P,K) = -ALPHA
        CALL HTAPP (D,UA,BETA,K-P,M)
        IF (K.EQ.N) GO TO 160
        KP1 = K+1
      DO 150 J = KP1,N
        CALL HTAPP (A(1,J),UA,BETA,K-P,M)
  150 CONTINUE
  160 CONTINUE
C
C SOLVE FOR X
C
      NMP = N-P
      NMP1 = NMP+1
      SSC = 0.0
      DO 165 I = NMP1,M
        SSC = SSC+D(I)**2
  165 CONTINUE
      DO 190 JB = 1,NMP
        J = NMP+1-JB
        X(J+P) = D(J)/A(J,J+P)
        XJ = X(J+P)
        IF (J.EQ.1) GO TO 175
        JM1 = J-1
      DO 170 I = 1,JM1
        D(I) = D(I)-A(I,J+P)*XJ
  170 CONTINUE
  175 DO 180 I = 1,P
        C(I) = C(I)-B(I,J+P)*XJ
  180 CONTINUE
  190 CONTINUE
C
      DO 200 JB = 1,P
        J = P+1-JB
        X(J) = C(J)/B(J,J)
        IF (J.EQ.1) GO TO 200
        XJ = X(J)
        JM1 = J-1
      DO 195 I = 1,JM1
        C(I) = C(I)-B(I,J)*XJ
  195 CONTINUE
  200 CONTINUE
C
C UNSCRAMBLE SOLUTION VECTOR
C
      DO 210 J = 1,N
        IF (JP(J).EQ.J) GO TO 210
        K = JP(J)
        JP(K) = K
        H = X(J)
        X(J) = X(K)
        X(K) = H
  210 CONTINUE
C
      RETURN
      END
      SUBROUTINE HTRAN (A,U,ALPHA,BETA,IP,M)
      DIMENSION A(M),U(M)
C
C COMPUTE HOUSEHOLDER TRANSFORMATION
C
      ALPHA = 0.0
      DO 10 I = IP,M
        U(I) = A(I)
        ALPHA = ALPHA+U(I)**2
   10 CONTINUE
      ALPHA = SQRT(ALPHA)
      IF (U(IP).LT.0.0) ALPHA = -ALPHA
      U(IP) = U(IP)+ALPHA
      BETA = ALPHA*U(IP)
C
      RETURN
      END
      SUBROUTINE HTAPP (B,U,BETA,IP,M)
      DIMENSION B(M),U(M)
C
C APPLY HOUSEHOLDER TRANSFORMATION TO VECTOR B
C
      IF (BETA.EQ.0.0) RETURN
      GAMMA = 0.0
      DO 10 I = IP,M
        GAMMA = GAMMA+U(I)*B(I)
   10 CONTINUE
      GAMMA = GAMMA/BETA
      DO 20 I = IP,M
        B(I) = B(I)-GAMMA*U(I)
   20 CONTINUE
C
      RETURN
      END
      SUBROUTINE SWAP (N,X,Y)
      DIMENSION X(N),Y(N)
C
C INTERCHANGE CONTENTS OF VECTORS X AND Y
C
      IF (N.LE.0) RETURN
      M = MOD(N,3)
      IF (M.EQ.0) GO TO 20
      DO 10 I = 1,M
        T = X(I)
        X(I) = Y(I)
        Y(I) = T
   10 CONTINUE
      IF (N.LT.3) RETURN
   20 MP1 = M+1
      DO 30 I = MP1,N,3
        T = X(I)
        X(I) = Y(I)
        Y(I) = T
        T = X(I+1)
        X(I+1) = Y(I+1)
        Y(I+1) = T
        T = X(I+2)
        X(I+2) = Y(I+2)
        Y(I+2) = T
   30 CONTINUE
C
      RETURN
      END
      SUBROUTINE CLSOUT (X,SSC,N,ISS,IDS)
      DIMENSION X(N)
C
C PRINT CONSTRAINED SOLUTION
C
      IF (ISS.EQ.1 .AND. IDS.EQ.1) GO TO 30
      IF (ISS.EQ.1 .AND. IDS.EQ.0) GO TO 40
      IF (ISS.EQ.0 .AND. IDS.EQ.1) GO TO 50
C
   30 WRITE (6,630)
      DO 35 I = 1,N,2
        WRITE (6,605) X(I),X(I+1)
   35 CONTINUE
      GO TO 60
C
   40 WRITE (6,640)
      DO 45 I = 1,N
        WRITE (6,605) X(I)
   45 CONTINUE
      GO TO 60
C
   50 WRITE (6,650)
      DO 55 I = 1,N
        WRITE (6,605) X(I)
   55 CONTINUE
C
   60 WRITE (6,620) SSC
      RETURN
C
  630 FORMAT (/14H SLIP ESTIMATE//
     1 1H ,6X,14HSTRIKE-SLIP(M),9X,11HDIP-SLIP(M)/)
  640 FORMAT (/14H SLIP ESTIMATE//1H ,6X,14HSTRIKE-SLIP(M)/)
  650 FORMAT (/14H SLIP ESTIMATE//1H ,9X,11HDIP-SLIP(M)/)
  605 FORMAT (1H ,2(13X,F7.3))
  620 FORMAT (/30H SUM OF SQUARED ERRORS (SSC) = ,G12.5)
C
      END
```