1986-09

Traverse adjustment.

Klangvichit, Supote

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

TRAVERSE ADJUSTMENT

by

Supote Klangvichit

September 1986

Thesis Co-Advisors: Muneendra Kumar
Glen R. Schaefer

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: Unclassified

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.

4 PERFORMING ORGANIZATION REPORT NUMBER(S)

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School

6b OFFICE SYMBOL (If applicable): Code 68

7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School

6c ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000

7b ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000

8a NAME OF FUNDING / SPONSORING ORGANIZATION

8b OFFICE SYMBOL (If applicable)

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS (City, State, and ZIP Code)

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification): TRAVERSE ADJUSTMENT

12 PERSONAL AUTHOR(S): Klangvichit, Supote

13a TYPE OF REPORT: Master's Thesis

13b TIME COVERED: FROM ______ TO ______

14 DATE OF REPORT (Year, Month, Day): 1986 September

15 PAGE COUNT: 87

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Surveying adjustment, least squares, observation equation method, UTM grid coordinates adjustment.

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A traverse is a series of consecutive lines whose lengths and directions have been determined from field measurements. It is chiefly used to determine the mutual location of survey lines and station positions.

Data reduction procedures have been applied to reduce slope distances to ellipsoidal distances to grid distances. Traverse computations were then performed in Universal Transverse Mercator grid coordinates. The computations included adjustment by the method of least squares observation equations. Three resection points adjacent to the traverse line were used to analyse the quality of the results. Adjusted traverse coordinates obtained by various methods were compared. The best results were obtained by the least squares method with selected weights incorporated for each observation.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL: Glen R. Schaefer

22b TELEPHONE (Include Area Code): (408) 646-3131

22c OFFICE SYMBOL: Code 68Sc

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Traverse Adjustment

by

Supote Klangvichit
Lieutenant, Royal Thai Navy
B.S., Royal Thai Naval Academy, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN HYDROGRAPHIC SCIENCES

from the

NAVAL POSTGRADUATE SCHOOL
September 1986

ABSTRACT

A traverse is a series of consecutive lines whose lengths and directions have been determined from field measurements. It is chiefly used to determine the mutual location of survey lines and station positions.

Data reduction procedures have been applied to reduce slope distances to ellipsoidal distances to grid distances. Traverse computations were then performed in Universal Transverse Mercator grid coordinates. The computations included adjustment by the method of approximation and by the method of least squares observation equations. Three resection points adjacent to the traverse line were used to analyse the quality of the results. Adjusted traverse coordinates obtained by various methods were compared. The best results were obtained by the least squares method with selected weights incorporated for each observation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. BACKGROUND

Surveying is the science and art of measurements which are necessary to determine the relative position of points above, on, or beneath the surface of the earth, or to establish the points in a specified position. Surveying operations are conducted not only on land, but also in the oceans and in space. The measurements of surveying consist of distances, horizontal and vertical, and directions. In order to provide a framework of survey points whose horizontal and vertical positions are accurately known, basic horizontal and vertical control surveys must be performed. A primary use of control surveys is for construction of control for a map or chart. The fundamental network of points whose horizontal positions have been accurately determined is called horizontal control [Schmidt, 1978, p. 122].

Horizontal control generally is established either by traverse, triangulation, or trilateration. Which one is to be used depends on the accuracy required and the factor of economy in the selection of survey method. Obviously, there are many degrees of precision possible in any measurement because no surveying measurement is exact. Each of these methods may be the best one to use for a given purpose. Ordinarily, it is a waste of time and money to attain unnecessarily high accuracy. On the other hand, if the measurements are not sufficiently precise, faulty survey results are produced. Therefore, the best surveyor is not the one who makes the most precise measurements, but the one who is able to choose and apply the appropriate measurement with precision requisite to the purpose.

Before 1950 the main framework of a first-order geodetic survey almost always consisted of triangulation, which could be replaced by traverse in cases where the topography made triangulation impracticable. Today, due to the development of electronic distance measuring (EDM) equipment, the first-order control points can be established by means of high accuracy traverse [Allan, 1968, p. 370]. Therefore, the horizontal control is frequently provided by traverse, especially for surveys in area of limited extent, mostly flat and jungle covered. Traverse in such cases is much more economic, convenient, and rapid than other methods and the results are equally accurate.

In order to achieve high precision of horizontal control points when distributed over a large area, first or second-order geodetic surveys are required. These types of survey treat the shape of the earth as ellipsoidal and require using the most accurate distance and angle measurements. Computation of such a survey is relatively complicated, based on long geodetic formulas for computing (with necessary precision) the exact horizontal and vertical position of widely distributed points on the earth's surface.

Disregarding ellipsoid shape, a third-order survey is used over earth areas of limited extent. In this type of a survey, the earth can be considered as flat and all angles are considered to be plane angles. Surveys of this type are used in the densification of geodetic control.

B. OBJECTIVES

As mentioned above, the traverse method has been used worldwide mostly for densification of control stations. However, there are many methods of traverse computations. The main objectives of this thesis are to (1) compute a closed-connecting traverse and adjust station positions by the Approximation Method with the compass (Bowditch) rule and Least Squares Method (adjustment by observation equations only), and (2) compare the results of the two methods.

All computations have been accomplished in the Universal Transverse Mercator (UTM) grid coordinates rather than geodetic coordinates.

II. TRAVERSES

A. GENERAL

The word traverse generally means to pass across. But in surveying, this word means the measurement in a specified sequence of the lengths and directions of lines between points on the earth whose position may be know or unknown. Traverse is the most widely employed method for densification of local horizontal control. Linear measurements are made either by direct observation using a tape or EDM equipment, or by indirect observation using tacheometric methods. The angular measurements are made with theodolite or transit. In this thesis, the only traverse operations considered are for angles measured by theodolite and distances measured by precise EDM equipment or tapes.

Two kinds of traverse exist in surveying, an open and a closed traverse.

1. Open Traverse

An open traverse normally originates at a point of known position and does not return to the starting point nor does it terminate on another point of known position (Figure 2.1). Open traverses should generally not be used because they can not be checked for errors.

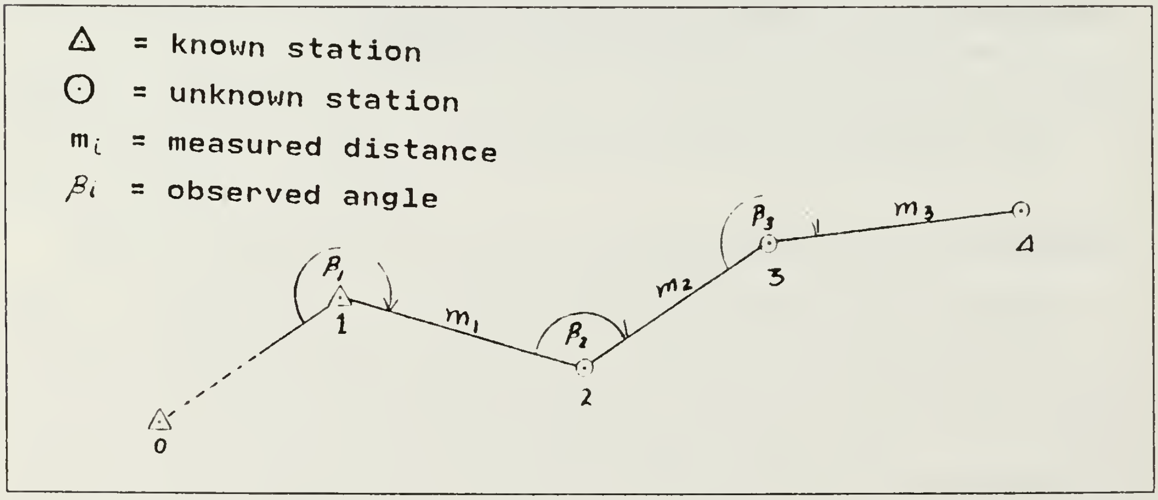

Figure 2.1 Open Traverse.

2. **Closed Traverse**

Closed traverses can further be sub-divided as, a closed-loop and a closed-connecting traverse.

A closed-loop traverse is one that originates and terminates on a single point of known position, thus forming a closed polygon (Figure 2.2). This type of traverse provides an internal check on angles but no check on systematic errors in distance. Also, if the starting azimuth (between stations 0 and 1 in Figure 2.2) has an error, it causes error in orientation of the entire traverse. A closed-loop traverse generally should not be used.

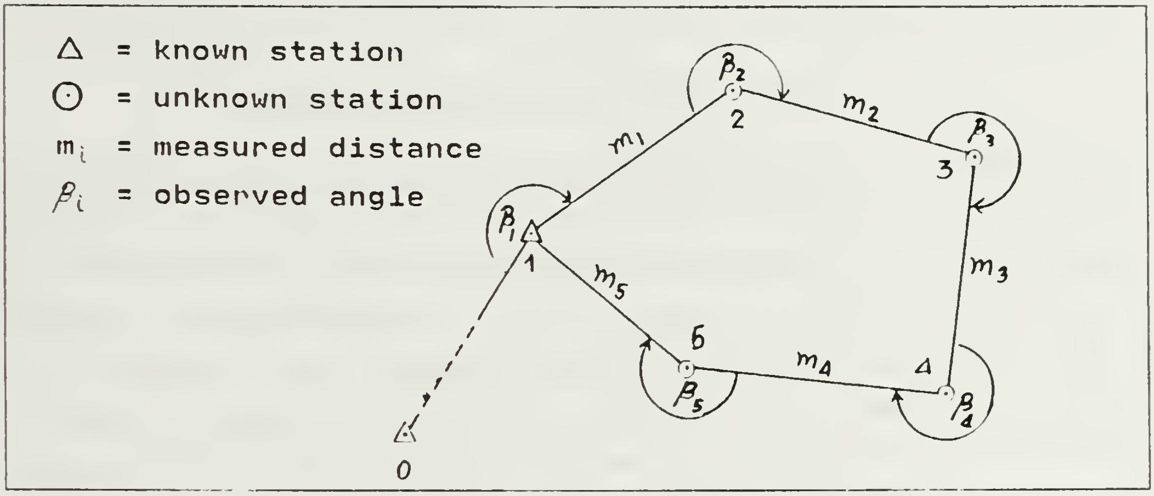

Figure 2.2 Closed-loop Traverse.

A closed-connecting traverse is one that begins and ends on two different points whose horizontal positions have been previously determined by a survey of higher or equal accuracy (Figure 2.3). This type of traverse is preferable to all others, since computational checks are possible to detect systematic errors in both distance and direction.

B. ANGULAR AND DIRECTIONAL MEASUREMENTS

The position of traverse points is determined by the direction and distance from the starting point. To obtain the direction by means of azimuth, the horizontal angle, or plane angle, must be measured in the field. Also, the determination of vertical angles, or zenith distances, may be required to reduce slope distances to horizontal distances.

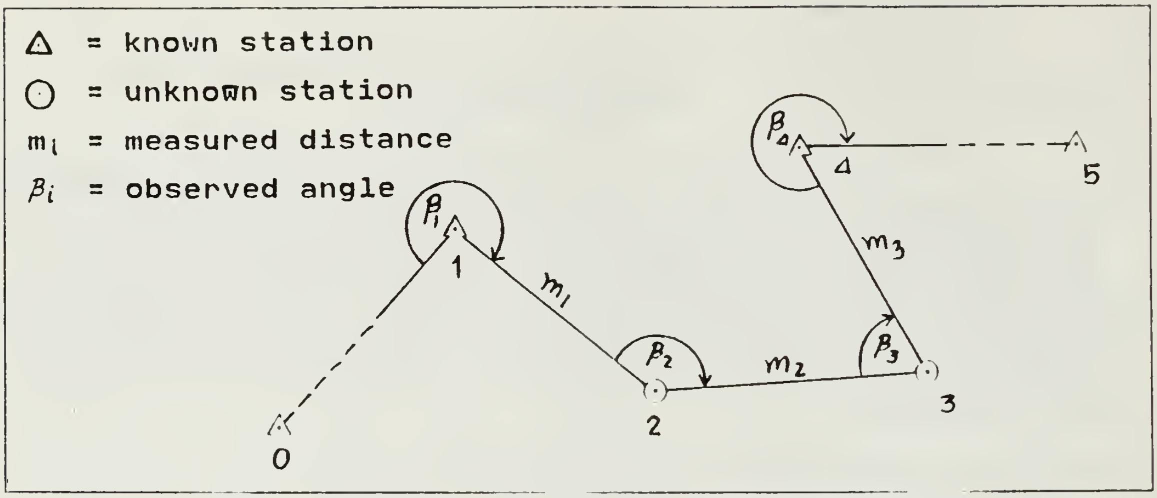

Figure 2.3 Closed-connecting Traverse.

Commonly, both horizontal and vertical angle measurements are accomplished with a transit or theodolite. The theodolite is employed especially for surveys of high precision. Two types of theodolite are a repeating theodolite and direction theodolite. A repeating theodolite reads directly to 20″ or 1′ and may be estimated to one-tenth of the corresponding direct reading. A direction theodolite usually reads directly to 1″ and may be estimated to 0.1″ [Davis et al., 1981, p. 215]. In general, a direction theodolite is more precise than a repeating theodolite and with it, plane angles are computed by subtracting one direction from another.

In all types of traverses, the horizontal angles can be measured by one or more of the following listed angle measurement methods.

1. **Interior Angle**

Interior angle is the angle measured within a closed figure at the intersection of two lines (Figure 2.4).

2. **Deflection Angle**

Deflection angle is the angle measured from the extension of the preceding line to the succeeding line (Figure 2.5). Such angles must be identified as right or left to express whether the angle is turned to the right or to the left from the preceeding line.

3. **Angle To The Right**

Angle to the right is the clockwise angle measured from the preceding line to the succeeding line (Figure 2.6)

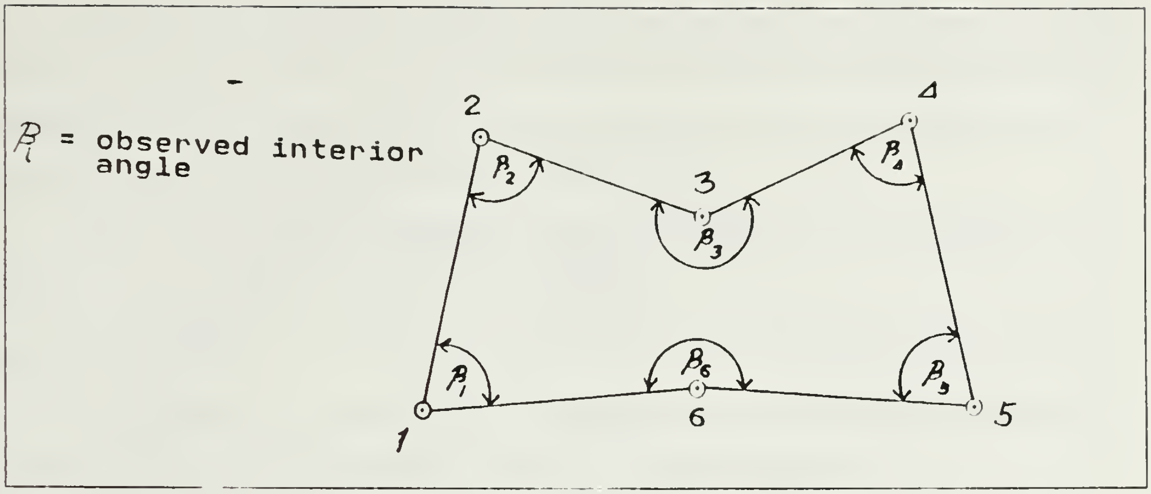

Figure 2.4 Measuring of Interior Angles.

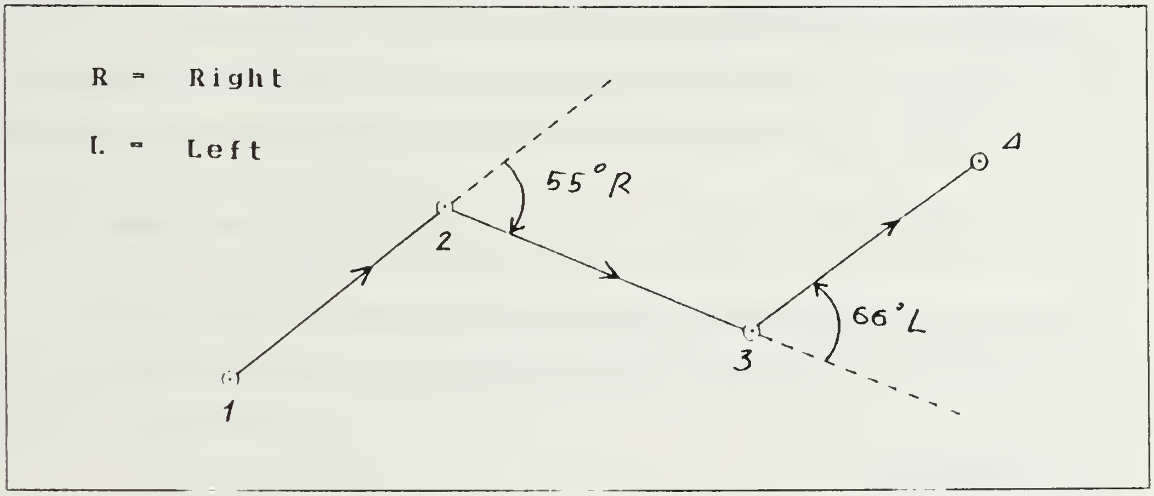

Figure 2.5 Measuring of Deflection Angles.

C. LINEAR MEASUREMENT

Direct linear measurements may be obtained in traversing by pacing, odometer reading, tacheometry (stadia), subtense bar, taping, and EDM. Of these methods, taping and EDM are most commonly used by surveyors. However, EDM equipment has a clear superiority over traditional taping for lines in excess of about 250 meters.

Distances measured using EDM equipment are subject to personal, instrumental, and natural errors. Personal errors include misreading, improper centering of the

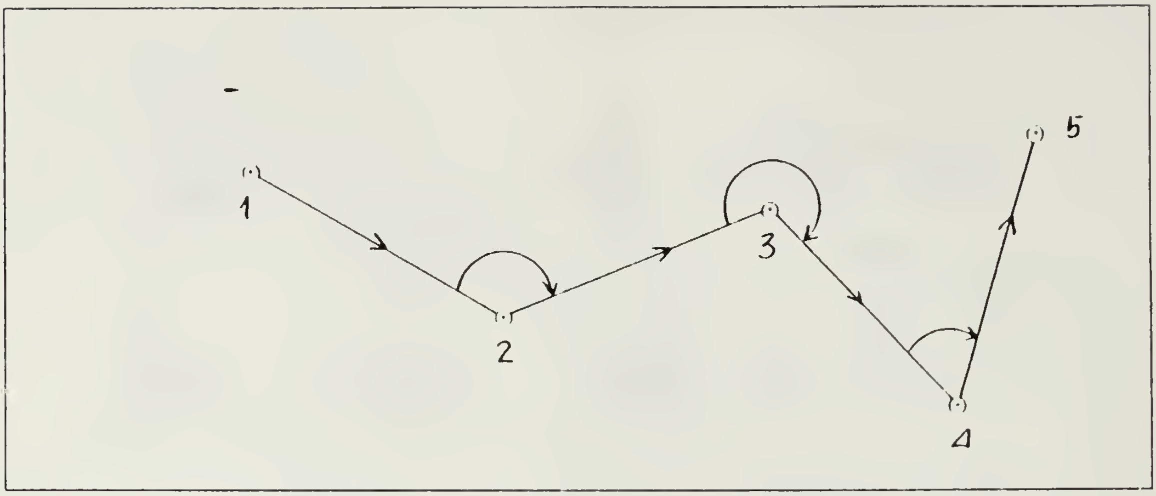

Figure 2.6 Measuring of Angles to the Right.

instrument over the stations, failing to exactly center the null meter, and incorrectly measuring meteorological factors and instrument heights. Instrumental errors, expressed in terms of the accuracy of the instrument specified by the manufacturer, contain two parts. For example, if the accuracy of an instrument is designated as ± (10 ppm + 5 mm), the constant error part is + 5 mm, which is independent of the distance, and the value of the proportional part is 10 ppm (parts per million) which is a function of the distance measured. Constant error is most significant for short distances. For very long distances the constant error becomes negligible, but the proportional part is important. Natural errors such as refraction are cause by changing of atmospheric conditions along the measured line between the end stations.

D. ACCURACY

In survey adjustment, a deviation from the 'true' value is considered as an observational error and the standard error designates the measure of accuracy of the observation. The meaning of an accuracy is then the degree of conformity or closeness of a measurement to true value.

The quality of traverse operations is dependent upon the accuracy of angular and linear measurements; thus, in checking the accuracy of traverse two quantities are considered, the angular misclosure and the linear misclosure. Although the positional closure (relative accuracy) is an indication of the overall quality of the traverse and is used for traverse classification, it does not yield information on the precision of point location determined in a traverse [Davis et al., p. 332].

The inherent weakness in a traverse is that the deviation of each measured line is determined by a single series of angular observations, further, any error in any angle will affect not only the adjoining line but all subsequent lines to a greater or lesser extent according to their lengths [Allan, 1968, p. 371].

Angular misclosure is expressed by standard error of the measured angle times the square root of the number of measured angles.

Linear misclosure is commonly expressed as a ratio of total misclosure to total length of traverse.

Finally, some of the most significant features of traverse classification by the U.S. Federal Geodetic Control Committee (1984) are shown in Table I.

E. ADJUSTMENTS

Adjustment of a traverse is carried out to ensure consistency within the known positions of the originating and terminating stations and to remove inconsistencies in observed angles and distances to compensate for random errors. For a more precise extended traverse, adjustments made on the basis of least squares are preferred. But a traverse of limited extent can be adjusted by simple approximation methods.

1. Approximation Methods

There are four methods for traverse adjustment by approximation.

a. Arbitary Method

This method does not conform to a fixed rule. Rather, the linear error of closure is distributed arbitarily according to arbitary preference of the surveyor.

b. Transit Rule

Transit rule is better for adjustment of the traverse where the angles are measured with greater accuracy than distances, and is valid only when the traverse lines are parallel with the grid system used for the traverse computations. Corrections are made by the following rules: the correction in latitude for any station is equal to the multiple of latitude in that section and total closure in latitude divided by the sum of all latitudes in traverse, and the correction in departure is equal to the multiple of departure in that section and total closure in departure divided by the sum of all departures in the traverse [Davis et al., 1981, p. 323].

c. Compass or Bowditch Rule

This method is suitable for adjustment of the traverse where the angles and distances are measured with equal precision and uses the following rules: the correction

TABLE I

TRAVERSE CLASSIFICATION

Order Class	First	Second I	Second II	Third I	Third II
Station spacing not less than (km)	10	4	2	0.5	0.5
Minimum number of network control points	4	3	2	2	2
Theodolite least count	0.2"	1.0"	1.0"	1.0"	1.0"
Direction number of positions	16	8 or 12	6 or 8	4	2
Standard deviation of mean not to exceed	0.4"	0.5"	0.8"	1.2"	2.0"
Rejection limit from the mean	4"	5"	5"	5"	5"
Azimuth closure at azimuth check point (second of arc)	$1.7\sqrt{N}$	$30\sqrt{N}$	$4.5\sqrt{N}$	$10.0\sqrt{N}$	$12\sqrt{N}$
Position closure after azimuth adjustment*	$0.04\sqrt{K}$ or 1:100000	$0.08\sqrt{K}$ or 1:50000	$0.2\sqrt{K}$ or 1:20000	$0.4\sqrt{K}$ or 1:10000	$0.8\sqrt{K}$ or 1:5000

N = route distance in km
K = number of segments

*The expression containing the square root is designed for longer lines where higher proportional accuracy is required and the results are in meter.
The closure (e.g., 1:50,000) is relative error of closure.

in latitude for any station is equal to the multiple of the length in that section and total closure in latitude divided by the total length in the traverse, and the correction in departure is equal to the multiple of the length in that section and total closure in departure divided by the the total length in the traverse [Schmidt, 1978, p. 150].

d. Crandall Method

Crandall method is a rather complicated procedure which is more rigorous than either the compass or transit rules but suitable for adjusting traverses where the linear measurements contain larger random errors than the angular measurement. In this method, the angular error of closure is first distributed in equal portions to all of the measured angles, then linear measurements are adjusted by using a weighted least squares procedure [Brinker, 1977, p. 228].

2. **Adjustment by Least Squares Method**

The method of least squares adjustment is based upon the theory of probability; it simultaneously adjusts the angular and linear measurements to make the sum of the square of the residuals (error) a minimum [Brinker, 1977, p. 228]. This method can be used for any type of traverse. Because of the availibility of fast computing devices at the present time, the least squares method is being widely used. Further, the least squares solution has the advantage that it determines, quite objectively, a unique solution for a given adjustment problem [Clark, 1973, p. 121].

In general, adjustment is needed whenever there are redundant observations (more observations than are necessary to solve the required unknowns). As an example, to determine the angles of a plane triangle, only two observed angles are required because the third angle can be obtained by subtraction from 180°. When three angles are observed, the sum of them will not be equal 180° due to error in measurements. Therefore, these three angles should be adjusted to fit the functional model.

The redundancy may be interpreted to mean that among n observations there exist r conditions or functions ($n > r$) that must satisfy the model.

Let n be a number of observations and n_0 the minimum number of observations to find the uniquely solution in the model, then redundancy or degree of freedom in the statistic, r, is

$$r = n - n_0 \tag{2.1}$$

Consequently, there are r redundant observations, which can also give a solution. To detect the error in each observation, the best estimated or the most probable value must be defined because the true value is not known exactly. Statistically, the best estimated value of a group of repeated observation is the average (arithmetic mean).

Once the difference between observed value (X_o) and estimated value (X_e) is determined, the adjusted value (X_a) is obtained through a least squares solution, then the residual (v) can be expressed as

$$v = X_a - X_o \quad (2.2)$$

and

$$X_a = X_e + dx \quad (2.3)$$

where dx is the correction to estimated value to obtain the adjusted value.

The least squares adjustment method is based upon the criterion of the sum of the squares of the observational residuals must be minimum.

When observations are considered as uncorrelated and of equal precision (with identity weight matrix), the least squares condition can be expressed as

$$\Phi = v_1^2 + v_2^2 + \ldots + v_n^2 = \sum v_i^2 = \text{minimum} \quad (2.4)$$

or in matrix form

$$\Phi = V^T V = \text{minimum} \quad (2.5)$$

where V is the vector of residuals.

In uncorrelated observations with unequal precision, such as distances and angles [Mikhail, 1981, p. 68], the Equations 2.4 and 2.5 become

$$\Phi = w_1 v_1^2 + w_2 v_2^2 + \ldots + w_n v_n^2 = \sum w_i v_i^2 = \text{minimum} \quad (2.6)$$

or in matrix form

$$\Phi = V^T W V = \text{minimum} \quad (2.7)$$

where w_i is the i^{th} element of the diagonal weight matrix W and v_i is the residual associated with the corresponding i^{th} observation.

Generally, the relative weights are inversely proportional to variance, thus the weight matrix is the inverse of cofactor matrix, Q (when it is square and nonsingular) and defined as

$$W = Q^{-1} \quad (2.8)$$

where the elements of cofactor matrix Q are

$$q_{ii} = \sigma_i^2/\sigma_0^2 \tag{2.9}$$

and

$$q_{ij} = \sigma_{ij}/\sigma_0^2 \tag{2.10}$$

where σ_i^2 the variance of the i^{th} observation, σ_{ij} is the covariance between the i^{th} and j^{th} observations, and σ_0^2 is variance of unit weight [Mikhail, 1981, p. 67].

For the case of uncorrelated weight observations, the cofactor matrice will be diagonal with all off-diagonal elements being equal to zero, thus the diagonal elements of weight matrix in this case are

$$w_{ii} = 1/q_{ii} = \sigma_0^2/\sigma_{ii}^2 \tag{2.11}$$

Generally, there are two types of equation in least squares adjustment: condition and observation equations. Condition equations include one or more observations but observation equations include parameters and only one observation.

The condition as well as the observation equations involved in an adjustment problem can be linear or nonlinear. However, least squares treatments are generally performed with linear functions, since it is rather difficult and often impractical to solve nonlinear models [Mikhail, 1978, p. 108]. Consequently, whenever the equations in the model are originally nonlinear, they have to be linearized. A method of series expansion, especially Taylor's series, is often used to obtain linear equations. Only the zero and first-order terms are used and all other higher-order terms are neglected. Thus, the linearized form for the general case of m functions of n variables becomes

$$F = F^0 + J_{mn}\Delta x \tag{2.12}$$

where F^0 is the zero-order terms, when $x = x^0$, and J_{mn} is a Jacobian matrix of coefficients of first order of n variable (Appendix A).

The choice between the observation equations (indirect observation) and condition equations (observation only) techniques will depend mainly on the mathematical model of the problem to be solved. However, the final answers are always the same.

a. Adjustment by Observation Equations

The method of adjustment by observation equations is performed with both observations and parameters. Therefore, the number of equations is equal to the number of observations. Using the example at the beginning of this section, if three measured angles and their residuals in a plane triangle are assummed to be α, β, γ, v_1, v_2, and v_3, respectively, and the adjusted value of those angles is x_{a1}, x_{a2}, and x_{a3} then the three condition equations in the normal form (zero at the right-hand side) are

$$(\alpha + v_1) - x_{a1} = 0$$
$$(\beta + v_2) - x_{a2} = 0$$
$$(\gamma + v_3) - x_{a3} = 0$$

by using Equation 2.3 for adjusted values

$$v_1 - dx_1 = x_{e1} - \alpha$$
$$v_2 - dx_2 = x_{e2} - \beta$$
$$v_3 - dx_3 = x_{e3} - \gamma$$

letting

$$V = \begin{vmatrix} v_1 \\ v_2 \\ v_3 \end{vmatrix}, \quad B = \begin{vmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{vmatrix}, \quad X = \begin{vmatrix} dx_1 \\ dx_2 \\ dx_3 \end{vmatrix}, \quad f = \begin{vmatrix} x_{e1} - \alpha \\ x_{e2} - \beta \\ x_{e3} - \gamma \end{vmatrix}$$

then, expressed in matrix notation as

$V_{31} + B_{33}X_{31} = F_{31}$ and the general form of the adjustment of observation equations become

$$V_{n1} + B_{nu}X_{u1} = F_{n1} \qquad (2.13)$$

where V is an n by 1 vector of residuals; B is an n by u matrix of coefficients; X is an u by 1 unknown vector in which u is the number of parameters; and f is an n by 1 vector and equals P - X_0, in which P is evaluated using the estimated value X_e.

b. Adjustment by Condition Equations

The method of adjustment of condition equations only has no parameters included in the condition equations. Thus, the number of condition equations is equal to the number of redundancies. From the example above, $n = 3$, $n_0 = 2$, and $r = 1$, which is the number of condition equations being set. In this case, because the sum of interior angles must equal 180°, the single condition equation can be expressed as

$$\alpha + \beta + \gamma + v_1 + v_2 + v_3 = 180°$$

$v_1 + v_2 + v_3 = 180° - \alpha - \beta - \gamma$

$$A = | \ 1, 1, 1 \ |, \quad V = \begin{vmatrix} v_1 \\ v_2 \\ v_3 \end{vmatrix}, \quad F = |180-\alpha-\beta-\gamma|$$

Then, the general form of this technique is

$$A_{rn}V_{nr} = F_{r1} \tag{2.14}$$

When the conditions are originally linear, the vector F is usually written in terms of the given observations as

$$F_{r1} = P_{r1} - A_{rn}X_{o,n1} \tag{2.15}$$

where A is the coefficient matrix V., P is a constant term (see Section II.E.a), X_o is observed values, r is redundancy, and n is a number of observation [Mikhail, 1976, p. 173].

III. DATA ACQUISITION AND REDUCTION

A. DATA ACQUISITION

Taverse data used in this thesis were obtained from field work accomplished from 25 September thru 9 October 1972 by CAPT Glen R. Schaefer, NOAA Corps, and Mr. Jim D. Shea, National Ocean Service (NOS), utilizing traverse methods in Pinellas County, Florida. Only the first 15 of 40 occupied stations and three intersection points will be used for analysis (Figure 3.1). The two pairs of known stations for this closed-connecting traverse are shown in Table II.

The known stations were observed by the US Coast and Geodetic Survey (now NOS) and adjusted by the National Geodetic Survey (NGS). Station Turtle 2 is of first-order and the other three stations are of third-order.

The horizontal angles measured by the method of angles to the right (Table III) were turned with a Wild T-2 theodolite, according to the specifications of third-order class I traverse, by starting at station Tomlinson and using Egmont Key Lt. House for a backsight. The traverse was closed on Turtle 2 with a check azimuth to Madeira Beach Tank.

The slope distances were measured in the field with a Model 76 Geodimeter in feet and corrected for temperature and pressure. Distance measurement by the Model 76 Geodimeter are reported to have an accuracy in the temperature range of -20°C to +50°C of ±(1 ppm + 1 cm) with a resolution of 1 mm. [Schmidt, 1978, p. 116]. all observed distances were converted to meters and reduced to horizontal by the procedure given later in this chapter. Finally, geodetic distances are reduced to grid distances by applying the scale factor correction (Table IV).

B. DATA REDUCTION

1. Computation of Ellipsoidal Distances

For the requirement of high precision in the first-order traverse, the measured distances obtained by EDM equipment are first corrected for atmospheric conditions and then reduced to the ellipsoid (Figure 3.2) by the equations

$$S = 2\, R_\alpha \, \mathrm{Sin}^{-1}(S_0/2R_\alpha) \tag{3.1}$$

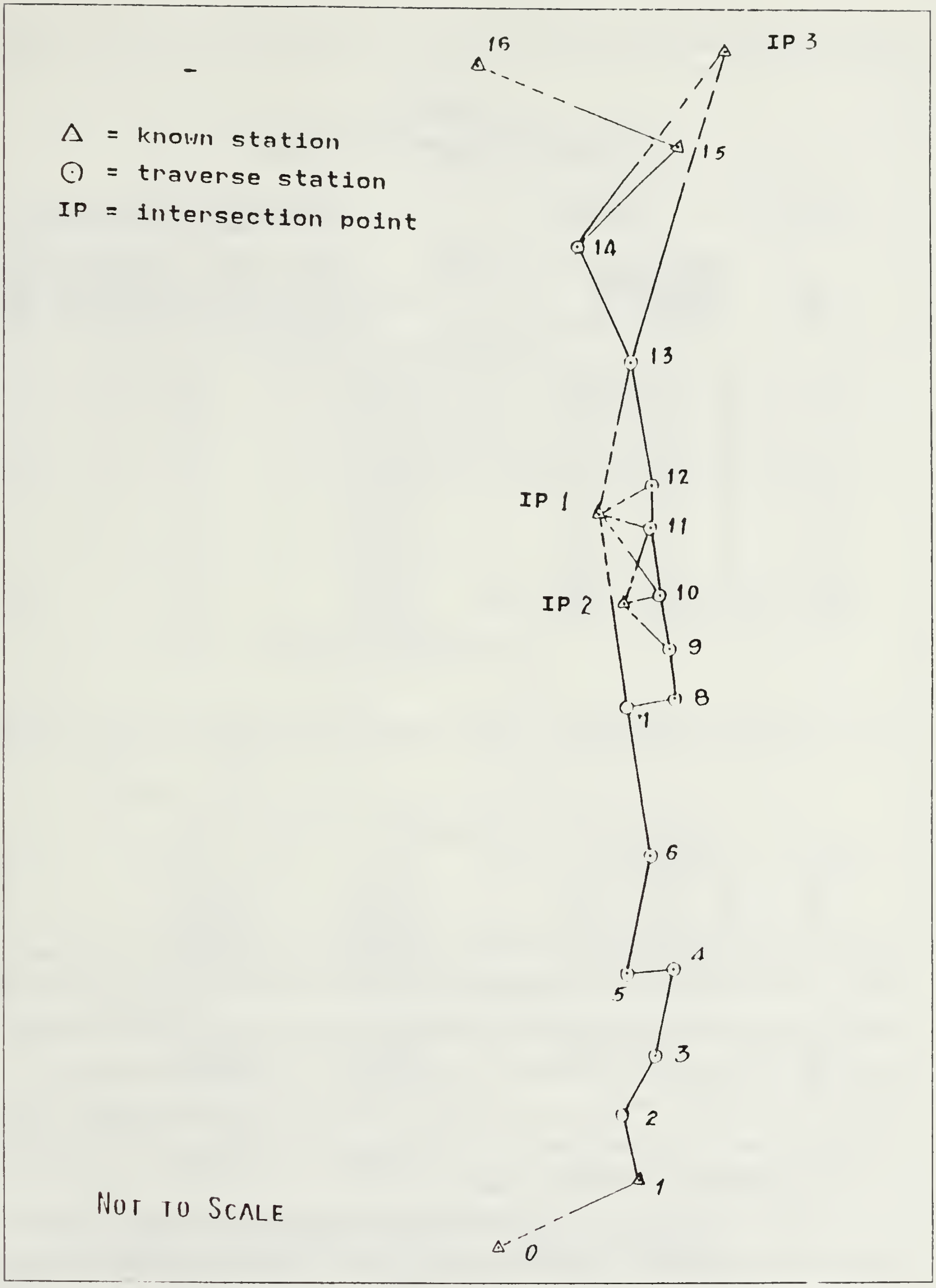

Figure 3.1 Traverse Layout.

TABLE II

DATA OF KNOWN STATIONS

Station No.	Station Name	Grid coordinate (m.) X/Easting	Grid coordinate (m.) Y/Northing
0	Egmont Key Lt. House	326214.833	3054014.779
1	Tomlinson	329400.420	3063485.483
15	Turtle 2	325348.292	3076179.297
16	Madeira Beach Tank	322698.706	3076362.405

$$S_0 = \{ [m^2-(h_2-h_1)] / (1+h_1/R_\alpha)(1+h_2/R_\alpha) \}^{1/2} \tag{3.2}$$

$$1/R_\alpha = \{(\text{Cos}^2\alpha)/M\} + \{(\text{Sin}^2\alpha)/N\} \tag{3.3}$$

$$M = a(1-e^2) / (1-e^2\text{Sin}^2\varphi)^{3/2} \tag{3.4}$$

$$N = a / (1 - e^2\text{Sin}^2\varphi)^{1/2} \tag{3.5}$$

where m is the measured distance, S is the ellipsoidal distance, S_0 is chord distance, h_1 and h_2 are the heights above the ellipsoid at each end of the line, R_α is the radius of curvature of the chord distance m at an azimuth α, α is geodetic azimuth clockwise from north, M is radius of curvature in the plane of the meridian, N is radius of curvature in the prime vertical, a is a semimajor axis of the reference ellipsoid, e is its eccentricity, and φ is the mean latitude of that line. [Torge, 1980, pp. 48,49,50,51,180]

2. **Computation of Geodetic Distances**

For a lower-order traverse, the measured distance can be reduced to mean sea level (MSL) or geoid only. Because the error of only 1 ppm in length results from an error of 6 m in separation between MSL and the spheroid [Bomford, 1980, pp. 42, 342, 345].

TABLE III

OBSERVED HORIZONTAL ANGLES

A) For Traverse

At Stn. No.	Stn. Name	Angles D	Angles M	Angles S	σ (±)
0	Egmont Key Lt. House				
1	Tomlinson	105	21	25	5"
2	TN-01	243	39	18	5"
3	TN-02	168	10	15	5"
4	TN-03	59	55	56	5"
5	Ruscue	291	28	29	5"
6	RE-01	160	43	42	5"
7	RE-02	269	55	50	5"
8	RE-03	92	43	29	5"
9	RE-04	178	22	51	5"
10	RE-05	182	31	19	5"
11	RE-06	196	54	42	5"
12	RE-08	168	51	46	5"
13	RE-09	161	06	42	5"
14	RE-10	236	58	14	5"
15	Turtle 2	78	37	24	5"
16	Madeira Beach Tank				

B) For Intersection Points (IP)

IP #1 *

Back Stn.	At Traverse Stn.	Angles D	Angles M	Angles S	σ (±)
RE-01	RE-02	196	23	46	5"
RE-04	RE-05	184	34	22	5"
RE-05	RE-06	186	29	15	5"
RE-06	RE-08	34	43	31	5"
RE-08	RE-09	2	41	22	5"

IP #2 **

Back Stn.	At Traverse Stn.	Angles D	Angles M	Angles S	σ (±)
RE-03	RE-04	175	35	01	5"
RE-04	RE-05	97	12	06	5"
RE-05	RE-06	2	32	22	5"

IP #3 ***

Back Stn.	At Traverse Stn.	Angles D	Angles M	Angles S	σ (±)
RE-08	RE-09	194	01	22	5"
RE-09	RE-10	232	11	34	5"

* St. Petersburg BCH CO Tank
** St. Petersburg BCH St. Johns CH Tower
*** Bay Pines Veterans Administration Hosp.

TABLE IV

MEASURED DISTANCES AND GRID DISTANCES

At Stn. To Stn.	Measured Distances ft	m	Grid Distances ft	m	σ (±m)
1	2300.98	701.340	2300.91	701.318	0.017
2	2605.41	794.131	2605.28	794.090	0.018
3	4452.48	1357.119	4452.31	1357.068	0.024
4	709.56	216.274	709.54	216.267	0.012
5	5050.37	1539.356	5050.21	1539.307	0.025
6	6620.54	2017.945	6620.33	2017.880	0.030
7	1655.17	504.497	1655.12	504.481	0.015
8	1605.85	489.464	1605.80	489.448	0.015
9	2114.68	644.556	2114.61	644.535	0.016
10	2362.13	719.979	2362.05	719.956	0.017
11	1360.16	414.578	1360.10	414.561	0.014
12	5060.98	1542.590	5060.41	1542.415	0.025
13	6172.26	1881.309	6171.67	1881.128	0.029
14	6664.72	2031.411	6664.51	2031.346	0.030
15					

m = measured distance

S_0 = chord distance

S = arc distance on ellipsoid

h_1 = elevation above ellipsoid at station 1

h_2 = elevation above ellipsoid at station 2

R_α = radius earth curvature along measured line

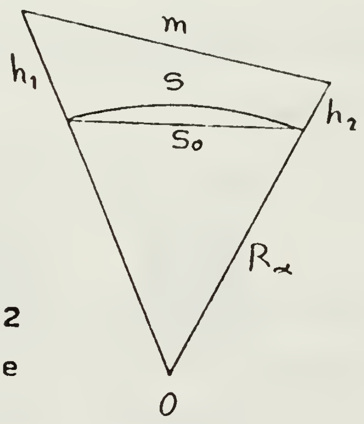

Figure 3.2 Ellipsoidal Distance.

However, the process of reduction requires three steps: (1) correct slope distances to horizontal distances, (2) reduce horizontal distance to geodetic distances, and (3) change the geodetic distances to grid distances.

The slope distance data used included correction for atmospheric conditions.

In a plane survey, such as a traverse, there are two considerations for the reduction of slope distances to horizontal distances, a short slope distance and a long slope distance.

a. Slope Reduction for Short Distances

Short slope distances ($\leq$ 2 mi or 3.3 km) measured by using EDM instruments separate from the theodolite, can be reduced to horizontal with a simple trigonometry process as

$$D = m \cos \theta \qquad (3.6)$$

where D is the horizontal distance, m is the slope distance, and θ is the vertical angle (Figure 3.3).

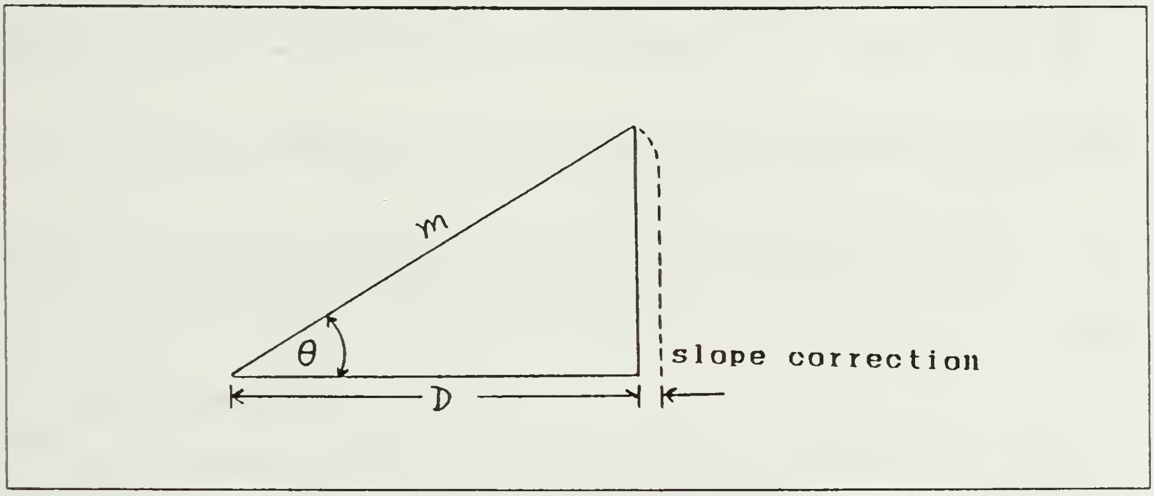

Figure 3.3 Slope Reduction for Typical Triangle.

The horizontal distance also may be determined by using the difference in elevation between the two ends of the line. The horizontal distance is

$$D = (m^2 - d^2)^{1/2} \qquad (3.7)$$

in which d is the difference in elevation between the two end points. The heights of the EDM instrument and reflector above the survey mark must be observed, and d becomes

$$d = (E_1 + I_1) - (E_2 + O_2) \qquad (3.8)$$

where E_1 and E_2 are the elevation at each end of the line respectively, I_1 is the height of the instrument, and O_2 is the height of the reflector. Then, expanding the right side of equation 3.7 with the binomial theorem yields

$$D = m-(d^2/2S + d^4/8S^3 + ...) \qquad (3.9)$$

The quantity in the parenthesis is designated slope correction. For moderate slope the first term is usually adequate. When the slope distances and vertical angles are obtained by separated EDM equipment from theodolite, the correction of the vertical angle must be determined. The corrected vertical angle θ_T is

$$\theta_T = \theta_0 + \Delta\theta \qquad (3.10)$$

where θ_0 is an observed vertical angle by theodolite, and $\Delta\theta$ is

$$\Delta\theta = H \text{ Cos } \theta_0 \,/\, S \text{ Sin } 1'' \qquad (3.11)$$

and

$$H = (H_r - H_t) - (H_i - H_e) \qquad (3.12)$$

where H_r is the height of the reflector, H_t is the height of the target, H_i is the height of the EDM, and H_e is the height of the theodolite [Davis et al., 1981, pp. 103-104]. Equations 3.11 and 3.12 are not needed when the slope distances and vertical angles are obtained simultaneously by using an EDM transmitter built into a theodolite.

b. Slope Reduction for Long Distances

Slope reduction for long distances (> 2 mi or 3.3 km) involves using vertical angles affected by curvature and refraction. By assuming a mean radius for the earth of 3959 mi or 6371 km, then the curvature correction (C), expressed as an angle in seconds, is 4.935″ per 1000 ft or 16.19″ per km and the horizontal distance, D, is

$$D = m \text{ Sin}(90° - \theta - C) \,/\, \text{Sin}(90° + C) \qquad (3.13)$$

for

$$\theta = (\gamma+\beta)/2 \tag{3.14}$$

where γ and β are the vertical angles at each end of the measured line [Davis et al., pp. 106-107].

When a single vertical angle (γ) is observed, θ is the corrected vertical angle for combined results of curvature and refraction (C&R), then, θ is γ + (C&R)″. The C&R correction is 4.244″ per 1000 ft or 13.925″ per 1000 m. The correction of C&R will be positive when the vertical angle is an elevation angle and negative in the case of a depression angle [Davis et al, 1981, p, 108].

3. Reduction of Horizontal Distances to Geodetic Distances

The horizontal distance at same elevation above the geoid, must be reduced to a geodetic distance. This can be done by the equation

$$D' = (R)(D) / (R+E) \tag{3.15}$$

where D′ is the geodetic distance, R is the mean radius of the earth′s surface at that section, D is the horizontal distance at elevation E above the geoid [Davis et al., 1981, p. 107].

C. GRID DISTANCES

The traverse computation, based on the UTM grid coordinate system, requires the reduction of geodetic distances to the plane of the projection by applying the projection scale factor and grid scale constant. Scale factor can be obtained from a graph or from a rigorous formula [Department of the Army Technical Manual, 1958, pp. 3,4,9,17]; i.e.,

$$k = k_0 [\, 1 + (XVIII)\, q^2 + 0.00003\, q^4 \,] \tag{3.16}$$

where k is the scale factor at scale working on the projection, k_0 is the central scale factor which is an arbitary reduction applied to all geodetic lengths to reduce the maximum scale distortion of the projection (for UTM, k_0 = 0.9996), and values for q and (XVIII) are obtained by the formulas which shown in Table V.

TABLE V

SPECIFICATION OF PARAMETERS

$$XVIII = \frac{1 + e' \, Cos^2\varphi}{2 \, \nu^2} \cdot \frac{1}{k_0^{\,2}} \cdot 10^{12}$$

$$\nu = \frac{a}{(1 - e^2 \, Sin^2\varphi)^{1/2}}$$

e^2	=	(ecentricity)^2 = $(a^2 - b^2) / a^2$
$e^{2'}$	=	$e^2 / (1 - e^2)$
q	=	0.000001 E'
E	=	grid easting = E' + 500,000 when point is east of central meridian, 500,000 - E' when point is west of central meridian
ν	=	radius of curvature in the prime vertical
a	=	semi-major axis of the ellipsoid
b	=	semi-minor axis of the ellipsoid
φ	=	latitude

IV. TRAVERSE COMPUTATION AND ADJUSTMENT

Linear measurements and angles must be checked by computation to determine the position of traverse stations and whether the traverse meets required precision. Traverse station coordinates are usually expressed in terms of geographical coordinates or rectangular coordinates such as those based on an UTM projection. Traverse computations are usually done in rectangular coordinates because of the ease of computation. In this thesis, only closed-connecting traverse computations in UTM coordinates were used. When specifications were satisfied, the traverse was adjusted for perfect geometric consistency among angles and lengths.

A. DATA PROCESSING

1. Set up of data base

Two files on IBM 370/3033AP main frame computer at NPS were established.

2. Modification of an Existing Program

The Fortran program TRAVADJ, originally written by LCDR Saman Aumchantr, RTN (1984) in Watfiv language for computing and adjusting the traverse station coordinates, was modified to be able to handle the distances reduction processes.

3. Writing a New Program

A Fortran program INDTRA was written for computing and simultaneously adjusting traverse station and intersection point coordinates by least squares observation equations method.

B. COMPUTATION OF STARTING AND CLOSING AZIMUTHS

The directions of lines by means of azimuth are used for traverse computation, because sines and cosines of azimuth angles automatically provide correct algebraic signs for latitudes and departures.

The terms latitude and departure are widely used in rectangular coordinate calculations of surveying. The latitude of a line is its projection on the reference meridian, which differs from geographic latitude. The departure of a line is its projection on the east-west line perpendicular to the reference meridian. In traverse calculations, north latitudes and east departures are considered plus; south latitudes and west departure, minus.

Latitudes are also sometimes termed 'northings' and 'Y differences' (ΔY); departures are similarly called 'eastings' and 'X differences' (ΔX).

Because the closure angle of traverse can be checked by azimuth of each consecutive line, starting and closing azimuths have to be determined in the first step of computation and azimuths can be computed from a pair of known coordinate station positions at the two ends of the traverse (Figure 3.1)

The azimuth of the line from A to B, α_{AB}, measured clockwise from north, is determined by the equation

$$\alpha_{AB} = \mathrm{Tan}^{-1}(\Delta X / \Delta Y) \qquad (4.1)$$

for

$$\Delta X = X_B - X_A \qquad (4.2)$$

and

$$\Delta Y = Y_B - Y_A \qquad (4.3)$$

where X_A and X_B are the grid easting coordinates, and Y_A and Y_B are the grid northing coordinates of stations A and B, respectively (Figure 4.1). The quadrant of the azimuth of line AB, α_{AB}, is dependent upon the sign of ΔX and ΔY (Table VI). The back azimuth α_{BA} (the azimuth from B to A) is obtained by adding 180° to the forward azimuth α_{AB}.

The length of the line AB (denoted as S_{AB} or S) can be determined by the Pythagorean theorem or by one of the trigonometric relationships

$$S = \Delta X / \mathrm{Sin}\,\alpha \qquad (4.4)$$

or

$$S = \Delta Y / \mathrm{Cos}\,\alpha \qquad (4.5)$$

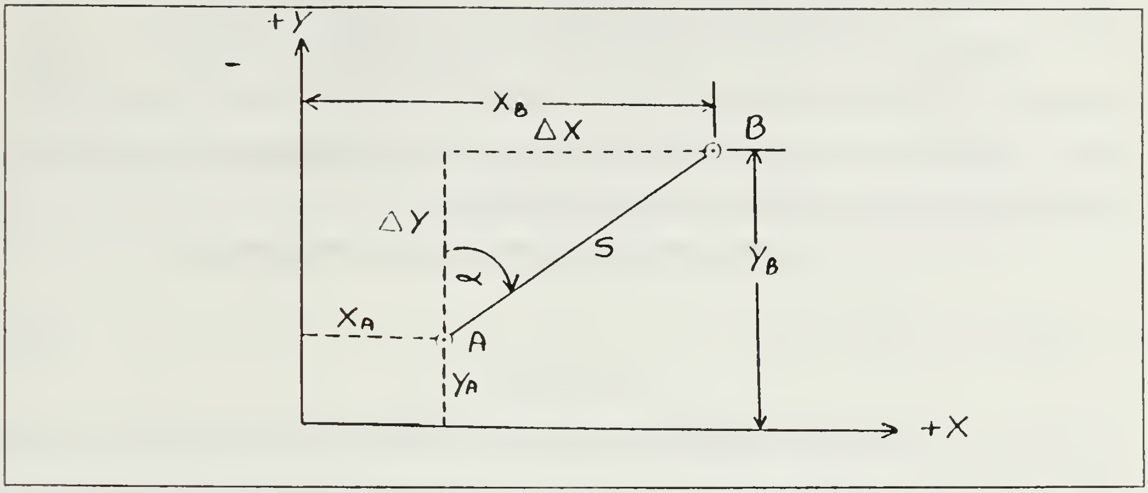

Figure 4.1 Azimuth Computation.

TABLE VI

QUADRANT OF AZIMUTH

Quadrant of azimuth measured clockwise from north	Sign of ΔX	Sign of ΔY
0 to 90	+	+
90 to 180	+	-
180 to 270	-	-
270 to 360	-	+

Substituting data from Table II into Equations 4.2 and 4.3, the ΔX and ΔY between Egmont Key Lt. House and Tomlinson are computed as ΔX = 3185.585 m and ΔY = 9470.704 m. The azimuth of the line from Egmont Key Lt. House to Tomlinson is 18° 35′ 27.6″ (Equation 4.1). Similarly, the azimuth from Turtle 2 to Madeira Beach Tank is computed to be 273° 57′ 12.0″. These starting and closing azimuths will be used for computing the coordinates of each traverse station and for checking the angular error.

C. COMPUTATION OF TRAVERSE STATION COORDINATES

Computation of traverse station coordinates is the reverse process of finding azimuth and distance from coordinates. Therefore, the rectangular coordinates for each closed-connecting traverse station can not be computed unless forward azimuth and distance from the previous station are known.

The azimuth is reckoned clockwise from north and obtained by

$$\alpha_{jk} = \alpha_{ij} + 180° + \beta_j \tag{4.6}$$

where α_{jk} is the forward azimuth from station j to station k, α_{ij} is the forward azimuth from station i to station j, and β_j is the horizontal angle at station j for j values of 1 to n, where the previous station i = j - 1 and the next station k = j + 1. The number j will increase from 1 (which designates the starting known station of the traverse) to number n (which was the last occupied and known station of the traverse).

Departures and latitudes are then computed by using Equations 4.4 and 4.5 which are rewritten as

$$\Delta X_{jk} = S_{jk} \text{Sin}\ \alpha_{jk} \tag{4.7}$$

and

$$\Delta Y_{jk} = S_{jk} \text{Cos}\ \alpha_{jk} \tag{4.8}$$

where S_{jk} is the distance between stations j and k. The coordinates of all other traverse stations can be determined by adding successive departures (ΔX_{jk}) and latitudes (ΔY_{jk}) to the previous station's X and Y coordinates, respectively.

Using the data in Tables II, III, and IV, the azimuth and coordinate computations at the first station are shown here by using Equations 4.6, 4.7, and 4.8. The result for other stations can be seen in Table VII.

Computing the azimuth from station 1 to 2

$$Az = 18°35'27.6'' + 180° + 105°21'25'' = 303°56'52.6''$$

Computing the coordinates at station 2

$$\text{X-easting} = 329400.420 + [701.318\ \text{Sin}(303°56'52.6'')]$$
$$= 328818.645\ \text{m}$$

$$\begin{aligned} \text{Y-northing} &= 3063485.483 + [701.318\ \text{Cos}(303°56'52.6'')] \\ &= 3063877.126\ \text{m} \end{aligned}$$

When the coordinates of all stations have been computed, they are still unadjusted coordinates and can then be adjusted by one of the two methods of Section II.E (approximation and/or least squares method).

TABLE VII

UNADJUSTED TRAVERSE STATION POSITIONS

Stn.	Angles			Forward Azimuths			Dist. (m)	Unadjusted Coordinates	
	D	M	S	D	M	S		X(m)	Y(m)
0									
				18	35	28			
1	105	21	25					*329400.420	*3063485.483
				303	56	53	701.318		
2	243	39	18					328818.645	3063877.126
				7	36	11	794.090		
3	168	10	15					328923.709	3064664.236
				355	46	26	1357.068		
4	59	55	56					328823.700	3066017.613
				235	42	22	216.267		
5	291	28	29					328645.029	3065895.760
				347	10	51	1539.307		
6	160	43	42					328303.493	3067396.699
				327	54	33	2017.880		
7	269	55	50					327231.464	3069106.259
				57	50	23	504.481		
8	92	43	29					327658.538	3069374.789
				330	33	52	489.448		
9	178	22	51					327418.001	3069801.054
				328	56	43	644.535		
10	182	31	19					327085.512	3070353.210
				331	28	2	719.956		
11	196	54	42					326741.616	3070985.723
				348	22	44	414.561		
12	168	51	46					326658.106	3071391.785
				337	14	30	1542.415		
13	161	6	42					326061.428	3072814.113
				318	21	12	1881.128		
14	236	58	14					324811.349	3074219.797
				15	19	26	2031.346		
15	78	37	24					325348.180	3076178.924
				273	56	50			
16									

* Coordinates for station 1 were known and held fixed.

D. ADJUSTMENT BY APPROXIMATION METHOD

In this thesis, the method of Compass or Bowditch rule was used to adjusted the data in Tables III and IV. Thus, the first step is to determine the angular error of closure and adjust the angles to obtain the proper closing azimuth (closed azimuth at fixed stations).

1. Angular Errors of Closure

In the closed-connecting traverse, when there are n stations of observed horizontal angles, n-1 lines will be measured. An angular error in traverse can be checked and obtained at the last station by comparing the computed azimuth and closing azimuth at the known station.

The closing azimuth computed (from the known station coordinates at the traverse end) at the station 15 is 273° 57′ 12.0″. But because of error in measurement, the azimuth computed through the traverse at this station is 273° 56′ 49.6″, which is a difficiency of 22.4″. This amount of angular error in 15 observed stations meets the limit for allowable error for a third-order class I traverse (allowable error from Table I is 38.7″).

The average correction (Table VIII) is distributed uniformly over all the 15 traverse angles (Table III).

2. Linear Errors of Closure

When all angles have been corrected, the process of calculating the improved coordinates of all traverse stations may be done. The check on angular closure for a closed traverse does not guarantee that the entire survey is correct, because there can be considerable errors in the linear measurement of individual lines. Such errors may not show up in the angular check. In order to check the closure of the traverse, it is necessary to determine linear error.

The linear error (LE), the departure error (δx), and latitude error (δy) in a traverse are determined by equations

$$LE = [\,(\delta x)^2 + (\delta y)^2\,]^{1/2} \tag{4.9}$$

$$\delta x = GE_n - GE_n' \tag{4.10}$$

$$\delta y = GN_n - GN_n' \qquad (4.11)$$

where δx and δy are the traverse closure in departure and latitude, GE_n and GE_n' are the known and computed grid easting, and GN_n and GN_n' are the known and computed grid northing at the closing station, respectively. By substituting the fixed and computed values of grid easting and northing in Equations 4.10 and 4.11 for the data of Tables II and VII

$$\delta x = 325348.292 - 325348.180 = +\ 0.112 \text{ m}$$
$$\delta y = 3076179.297 - 3076178.924 = +\ 0.373 \text{ m}$$
$$LE = [(0.112)^2 + (0.373)^2]^{1/2} = 0.389 \text{ m}$$

The relative error of closure provides a better assessment of the quality of a traverse than the linear error of closure. Therefore, it is common practice to calculate the relative error of closure, which is the linear error of closure divided by total distances of traverse, and to express the result in the form of a ratio with unity as the numerator. For the data of Tables II and VII, this computation is 0.389 / 14853.800 or 1 : 38,185.

Using the Compass or Bowditch rule, the computed traverse closures and corrections were obtained (Tables VIII and IX) and then the adjusted station coordinates (Table X) were computed.

E. LEAST SQUARES ADJUSTMENT BY OBSERVATION EQUATIONS

The adjustment by observation equations, shown in general form by Equation 2.13, can be done more directly than the adjustment by condition equations. To achieve this, the explicit expression for the residual vector V from Equation 2.13 is substituted in Equation 2.7 to obtain the following equation:

$$\begin{aligned} \Phi &= (F - BX)^T W (F - BX) \qquad (4.12) \\ &= (F^T - B^T X^T)(WF - WBX) \\ &= X^T B^T WBX - X^T B^T WF + F^T WF - F^T WBX \\ &= X^T B^T WBX - 2F^T WBX + F^T WF \end{aligned}$$

where $X^T B^T WF = F^T WBX$ are scalar quantities.

TABLE VIII

TRAVERSE CLOSURE

I) Angular error

Known azimuth at last station	273° 57' 11.95"
Computed azimuth	273° 56' 49.58"
Angular error	22.37"
Angular correction per station	1.49"

II) Linear error before adjusting azimuths

	X(m)	Y(m)
Known Coordinate at Turtle 2	325348.292	3076179.297
Computed Coordinate at Turtle 2	325348.180	3076178.924
Error	δx = + 0.112	δy = + 0.373

Linear error of closure	0.389 m
Total distances	14853.800 m
Relative error of closure	1 / 38,185

TABLE IX

LATITUDE AND DEPARTURE CORRECTIONS

Traverse lines	X-correction (m)	Y-correction (m)
1	- 0.031	+ 0.007
2	- 0.035	+ 0.008
3	- 0.061	+ 0.012
4	- 0.010	+ 0.002
5	- 0.069	+ 0.015
6	- 0.090	+ 0.019
7	- 0.022	+ 0.005
8	- 0.022	+ 0.005
9	- 0.029	+ 0.006
10	- 0.032	+ 0.007
11	- 0.018	+ 0.004
12	- 0.069	+ 0.015
13	- 0.084	+ 0.018
14	- 0.090	+ 0.020

TABLE X

ADJUSTED COORDINATES BY COMPASS RULE

stn.	Adj.Az. D	M	S	Adj. Dist. (m)	Departure (m)	Latitude (m)	Adj. Coordinates X (m)	Y(m)
1							329400.420*	3063485.483*
	303	56	54	701.318				
2					- 581.803	391.654	328818.617	3063877.137
	7	36	14	794.090				
3					105.040	787.116	328923.657	3064664.253
	355	46	30	1357.068				
4					- 100.040	1353.393	328823.617	3066017.646
	235	42	28	216.267				
5					- 178.684	-121.847	328644.933	3065895.799
	374	10	58	1539.307				
6					- 341.551	1500.967	328303.382	3067396.766
	327	54	42	2017.880				
7					-1072.044	1709.626	327231.338	3069106.392
	57	50	33	504.481				
8					427.065	268.514	327658.403	3069374.906
	303	34	4	489.448				
9					- 240.535	426.283	327417.868	3069801.189
	328	56	56	644.535				
10					- 332.482	552.184	327085.386	3070353.373
	331	28	16	719.956				
11					- 343.882	632.544	326741.504	3070985.917
	348	23	00	414.561				
12					- 83.496	406.073	326658.008	3071391.990
	337	14	47	1542.415				
13					- 596.624	1422.395	326061.384	3072814.385
	318	21	31	1881.128				
14					-1250.030	1405.819	324811.354	3074220.204
	15	19	46	2031.346				
15					536.938	1959.093	325348.292*	3076179.297*
	273	57	12					

* Coordinates of stations 1 and 15 were known and held fixed.

The minimization of Equation 4.12 can be done by taking partial derivative with respected to each unknown variable (X)

$$\Phi' = \partial\Phi/\partial X = 2X^TB^TWB - 2F^TWB = 0 \qquad (4.13)$$

Transposing Equation 4.13 and rearranging yields

$$(B^TWB)X = B^TWF \qquad (4.14)$$

or

$$NX + U = 0 \qquad (4.15)$$

where $N = (B^TWB)$ is the normal matrix of dimension u by u, and U is B^TWF. Then

$$X = -N^{-1}U \qquad (4.16)$$

For the adjustment in traverse, the vector V in equation 2.13 is represented the residual of observed angles and distances. If there are n observed angles in a traverse, there will be n - 1 observed distances and the number of residuals becomes 2n+1 which includes the residual of angles and distances.

There are two types of condition equations in the adjustment of observation equations: the angle condition and distance condition.

From Figure 4.2 the angle condition can be expressed by

$$v_{ia} = \beta_i - (\alpha_{ij} - \alpha_{ik}) \qquad (4.17)$$

where v_{ia} is a residual of observed angle, α_{ij} is a forward azimuth, α_{ik} is a backward azimuth and β_i is an observed angle. Equation 4.17 is suitable when $\alpha_{ij} \geq \alpha_{ik}$. If $\alpha_{ij} < \alpha_{ik}$, the quantities in parenthesis must be added by 360°.

The Equation 4.17 can be expressed in coordinates of the two stations as

$$v_{ia} = \beta_i - [\tan^{-1}(\frac{X_j - X_i}{Y_j - Y_i}) - \tan^{-1}(\frac{X_k - X_i}{Y_k - Y_i})] \qquad (4.18)$$

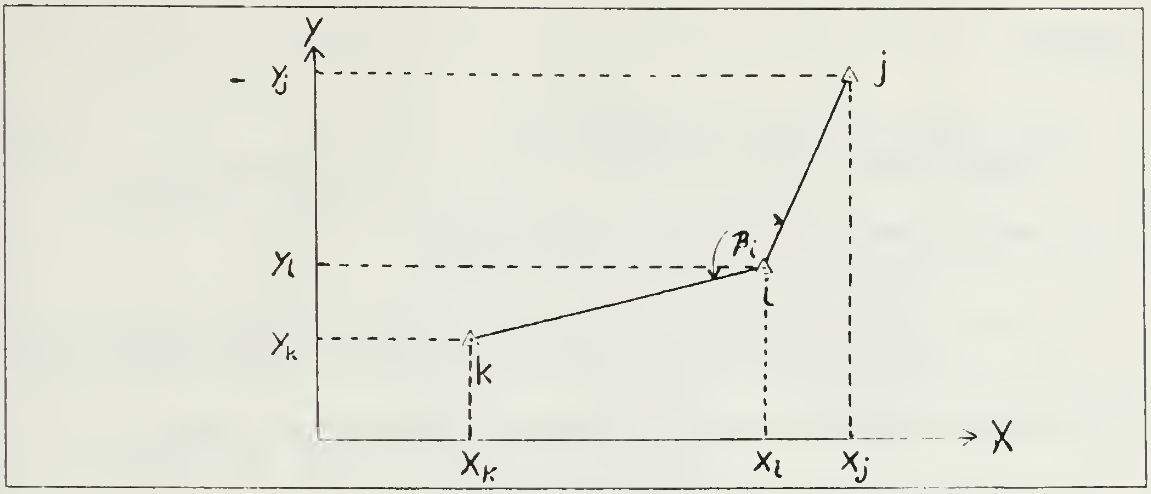

Figure 4.2 Determination of Angle and Distance from Coordinates.

and

$$g_i = \left[\tan^{-1}\left(\frac{X_j - X_i}{Y_j - Y_i}\right) - \tan^{-1}\left(\frac{X_k - X_i}{Y_k - Y_i}\right) \right] \qquad (4.19)$$

When linearized by using Equation 2.12, Equation 4.19 yields

$$g_i = g_i(X_i', Y_i', X_j', Y_j', X_k', Y_k') + a_1\delta X_i + a_2\delta Y_i + a_3\delta X_j + a_4\delta Y_j + a_5\delta X_k + a_6\delta Y_k \qquad (4.20)$$

where X_i', Y_i', X_j', Y_j', X_k', and Y_k' are the estimated station coordinates. By substituting the linearization of function g_i, Equation 4.18 can be written as

$$v_{ia} + a_1\delta X_i + a_2\delta Y_i + a_3\delta X_j + a_4\delta Y_j + a_5\delta X_k + a_6\delta Y_k = F_1 \qquad (4.21)$$

where δX_i, δY_i,.......,δX_k, and δY_k are unknown parameters (correction in X and Y coordinates), coefficeints a_1, a_2,.......,a_5, and a_6 are the partial derivative of function g_i with respected to X_i, Y_i,......., X_k, and Y_k, respectively.

For the distance condition, it can be expressed as

$$v_{id} = d_i - \left[(X_j - X_i)^2 + (Y_j - Y_i)^2 \right]^{1/2} \qquad (4.22)$$

where

$$h_i = [(X_j - X_i)^2 + (Y_j - Y_i)^2]^{1/2} \tag{4.23}$$

The linerized form of Equation 4.23 is then given as

$$h_i = h_i(X_i', Y_i', X_j', Y_j', X_k', Y_k') + b_1\delta X_i + b_2\delta Y_i + b_3\delta X_j + b_4\delta Y_j \tag{4.24}$$

Substituting the linearized form from Equation 4.24, Equation 4.22 becomes

$$v_{id} + b_1\delta X_i + b_2\delta Y_i + b_3\delta X_j + b_4\delta Y_j = F_2 \tag{4.25}$$

where F1 and F2 represent the constant form for Equations 4.21 and 4.25. Table XI shows the partial derivatives for Equations 4.21 and 4.25.

Sometimes, the station positions which is determined by intersection from traverse station is also to be adjusted simultaneously with traverse station positions. In this case only the angle conditions are added to observation equations of traverse.

In the data adjusted under this thesis, there are 15 observed angles and 14 oberved distances which consisted of 13 unknown points, consequently, there will be 29 observation equations included 26 parameters of δX and δY. Thus, they can be expressed in matrix form of Equation 2.13 as

$$V_{29,1} + B_{29,26} X_{26,1} = F_{29,1} \tag{4.26}$$

where $v_{1,1}$, $v_{2,1}$, ...$v_{15,1}$ are the residuals of angles; $v_{16,1}$, $v_{17,1}$, ...$v_{29,1}$ are the residuals of distances; B is the coefficeint matrix (consistings of a's and b's) of parameter X (Table XI); and F is the constant vector.

When three intersection points with 10 observed angles were added (Figure 3.1) to adjust the coordinates, the observation equations have 39 equations including 32 unknown parameters.

$$V_{39,1} + B_{39,32} X_{32,1} = F_{39,1} \tag{4.27}$$

TABLE XI

THE COEFFICIENTS OF ANGLE AND DISTANCE CONDITIONS

$$a_1 = \frac{\partial g_i}{\partial X_i} = - \frac{Y_j - Y_i}{(S_{ij})^2} + \frac{Y_k - Y_k}{(S_{ik})^2}$$

$$a_2 = \frac{\partial g_i}{\partial Y_i} = + \frac{X_j - X_i}{(S_{ij})^2} - \frac{X_k - X_k}{(S_{ik})^2}$$

$$a_3 = \frac{\partial g_i}{\partial X_j} = + \frac{Y_j - Y_i}{(S_{ij})^2}$$

$$a_4 = \frac{\partial g_i}{\partial Y_j} = - \frac{X_j - X_i}{(S_{ij})^2}$$

$$a_5 = \frac{\partial g_i}{\partial X_k} = - \frac{Y_k - Y_i}{(S_{ik})^2}$$

$$a_6 = \frac{\partial g_i}{\partial Y_k} = + \frac{X_k - X_i}{(S_{ik})^2}$$

$$b_1 = \frac{\partial h_i}{\partial X_i} = + \frac{X_j - X_i}{S_{ij}}$$

$$b_2 = \frac{\partial h_i}{\partial Y_i} = + \frac{Y_j - Y_i}{S_{ij}}$$

$$b_3 = \frac{\partial h_i}{\partial X_j} = - \frac{X_j - X_i}{S_{ij}}$$

$$b_4 = \frac{\partial h_i}{\partial Y_j} = - \frac{Y_j - Y_i}{S_{ij}}$$

By using Equation 4.16, N^{-1} is the cofactor matrix and the diagonal terms of this matrix gives the variances of the adjusted coordinates. To obtain the residuals of all observed quantities, the reverse process must be done. The correction of X and Y adjusted coordinates and their standard deviation (σ's) are shown in Table XII. And the adjusted standard deviation of observed quantities were obtained by multiplying standard deviation of unit weight $[(V^TWV / r)^{-1/2}]$ to a squares root of diagonal element of $B(B^TWB)^{-1}B^T$ matrix (Table XIII).

TABLE XII

ADJUSTED COORDINATES BY OBSERVATION EQUATION

Stn.	Estimated Coordinates X(m)	Y(m)	Corrections dx	dy	Adjusted Coordinates X(m)	σ_x(±)	Y(m)	σ_y(±)
1					329400.420		3063485.483	
2	328818.645	3063877.126	-0.009	+0.018	328818.636	0.030	3063877.144	0.032
3	328923.709	3064664.234	-0.037	+0.034	328923.672	0.047	3064664.270	0.057
4	328823.700	3066017.613	-0.087	+0.055	328823.613	0.062	3066017.668	0.110
5	328645.029	3065895.760	-0.083	+0.052	328644.946	0.064	3065895.811	0.108
6	328303.493	3067396.699	-0.112	+0.075	328303.380	0.078	3067396.774	0.137
7	327231.464	3069106.259	-0.135	+0.110	327231.330	0.155	3069106.368	0.155
8	327658.538	3069374.789	-0.136	+0.111	327658.402	0.161	3069374.901	0.161
9	327418.001	3069801.054	-0.138	+0.122	327417.863	0.165	3069801.176	0.165
10	327085.512	3070353.210	-0.133	+0.140	327085.378	0.102	3070353.350	0.168
11	326741.616	3070985.723	-0.125	+0.161	326741.491	0.104	3070985.883	0.092
12	326658.106	3071391.785	-0.112	+0.172	326657.994	0.104	3071391.957	0.161
13	326061.428	3072814.113	-0.049	+0.231	326061.379	0.083	3072814.344	0.113
14	324811.349	3074219.797	+0.024	+0.344	324811.373	0.058	3074220.142	0.068
15					325348.180		3076178.924	

TABLE XIII

ADJUSTED STANDARD DEVIATION OF ANGLES AND DISTANCES

Number	Angles seconds (±)	Distances meter (±)
1	9.8	0.029
2	8.9	0.035
3	9.1	0.045
4	9.2	0.024
5	9.3	0.048
6	9.4	0.056
7	8.4	0.030
8	8.5	0.029
9	7.8	0.032
10	7.1	0.034
11	9.6	0.028
12	9.7	0.048
13	9.8	0.054
14	6.8	0.058
15	6.8	

V. ANALYSIS OF RESULTS

A. COMPARISON OF ADJUSTED COORDINATES

If the given coordinates of the control points are assumed to be error free, then the accuracy of the traverse station coordinates depends only on the accuracy of distance and angle measurements. The adjusted traverse coordinates obtained by the approximation method are of a lower order of accuracy as only the errors in the misclosure in azimuths and distances were determined. These errors were distributed by assuming that all observed quantities had an equal probable occurrence.

The least squares adjustment method provides a better approximation of the true value. Therefore, the adjusted traverse coordinates obtained by this technique provided better estimates for position of all traverse stations and the accuracy of the adjustment can be checked and statistically tested.

After the 13 adjusted traverse station (from stations 2 to 14) coordinates obtained by the approximation method and by the least squares method were compared, the difference in coordinates at each station were computed and plotted (Table XIV and Figure 5.1). The largest difference was at station 14. Because stations 1 and 15 are held fixed, the least squares techniques adjusts simultaneously errors in azimuths and distances, while the adjustment by approximation adjusts errors in azimuths and distances sequentially. Consequently, the largest difference in traverse distances occurs at the last station (station 14) before closing of traverse at the fixed station.

When the standard deviation of observed quantities before the adjustment (Tables III and IV) were compared with those obtained through adjustment (Table XIII), the standard deviation of all observed quantities in Table XIII showed increments. That means, the estimated standard deviations were optimistic.

In this thesis, the three intersection points were also adjusted. The adjusted coordinates of these were compared to NOS results. The largest difference occurs at point no. 3 ($\delta x = + 0.140$ m; $\delta y = - 0.158$ m). The standard deviation of adjusted coordinates at this point are $\sigma x_3 = \pm 1.87$, $\sigma y_3 = \pm 0.96$ m (Table XV).

TABLE XIV

COMPARISON OF ADJUSTED COORDINATES/DISTANCES OBTAINED BY APPROXIMATION AND LEAST SQUARES METHODS

	Differences*		
	Coordinates		Distances
stn.	dx (m)	dy (m)	(m)
2	- 0.019	- 0.007	0.020
3	- 0.015	- 0.017	0.023
4	+ 0.004	- 0.022	0.022
5	- 0.013	- 0.012	0.018
6	+ 0.002	- 0.008	0.008
7	+ 0.008	+ 0.024	0.025
8	+ 0.001	- 0.005	0.005
9	+ 0.005	- 0.013	0.014
10	- 0.008	- 0.023	0.024
11	- 0.003	- 0.034	0.034
12	- 0.014	+ 0.003	0.014
13	- 0.005	+ 0.041	0.041
14	- 0.019	- 0.101	0.103

* Approximation minus least squares solution

B. ANALYSIS OF THE REFERENCE VARIANCE OF UNIT WEIGHT

The weight matrix was set for the least squares adjustment by using Equation 2.11. The σ_0^2 (a priori reference variance of unit weight) was assumed as 1. The result of $\hat{\sigma}_0^2 = (V^TWV / r)$ was obtained after adjustment. The value of $\hat{\sigma}_0^2$ (a posteriori reference variance of unit weight) can be used to evaluate the weighting scheme used in the least squares adjustment.

The standard deviation of the observed angles (σ_a) and distances (σ_d) used in the least squares adjustments (solutions 1, 2, and 3) and the corresponding a posteriori $\hat{\sigma}_0^2$ obtained for these solutions are listed in Table XVI. As the a posteriori $\hat{\sigma}_0^2$ for the solution no. 3 is closest to the assumed a priori σ_0^2 (= 1), the weight (or the standard deviations) used for observed angles and distances in this case seem the most realistic. Further statistical testing for $\hat{\sigma}_0^2$ done by Chi-squares or F-test was not carried out under this thesis.

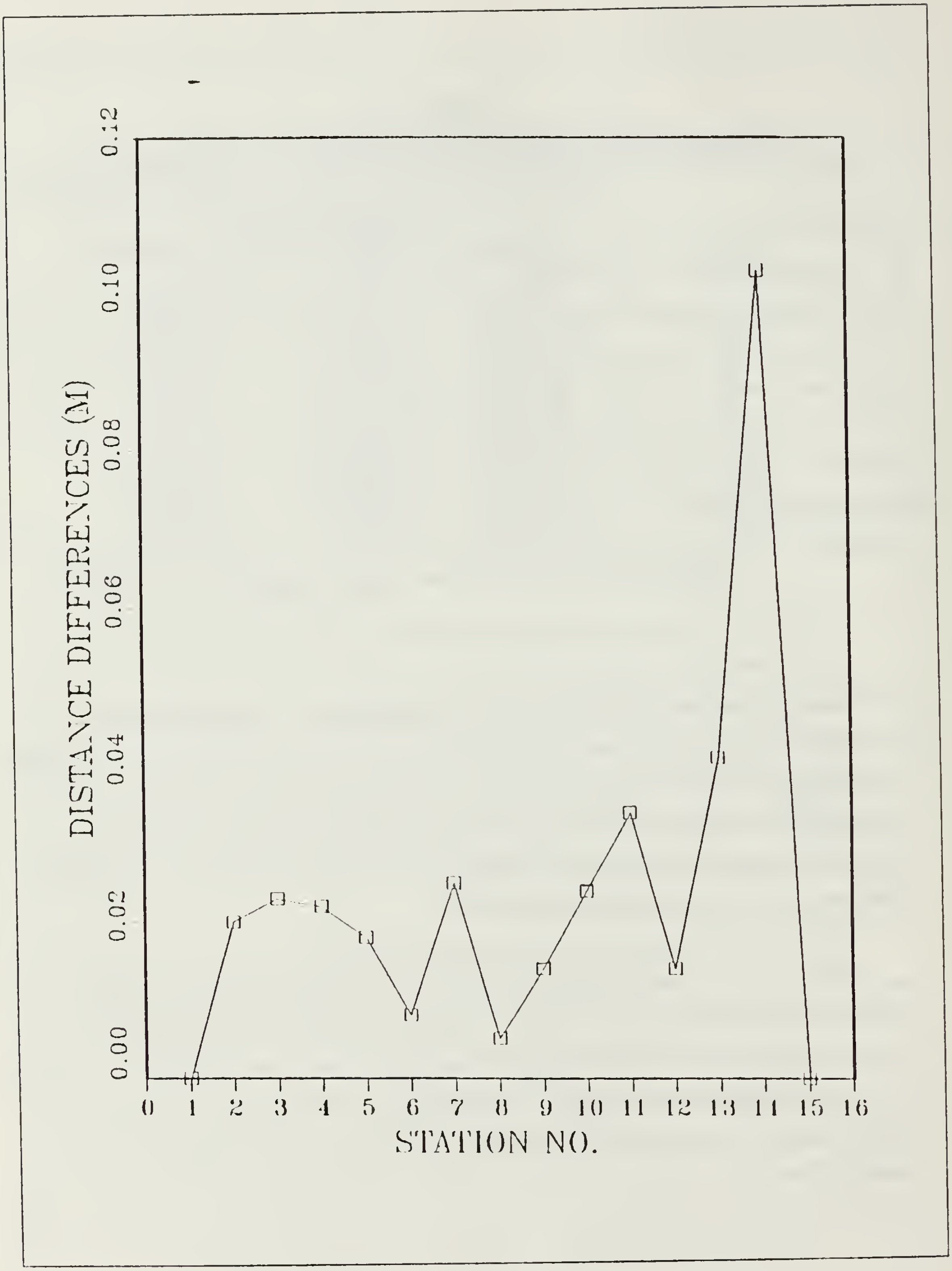

Figure 5.1 Comparison of Adjusted Distances Obtained by Approximation and Least Squares Methods.

TABLE XV

COMPARISON OF COORDINATES AT INTERSECTION POINTS

	X(m)	σx(±m)		Y(m)	σy(±m)
At Point No. 1					
NOS	326616.356			3071294.683	
Adjusted	326616.470	1.72		3071294.635	1.03
δx	- 0.114		δy	+ 0.048	
At Point No. 2					
NOS	327056.079			3070340.474	
Adjusted	327056.183	1.68		3070340.444	1.14
δx	- 0.104		δy	+ 0.030	
At Point No. 3					
NOS	325378.101			3077263.378	
Adjusted	325377.961	1.87		3077263.536	0.96
δx	+ 0.140		δy	- 0.158	

TABLE XVI

COMPARISION OF VARIANCES OF UNIT WEIGHT

Solutions	Variance used in Adjustment			V^TWV/r
	σ_a	σ_d	σ_0^2	$\hat{\sigma}_0^2$
1	2"	(10ppm + 0.5 cm)	1	15.08
2	5"	(10ppm + 1.0 cm)	1	3.98
3	10"	(10ppm + 2.0 cm)	1	1.32

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

By using the weight of observed quantities for the adjustment, the traverse station coordinates computed and adjusted by the least squares observation equations method were more accurate than those obtained by the approximation methods. Even though the observation equations method may require a greater number of equations than the condition equations method, the processing of the data for adjustment is easier and the corrections in X and Y coordinates can be directly obtained through iterative solution. This method is suitable when a computer with a memory capacity of over 500 K bytes is available. However, for local work or a relatively short traverse, an approximation method is commonly utilized when economic and logistic criteria are considered.

B. RECOMMENDATION

The INDTRA Fortran program written for this thesis is automated for handling only two kinds of survey techniques: traversing and intersection. With the computers available at NPS, the development of adjustment programs for covering a wide range of survey techniques should be done to use and continue analysis of the mixed kind of survey techniques including traverse, triangulation, trilateration, resection, and intersection.

APPENDIX A
LINEARIZATIONS

This is a linearized form of m functions in n unknowns.

$$
\begin{aligned}
y_1 &= f_1(x_1, x_2, \ldots, x_n) \\
y_2 &= f_2(x_1, x_2, \ldots, x_n) \\
&\vdots \\
y_m &= f_m(x_1, x_2, \ldots, x_n)
\end{aligned}
$$

$$
Y^0 = \begin{vmatrix} y_1^0 \\ y_2^0 \\ \vdots \\ y_m^0 \end{vmatrix} = \begin{vmatrix} f_1(x_1^0, x_2^0, \ldots, x_n^0) \\ f_2(x_2^0, x_2^0, \ldots, x_n^0) \\ \vdots \\ f_m(x_m^0, x_m^0, \ldots, x_n^0) \end{vmatrix}
$$

$$
J_{yx} = \frac{\partial Y}{\partial X} = \begin{vmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \cdot & \cdot & \cdots & \cdot \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial y_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{vmatrix}
$$

$$
\Delta X = \begin{vmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_1 \end{vmatrix}
$$

The general form of linearized functions becomes

$$
Y = Y^0 + J_{yx}\Delta X
$$

APPENDIX B

TRAVADJ FORTRAN PROGRAM

This program is used for computing and adjusting the traverse station position by approximation method (Compass rule).

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C                 FORTRAN PROGRAM "TRAVADJ"           C
C                                                     C
C  THIS PROGRAM IS USED FOR                           C
C    1) REDUCE SLOPE DISTANCE TO ELLIPSOIDAL DISTANCE C
C    2) DETERMINE GRID DISTANCE                       C
C    3) COMPUTE CLOSE TRAVERSE                        C
C    4) ADJUST COORDINATE BY COMPASS RULE             C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C  INPUT DATA                                         C
C   1) SEMIMAJOR AXIS OF REFERENCE ELLIPSOID USED     C
c      "a"                                            c
C   2) VALUE OF 1/F  (EX. 1/F = 294.978698            C
c       "f"                                           c
C   3) CENTRAL SCALE FACTOR                           C
c       "ko"                                          c
C   4) LAT. AT STARTING AND CLOSING POINT             C
c       "id1,im1,s1,id2,im2,s2"                       c
C   5) GRID N. AND E. OF 4 KNOWN STATION              C
c       "gn1,ge1,gn2,ge2,gn3,ge3,gn4,gn4"             c
C   6) NUMBER OF MEASURE DISTANCE                     C
c       "n"                                           c
C   7) ELEVATION AT FIRST OCCUPIED POINT              C
c       "elev"                                        c
C   8) NAME OF ALL STATIONS                           C
C   9) INDICATOR VALUE 1 = NO VERTICAL ANGLE          C
C                      2 = VERTICAL ANGLE             C
C  10) DIFFERENT IN ELEVATION BETWEEN TWO STATIONS    C
C  11) SLOPE DISTANCE (WITH UNIT FEET OR METER)       C
c        "dist"                                       c
C  12) HORIZONTAL ANGLES                              C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                     c
c     sfac = scale factor coorection
c     hgd  = horizontal distance                      c
c     todis= total distance in traverse               c
c     cegrid = grid east of traverse station          c
c     cngrid = grid north of traverse station         c
c     difazi = azimuth misclosure                     c
c     difdis = distance misclosure                    c
c     coraz  = angular correction per station         c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                     c
c     gridaz = subroutine for computing azimuth       c
```

```
c               between two traverse stations          c
c     utm     = subroutine for computing the grid      c
c               coordinates from known distance        c
c               and azimuth                            c
c     dmsr    = subroutine for converting the angle    c
c               from degrees, minutes, and seconds to  c
c               radians                                c
c     rdms    = subroutine for converting the angle    c
c               from radians to degrees, minutes, and  c
c               seconds                                c
c                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
      DIMENSION LAT(3),K(3),HGD(30),HAGL(30),HANG(30)
      DIMENSION DHAGL(30),MHAGL(30),SHAGL(30),NAME(3,30),NAME1(5)
      DIMENSION NAME3(2),NAME4(5),MDIST(30)
C
C     *                          *
C     *  VARIABLE DECLARATION    *
C     *                          *
      DOUBLE PRECISION  AN90R,AN180R,AN360R,AZC,AZFIR,AZIMUT,AZLAS,
     *                  CEGRID(30),COAZR(30),CORAZ,CNGRID(30),
     *                  DISTAN,DIFAZI,DELTAX(30),DELTAY(30),DUMMY1,
     *                  DUMMY2,DUMDIS,DIFDIS,TODIS
C
      DOUBLE PRECISION  NUM0,NUM360,NUM1,SUMDX,SUMDY,STDD(30),STDA(30),
     *                  XGD,YGD,NUM180,NUM90,WPRED(3)
C
      DOUBLE PRECISION  ANGS(30),CFAZS(30),DUM1D,DUM1M,DUM1S
     *                  ,EAZ1S,EAZ2S,CORDX1,CORDY1,TEMS,AZS21,AZS34
C
      DOUBLE PRECISION  SUMIX1,SUMIX2,SUMFY,SUNFX,SUMCOY,SUMCOX,
     *                  SSTA,SSTD,SIGX,SIGY,SIGXY,SEMAJ,SEMIN,
     *                  SETA,SETAA,SSIGX,SSIGY,SSIGXY,DEG1,MIN1,
     *                  SECON1,SEMAJ1,SEMIN1,TEMPO1(30),TEMPO2(30),
     *                  DDIFFX,DDIFFY
C
      REAL*8 A,C,D,H,R,V,X,Y,GD,DH,ZD,CON,CUV,LAT,PHE,CUVRE,ELEV
      REAL*8 HDIST,SDIST,MDIST,ESQR,GRAMMA,ANG,ANGV,PI,AGU,F1,F2
      REAL*8 K,KO,E1,E2,Q1,Q2,RR,VV,ECEN,FAC,SFAC,HGD
      REAL*8 F,CHGD,CSDIST,DDM,HH,HAGL,CHDIST,DG,DMS,DS,DDANG,MMANG
      REAL*8 GN1,GN2,GN3,GN4,GE1,GE2,GE3,GE4,AZ21,AZ34,DIST21,DIST34
      REAL*8 S1,S2,D1,D2,SV,VANG,SHAGL,SANG,DD,MM,SS,HANG,RRR
      REAL*8 DUM1,DUM2,DUM3,DUM4,XXXX,YYYY
      REAL*8 IDD1,IMM1,IDD2,IMM2,DVV,MVV,CUVRER,CUVR,CURVR
      REAL*8 ZINE,KOSE
C
      INTEGER I,N,ID1,ID2,IM1,IM2,DV,MV,DHAGL,MHAGL,DANG,MANG,UNIT
      INTEGER ANGD(30),ANGM(30),ADJ1,ERR,AZD21,AZM21,AZD34,AZM34
      INTEGER SSHAGL(30),CFAZD(30),CFAZM(30),MIN2,DEG2,
     *        EAZ1D,EAZ1M,EAZ2D,EAZ2M,TEMD,TEMM
C
C
      DATA DUM1/0.30480061D0/,DUM2/500000.0D0/,DUM3/0.000001D0/
      DATA DUM4/3600.0D0/,SIGANG/5.0D0/,DUM5/0.000005D0/,DUM6/0.005D0/
      DATA TODIS/0.0D0/,DUM10/0.0000048481368D0/
C
C
      DATA NUM0/0.0D0/,NUM360/360.0D0/,NUM1/1.0D0/,SUMDX/0.0D0/,
     *     SUMDY/0.0D0/,NUM180/180.0D0/,NUM90/90.0D0/
C
      READ(5,10) A,F,KO
      READ(5,15) ID1,IM1,S1
      READ(5,15) ID2,IM2,S2
      READ(5,16) GN1,GE1
      READ(5,16) GN2,GE2
      READ(5,16) GN3,GE3
      READ(5,16) GN4,GE4
```

```
      READ(5,11)N,ELEV
C
C
      CALL GRIDAZ (GE1,GN1,GE2,GN2,AZ21,DIST21)
      CALL GRIDAZ (GE3,GN3,GE4,GN4,AZ34,DIST34)
C
            IDD1 = FLOAT(ID1)
            IMM1 = FLOAT(IM1)
      CALL DMSR (IDD1,IMM1,S1,D1)
            LAT(1) = D1
            IDD2 = FLOAT(ID2)
            IMM2 = FLOAT(IM2)
      CALL DMSR (IDD2,IMM2,S2,D2)
            LAT(2) = D2
      LAT(3) = (D1+D2)/2.0D0
      ESQR   = 2.0D0*(1.0D0/F)-(1.0D0/F)**2
        X = A*DSQRT(1.0D0-ESQR)
        Y = 1.0D0-(ESQR*(DSIN(LAT(3)))**2)
        R = X/Y
C
C
C DETERMINATION OF THE SCALE FACTOR FOR UTM.
C
C
C
      E1 = DABS(DUM2-GE2)
      E2 = DABS(DUM2-GE3)
C
      ECEN = ESQR/(1.0D0-ESQR)
C
C
      Q1 = DUM3*E1
      Q2 = DUM3*E2
      QPRIME = ((Q1**2)+(Q1*Q2)+(Q2**2))/3.0D0
C
C
      DO1 M=1,3
        RR=A/(1.0D0-ESQR*(DSIN(LAT(M)))**2)**0.5
C
        F1 = (1.0D0+ECEN*DCOS(LAT(M)))*(10.0D0**12)
        F2 = 2.0D0*(RR**2)*(KO**2)
        FAC= F1/F2
C
C
        K(M)=KO*(1.0D0+FAC*QPRIME+(0.00003D0*(QPRIME**2)))
 1    CONTINUE
C
      SFAC = K(3)
C     SFAC = 6.0D0/((1.0D0/K(1))+(4.0D0/K(3))+(1.0D0/K(2)))
C
C
          READ(5,20) UNIT
          READ(5,18)(NAME1(L),L=1,5)
C
C  DETERMINATION OF THE HORIZONTAL DISTANCES
C
      DO 1000 J=1,N
C
          READ(5,14)(NAME(L,J),L=1,3)
          READ(5,12)I,DH,DV,MV,SV
          READ(5,13) SDIST
          MDIST(J) = SDIST
C
C  IN CASE OF THE LENGTH'S UNIT IS IN FEET, THEN CONVERSE TO METER
C
C
          IF(UNIT.EQ.1) THEN
             SDIST = SDIST*DUM1
```

```
              DH    = DH*DUM1
              ELEV  = ELEV*DUM1
           END IF
C
           READ(5,17)DHAGL(J),MHAGL(J),SHAGL(J)
               DD = DFLOAT(DHAGL(J))
               MM = DFLOAT(MHAGL(J))
               SS = SHAGL(J)
C
               CALL DMSR ( DD,MM,SS,RRR )
                   HANG(J) = RRR
                   STDA(J) = SIGANG
C
C
C
C   DETERMINATION OF HORIZONTAL DISTANCES WHEN THE DIFFERENT IN
C   ELEVATION IS APPROXIMATELY KNOWN.
C
           IF(I.EQ.1)THEN
C
             IF(DH.NE.0.0)THEN
               DDM = SDIST
               IF(DDM.GE.3300.0)THEN
                 DO 100 KK=1,3
C
                   CURV  =(0.016192D0*DDM)
                   CALL DMSR (NUMO,NUMO,CURV,CURVR)
                   D=DSQRT(DDM**2-(DH*DCOS(CURVR))**2)-DH*DSIN(CURVR)
                   DDM = D
C
 100           CONTINUE
C
                 HDIST = DDM
               ELSE
                 HDIST = DSQRT(DDM**2-DH**2)
               END IF
C
             ELSE
C
               HDIST = SDIST
             END IF
           END IF
C
C  DETERMINATION OF HORIZONTAL DISTANCE WHEN ZENITH DISTANCE IS KNOWN
C
           IF(I.EQ.2)THEN
C
             DVV = FLOAT(DV)
             MVV = FLOAT(MV)
             CALL DMSR (DVV,MVV,SV,VANG)
             CALL DMSR (NUM90,NUMO,NUMO,AN90R)
             CALL DMSR (NUM180,NUMO,NUMO,AN180R)
             ZD = AN90R-VANG
             DH = SDIST*DSIN(ZD)
             DS = SDIST*DCOS(ZD)
             IF(SDIST.GE.3300.0) THEN
               CUVRE = 0.013925D0*SDIST
               CUV   = 0.016192D0*SDIST
               CALL DMSR (NUMO,NUMO,CUVRE,CUVRER)
               CALL DMSR (NUMO,NUMO,CUV,CUVR)
               PHE      = (AN90R-ZD)+CUVRER
               C        = (AN90R+CUVR)
               GRAMMA   = AN180R-(C+PHE)
               HDIST    = (SDIST*DSIN(GRAMMA))/DSIN(C)
             ELSE
               HDIST = DS
             END IF
           END IF
C
```

```
          HH= (ELEV+DH)
C
C DETERMINATION OF THE HORIZONTAL DISTANCE ON GEOID
C
          GD = R*HDIST/(R+H)
C
C DETERMINATION OF THE GRID DISTANCES
          HGD(J) = GD*SFAC
C
C
          TODIS = TODIS+HGD(J)
          STDD(J) = HGD(J)*DUM5+DUM6
C
          WRITE(6,*) 'GRID DISTANCE =',HGD(J)
          WRITE(6,*)' '
C
          ELEV = HH
C
 1000 CONTINUE
C
      READ(5,19)(NAME3(L),L=1,2),DANG,MANG,SANG
      READ(5,18)(NAME4(L),L=1,5)
C
      CALL OUTPUT(N,AZ21,AZ34,GN2,GE2,GN3,GE3,HGD,DHAGL,MHAGL,SHAGL,
     *DANG,MANG,SANG,NAME1,NAME3,NAME4,NAME,MDIST,GN1,GE1,GN4,GE4)
          DDANG = DFLOAT(DANG)
          MMANG = DFLOAT(MANG)
          CALL DMSR (DDANG,MMANG,SANG,RRR )
C
               NN = N+1
               HANG(NN) = RRR
               STDA(NN) = SIGANG
C
C
C
C
      CALL DMSR ( NUM360,NUM0,NUM0,AN360R )
      CALL DMSR ( NUM180,NUM0,NUM0,AN180R )
C
      DO 2000 ADJ1 = 1,2
          XGD = GE2
          YGD = GN2
          AZFIR = AZ21
        DO 200 I = 1,N
            AZIMUT = AZFIR+HANG(I)
C
                IF (AZIMUT.GE.AN360R) THEN
                  AZIMUT = AZIMUT-AN360R
                END IF
                IF(AZIMUT.GE.AN180R) THEN
                  AZIMUT = AZIMUT-AN180R
                ELSE
                    AZIMUT = AZIMUT+AN180R
                END IF
C
            COAZR(I) = AZIMUT
C
C
            CALL RDMS ( AZIMUT,DUM1D,DUM1M,DUM1S )
                 CFAZD(I) = IDINT(DUM1D)
                 CFAZM(I) = IDINT(DUM1M)
                 CFAZS(I) = DUM1S
C
C
            DISTAN = HGD(I)
            CALL UTM ( XGD,YGD,DISTAN,AZIMUT,DUMMY1,DUMMY2 )
C
             WRITE(6,*)'I =',I
```

```
             WRITE(6,*)'GRID E=',DUMMY1,'      GRID N=',DUMMY2
                 DELTAX(I) = DUMMY1-XGD
                 DELTAY(I) = DUMMY2-YGD
             WRITE(6,*)'DEPARTure =',DELTAX(I),'  LATITUde =',DELTAY(I)
                  ZINE = DSIN(AZIMUT)
                 KOSE = DCOS(AZIMUT)
C
                 SUMDX     = SUMDX+DELTAX(I)
                 SUMDY     = SUMDY+DELTAY(I)
                 CEGRID(I) = DUMMY1
                 CNGRID(I) = DUMMY2
             CALL GRIDAZ ( XGD,YGD,DUMMY1,DUMMY2,AZC,DUMDIS )
C
C          CALL RDMS ( AZC,DUM1D,DUM1M,DUM1S )
C               CBAZD(I) = IDINT(DUM1D)
C               CBAZM(I) = IDINT(DUM1M)
C               CBAZS(I) = DUM1S
C
C
C
               IF (ADJ1.EQ.1) THEN
                   TEMPO1(I) = CEGRID(I)
                   TEMPO2(I) = CNGRID(I)
               END IF
C
             AZFIR = AZC
             XGD   = DUMMY1
             YGD   = DUMMY2
 200     CONTINUE
C
          IF (ADJ1.NE.1) THEN
C
             CORDX1 = ( SUMDX-(GE3-GE2) )/TODIS
             CORDY1 = ( SUMDY-(GN3-GN2) )/TODIS
             XGD    = GE2
             YGD    = GN2
C
C
             DO 230 I=1,N
C
             DISTAN = HGD(I)
C
            XXXX = CORDX1*HGD(I)
            YYYY = CORDY1*HGD(I)
                DELTAX(I) = DELTAX(I) - XXXX
                DELTAY(I) = DELTAY(I) - YYYY
C
C
C
                CEGRID(I) = XGD+ DELTAX(I)
                CNGRID(I) = YGD+ DELTAY(I)
                DDIFFX    = CEGRID(I)-TEMPO1(I)
                DDIFFY    = CNGRID(I)-TEMPO2(I)
C
C
C
                XGD       = CEGRID(I)
                YGD       = CNGRID(I)
 230       CONTINUE
C
C
          END IF
          AZLAS = AZFIR+HANG(NN)
c
          IF ( AZLAS.GE.AN360R ) THEN
             AZLAS = AZLAS-AN360R
          END IF
c
          IF (AZLAS.GT.AN180R) THEN
```

```
            AZLAS = AZLAS-AN180R
         ELSE
            AZLAS = AZLAS+AN180R
         END IF
C
         CALL RDMS ( AZLAS,DUM1D,DUM1M,DUM1S )
               TEMD = IDINT(DUM1D)
               TEMM = IDINT(DUM1M)
               TEMS = DUM1S
C
C
         DIFAZI = AZLAS-AZ34
         WPRED(1) = DIFAZI
         CORAZ    = DIFAZI/DFLOAT(NN)
C
         CALL RDMS ( DIFAZI,DUM1D,DUM1M,DUM1S )
               EAZ1D = IDINT(DABS(DUM1D))
               EAZ1M = IDINT(DABS(DUM1M))
               EAZ1S = DABS(DUM1S)
C
         CALL RDMS ( CORAZ,DUM1D,DUM1M,DUM1S )
               EAZ2D = IDINT(DABS(DUM1D))
               EAZ2M = IDINT(DABS(DUM1M))
               EAZ2S = DABS(DUM1S)
C
         IF(ADJ1.EQ.1)THEN
           WPRED(2) = SUMDX - ( GE3-GE2 )
           WPRED(3) = SUMDY - ( GN3-GN2 )
           DIFDIS   = DSQRT( WPRED(2)**2 + WPRED(3)**2 )
           ERR      = IDINT( TODIS / DIFDIS )
         END IF
C
C
C     PRINT RESULTS OF TRAVERSE COMPUTATION
C
C
         CALL RDMS (AZ21,DUM1D,DUM1M,DUM1S)
               AZD21  = IDINT(DUM1D)
               AZM21  = IDINT(DUM1M)
               AZS21  = DUM1S
         CALL RDMS (AZ34,DUM1D,DUM1M,DUM1S)
               AZD34  = IDINT(DUM1D)
               AZM34  = IDINT(DUM1M)
               AZS34  = DUM1S
C
         IF (ADJ1.EQ.1) THEN
            WRITE(8,29)
            WRITE(8,27)
            WRITE(8,30)
            WRITE(8,31)
         ELSE
            WRITE(8,28)
            WRITE(8,27)
            WRITE(8,30)
            WRITE(8,40)
         END IF
         WRITE(8,32)
         WRITE(8,33)
         WRITE(8,30)
         WRITE(8,34) AZD21,AZM21,AZS21
         WRITE(8,35) GE2,GN2
C
         DO 240 I=1,N
            II = I+1
            CALL RDMS(HANG(I),DUM1D,DUM1M,DUM1S)
                  ANGD(I)   = IDINT(DUM1D)
                  ANGM(I)   = IDINT(DUM1M)
                  ANGS(I)   = DUM1S
                  SSHAGL(I) = IDINT(SHAGL(I))
```

```
            IF (ADJ1.EQ.1) THEN
               WRITE(8,36)DHAGL(I),MHAGL(I),SSHAGL(I),CFAZD(I),
     *                    CFAZM(I),CFAZS(I),HGD(I)
            ELSE
               WRITE(8,37)ANGD(I),ANGM(I),ANGS(I),CFAZD(I),CFAZM(I),
     *                    CFAZS(I),HGD(I)
            END IF
            WRITE(8,38) II,CEGRID(I),CNGRID(I)
 240     CONTINUE
C
C         CALL RDMS ( HANG(NN),DUM1D,DUM1M,DUM1S)
            IF (ADJ1.EQ.1) THEN
               DHAGL(NN) = DANG
               MHAGL(NN) = MANG
               SSHAGL(NN) = IDINT(SANG)
C
               WRITE(8,36)DHAGL(NN),MHAGL(NN),SSHAGL(NN),TEMD,TEMM,TEMS
            ELSE
               CALL RDMS (HANG(NN),DUM1D,DUM1M,DUM1S)
                     ANGD(NN) = IDINT(DUM1D)
                     ANGM(NN) = IDINT(DUM1M)
                     ANGS(NN) = DUM1S
               WRITE(8,37)ANGD(NN),ANGM(NN),ANGS(NN),TEMD,TEMM,TEMS
            END IF
C
          WRITE(8,30)
          WRITE(8,39)AZD34,AZM34,AZS34,GE3,GN3
          WRITE(8,*) ' '
          IF (ADJ1.EQ.1) THEN
             WRITE(8,41)EAZ1D,EAZ1M,EAZ1S
             WRITE(8,42)EAZ2D,EAZ2M,EAZ2S
             WRITE(8,43)TODIS
             WRITE(8,44)DIFDIS
             WRITE(8,45) ERR
          END IF
C
C   CORRECTED OBSERVED ANGLES
C
          DO 250 I=1,NN
             HANG(I) = HANG(I)-CORAZ
 250     CONTINUE
C
          SUMDX = 0.0D0
          SUMDY = 0.0D0
C
C
 2000 CONTINUE
C
C
       SUMDX = GE3-CEGRID(N)
       SUMDY = GN3-CNGRID(N)
       DIFDIS= DSQRT ( SUMDX**2 + SUMDY**2 )
C
       WRITE(8,41) EAZ1D,EAZ1M,EAZ1S
       WRITE(8,44) DIFDIS
C
 10    FORMAT(5X,F15.5,5X,F15.10,15X,F12.8)
 11    FORMAT(26X,I4,35X,F7.3)
 12    FORMAT(20X,I1,4X,F9.3,11X,2I3,F10.4)
 13    FORMAT(25X,F15.6)
 14    FORMAT(6X,3A4)
 15    FORMAT(5X,2I3,F14.8)
 16    FORMAT(10X,F15.5,15X,F15.5)
 17    FORMAT(25X,2I3,F10.4)
 18    FORMAT(5A4)
 19    FORMAT(2A4,17X,2I3,F10.4)
 20    FORMAT(I1)
C
 27    FORMAT(16X,'::::::::::::::::::::::::::::::',///)
```

```
 28    FORMAT(16X,'ADJUSTED TRAVERSE COMPUTATION')
 29    FORMAT(15X,'UNADJUSTED TRAVERSE COMPUTATION')
 30    FORMAT('--------------------------------------------------------------
      *--------')
 31    FORMAT(': STN:  OBSERVED  : FORWARD    :  GRID  :     COORDINATE (M
      *.)      :')
 40    FORMAT(': STN:  CORR. OBS.: FORWARD    :  GRID  : ADJUSTED COORDINA
      *TE(M)  :')
 32    FORMAT(':      :  HOR. ANG. : AZIMUTHS   :  DIST. :-----------------
      *-------:')
 33    FORMAT(': NO.:  D  M  S   :  D  M  S   :  (M.)  : GRID EAST :  GRID
      *NORTH  :')
 34    FORMAT('I    :',13X,I3,1X,I2,1X,F5.2,34X,'I')
 35    FORMAT('I  1:',36X,F10.3,1X,F11.3,1X,'I')
 36    FORMAT('I',3X,':',I3,1X,I2,1X,I2,4X,I3,1X,I2,1X,F5.2,1X,F8.3,25X
      * ,'I')
 37    FORMAT('I',3X,':',I3,1X,I2,1X,F5.2,1X,I3,1X,I2,1X,F5.2,1X,F8.3,
      *25X,'I')
 38    FORMAT('I',I3,':',36X,F10.3,1X,F11.3,1X,'I')
C39    FORMAT(1X,'KNOWN DATA',7X,I3,1X,I2,1X,F5.2,11X,F10.3,1X,F11.3,//)
C41    FORMAT('ANGULAR ERROR            =',2X,I3,1X,'D',1X,I2,1X,'M',1X,F5.
C     *2,1X,'S')
C42    FORMAT('ANGULAR CORR./ STATION =',2X,I3,1X,'D',1X,I2,1X,'M',1X,F5.
C     *2,1X,'S')
C43    FORMAT('TOTAL DISTANCES          =',2X,F10.3)
C44    FORMAT('LINEAR ERROR             =',2X,F10.3)
C45    FORMAT('LINEAR CLOSURE           =',3X,'1 /',I6,//////////)
        STOP
        END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                              C
C  SUBROUTINE TO CALCULATE GRID AZIMUTH AND DISTANCE BETWEEN C
C  TWO KNOWN STATIONS WITH THEIRS GRID NORTH AND GRID EAST.  C
C                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        SUBROUTINE GRIDAZ (X1,Y1,X2,Y2,ANGLR,DIST12)
C
C
        REAL*8 X1,X2,Y1,Y2,ANGLR,ANG0,ANG90,ANG180,ANG270,DIFX,DIFY,DIST12
        REAL*8 ANG90R,AN180R,AN270R
C
        DATA ANG0/0.0D0/,ANG90/90.0D0/,ANG180/180.0D0/,ANG270/270.0D0/
C
        CALL DMSR ( ANG90,ANG0,ANG0,ANG90R )
        CALL DMSR ( ANG180,ANG0,ANG0,AN180R )
        CALL DMSR ( ANG270,ANG0,ANG0,AN270R )
C
        DIFX = X2-X1
        DIFY = Y2-Y1
C
        IF((DIFX.EQ.ANG0).AND.(DIFY.EQ.ANG0)) THEN
           ANGLR = ANG0
        ELSE IF(DIFX.EQ.ANG0) THEN
              IF(DIFY.GT.ANG0) THEN
                 ANGLR = ANG0
              END IF
                 ANGLR = AN180R
        ELSE
              IF(DIFX.GT.ANG0) THEN
                 ANGLR = ANG90R-DATAN(DIFY/DIFX)
              ELSE
                 ANGLR = AN270R-DATAN(DIFY/DIFX)
              END IF
        END IF
C
C
        DIST12 = DSQRT(DIFX**2+DIFY**2)
```

```
C
      RETURN
      END
C
C
C
C   @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C   @                                                              @
C   @ SUBROUTINE OUTPUT TO PRINT OUT THE RESULT FOR USING IN        @
C   @ THE NEXT COMPUTATION.                                        @
C   @                                                              @
C   @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
      SUBROUTINE OUTPUT(N,A,B,G2,E2,G3,E3,RD,DH,MH,SH,DA,MA,SA,NAME1,
     *NAME3,NAME4,NAME,MDIST,G1,E1,G4,E4)
C
      DIMENSION RD(N),DH(N),MH(N),SH(N),NAME(3,N),NAME1(5),NAME3(2)
     * ,NAME4(5),MDIST(N)
      REAL*8 A,B,DG,G2,G3,E2,E3,RD,AA,BB,HH,SH,SA,SSA,SIGA,SIGD
      REAL*8 MDIST,MDIS,RRD,G1,G4,E1,E4,DUM1
      INTEGER N,DH,MH,DA,MA
C
      DATA SIGA/5.0D0/,DUM1/0.30480061D0/
C
      WRITE(7,240)
      WRITE(7,250)
      WRITE(7,260)
      WRITE(7,250)
      WRITE(7,270)
      WRITE(7,280)
      WRITE(7,290)
      WRITE(7,280)
      WRITE(7,240)
      WRITE(7,280)
      WRITE(7,300)(NAME1(L),L=1,5),G1,E1
      WRITE(7,310)(NAME(L,1),L=1,3),G2,E2
      WRITE(7,310)(NAME3(L),L=1,3),G3,E3
      WRITE(7,300)(NAME4(L),L=1,5),G4,E4
      WRITE(7,280)
      WRITE(7,240)
      NN = N+1
C
      WRITE(6,230)
      WRITE(6,100)
      WRITE(6,110)
      WRITE(6,120)
      WRITE(6,110)
      WRITE(6,125)
      WRITE(6,140)
      WRITE(6,130)
      WRITE(6,140)
      WRITE(6,100)
      WRITE(6,140)
C
      N1 = N+1
C
      DO 10 I=2,N
         WRITE(6,150)(NAME(L,I-1),L=1,2),(NAME(L,I),L=1,2),DH(I-1),MH(I-
     *   1),SH(I-1)
 10   CONTINUE
      WRITE(6,150)(NAME(L,N),L=1,2),(NAME3(L),L=1,2),DH(N),MH(N),SH(N)
      WRITE(6,150)(NAME3(L),L=1,2),(NAME4(L),L=1,2),DA,MA,SA
      WRITE(6,140)
      WRITE(6,100)
C
C
      WRITE(6,230)
      WRITE(9,170)
```

```
      WRITE(9,180)
      WRITE(9,190)
      WRITE(9,180)
      WRITE(9,170)
      WRITE(9,200)
      WRITE(9,210)
      WRITE(9,200)
      WRITE(9,170)
      WRITE(9,200)
C
      DO 20 J=1,N-1
         MDIS = MDIST(J)*DUM1
         RRD  = RD(J)/DUM1
         SIGD = (RD(J)*0.00001D0)+0.01D0
         WRITE(9,220)(NAME(L,J),L=1,2),(NAME(L,J+1),L=1,2),MDIS,
     *   MDIST(J),RD(J),RRD
 20   CONTINUE
      MDIS = MDIST(N)*DUM1
      RRD  = RD(N)/DUM1
      WRITE(9,220)(NAME(L,N),L=1,2),(NAME3(L),L=1,2),MDIS,MDIST(N)
     *,RD(N),RRD
      SIGD = (RD(N)*0.00001D0)+0.01D0
      WRITE(9,200)
      WRITE(9,170)
C
C
 40   FORMAT(2I3,F6.2)
100   FORMAT(9X,'*****************************************************
     *******')
110   FORMAT(9X,'*                              *         *         *
     *     *')
120   FORMAT(9X,'*  MEASURED HORIZONTAL ANGLE  *         *         *
     *     *')
125   FORMAT(9X,'****************************** DEGREES * MINUTES * SEC
     *ONDS *')
130   FORMAT(9X,'*      AT      *      TO      *         *         *
     *     *')
140   FORMAT(9X,'*              *              *         *         *
     *     *')
150   FORMAT(9X,'*',3X,2A4,3X,'*',3X,2A4,3X,'*',3X,I3,3X,'*',4X,I2,3X,
     *'*',2X,F5.2,2X,'*')
170   FORMAT(5X,'#####################################################
     *###############')
180   FORMAT(5X,'#                 #                          #
     *           #')
190   FORMAT(5X,'#    DISTANCES    #    MEASURED DISTANCES    #    REDUCE
     *D DISTANCES    #')
200   FORMAT(5X,'#        #        #            #             #
     * #         #')
210   FORMAT(5X,'#  FROM  #   TO   #   METERS   #    FEET     #   METERS
     * #    FEET     #')
220   FORMAT(5X,'#',2A4,'#',2A4,'#',2X,F8.3,2X,'#',2X,F8.2,2X,'#',2X,
     *F8.3,1X,'#',2X,F8.2,2X,'#')
230   FORMAT('1')
240   FORMAT(9X,'*****************************************************'
     *)
250   FORMAT(9X,'*                    *                               *'
     *)
260   FORMAT(9X,'*       NAME         *          COORDINATES          *'
     *)
270   FORMAT(9X,'*        OF          *********************************'
     *)
280   FORMAT(9X,'*                    *               *               *'
     *)
290   FORMAT(9X,'*   KNOWN POSITIONS  *  GRID NORTH   *   GRID EAST   *'
     *)
300   FORMAT(9X,'*',5A4,'*',2X,F11.3,2X,'*',2X,F11.3,2X,'*')
310   FORMAT(9X,'*',4X,3A4,4X,'*',2X,F11.3,2X,'*',2X,F11.3,2X,'*')
      RETURN
```

```
      END
C
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @                                 @
C     @    CONVERT DEGREES, MINUTES,    @
C     @    AND SECONDS TO RADIANS       @
C     @                                 @
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
      SUBROUTINE  DMSR ( D1,M1,S1,R1 )
C
      REAL*8  D1,M1,S1,R1,PI1,N60,N3600,N180
C
      DATA N60/60.0D0/,N3600/3600.0D0/,N180/180.0D0/
C
      PI1 = 4.0D0*DATAN(1.0D0)
      R1  = ((D1+(M1/N60)+(S1/N3600))*PI1)/N180
C
      RETURN
      END
C
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @                                 @
C     @   COMPUTE THE COORDINATES       @
C     @   FROM ONE KNOWN POINT WITH     @
C     @   MEASURED DISTANCE AND ANGLE   @
C     @                                 @
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
      SUBROUTINE  UTM (X1,Y1,DIS12,ANG12R,X2,Y2 )
C
      REAL*8  X1,Y1,X2,Y2,DIS12,ANG12R,DIFX,DIFY
C
      DIFX = DIS12*(DSIN(ANG12R))
      DIFY = DIS12*(DCOS(ANG12R))
      X2   = X1+DIFX
      Y2   = Y1+DIFY
C
      RETURN
      END
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @                                 @
C     @    CONVERT ANGLE FROM RADIAN    @
C     @    TO DEGREE,MINUTES,SECONDS    @
C     @                                 @
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
      SUBROUTINE  RDMS ( RR1,D2,M2,S2 )
C
      REAL*8  RR1,D2,M2,S2,PI1,N60,N180,TDEG,DIF,TMIN
C
      DATA  N60/60.0D0/,N180/180.0D0/
C
      PI1 = 4.0D0*DATAN(1.0D0)
      TDEG= (N180*RR1)/PI1
      D2  = DFLOAT(IDINT(TDEG))
      DIF = TDEG-D2
      TMIN= DIF*N60
      M2  = DFLOAT(IDINT(TMIN))
      DIF = TMIN-M2
      S2  = DIF*N60
C
      RETURN
      END
C
```

APPENDIX C

INDTRA FORTRAN PROGRAM

INDTRA Fortran program is used for computing and adjusting traverse station position by least squares method of observation equations (indirect observation) and simultaneously adjust the intersection points which are observed at each traverse station.

```
C      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C      @@                                   @@
C      @@   LEAST SQUARE ADJUSTMENT OF      @@
C      @@   OBSERVATION EQUATIONS METHOD    @@
C      @@                                   @@
C      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
C      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C      @                                                          @
C      @  N,NA = NUMBER OF OBS. ANGLES                            @
C      @  NR1  = NUMBER OF RESECTION PT.                          @
C      @  ND   = OBS. DISTANCES                                   @
C      @  NT   = OBS. DIST + OBS. ANG.                            @
C      @  NC   = TRAV. STN. + RESECT. PT.                         @
C      @         (NOT INCLUDE KNOWN STN.)                         @
C      @  NZ1  = NUMBER OF RESECT. ANGLE AT EACH TRAV. STN.       @
C      @  EGRID = GRID EAST OF KNOWN POINT                        @
C      @  NGRID = GRID NORTH OF KNOWN POINT                       @
C      @  D,M,S = READ ANGLE IN DEGREES, MINUTES, SECONDS         @
C      @  RANG, DIST = GRID DISTANCES                             @
C      @  NCO,ECO = GRID NORTH AND EAST OF INTERSECTION PT.       @
C      @  STDA  = STANDARD DEVIATION OF ANG.                      @
C      @  STDD  = STANDARD DEVIATION OF DIST.                     @
C      @  NP2   = NUMBER OF UNKNOWN DX,DY                         @
C      @  CEGRID,CNGRID = TRAV. STN. COORDI.                      @
C      @  INDLSQ = SUBROUTINE ADJUSTING THE COORDINATES           @
C      @           BY OBSERVATION EQUATIONS                       @
C      @  CALFM = SUBROUTINE FOR COMPUTING CONSTANT VECTOR        @
C      @  CALAM = SUBROUTINE FOR COMPUTING THE COEFFICIENTS       @
C      @          OF UNKNOWN PARAMETERS                           @
C      @                                                          @
C      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
C
       DOUBLE PRECISION  NGRID(4),EGRID(4),AZ21,DIST21,AZ34,
      *                  DIST34,XGD,YGD,AZFIR,NUM360,NUMO,
      *                  AN360R,DD,MM,SS,R,ANG1(40),STDA(40),
      *                  DUM10,RANG,DIST(40),NCO,ECO,CNGRID(40),
      *                  CEGRID(40),STDD(40),AZIMUT,DIS(40),
      *                  COAZR(40),DUMMY1,DUMMY2,AZC,CFAZS(40),
      *                  DUM1D,DUM1M,DUM1S,DELX(40),DELY(40)
       DOUBLE PRECISION  SUMDX,SUMDY,CBAZS(40),TTCOX(40),TTCOY(40),
      *                  AZLAS,DIFAZI,WPRED(3),EAZ1S,CORAZ,DIFDIS,
      *                  EAZ2S,TODIS,ASNY(40),ASEX(40),DUM11,
      *                  DUM12
       DOUBLE PRECISION  WM(80,80),STDO(80),FM(80),BM(80,80),
      *                  BMT(80,80),BMTWM(80,80),NM(80,80),TM(80),
      *                  DELTA(80),NMI(80,80),WK10(4288),VM(1,80),
```

```
     *                   STDAD,XCOX(80),YCOY(80)
C
C
      INTEGER            NT,NA,NP,NP2,ND,NI,PN,ASP,PRINTD,N,NSTA,
     *                   D,M,S,CFAZD(40),CFAZM(40),CBAZD(40),K,KK,
     *                   K4,CBAZM(40),EAZ1D,EAZ1M,EAZ2D,EAZ2M,N2,
     *                   NUM(40),ERR,NR1,NZ1(40),K10,KON,II,II1,NC
C
C
      DATA NUM360/360.0D0/,NUM0/0.0D0/,DUM10/2.0D0/
      DATA DUM11/0.00001D0/,DUM12/0.02D0/,SUMDX/0.0D0/
      DATA SUMDY/0.0D0/,TODIS/0.0D0/
C
      DATA PW/0/,PRINTD/1/
C
C@@   PW = 1 UNWEIGHT
C@@   PRINTD = 0 NOT PRINT DETAILS OF MATRIX
C
C
C     DETERMINE BACK AZIMUT AT STARTING STATION AND FORWARD
C     AZIMUTH AT CLOSING STATION
C
C
      DO 1 I=1,4
         READ(5,*) NGRID(I)
         READ(5,*) EGRID(I)
 1    CONTINUE
C
      CALL GRIDAZ (EGRID(2),NGRID(2),EGRID(1),NGRID(1),AZ21,DIST21)
      CALL GRIDAZ (EGRID(3),NGRID(3),EGRID(4),NGRID(4),AZ34,DIST34)
C
C
C     COMPUTE APPROXIMATED COORDINATE OF TRAVERSE STATION POSITIONS
C
C
C@@@  READ NUMBER OF OBSERVED ANGLES
C
      READ(5,*) N
C
C@@@@ READ NUMBER OF RESECTION POINT
C
      READ(5,*) NR1
C
C
      NSTA  =  N-1
      XGD   =  EGRID(2)
      YGD   =  NGRID(2)
      AZFIR =  AZ21
C
      CALL DMSR (NUM360,NUM0,NUM0,AN360R)
C
      K = 0
C
      DO 100 I=1,NSTA
C       READ ANGLE IN DEGREE, MINUTE, SECOND
          READ(5,105) D,M,S
               DD = DFLOAT(D)
               MM = DFLOAT(M)
               SS = DFLOAT(S)
C
               CALL DMSR (DD,MM,SS,R)
C
                      ANG1(I) = R
                      STDA(I) = DUM10
C
C            READ REDUCE GRID DISTANCES
               READ(5,*) RANG
                    DIST(I) = RANG
C
```

```
                READ(5,*) NZ1(I)
C
C
C
                IF (RANG.EQ.0.0) THEN
C
C                READ APPROX. GRID COORDINATES OF RESECTION STATION
                   READ(5,*) NCO
                   READ(5,*) ECO
C
                       CNGRID(I) = NCO
                       CEGRID(I) = ECO
                       GO TO 100
                 END IF
C
                 K = K+1
                 DIS(K)  =  DIST(I)
                 STDD(K) = (DIS(K)*DUM11) + DUM12
C
                 AZIMUT  = AZFIR + ANG1(I)
                 IF (AZIMUT.GT.AN360R) THEN
                    AZIMUT = AZIMUT - AN360R
                 END IF
C
                 COAZR(K) = AZIMUT
C
C@@@@@ FOR PRINTING
C
                 CALL RDMS (AZIMUT,DUM1D,DUM1M,DUM1S)
                      CFAZD(K) = IDINT(DUM1D)
                      CFAZM(K) = IDINT(DUM1M)
                      CFAZS(K) = DUM1S
C@@@@@
C
                 CALL UTM (XGD,YGD,RANG,AZIMUT,DUMMY1,DUMMY2)
C
                      DELX(K)  =  DUMMY1-XGD
                      DELY(K)  =  DUMMY2-YGD
C
                      SUMDX    =  SUMDX + DELX(K)
                      SUMDY    =  SUMDY + DELY(K)
C
                      CEGRID(I)=  DUMMY1
                      CNGRID(I)=  DUMMY2
C
C
                CALL GRIDAZ (DUMMY1,DUMMY2,XGD,YGD,AZC,DUM1S)
C
C@@@@@ FOR PRINTING
                CALL RDMS (AZC,DUM1D,DUM1M,DUM1S)
C
                     CBAZD(K) = IDINT(DUM1D)
                     CBAZM(K) = IDINT(DUM1M)
                     CBAZS(K) = DUM1S
C@@@@@
C
                AZFIR = AZC
                XCOX(K) = DUMMY1
                YCOY(K) = DUMMY2
C
C
                XGD      = DUMMY1
                YGD      = DUMMY2
                TODIS    = TODIS + RANGE
C
C
C
C
 100  CONTINUE
```

```
C
C
      K10 = K-1
C
      DO 9 I =1,K10
         TTCOX(I) = XCOX(I)
         TTCOY(I) = YCOY(I)
 9    CONTINUE
C
C
      KK = K+1
C
      READ (5,105) D,M,S
            DD = DFLOAT(D)
            MM = DFLOAT(M)
            SS = DFLOAT(S)
C
      READ (5,*) RANG
            DIST(N) = RANG
C
C
      CALL DMSR (DD,MM,SS,R)
C
            ANG1(N) = R
            STDA(N) = DUM10
C
      AZLAS = AZFIR + ANG1(N)
      IF (AZLAS.GT.AN360R) THEN
         AZLAS = AZLAS - AN360R
      END IF
C
      DIFAZI  =  AZLAS - AZ34
      WPRED(1)=  DIFAZI
      CORAZ   =  DIFAZI / DFLOAT(KK)
C
C@@@@@ FOR PRINTING
C
      CALL RDMS (DIFAZI,DUM1D,DUM1M,DUM1S)
C
            EAZ1D  =  IDINT( ABS (DUM1D) )
            EAZ1M  =  IDINT( ABS (DUM1M) )
            EAZ1S  =  ABS (DUM1S)
C
      CALL RDMS (CORAZ,DUM1D,DUM1M,DUM1S)
C
            EAZ2D  =  IDINT( ABS (DUM1D) )
            EAZ2M  =  IDINT( ABS (DUM1M) )
            EAZ2S  =  ABS (DUM1S)
C@@@@@
C
      WPRED(2)  =  SUMDX - (EGRID(3) - EGRID(2))
      WPRED(3)  =  SUMDY - (NGRID(3) - NGRID(2))
C
      DIFDIS    =  DSQRT( (WPRED(2))**2 + (WPRED(3))**2 )
      ERR       =  IDINT( TODIS / DIFDIS )
C
C     READ NUMBER OF MEASURED ANGLES AT EACH TRAVERSE STATION
C
      DO 101 J=1,KK
         READ(5,*) N2
            NUM(J) = N2
 101  CONTINUE
C
C
      DO 108 I = 1,NR1
         II = K10+I
         READ(5,*) DUMMY1
         READ(5,*) DUMMY2
         TTCOY(II) = DUMMY1
```

```
            TTCOX(II) = DUMMY2
C
 108   CONTINUE
C
C
C
        ASP = NSTA - 1
C
C@@    KEPP APPROX. GRID COORDINATES OF EACH STATION IN
C@@    ASNY AND ASEX
C
        DO 102 I=1,ASP
            ASNY(I) = CNGRID(I)
            ASEX(I) = CEGRID(I)
C
 102   CONTINUE
C
C@@    NUMBER OF OBSERVED ANGLES PLUS OBSERVED DISTANCES
C
        NT = N+K
C
C@@    NUMBER OF OBSERVED ANGLES
C
        NA = N
C
C@@    NUMBER OF OBSERVED DISTANCES
C
        ND = K
C
C@@    NUMBER OF TRAVERSE STATIONS NOT INCLUDED KNOWN STATIONS
C
        NP = ASP
C
C@@    NUMBER OF UNKNOWN DX AND DY
C
        NP2 = K10*2 + NR1*2
C
C
C@@
        NI = NP2**2 + 3*NP2
C
C@@    NUMBER OF TRAVERSE STATION (NOT INCLUDE KNOWN STATION)
C      PLUSE NUMBER OF INTERSECTION POINT
C
        NC = K10+NR1
C
C@@@@@ CALL SUBROUTINE TO ADJUST STATION POSITIONS
C       BY LEAST SQUARE METHOD OF OBSERVATION EQUATION
C
        CALL INDLSQ (NC,NR1,NZ1,NUM,KK,K,DIST,TTCOX,TTCOY,NT,NA,ND,NP,
     *                NP2,NI,AZ21,AZ34,NGRID,EGRID,ANG1,STDA,DIS,STDD,
     *                PRINTD,PW,WM,STDO,FM,BM,BMT,BMTWM,NM,TM,
     *                DELTA,NMI,WK10,ASNY,ASEX,VM,STDAD)
C
C
C@@@@   PRINTING DETAILS
C
        WRITE(6,*)'STANDARD DEVIATION OF UNIT WEIGHT =',STDAD
        WRITE(6,*)' '
        WRITE(6,*)'STANDARD DEVIATION OF ADJ. ANGLES'
C
        DO 103 I=1,NA
            CALL RDMS (VM(1,I),DUM1D,DUM1M,DUM1S)
                DUMMY1 = DUM1D*3600.0D0 + DUM1M*60.0D0 + DUM1S
                WRITE(6,*)'I =',I,'     ADJ. STDA. =',DUMMY1
 103   CONTINUE
C
        WRITE(6,*)'STANDARD DEVIATION OF ADJ DISTANCES'
C
```

```
      DO 104 I=1,ND
          K4 = NA+I
          WRITE(6,*)' '
          WRITE(6,*) 'I =',I,'ADJ. STDD. =',VM(1,K4)
 104  CONTINUE
C
C
 105  FORMAT(I3,1X,I2,1X,I2)
C
      STOP
      END
C
C
C
C
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @@                                    @@
C     @@   SUBROUTINE LEAST SQUARE          @@
C     @@   ADJUSTMENT OF INDIRECT           @@
C     @@   OBSERVATIONS                     @@
C     @@                                    @@
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
      SUBROUTINE INDLSQ (NC1,NR,NZZ,NUMM,KK,K,DIST1,TCOX,TCOY,NT,NA,
     *                   ND,NP,NP2,NI,AZ21,AZ34,NGRID,EGRID,ANG,
     *                   STDA,DIS1,STDD,PRINTD,PW,WM,STDO,FM,
     *                   BM,BMT,BMTWM,NM,TM,DELTA,NMI,WK10,
     *                   ASNG,ASEG,VM,STAND)
C
      DOUBLE PRECISION  DUMMY1,NGRID(4),EGRID(4),DUM1D,DUM1M,
     *                  DUM1S,ANG(NA),DIS1(ND),WM(NT,NT),STDO(NT),
     *                  ASNG(NP),ASEG(NP),FM(NT),BM(NT,NP2),TEST1,
     *                  TEST2,BMT(NP2,NT),BMTWM(NP2,NT),NM(NP2,NP2),
     *                  TM(NP2),DELTA(NP2),NMI(NP2,NP2),WK10(NI),
     *                  STDA(NA),STDD(ND),DUMMY2,NUMO,AZ21,AZ34,
     *                  VM(1,NT),STAND,DIST1(NA),TCOX(NC1),
     *                  TCOY(NC1)
C
C
      INTEGER           I,J,K1,IER,PRINTD,PW,CHECK,NUMM(KK),K,KK,
     *                  NR,NZZ(NA),KON,III,NC1,NNDI
C
C
      DATA K1/0/,TEST1/0.0001000000D0/,NUMO/0.0D0/
C
C
      NC4  =  4
      NP5  =  6
      N1   =  1
C
C@@  SET WEIGHT MATRIX
C
      DO 30 I=1,NT
          DO 20 J=1,NT
              WM(I,J) = 0.0D0
              IF (I.EQ.J) THEN
                  WM(I,J) = 1.0D0
              END IF
 20       CONTINUE
 30   CONTINUE
C
C@@ FOR STD. ANGLE
C
      IF (PW.NE.1) THEN
          DO 40 I=1,NA
              DUMMY1 = STDA(I)
```

```
                CALL DMSR (NUMO,NUMO,DUMMY1,DUMMY2)
                    STDO(I) = 1.0DO / ( (DSIN(DUMMY2) )**2)
 40      CONTINUE
C
C@@ FOR STD. DISTANCE
C
          DO 50 I=1,K
              J = NA+I
              DUMMY1 = STDD(I)
              STDO(J)= 1.0DO / (DUMMY1**2)
 50      CONTINUE
C
C@@ SET UNEQUAL WEIGHT
C
          DO 70 I=1,NT
              DO 60 J=1,NT
                  IF (I.EQ.J) THEN
                      WM(I,J) = STDO(I)
                  END IF
 60        CONTINUE
 70      CONTINUE
C
        END IF
C
C@@ PRINT WEIGHT MATRIX
C
        IF (PRINTD.NE.0) THEN
            WRITE(6,*) 'WEIGHT MATRIX'
            WRITE(6,*) ' '
            CALL USWFM ('R-C.',NC4,WM,NT,NT,NT,NP5)
        END IF
C
C
 99    CONTINUE
C
C
        K1 = K1+1
C
        IF (K1 .GT. 2) THEN
            GO TO 999
        END IF
C
        CHECK = 0
C
C@@ CALL SUBROUTINE TO CALCULATE  "F" MATRIX
C
C
        CALL CALFM (NUMM,KK,K,DIST1,TCOX,TCOY,AZ21,AZ34,NA,ND,
     *             NT,NGRID,EGRID,ANG,DIS1,NP,ASNG,ASEG,FM)
C
C
C
C
C@@ CALL SUBROUTINE TO CALCULATE "A" MATRIX (COEFFICEINT DX,DY)
C
        CALL CALAM (NR,NZZ,NUMM,KK,K,DIST1,TCOX,TCOY,NT,NA,ND,NP,
     *             NP2,NGRID,EGRID,ASNG,ASEG,BM)
C
C
C@@ TRANSPOSE OF "A" MATRIX
C
        DO 120 I=1,NT
            DO 110 J=1,NP2
                BMT(J,I) = BM(I,J)
C
 110      CONTINUE
 120   CONTINUE
C
C@@ "A" TRANSPOSE * W
```

```
C
        CALL VMULFF (BMT,WM,NP2,NT,NT,NP2,NT,BMTWM,NP2,IER)
C
C@@ "A" TRANPOSE * W * A
C
        CALL VMULFF (BMTWM,BM,NP2,NT,NP2,NP2,NT,NM,NP2,IER)
C
C@@ "A" TRANSPOSE * W * F
C
        CALL VMULFF (BMTWM,FM,NP2,NT,N1,NP2,NT,TM,NP2,IER)
C
C
C
C@@ (INVERSE OF ("A" TRANSPOSE W * A) )
C
        CALL LINV2F (NM,NP2,NP2,NMI,N1,WK10,IER)
C
C
C@@ CALCULATE DELTA MATRIX (COORECTION VECTOR)
C
C    (INVERSE OF ("A" TRANSPOSE W * A) * "A" TRANSPOSE * W * F )
        CALL VMULFF (NMI,TM,NP2,NP2,N1,NP2,NP2,DELTA,NP2,IER)
C
C@@ UPDATE APPROX. VALUE
C
C
C      KON = K-1+NR
        DO 130 I=1,NC1
            TCOX(I) = TCOX(I) + DELTA((I-1)*2 + 1)
            TCOY(I) = TCOY(I) + DELTA((I-1)*2 + 2)
C
 130  CONTINUE
C
C
        II1 = (K-1)*2
        DO 131 I=1,NP
C
            IF (NZZ(I) .EQ. 0) THEN
                ASEG(I) = ASEG(I) + DELTA( (I-1)*2 + 1)
                ASNG(I) = ASNG(I) + DELTA( (I-1)*2 + 2)
            ELSE
                DO 132 J =1,NR
                    IF (NZZ(I) .EQ. J) THEN
                        ASEG(I) = ASEG(I) + DELTA( II1+(J-1)*2 +1)
                        ASNG(I) = ASNG(I) + DELTA( II1+(J-1)*2 +2)
                        GO TO 131
                    END IF
 132          CONTINUE
            END IF
C
 131  CONTINUE
C
C
        IF (PRINTD .NE. 0) THEN
            WRITE(6,1020) K1
            WRITE(6,1030)
                CALL USWFM ('R-C.',NC4,FM,NT,NT,N1,NP5)
            WRITE(6,1000)
            WRITE(6,1040)
                CALL USWFM ('R-C.',NC4,BM,NT,NT,NP2,NP5)
            WRITE(6,1000)
            WRITE(6,1050)
                CALL USWFM ('R-C.',NC4,BMT,NP2,NP2,NT,NP5)
            WRITE(6,1000)
            WRITE(6,1060)
                CALL USWFM ('R-C.',NC4,BMTWM,NP2,NP2,NT,NP5)
            WRITE(6,1000)
            WRITE(6,1070)
                CALL USWFM ('R-C.',NC4,NM,NP2,NP2,NP2,NP5)
```

```
          WRITE(6,1000)
          WRITE(6,1080)
               CALL USWFM ('R-C.',NC4,TM,NP2,NP2,N1,NP5)
          WRITE(6,1000)
          WRITE(6,1090)
               CALL USWFM ('R-C.',NC4,NMI,NP2,NP2,NP2,NP5)
          WRITE(6,1000)
          WRITE(6,1100)
               CALL USWFM ('R-C.',NC4,DELTA,NP2,NP2,N1,NP5)
          WRITE(6,1000)
          WRITE(6,1110)
C
          DO 140 I=1,NP
             WRITE(6,*) I,' N = ', ASNG(I),' E = ',ASEG(I)
             WRITE(6,*)
 140     CONTINUE
C
       END IF
C
C
C@@ CHECK DELTA MATRIX
C
C
       DO 180 I=1,NP2
          TEST2 = DABS (DELTA(I))
C
          IF ( TEST2 .GE. TEST1) THEN
             CHECK = 1
          END IF
 180  CONTINUE
C
C
       IF (CHECK .NE. 0) THEN
         GO TO 99
       END IF
C
C@@ CALCULATE "A" * X  MATRIX
C
       CALL VMULFF (BM,DELTA,NT,NP2,N1,NT,NP2,STDO,NT,IER)
C
C@@ CALCULATE "V" MATRIX
C
       DO 190 I=1,NT
          STDO (I) = STDO(I) - FM(I)
          VM(1,I)  = STDO(I)
 190  CONTINUE
C
C
C     CORRECT OBSERVED ANGLES AND DISTANCES
C
       DO 139 I = 1,NA
          ANG(I) = ANG(I)+(STDO(I))
139   CONTINUE
       DO 149 I = 1,ND
         NND1 = NA+I
          DIS1(I) = DIS1(I)+STDO(NND1)
149   CONTINUE

       IF (PRINTD .NE.0) THEN
          WRITE(6,1000)
          WRITE(6,1120)
               CALL USWFM ('R-C.',NC4,STDO,NT,NT,N1,NP5)
          WRITE(6,1000)
          WRITE(6,1130)
               CALL USWFM ('R-C.',NC4,VM,N1,N1,NT,NP5)
       END IF
C
C
C@@ CALCULATE "V" TRANSPOSE * W * V  MATRIX
```

```
C
      CALL VMULFF (WM,STDO,NT,NT,N1,NT,NT,FM,NT,IER)
      CALL VMULFF (VM,FM,N1,NT,N1,N1,NT,STAND,N1,IER)
C
      IF (PRINTD . NE. 0) THEN
         WRITE(6,1000)
         WRITE(6,1140)
                CALL USWFM ('R-C.',NC4,FM,NT,NT,N1,NP5)
         WRITE(6,1150)
         WRITE(6,1160) STAND
      END IF
C
C
      CALL VMULFF (NMI,BMT,NP2,NP2,NT,NP2,NP2,BMTWM,NP2,IER)
C
      STAND = DSQRT ( STAND / (DFLOAT(NT-NP2)) )
C
      DO 240 I=1,NP2
         DO 230 J=1,NP2
            NMI (I,J) = STAND * (DSQRT(DABS (NMI(I,J)) ) )
C
            IF (I.EQ.J) THEN
               TM(I) = NMI(I,J)
            END IF
C
 230     CONTINUE
 240  CONTINUE
C
C
C@@ CALCULATE "A"*(INVERSE "A"TRANSPOSE*W*A) * "A"TRANSPOSE
C
      CALL VMULFF (BM,BMTWM,NT,NP2,NT,NT,NP2,WM,NT,IER)
C
      IF (PRINTD .NE.0) THEN
         WRITE(6,1000)
         WRITE(6,1170)
         CALL USWFM ('R-C.',NC4,WM,NT,NT,NT,NP5)
      END IF
C
C
      DO 280 I=1,NT
         DO 270 J=1,NT
            WM (I,J) = STAND * ( DSQRT( DABS(WM(I,J)) ) )
C
            IF (I.EQ.J) THEN
               VM(I,I) = WM(I,J)
            END IF
C
 270     CONTINUE
 280  CONTINUE
C
999   RETURN
C
C
1000  FORMAT('1H1')
1020  FORMAT(///,5X,I2,'ITERATED',/)
1030  FORMAT(///,5X,'F MATRIX',/)
1040  FORMAT(///,5X,'A MATRIX',/)
1050  FORMAT(///,5X,'TRANSPOSE OF A MATRIX',/)
1060  FORMAT(///,5X,'A TRANSPOSE * W MATRIX',/)
1070  FORMAT(///,5X,'A TRANSPOSE * W * A MATRIX',/)
1080  FORMAT(///,5X,'A TRANSPOSE * W * F MATRIX',/)
1090  FORMAT(///,5X,'THE INVERSION OF  A TRANSPOSE * W * A',/)
1100  FORMAT(///,5X,'X MATRIX OR DELTA MATRIX',/)
1110  FORMAT(///,5X,'UP-DATE THE APPROX.  VALUES',/)
1120  FORMAT(///,5X,'V MATRIX',/)
1130  FORMAT(///,5X,'V  TRANSPOSE MATRIX',/)
1140  FORMAT(///,5X,'W * V  MATRIX',/)
1150  FORMAT(///,5X,'V TRANSPOSE * W * V  MATRIX',/)
```

```
1160  FORMAT(///,5X,F20.15,/)
1170  FORMAT(///,5X,'A*(INVERSE A TRANSPOSE *W*A)*A TRANSPOSE',/)
C
       END
C
C
C
C
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @@@                                  @@@
C     @@@   SUBROUTINE FOR COMPUTING       @@@
C     @@@   F  MATRIX                      @@@
C     @@@                                  @@@
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
C
       SUBROUTINE CALFM (KOUNT,KK,K,DIS,COX,COY,AZ21,AZ34,
     *                   NNA,NND,NNT,GRY,GRX,OBANG,OBDIS,
     *                   NOS,ASY,ASX,F)

       DOUBLE PRECISION  GRY(4),GRX(4),OBANG(NNA),OBDIS(NND),
     *                   N360,N360R,DUM1,NUMO,CDIS(100),TEMP1,
     *                   TEMP2,TEMP3,TEMP4,ASY(NOS),AZF,AZB,
     *                   ASX(NOS),F(NNT),AZ21,AZ34,DIS(NNA),
     *                   COX(K),COY(K),DISTAN,ADIS(100)
C
C
       INTEGER           I1,I2,K,KK,KKK,N,STN,KOUNT(KK)
C
C
       DATA N360/360.0D0/,NUMO/0.0D0/,N/0/,STN/1/
C
C
       CALL DMSR (N360,NUMO,NUMO,A360R)
C
       DO 500 I=1,NNA
          DISTAN = DIS(I)
C
          IF (STN .EQ. 1) THEN
             TEMP1 = GRX(2)
             TEMP2 = GRY(2)
             AZB   = AZ21
             TEMP3 = ASX(I)
             TEMP4 = ASY(I)
C
             CALL GRIDAZ (TEMP1,TEMP2,TEMP3,TEMP4,AZF,DUM1)
C
                  CDIS(I) = DUM1
                  IF (AZF .GT. AZB) THEN
                     F(I) = OBANG(I)-(AZF-AZB)
                  ELSE
                     F(I) = OBANG(I)-(A360R+AZF-AZB)
                  END IF
C
               N = N+1
               GO TO 400
          END IF
C
          IF (STN .EQ. 2) THEN
             TEMP1 = COX(STN-1)
             TEMP2 = COY(STN-1)
             TEMP3 = GRX(2)
             TEMP4 = GRY(2)
C
             CALL GRIDAZ (TEMP1,TEMP2,TEMP3,TEMP4,AZB,DUM1)
C
```

```
                IF (KK .EQ. 3) THEN
                   TEMP3 = GRX(3)
                   TEMP4 = GRY(3)
                END IF
C
                TEMP3 = ASX(I)
                TEMP4 = ASY(I)
C
          CALL GRIDAZ (TEMP1,TEMP2,TEMP3,TEMP4,AZF,DUM1)
C
C
                CDIS(I) = DUM1
                IF (AZF .GT. AZB) THEN
                   F(I) = OBANG(I)-(AZF-AZB)
                ELSE
                   F(I) = OBANG(I)-(A360R+AZF-AZB)
                END IF
C
           N = N+1
           GO TO 400
C
      END IF
C
C
      IF (STN .EQ. KK) THEN
         AZF   = AZ34
         TEMP1 = GRX(3)
         TEMP2 = GRY(3)
C
C      IF (DISTAN .EQ. 0.) THEN
C         TEMP3 = ASX(I)
C         TEMP4 = ASY(I)
C      ELSE
            TEMP3 = COX(K-1)
            TEMP4 = COY(K-1)
C      END IF
C
         CALL GRIDAZ (TEMP1,TEMP2,TEMP3,TEMP4,AZB,DIS1)
C
                IF (AZF .GT. AZB) THEN
                   F(I) = OBANG(I)-(AZF-AZB)
                ELSE
                   F(I) = OBANG(I)-(A360R+AZF-AZB)
                END IF
C
           N = N+1
           GO TO 400
      END IF
C
C
      IF (STN .EQ. K) THEN
         TEMP1 = COX(K-1)
         TEMP2 = COY(K-1)
         TEMP3 = COX(K-2)
         TEMP4 = COY(K-2)
C
         CALL GRIDAZ (TEMP1,TEMP2,TEMP3,TEMP4,AZB,DUM1)
C
                IF (DISTAN .EQ. 0.) THEN
                   TEMP3 = ASX(I)
                   TEMP4 = ASY(I)
                ELSE
                   TEMP3 = GRX(3)
                   TEMP4 = GRY(3)
                END IF
C
         CALL GRIDAZ (TEMP1,TEMP2,TEMP3,TEMP4,AZF,DUM1)
C
                CDIS(I) = DUM1
```

```
                IF (AZF .GT. AZB) THEN
                   F(I) = OBANG(I)-(AZF-AZB)
                ELSE
                   F(I) = OBANG(I)-(A360R+AZF-AZB)
                END IF
C
             N = N+1
             GO TO 400
          END IF
C
C
          IF (STN .GT. 2) THEN
             TEMP1 = COX(STN-1)
             TEMP2 = COY(STN-1)
             TEMP3 = COX(STN-2)
             TEMP4 = COY(STN-2)
C
             CALL GRIDAZ (TEMP1,TEMP2,TEMP3,TEMP4,AZB,DUM1)
C
                TEMP3 = ASX(I)
                TEMP4 = ASY(I)
C
C
             CALL GRIDAZ (TEMP1,TEMP2,TEMP3,TEMP4,AZF,DUM1)
C
                CDIS(I) = DUM1
                IF (AZF .GT. AZB) THEN
                   F(I) = OBANG(I)-(AZF-AZB)
                ELSE
                   F(I) = OBANG(I)-(A360R+AZF-AZB)
                END IF
                N = N + 1
C
           END IF
C
C
 400      IF ( (DISTAN .NE. 0.) .AND. (STN .LE. K) ) THEN
             ADIS(STN) = CDIS(I)
C             WRITE(6,*)'STN=',STN,'  CDIS=',CDIS(I)
          END IF
C
C
          KKK = KOUNT(STN)
          IF (N .LT. KKK) THEN
             STN = STN
          ELSE
             STN = STN+1
             N   = 0
          END IF
C
C
 500  CONTINUE
C
C
C
        I1 = NNA
        DO 600 I2=1,NND
           I1 = I1+1
           F(I1) = OBDIS(I2) - ADIS(I2)
 600  CONTINUE
C
        RETURN
C
        END
C
C
C
C
C
```

```
C       @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C       @@                                   @@
C       @@     SUBROUTINE FOR CALCULATING    @@
C       @@     A MATRIX                      @@
C       @@                                   @@
C       @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
C
        SUBROUTINE CALAM (NR,NZ,COUNT,KK,K,DISS,COXX,COYY,NT,NA,
      *                   ND,NP,NP2,KNY,KNX,ANY,ANX,AM)
C
C
        DOUBLE PRECISION  KNY(4),KNX(4),ANY(NP),ANX(NP),DISTAN,
      *                   AM(NT,NP2),DISS(NA),COXX(K),COYY(K),
      *                   DU1,DU2,DU3,DU4,DU5,DU6
C
C
        INTEGER           I,I1,I2,I3,I4,I5,J1,J2,K,KK,K1,K2,K3,
      *                   K4,N,NN,STN,TEM,TEM1,TEM2,JJ,JJ1,JJ2,
      *                   NOM,COUNT(KK),TT,TT1,TEM3,NUSED,NZ(NA),
      *                   NR,L,I10
C
C
        DATA STN/1/,N/1/,NN/0/,TT/0/,JJ/1/
C
C
        NP1 = NP2-1
        K1  = NA-1
        K2  = K-1
        K3  = K+1
        I10 = ((K-1)*2) - 1
C
C       @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C       @                            @
C       @ COEFFICEINT OF COORDINATES @
C       @ FOR ANGLE CONDITION        @
C       @                            @
C       @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
        DO 800 I = 1,NA
           DISTAN = DISS(I)
C
           DO 700 J1 = 1,NP1,2
              J2 = J1+1
              AM(I,J1) = 0.0D0
              AM(I,J2) = 0.0D0
C
              IF (STN .EQ. 1) THEN
C
C
                 IF (DISTAN .EQ. 0) THEN
C
                    DO 231 L=1,NR
                       IF (NZ(I) .EQ. L) THEN
                          JJ = I10+2*L
                          GO TO 232
                       END IF
 231             CONTINUE
C
                 ELSE
                    JJ = 1
                    TEM = JJ
                    GO TO 232
                 END IF
C
C
232           IF (J1 .EQ. JJ) THEN
```

```
                  DU1 = ANY(I)-KNY(2)
                  DU2 = ANX(I)-KNX(2)
                  DU3 = (DU1**2)+(DU2**2)
C
                  AM(I,J1) = DU1/DU3
                  AM(I,J2) = -DU2/DU3
C
                  N = N+1
                  GO TO 700
               END IF
C
               GO TO 700
            END IF
C
C     *******************
C
            IF (STN .EQ. 2) THEN
C
               IF (J1 .EQ. JJ) THEN
                  DU1 = ANY(I) - COYY(STN-1)
                  DU2 = ANX(I) - COXX(STN-1)
                  DU3 = (DU1**2)+(DU2**2)
                  DU4 = KNY(2) - COYY(STN-1)
                  DU5 = KNX(2) - COXX(STN-1)
                  DU6 = (DU4**2)+(DU5**2)
C
                  AM(I,J1) = -(DU1/DU3)+(DU4/DU6)
                  AM(I,J2) =  (DU2/DU3)+(DU5/DU6)
C
C
                  IF (DISTAN .EQ. 0) THEN
                     DO 233 L = 1,NR
                        IF ( NZ(I) .EQ. L) THEN
                           JJ1 = I10+2*L
                           GO TO 700
                        END IF
 233                 CONTINUE
C
                   ELSE
                     JJ1 = TEM + 2
                     TEM1 = JJ1
                     GO TO 700
                   END IF
                  GO TO 700
                END IF
C
                IF (J1 .EQ. JJ1) THEN
C
                   AM(I,J1) = (DU1/DU3)
                   AM(I,J2) =-(DU2/DU3)
C
                   N = N+1
                   GO TO 700
                END IF
C
                GO TO 700
            END IF
C
C     ***************
C
            IF ( (STN .GT. 2) .AND. (STN .LE. K2) ) THEN
               K4 = STN - 2
C
               IF (J1 .EQ. JJ) THEN
                   DU1 = ANY(I) - COYY(STN-1)
                   DU2 = ANX(I) - COXX(STN-1)
                   DU3 = (DU1**2)+(DU2**2)
                   DU4 = COYY(K4) - COYY(STN-1)
                   DU5 = COXX(K4) - COXX(STN-1)
```

```
                 DU6 = (DU4**2)+(DU5**2)
C
                 AM(I,J1) = -DU4/DU6
                 AM(I,J2) =  DU5/DU6
C
                 JJ1      =  TEM1
                 GO TO 700
              END IF
C
              IF (J1 .EQ. JJ1) THEN
C
                 AM(I,J1) = -(DU1/DU3) + (DU4/DU6)
                 AM(I,J2) =  (DU2/DU3) - (DU5/DU6)
C
                 IF (DISTAN .EQ. 0) THEN
                    DO 234 L = 1,NR
                       IF (NZ(I) .EQ. L) THEN
                         JJ2 = I10+2*L
                          GO TO 700
                       END IF
  234            CONTINUE
C
                 ELSE
                    JJ2 = TEM1 + 2
                    TEM2 = JJ2
                    GO TO 700
                 END IF
C
C
                 GO TO 700
              END IF
C
              IF (J1 .EQ. JJ2) THEN
C
                  AM(I,J1) =  DU1/DU3
                  AM(I,J2) = -DU2/DU3
C
                  N = N+1
                  GO TO 700
              END IF
              GO TO 700
C
          END IF
C
C         *******************
C
          IF (STN .EQ. K) THEN
             IF (J1 .EQ. JJ) THEN
                DU1 = COYY(K-2) - COYY(K-1)
                DU2 = COXX(K-2) - COYY(K-1)
                DU3 = (DU1**2) + (DU2**2)
C
                AM(I,J1) = -DU1/DU3
                AM(I,J2) =  DU2/DU3
C
                JJ1 = TEM1
                GO TO 700
             END IF
C
             IF (J1 .EQ. JJ1) THEN
                IF (DISTAN .EQ. 0.) THEN
                    DU4 = ANY(I) - COYY(K-1)
                    DU5 = ANX(I) - COXX(K-1)
                    DU6 = (DU4**2) + (DU5**2)
                ELSE
                    TEMP2 = J1
                    DU4 = KNY(3) - COYY(K-1)
                    DU5 = KNX(3) - COXX(K-1)
                    DU6 = (DU4**2) + (DU5**2)
```

```
C
                                AM(I,J1) = -(DU4/DU6) + (DU1/DU3)
                                AM(I,J2) =  (DU5/DU6) - (DU2/DU3)
C
                                N = N + 1
C
                                NUSED = J1

                                JJ2 = 0
                                GO TO 700
                           END IF
                           NN = NN + 1
C
                           AM(I,J1) = -(DU4/DU6) + (DU1/DU3)
                           AM(I,J2) =  (DU5/DU6) - (DU2/DU3)
C
                           DO 235 L = 1,NR
                               IF ( NZ(I) .EQ. L) THEN
                                  JJ2 = I10+2*L
                                  GO TO 700
                               END IF
 235                    CONTINUE
C
C
                           GO TO 700
                        END IF
C
                        IF (J1 .EQ. JJ2) THEN
                           AM(I,J1) =  DU4/DU6
                           AM(I,J2) = -DU5/DU6
C
                           N = N + 1
                           GO TO 700
                         END IF
                         GO TO 700
C
                   END IF
C
C               ******************
C
                   IF (STN .EQ. KK) THEN
                      IF (J1 .EQ. JJ) THEN
                         DU1 = COYY(K-1) - KNY(3)
                         DU2 = COXX(K-1) - KNX(3)
                         DU3 = (DU1**2) + (DU2**2)
C
                         AM(I,J1) = -(DU1/DU3)
                         AM(I,J2) =  (DU2/DU3)
C
                         N = N + 1
                         GO TO 700
                      END IF
C
                      GO TO 700
                    END IF
C
 700       CONTINUE
C
             NOM = COUNT(STN)
C
             IF (N .LE. NOM) THEN
C
                 STN= STN
C
             ELSE
C
                 STN = STN + 1
                 JJ  = TEM
                 N   = 1
```

```
C
              IF (STN .GT. 3) THEN
                 JJ   = TEM1
                 TEM1 = TEM2
              END IF
           END IF
C
 800  CONTINUE
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @ COEFF. OF COORDINATES   @
C     @ FOR DISTANCE CONDITION @
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
      DO 3000 I1 = 1,ND
         I2 = NA + I1
         I3 = I1 - 1
         I4 = I3 + (I3 - 1)
         I5 = I3 + (I3 + 1)
C
         DO 2000 J1 = 1,NP1,2
C
            J2 = J1 + 1
            AM(I2,J1) = 0.0D0
            AM(I2,J2) = 0.0D0
C
            IF (I1 .EQ. 1) THEN
               IF (J1 .EQ. 1) THEN
                  DU1 = KNY(2) - COYY(1)
                  DU2 = KNX(2) - COXX(1)
                  DU3 = DSQRT ( DU1**2 + DU2**2)
C
                  AM(I2,J1) = -(DU2/DU3)
                  AM(I2,J2) = -(DU1/DU3)
C
               END IF
               GO TO 2000
            END IF
C
C
            IF (I1 .EQ. ND) THEN
               IF (J1 .EQ. NUSED) THEN
                  DU1 = COYY(K-1) - KNY(3)
                  DU2 = COXX(K-1) - KNX(3)
                  DU3 = DSQRT ( DU1**2 + DU2**2)
                  AM(I2,J1) =  (DU2/DU3)
                  AM(I2,J2) =  (DU1/DU3)
C
               END IF
               GO TO 2000
            END IF
C
C
            IF (I4 .EQ. J1) THEN
               DU1 = COYY(I3) - COYY(I1)
               DU2 = COXX(I3) - COXX(I1)
               DU3 = DSQRT ( DU1**2 + DU2**2 )
C
               AM(I2,J1) = (DU2/DU3)
               AM(I2,J2) = (DU1/DU3)
C
               GO TO 2000
            END IF
C
            IF (I5 .EQ. J1) THEN
               DU1 = COYY(I3) - COYY(I1)
               DU2 = COXX(I3) - COXX(I1)
               DU3 = DSQRT ( DU1**2 + DU2**2 )
```

```
C
            -   AM(I2,J1) = -(DU2/DU3)
                AM(I2,J2) = -(DU1/DU3)
C
            END IF
C
 2000    CONTINUE
C
 3000 CONTINUE
C
      RETURN
C
C
      END
C
C
C
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @@                                    @@
C     @@  SUBROUTINE TO CALCULATE GRID      @@
C     @@  AZIMUTH AND DISTANCE              @@
C     @@                                    @@
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
      SUBROUTINE GRIDAZ (X1,Y1,X2,Y2,ANGLR,DIST12)
C
C
      REAL*8 X1,X2,Y1,Y2,ANGLR,ANG0,ANG90,ANG180,ANG270
      REAL*8 ANG90R,AN180R,AN270R,DIFX,DIFY,DIST12
C
      DATA ANG0/0.0D0/,ANG90/90.0D0/,ANG180/180.0D0/
      DATA ANG270/270.0D0/
C
      CALL DMSR ( ANG90,ANG0,ANG0,ANG90R )
      CALL DMSR ( ANG180,ANG0,ANG0,AN180R )
      CALL DMSR ( ANG270,ANG0,ANG0,AN270R )
C
      DIFX = X2-X1
      DIFY = Y2-Y1
C
      IF((DIFX.EQ.ANG0).AND.(DIFY.EQ.ANG0)) THEN
         ANGLR = ANG0
      ELSE IF(DIFX.EQ.ANG0) THEN
            IF(DIFY.GT.ANG0) THEN
              ANGLR = ANG0
            END IF
              ANGLR = AN180R
      ELSE
            IF(DIFX.GT.ANG0) THEN
              ANGLR = ANG90R-DATAN(DIFY/DIFX)
            ELSE
              ANGLR = AN270R-DATAN(DIFY/DIFX)
            END IF
      END IF
C
C
      DIST12 = DSQRT(DIFX**2+DIFY**2)
C
      RETURN
      END
C
C
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @@                                    @@
C     @@   SUBROUTINE FOR CHANGING THE      @@
C     @@   ANGLE FROM DEGREE, MINUTE, AND   @@
C     @@   SECONDS TO RADIAN                @@
```

```
C     @@                                     @@
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
      SUBROUTINE  DMSR ( D1,M1,S1,R1 )
C
      REAL*8  D1,M1,S1,R1,PI1,N60,N3600,N180
C
      DATA N60/60.0D0/,N3600/3600.0D0/,N180/180.0D0/
C
      PI1 = 4.0D0*DATAN(1.0D0)
      R1  = ((D1+(M1/N60)+(S1/N3600))*PI1)/N180
C
      RETURN
      END
C
C
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @@                                     @@
C     @@   SUBROUTINE FOR COMPUTING GRID     @@
C     @@   COORDINATES FROM KNOWN AZIMUTH    @@
C     @@   DISTANCE AND STARTING COORDI-     @@
C     @@   NATES                             @@
C     @@                                     @@
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C
      SUBROUTINE  UTM (X1,Y1,DIS12,ANG12R,X2,Y2 )
C
      REAL*8  X1,Y1,X2,Y2,DIS12,ANG12R,DIFX,DIFY
C
      DIFX = DIS12*(DSIN(ANG12R))
      DIFY = DIS12*(DCOS(ANG12R))
      X2   = X1+DIFX
      Y2   = Y1+DIFY
C
      RETURN
      END
C
C
C
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     @@                                     @@
C     @@   SUBROUTINE FOR CHANGING THE       @@
C     @@   ANGLE FROM RADIANS TO DEGREE,     @@
C     @@   MINUTES, AND SECONDS              @@
C     @@                                     @@
C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
      SUBROUTINE  RDMS ( RR1,D2,M2,S2 )
C
      REAL*8  RR1,D2,M2,S2,PI1,N60,N180,TDEG,DIF,TMIN
C
      DATA  N60/60.0D0/,N180/180.0D0/
C
      PI1 = 4.0D0*DATAN(1.0D0)
      TDEG= (N180*RR1)/PI1
      D2  = DFLOAT(IDINT(TDEG))
      DIF = TDEG-D2
      TMIN= DIF*N60
      M2  = DFLOAT(IDINT(TMIN))
      DIF = TMIN-M2
      S2  = DIF*N60
C
      RETURN
      END
```

LIST OF REFERENCES

Allan, Arthur L., J.R. Hollwey, and J.H.B. Maynes, *Practical Field Surveying and Computations*, Fifth Edition, Heinemann Publishing Co., 1968.

Bomford, G., *Geodesy*, Fourth Edition, Clarendon Press, Oxford, 1980.

Brinker, Russell C., and Paul R. Wolf, *Elementary Surveying*, Sixth Edition, Harper & Row, Publishers, 1977.

Clark, Late David, and J.E. Jackson, *Plane and Geodetic Surveying, Volume II Higher Surveying*, Sixth Edition, Constable & Company Ltd., 1976.

Davis, Raymond E., Francis S. Foote, James M. Anderson, and Edward M. Mikhail, *Surveying Theory and Practice*, Sixth Edition, McGraw-Hill Company, 1981.

Department of the Army Technical Manual, *TM 5-241-8, Universal Transverse Mercator Grid*, Headquarters, Department of the Army, 1958.

Evett, Jack B., *Surveying*, First Edition, John Wiley & Son, 1979.

Federal Geodetic Control Committee, *Standards and Specifications for Geodetic Control Networks*, Rockville, Maryland, 1984.

Mikhail, Edward M., and Gordon Gracie, *Analysis and Adjustment of Survey Measurements*, First Edition, Van Norstrand Reinhold Co., 1981.

Mikhail, Edward M., and F. Ackermann, *Observations and Least squares*, First Edition, Dun-Donnelley Publisher, 1976.

Schmidt, Milton O. and William H. Rayner, *Fundamentals of Surveying*, Second Edition, D. Van Nostrand Company, 1978.

Torge, Wolfgang, *Geodesy*, First Edition, Walter de Gruyter Berlin, New York, 1980.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Chairman, Department of Oceanography Code 68 Naval Postgraduate School Monterey, California 93943-5000	1
4.	NOAA Liaison Officer Post Office Box 8688 Monterey, California 93943-0688	1
5.	Dr. Muneendra Kumar Defense Mapping Agency Hydrographic/Topographic Center Geodesy and Surveys Department 6500 Brookes Lane Washinton, D.C. 20315	1
6.	Office of the Director Naval Oceanography Division (OP-952) Department of the Navy Washington, D.C. 20350	1
7.	Commander Naval Oceanography Command NSTL, Mississippi 39529	1
8.	Commanding Officer Naval Oceanographic Office Bay St. Louis NSTL, Mississippi 39522	1
9.	Commanding Officer Naval Ocean Research and Development Activity Bay St. Louis NSTL, Mississippi 39522	1
10.	Chief of Naval Research 800 N. Quincy Street Arlington, Virginia 22217	1
11.	Chairman, Oceanography Department U.S. Naval Academy Annapolis, Maryland 21402	1
12.	Topographic Sciences Department Code TSD-MC Defense Mapping School Ft. Belvoir, Virginia 22060-5828	1

13. CAPT Glen R. Schaefer 1
Code 68 Sc
Naval Postgraduate School
Monterey, California 93943-5000

14. Supote Klangvichit 4
Hydrographic Department
Royal Thai Navy
Bangkokyai, Bangkok
Thailand

15. Hydrographic Department Library 1
Hydrographic Department
Royal Thai Navy
Bangkokyai, Bangkok
Thailand

16. Yu, Ta-Te 1
SMC 1343, NPS
Monterey, California 93943

17. Wang, Chih-Ping 1
SMC 1342, NPS
Monterey, California 93943

220297
Thesis
K57282 Klangvichit
c.1 Traverse adjustment.