To the Graduate Council:

I am submitting herewith a thesis written by Joseph J. Shaw, Jr., entitled "The Transformation of Geographic Coordinates from Universal Transverse Mercator Coordinates." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

F. W. Donaldson

F. W. Donaldson, Major Professor

We have read this thesis
and recommend its acceptance:

W. C. Armstrong

C. J. N. Paludan

Accepted for the Council:

L. Evans Roth

Vice Chancellor
Graduate Studies and Research

A

THE TRANSFORMATION OF GEOGRAPHIC COORDINATES FROM UNIVERSAL TRANSVERSE MERCATOR COORDINATES

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Joseph J. Shaw, Jr.

December 1981

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to his Major Professor, Dr. F. W. Donaldson, Professor of Computer Science at The University of Tennessee Space Institute, Dr. C. C. Limbaugh, Mr. W. C. Armstrong, and Dr. T. Paludan for their time and effort and many helpful comments offered during the preparation of this thesis.

The author is also indebeted to Messrs. J. Kamus, M. Weldon, and R. Lampley of the Computer Science Corporation, Huntsville, Alabama, for their technical advice. Appreciation is also extended to the Computer Science Corporation for allowing the use of their project theory, data, software, and logic; to the Army Engineer Topographic Laboratories (AETL) at Fort Belvoir, Virginia, for permission to use Project H194 for this study; and to the National Aeronautics and Space Administration (NASA) at Redstone Arsenal, Huntsville, Alabama, for the use of their computer facilities. The work was done under General Services Administration Contract Number GS-045-22715. Special thanks go to Mrs. Jeannie Harper for typing the final manuscript.

Sincere appreciation is extended to my wife, Vickie, for her continuing support and encouragement throughout the preparation of the thesis.

ABSTRACT

The Army Engineer Topographic Laboratories (AETL) at Fort Belvoir, Virginia, implemented special purpose software to handle their map data. One of the functions of the software was to convert Universal Transverse Mercator (UTM) map coordinates to geographic map coordinates. The conversion was time-consuming and expensive due to a lengthy iterative process; therefore, a different method was sought on the computer.

One method for reducing the computer time is based on a curve-fitting procedure, suggested by an AETL engineer. This procedure is used to provide data to geographic coordinates (X_G, Y_G) from three UTM coordinates (X_T, Y_T). Each of the X_G's is then represented by a quadratic equation dependent on the X_T's; the three X_T points are used to determine the unknown quadratic coefficients; and the resultant equation is then a faster conversion process. The Y_T transformation is done similarly.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The Army Engineer Topographic Laboratories (AETL) at Fort Belvoir, Virginia, does the automatic compilation of cartographic data for the Army. (Cartography is the study of maps and map-making.) The AETL map data are obtained from existing maps by manually tracing contour lines on a machine which digitizes them into vector form. The digitizing machine has a plane rectangular grid on which the elevation readings are recorded. Because the machine coordinate system and the projection coordinate system lie in the same plane, the coordinate conversions are simple [1].[1] The elevation readings become matrix values in the computer. This grid of elevation estimates is called a digital terrain model. The elevations are stored on tape and their sequence position on the tape determines the (X,Y) coordinates to which the values refer. The map identification, projection, and scale information are stored in the header data.

After the elevation matrix is on tape, the digital terrain model is divided into square subsets [2]. These square-shaped subsets are approximated by polynomials using the least squares technique. By simple statistical measures, extreme errors and the closeness of the fit are examined. The polynomial has the following form:

---

[1]Numbers in brackets refer to similarly numbered references in the List of References.

$$Z = C_0 + C_1 X + C_2 Y + C_3 XY . \qquad (1.1)$$

The polynomials are made by use of the weighted integer technique.

To handle the map data, the AETL implemented special purpose hardware with a unified system of software. The software is called the Digital Terrain Elevation Database Software (DTEDS). One of the functions of the software is the transformation of coordinates from two projections. The transformation software was expensive and time-consuming on the computer. For the present investigation, the transformation of the Universal Transverse Mercator (UTM) coordinates to geographic coordinates is examined.

One method of reducing the computer time is based on a curve-fitting procedure, suggested by an AETL engineer [3]. In this curve-fitting technique, the subroutine, UTMFLH, is used to provide data of geographic coordinates (X_G, Y_G) from three UTM (X_T, Y_T) coordinates. Each X_G is then represented by a quadratic equation dependent on the X_T. The three X_T points are used to determine the unknown quadratic coefficients. The resultant equation provides a faster conversion from UTM to geographic. The Y_G transformation is done similarly. (UTMFLH is the subroutine used by AETL.)

The UTMFLH was slow because of a lengthy iterative process in its calculations. For several points the computer time would be expensive. For example, it took AETL 15 minutes on a Univac 1108 computer to convert $5x10^6$ points using UTMFLH.

The research for this study was done on a PDP 11/70 computer at Redstone Arsenal, Huntsville, Alabama. Fortran IV was used in the testing.

The main purpose of the present study was to see the advantage of the curve-fitting procedure over the slow subroutine, UTMFLH. The accuracy of the approximations was to be evaluated. If the curve-fitting procedure proved satisfactory, it would replace the method of direct computation.

The remainder of the study is composed of five chapters. Chapter II describes in detail the theoretical concepts of the two maps and their mapping function. Chapter III discusses the direct calculation of geographic coordinates from UTM coordinates. Chapter IV describes in detail the approximating method for determination of geographic coordinates. Chapter V shows the input for, and the results of, running the test cases. Chapter VI presents conclusions and recommendations on the approximating system.

CHAPTER II

THEORETICAL CONCEPTS

In this chapter, attention is focused on the definitions of the geographic map, the Universtal Transverse Mercator (UTM) map, and their mapping functions. In cartography, a map is a systematic projection of the Earth's surface upon a plane; however, in calculations, the irregular Earth's surface is difficult to handle. The irregular shape of the Earth approximates a spheroid, an oblate ellipsoid of revolution [4]. A spheroid is a mathematical approximation of the Earth's surface, used by cartographers [5]. The axes of the spheroid are seen in Figure A.1.[1] The major axis and the minor axis are the equator radii and the polar radii, respectively. There are about nine different types of spheroids used in cartography [5]. The spheroid in use depends on the part of the world in which one is located. For this study, the Clarke 1866 spheroid was used. The Clarke spheroid has a major axis of 6,378,206.4 meters and a flattening factor of 1/294.978698. The flattening factor indicates how well the spheroid approaches a true sphere. The minor axis is calculated by

$$b = -a\left(\frac{1}{f} - 1\right), \tag{2.1}$$

[1]Figures for Chapters II through V are located in Appendix A.

where b, a, and f are the minor axis, the major axis, and the flattening factor, respectively.

The spheroid allowed easier calculations on map projections. The spheroid played a major role in the transformations of the UTM coordinates to geographic coordinates. In the following discussion, the two maps and their mapping function will be analyzed and discussed.

A. The Geographic Map

The geographic graticule coordinates are latitude and longitude. (A graticule is a system of meridians and parallels intended to aid the measurement of position on a map; a meridian is any circle on the Earth that goes through both poles; a parallel is any circle on the Earth that runs parallel with the equator.) In this study, latitude and longitude are represented by the Greek letters, phi and lamda, respectively. Figure A.2 shows a point on the Earth's surface and the lines of longitude and latitude. The latitude of a point is the angle between the plane normal to the Earth's surface and the equatorial plane [6]. Latitudes vary from 0^{0} to 90^{0}, with the equator being 0^{0}. Points in the northern hemisphere have north latitude values and points in the southern hemisphere have south latitude values. The point's longitude is the angle between the Greenwich meridian and the point meridian plane. Longitude values vary from 0^{0} to 180^{0} in both eastward and westward directions. The Greenwich meridian and all points that lie on it have 0^{0} longitude. Figure A.3 shows how a point is located by latitude and longitude.

The geographic map used in the present investigation is in the form of the elevation matrix. Figure A.4 shows the geographic map model. The southwest corner (labeled "A") of the map is the origin of the map graticule. Longitude and latitude of the geographic map origin are given in whole degrees. Increments across the axes are in terms of tenths of a second. In terms of a matrix, the longitude lines and the latitude lines are columns and rows, respectively; so each line of longitude will have a number of latitude points. These longitude lines are called profile lines by the Army Engineer Topographic Laboratories (AETL).

B. The Universal Transverse Mercator Map

The UTM grid and projection was developed by the United States and its allies after World War II. The system suited the U.S. Army criterion, which would be a continuous system encompassing the globe and have as few zone boundaries as possible. The UTM map system divides the Earth into 60 zones. Each zone is 6^{o} wide in longitude, 3^{o} on either side of the central meridian of the zone. The UTM zone numbers begin with 1 on the zone from 180^{o} west longitude to 174^{o} west longitude and increase eastward. The last zone is from 174^{o} east longitude to 180^{o} east longitude with a number of 60. The zone numbering scheme originated from the specifications of the International Map of the World on the Millionth Scale, established by international agreements. The zones are bounded by latitudes 84^{o} north and 80^{o} south [7]. Figure A.5 shows a UTM zone on the Earth's surface. The origin of the zone is the junction of the equator with the zone's central meridian. The UTM map scale varies from 0.99960 at the central meridian to 1.00158 at the zone borders. At times the zone does not follow the contour of the land with the above scales. The scale of the map is adjusted to fit the land elevations. This technique is called "scale factor mapping" [7].

A rectangular metric grid can be placed over each zone and any point on the zone grid has coordinates of easting and northing. The coordinates are subtracted from false origins to insure that all values are positive. The false easting value, the false northing for the Northern Hemisphere, and the false northing for the Southern Hemisphere are 500,000 meters, 0 meter, and 10,000,000 meters, respectively. Figure A.6 shows a UTM grid. (A grid is a system of coordinates on the Earth's surface expressed in linear units.)

The UTM map is made from a variation of the transverse mercator (TM) projection. (Normally, the globe is placed inside a cylinder in which the cylinder is tangent to a great meridian circle. The meridional great circle is known as a standard line for the projection. A standard line is where map deformation is the least [8].) For the UTM projection, the cylinder is secant to the Earth, creating two standard lines. The standard lines are two small circles parallel to the central meridian and are 180 kilometers east and west of it. Figure A.7 shows the difference in the two projections.

In Figure A.8, the UTM grid system is in the form of a diagram. In the diagram the equator, OM, and the central meridian, OG, are shown as straight lines. The intersection of the two lines defines a rectangular grid. The grid is parallel to the two lines and the grid origin is Point O. The point, P(E,N), represents a point in the northern hemisphere, east of the central meridian. The point can be on either side of the central meridian by inverting the diagram. The diagram cannot picture the Southern Hemisphere because of the Southern Hemisphere's false northing of 10,000,000 meters at the equator. Line FP is the point easting coordinate, and the line, OF, is the point northing coordinate. Point P(E,N)'s meridian and latitude are arcs--MG and LP, respectively. The point, F, is called the foot of the perpendicular from the point, P(E,N), to the central meridian. The line, OL, is the meridional arc, while line LF is the ordinate of curvature. The distance on the meridional arc, OL, is denoted S_ϕ.

In Figure A.8 the meridional arc is described as the distance from the equator to the point's parallel. For any point on the meridional ellipse the distance is

$$S_\phi = \int_0^P P d\phi \ , \tag{2.2}$$

where P is the radius of curvature of the meridional ellipse, and ϕ is the angle of longitude. The equation for P is

$$P = (1 + (dz/dr)^2)^{3/2} / (d^2z/dr^2) \ , \tag{2.3}$$

where r is the radius of the parallel at the point. The variable, r, is defined as

$$r = V \cos\phi \ , \tag{2.4}$$

where V denotes the plane normal to the meridional ellipse. The expression for z is

$$z = \frac{a(1-e^2)\ \sin\phi}{\sqrt{1-e^2\ \sin^2\phi}} \quad .$$

Finally, V, P, and S_ϕ are put in terms of latitude, longitude, and spheroidal constants. The meridional arc function is

$$\begin{aligned} S_\phi = a \int_0^\phi [& (1 - \frac{1}{4} e^2 - \frac{3}{64} e^4 - \frac{5}{256} e^6 - \ldots) \phi \\ & - (\frac{3}{4} e^2 + \frac{3}{16} e^4 + \frac{45}{512} e^6 + \ldots) \cos 2\phi \\ & + (\frac{15}{64} e^4 + \frac{45}{256} e^6 + \ldots) \cos 4\phi - (\frac{35}{512} e^6 + \ldots) \\ & \cos 6\phi + \ldots]\, d\phi \ , \end{aligned} \qquad (2.5)$$

where e is the eccentricity of the spheroid. The formula of the spheroid's eccentricity is

$$e = \left(\frac{a^2 - b^2}{a^2} \right)^{1/2} \quad . \qquad (2.6)$$

The meridional arc function is part of the transformation of geographic coordinates to be discussed in Chapter III.

C. The Mapping Function

This section is intended to explain the relationships between the two map systems. The relationship of the UTM map to the geographic map can be expressed as:

$$P(E,\ N,\ Z) = f\ (S,\ \phi,\ N/S,\ \lambda,\ E/W)\ , \tag{2.7}$$

known as the mapping function. The variables of the UTM map are the easting, E; the northing, N; and the UTM zone number, Z. From the geographic map the variables are the spheroid, S; the hemisphere indicators; and latitude and longitude. The map variables are modified according to the map from which they come. For example, the easting and the northing are adjusted by their false origins. (This will be shown in Chapter III.) The UTM zone is determined by the point longitude and the E/W hemisphere indicator. If the zone lies in the west, the UTM zone number is determined by

$$Z = \text{largest integer} \leq [\frac{(360-\lambda)}{6}] - 29\ , \tag{2.8}$$

and the central meridian of the zone is

$$\lambda_{CM} = 183 - 6z\ . \tag{2.9}$$

If the point lies in the east, the UTM zone number is

$$Z = \text{largest integer} \leq \frac{360+\lambda}{6} - 29\ , \tag{2.10}$$

and the central meridian zone is

$$\lambda_{CM} = 6z - 183\ . \tag{2.11}$$

In the geographic map, the variables are modified also. The spheroid indicator simply tells the values of the spheroid axes, A and B. The hemisphere indicators, N/S and E/W, tell the hemisphere in which the geographic map lies. The longitude is modified to reflect its relationship to the central meridian by

$$\Delta\lambda = |\lambda - \lambda_{CM}| . \tag{2.12}$$

When all the modifications are done, the mapping function is condensed to a function of the UTM coordinates, easting and northing, to form an expression from the geographic coordinates, longitude and latitude, as follows:

$$P(E', N') = f(\phi, \Delta\lambda) , \tag{2.13}$$

where the primes indicate modified easting and northing. The mapping equation is the complex function:

$$E' + iN' = f(\Delta\lambda + i\tau) = F(\Delta\lambda + i \int_0^\phi V \cos\phi \, d\tau) , \tag{2.14}$$

where

$$\tau = \int_0^\phi \frac{P d\phi}{V \cos} ,$$

and where i is defined as $i = \sqrt{-1}$. The proof of Eq. (2.14) can be found in Reference [9]. The formulas for changing between UTM coordinates and geographic coordinates are derived from Eq. (2.14).

CHAPTER III

THE DIRECT COMPUTATION OF GEOGRAPHIC COORDINATES

In this chapter, attention is focused on the direct transformation of geographic coordinates from Universal Transverse Mercator (UTM) coordinates. The subroutine, UTMFLH, is analyzed and discussed. The direct calculation method of UTMFLH can be divided into six steps: (1) unit check; (2) central meridian is determined; (3) easting and northing are corrected; (4) use of the meridian arc function; (5) the latitude and the longitude are calculated; and (6) the longitude is adjusted.

First, UTMFLH checks to see if the easting and the northing are in units of feet or meters. If the units are feet, the units are changed to meters.

Next, the central meridian of the point zone is determined. If the zone number is greater than 30, then the central meridian is

$$\lambda_{CM} = 6z - 183 \ , \tag{3.1}$$

where z is the UTM zone number. If the zone is less than or equal to 30, the zone's central meridian is

$$\lambda_{CM} = 183 - 6z \ . \tag{3.2}$$

Next, the central meridian is changed to units of seconds of a degree by multiplying it by 3,600 seconds.

Next, the easting and the northing of the UTM point are corrected by their false origins. If

$$E \geq 5 \times 10^5 , \tag{3.3}$$

the corrected easting is

$$E' = \frac{E-500,000}{k} , \tag{3.4}$$

where E' and k are the corrected easting and the central scale factor, respectively. (The central scale factor is defined as the relationship between a scale distance on the map and the equivalent distance on the ground. For the present study, k = 0.9996) If

$$E < 5 \times 10^5 , \tag{3.5}$$

the corrected easting is

$$E' = \frac{500,000-E}{k} . \tag{3.6}$$

The point northing correction depends on the hemisphere. If the point is in the Northern Hemisphere, the corrected northing is

$$N' = \frac{N}{k} , \tag{3.7}$$

where N' is the corrected northing. If the point is in the Southern Hemisphere, the corrected northing is

$$N' = \frac{10,000,000 - N}{k} . \tag{3.8}$$

Next, the eccentricity of the spheroid and the spheroidal constants are calculated. The spheroidal constants are used in the meridional arc function. In the meridional arc function the parameters,

$$\phi \text{ and } S_\phi ,$$

are replaced by

$$\phi_F \text{ and } S_F .$$

The variable, ϕ_F, is the latitude of the foot of the perpendicular line from the point to the central meridian. The starting value for ϕ_F is

$$\phi_F = \frac{N'}{a\,(1 - \frac{1}{4}e^2 - \frac{3}{34}e^4 - \frac{5}{256}e^6)} , \tag{3.9}$$

where a is the semi-major axis, and e is the eccentricity of the sphere. The value is derived from the first order of the meridional arc function. A correction factor,

$$\Delta\phi_F ,$$

is added to each iteration afterwards. The correction factor is

$$\Delta\phi_F = \frac{S_F - N'}{P_F} , \tag{3.10}$$

$$P_F = \frac{a\,(1 - e^2)}{(1 - e^2 \sin^2\phi)^{3/2}} . \tag{3.11}$$

Equation (3.11) is derived from Eq. (2.3).

As the correction factor approaches zero, N' and S_F become the same distance along the meridional ellipse from the foot to the equator. The distance on the meridional ellipse becomes nearly equal to the distance from the latitude of the point foot to the central meridian. (The above iterative process is the factor that makes UTMFLH slow.)

Next, the latitude and the longitude of the point are determined with precision. To do this, more constants are calculated. The formulas for obtaining longitude and latitude are

$$\Delta\lambda = \frac{1}{\cos\phi_{F_2}}\left[\frac{E'}{V_F} - (1 - 2t^2 + \eta^2)\frac{\left(\frac{E'}{V_F}\right)^3}{6} + 5 + 6\eta^2 + 28t^2 + 8t\eta^2 + 24t^2\frac{\left(\frac{E'}{V_F}\right)^5}{120}\right] \quad (3.12)$$

and

$$\phi = \phi_F + t(1 + \eta^2)\left[\frac{\left(\frac{E'}{V_F}\right)^2}{2} + (5 + \eta^2 + 3t^2 - 9t^2)\frac{\left(\frac{E'}{V_F}\right)^4}{24} + (61 + 90t^2 + 46\eta^2 + 45t^4 - 252t^2\eta^2 - 90t^4)\frac{\left(\frac{E'}{V_F}\right)^6}{720}\right], \quad (3.13)$$

respectively. The formulas are derived from the mapping function. The variable, η, is defined as

$$\eta = e \quad \cos\phi_F ,$$

where e is the spheroid eccentricity. The variable, V_F, is defined as

$$V_F = a/\sqrt{1 - e^2 \sin^2\phi_F} \ .$$

Finally, the longitude is added to, or subtracted from, the central meridian of the point. If

$$E < 500{,}000 \ ,$$

the longitude is

$$\lambda = \lambda_{CM} - \Delta\lambda \ . \tag{3.14}$$

If

$$E \geq 500{,}000 \ ,$$

the longitude is

$$\lambda = \lambda_{CM} + \Delta\lambda \ , \tag{3.15}$$

if the point is in the Western Hemisphere. For the Eastern Hemisphere, the sign of $\Delta\lambda$ reverses.

Next, the coordinates are converted to radians from degrees. (For a fuller explanation of the conversion process, see Reference [9].)

CHAPTER IV

THE APPROXIMATION OF GEOGRAPHIC COORDINATES

The alternative method of conversion is the curve-fitting method, suggested by Jancaitis [3]. The Universal Transverse Mercator (UTM) map used in the present investigation was limited to a 50 by 50 grid by the Army Engineer Topographic Laboratories (AETL). The UTM map follows the same pattern as the geographic map in being in the form of an elevation matrix. The easting axis has profile lines on it. Each profile line has 50 points of northing on it. The AETL named the UTM grid the "target system." The target system definition results from a query by the user, as in Figure A.9.

A. Initiation

The subroutine, UTMFLH, is used to gain data to help derive the approximating equations. To do this, three points are chosen from the first profile line on the UTM grid. The first, middle, and last points of the line are converted to geographic points by UTMFLH. If the line's three point coordinates are

$$\left(X_{t_1}, Y_{t_1}\right) ; \left(X_{t_2}, Y_{t_2}\right) ; \text{and} \left(X_{t_3}, Y_{t_3}\right) , \qquad (4.1)$$

then, by definition [10],

$$X_{t_1} = X_{t_2} = X_{t_3} .$$

B. The Determination of the Approximation Equations

After the points are converted by UTMFLH, there are three known points in each coordinate system. The transform locations of the three UTM points on the geographic map are:

$$\left(X_{i_1}, Y_{i_1}\right);\left(X_{i_2}, Y_{i_2}\right);\text{ and }\left(X_{i_3}, Y_{i_3}\right). \tag{4.3}$$

Figure A.10 shows these new relationships. The transformed coordinates are fitted deterministically with the quadratic polynomials,

$$X_{i_k} = C_0 + C_1 Y_{t_k} + C_2 Y_{t_k}^2 \tag{4.4}$$

and

$$Y_{i_k} = D_0 + D_1 Y_{t_k} + D_2 Y_{t_k}^2 , \tag{4.5}$$

where k = 1,2,3.

By substituting three known geographic points and the original UTM points, two sets of equations are formed. These are:

$$\begin{aligned} X_{i_1} &= C_0 + C_1 Y_{t_1} + C_2 Y_{t_1}^2 \\ X_{i_2} &= C_0 + C_1 Y_{t_2} + C_2 Y_{t_2}^2 \\ X_{i_3} &= C_0 + C_1 Y_{t_3} + C_2 Y_{t_3}^2 \end{aligned} \tag{4.6}$$

and

$$Y_{i_1} = D_0 + D_1 Y_{t_1} + D_2 Y_{t_1}^2$$

$$Y_{i_2} = D_0 + D_1 Y_{t_2} + D_2 Y_{t_2}^2 \quad . \tag{4.7}$$

$$Y_{i_3} = D_0 + D_1 Y_{t_3} + D_2 Y_{t_3}^2$$

Next, the equations are translated into matrix form:

$$\bar{A}\,\bar{X} = \bar{B} \quad . \tag{4.8}$$

The X-vector can readily be solved. Using the equation matrix form, the equations become

$$\begin{pmatrix} 1 & Y_{t_1} & Y_{t_1}^2 \\ 1 & Y_{t_2} & Y_{t_2}^2 \\ 1 & Y_{t_3} & Y_{t_3}^2 \end{pmatrix} \times \begin{pmatrix} C_0 \\ C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} X_{i_1} \\ X_{i_2} \\ X_{i_3} \end{pmatrix} \quad . \tag{4.9}$$

Solving for C_0, C_1, and C_2 yields

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 1 & Y_{t_1} & Y_{t_1}^2 \\ 1 & Y_{t_2} & Y_{t_2}^2 \\ 1 & Y_{t_3} & Y_{t_3}^2 \end{pmatrix}^{-1} \times \begin{pmatrix} X_{i_1} \\ X_{i_2} \\ X_{i_3} \end{pmatrix} \quad . \tag{4.10}$$

The C-coefficients are found by Gauss elimination. Similarly, the coefficients are determined for the northing of the points. The C-coefficients gave the equation for the easting of the UTM points.

Once the C- and D-coefficients are found, the two approximating equations are determined. The geographic points are found by substituting the original profile line points into the two equations. Similarly, the procedure is used on the 49 UTM profile lines.

The system flowcharts and code listings are presented in Appendix B.

CHAPTER V

TEST CASES

This chapter discusses the test cases of the curve-fitting method. The Universal Transverse Mercator (UTM) map target system data are:

$$N = 1437489.0$$

$$E = 391533.0$$

$$\Delta E = AA$$

$$\Delta N = BB$$

$$H = N \text{ (or S)}$$

$$S = \text{Clarke 1866}$$

$$Z = 39$$

The central meridian of the UTM zone is 51^{0} east longitude. Terms AA and BB are the increments of the easting and the northing, respectively. For Cases 1 through 6, the increments were 100, 1,000, 10,000, 37,000, 63,000, and 150,000 meters, respectively. The delta symbol, Δ, refers to the increment along the axes

The geographic map data were in the form of the elevation matrix. The southwest corner of the map is 13^{0} north latitude and 50^{0} east longitude. There are six profile lines with 79 points on each line in the geographic map. The area of two maps is pictured in Figure A.11.

The speed of calculation of the two maps is listed in Table 1. Each method did 2,500 point conversions (50 profile lines with 50 points on each line on the UTM map.) The Army Engineer Topographic Laboratories (AETL) subroutine, UTMFLH, of calculating the points is first. The total time of the approximating method appears next. The time of finding the approximation equations and the time to do the substitutions are added together for the total time of the approximation method. The ratio of the times was determined by the following equation:

$$\text{Ratio} = AT/TT \ ,$$

where TT and AT are the UTMFLH time and the approximation time, respectively. The ratio shows that the approximating method is at least 100 times faster than the UTMFLH method. When the increment was 1,000 meters in Case 2, the approximation time, the UTMFLH time, and their ratio were 37.82 seconds, 5584 seconds, and 1/148, respectively.

The accuracy of the approximations depends on the size of the increment. As the increment along the axis increases, the error in the approximations increases. The coordinates' approximation deviation from the true point value and the percent of deviation is shown in Table 2. The deviation for the X-coordinate is determined by

$$\text{Deviation} = TX - AX \ ,$$

where TX and AX are the direct calculated coordinate and the

Table 1. Approximation Times of the Points

Case	UTMFLH Total Time (sec)	Approximation Total Time (sec)	Ratio
1	2151	15.16	1/142
2	5584	37.82	1/148
3	9161	44.14	1/208
4	4110	32.49	1/126
5	4766	27.67	1/172
6	5682	35.59	1/160

Table 2. Deviation of the Approximation Point

Case	X-Deviation (meters)	Y-Deviation (meters)	X-Deviation (percent)	Y-Deviation (percent)
1	0.1906×10^{-5}	0.4724×10^{-6}	0.3812×10^{-5}	0.3634×10^{-5}
2	0.1905×10^{-5}	0.4636×10^{-6}	0.3795×10^{-5}	0.3548×10^{-5}
3	0.2141×10^{-4}	0.4037×10^{-5}	0.3923×10^{-4}	0.2921×10^{-4}
4	0.7873×10^{-2}	0.1609×10^{-3}	0.1160×10^{-1}	0.9871×10^{-3}
5	0.1116	0.6594×10^{-3}	0.1350	0.3653×10^{-2}
6	0.5455×10^{-2}	0.3074×10^{-2}	0.2707×10^{-2}	0.9644×10^{-2}

approximated coordinate, respectively. The percentage of the X-coordinate deviation is

$$\text{Percentage of deviation} = ((TX - AX)/TX) * 100 .$$

The Y-coordinate deviation and the percentage of deviation are done similarly.

The accuracy of approximations due to the increments is reflected in the deviations. At the lowest increment of 100 meters, the percentages of deviation for X and Y are 381×10^{-8} and 363×10^{-8}, respectively. The deviation percentages for X and Y, when the increment is 150,000 meters, are 27.07 and 0.00964, respectively. The deviation of the coordinates for Case 1 is well below the 1-percent margin.

The graphs of the 50 profile lines (Appendix C) show the deviation in accuracy of the two values. In the cases with low increments, the two points seem to coincide. In the cases with increments greater than 10,000 meters, the points seem to diverge from each other. A close look at the graphs will show that the first, middle, and the last points are always close. This is because these points were used to determine the approximating equations. For example, in Case 6, the fiftieth profile line has values for five points that show the contrast in accuracy of the points (Table 3). The reason that the error increases is that the points tend to fall outside the zone boundaries because of the large increment. All points other than the three chosen tend to have a high percentage of error in them.

Table 3. Approximation of the Points

Point	True X (degrees)	Approximate X (degrees)	True Y (degrees)	Approximate Y (degrees)
1	108.0041	108.0041	13.0040	13.0040
10	112.3953	95.0739	25.2050	25.2069
25	130.2713	130.2713	45.4929	45.4929
44	222.5529	276.6851	71.0920	71.0950
50	346.5658	346.5658	79.1581	79.1581

As the above error indicates, poor input gives poor data. If the zone does not match the exact location of the geographic map, the answers will have no meaning. For example, if the zone number given is 26, the zone boundaries are 30^{0} west longitude and 24^{0} west longitude. The geographic map area must fall some place in Zone 26; otherwise, the logic of the program fails. Also, a zone number greater than 60 or less than 0 will be in error. If the increment chosen is large, the profile lines may fall outside the zone borders. The UTM zone is a specific number of kilometers across, so there is a limit on the size of the increment.

The amount of core for the approximating method is more than for UTMFLH. On the computer, the UTMFLH core was 2,156 words. The approximating system subroutine, TRANSF, used 9,796 words of core, but only part of TRANSF did the approximating method.

By looking at the flow charts (Appendix B), one can see the flow of the logic. The subroutine, IINPUT, put the parameters of the geographic map into the system. The UTM map is defined by the user in subroutine UTMQRY. The main calculation of the approximations plus the determination of the equations is handled in subroutine TRAREG. In TRAREG, the use of the Gauss method of solving the system of equations is found in subroutine GAUSS. As mentioned previously, UTMFLH is used to derive needed data. Some of the software code listings have been omitted because they are not pertinent to the present study.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

The curve-fitting method has its advantages over the Army Engineer Topographic Laboratories (AETL) subroutine, UTMFLH. The approximating method uses 1/100 or less of the time used by UTMFLH. The approximating method will be less expensive in the future. The accuracy of the approximating method is enough to warrant its use instead of UTMFLH.

The approximating of geographic coordinates could be improved upon. For example, four or more points could be used in determining the approximating equations; thus, improving the accuracy of the approximations. The limits of the increments on the Universal Transverse Mercator (UTM) axes could be searched for; thus, decreasing the high percentage of error in the approximations. The idea of approximations could be extended to other map systems, such as local orthogonal to geographic. The use of approximating map coordinates deserves further research for development.

LIST OF REFERENCES

LIST OF REFERENCES

1. Codd, John F. "Transformations of Coordinates of Cartographic Digital Data," U.S. Army Engineer Topographic Laboratories ETL-TR-74-8, Fort Belvoir, Virginia, 1974.

2. Jancaitis, James R. "Element Data Compaction by Polynomial Modeling," U.S. Army Engineer Topographic Laboratories ETL-0140, Fort Belvoir, Virginia, April 1977.

3. Jancaitis, James R. "Statement of Work for Experimental Elevation Transformation and Regridding Software," U.S. Army Engineer Topographic Laboratories ETL-TD-MA, Fort Belvoir, Virginia, November 1978.

4. Burkard, Richard K. Geodesy for the Layman. St. Louis: Aeronautical Chart and Information Center, 1959.

5. Smirnoff, Michael V. Measurements for Engineering and Other Surveys. Englewood Cliffs, New Jersey: Prentice-Hall, 1961.

6. Universal Transverse Mercator Grid. Washington, D.C.: Headquarters, Department of the Army, 1973.

7. Colvocoresses, Alden P. "A Unified Plane Coordinate Reference System," World Cartography. Vol. IX. New York: United Nations Publications, 1969.

8. Robinson, Arthur H., and Randall D. Sale. Elements of Cartography. Third edition. New York: John Wiley and Sons, Inc., 1969.

9. Weldon, Mark R. "A General Automated Terrain Simulation," U.S. Army Missile Command TR-WS-74-1, Redstone Arsenal, Alabama, March 1974.

10. System Documentation: Transformation and Regridding Software for Polynomial Terrain Models H194. Prepared for the U. S. Army Engineer Topographic Laboratory under General Services Administration Contract No. GS-045-22715. Huntsville, Alabama: Computer Sciences Corporation, 1980.

APPENDICES

APPENDIX A

FIGURES

This appendix contains Figures A.1 through A.11, described in Chapters II through V.

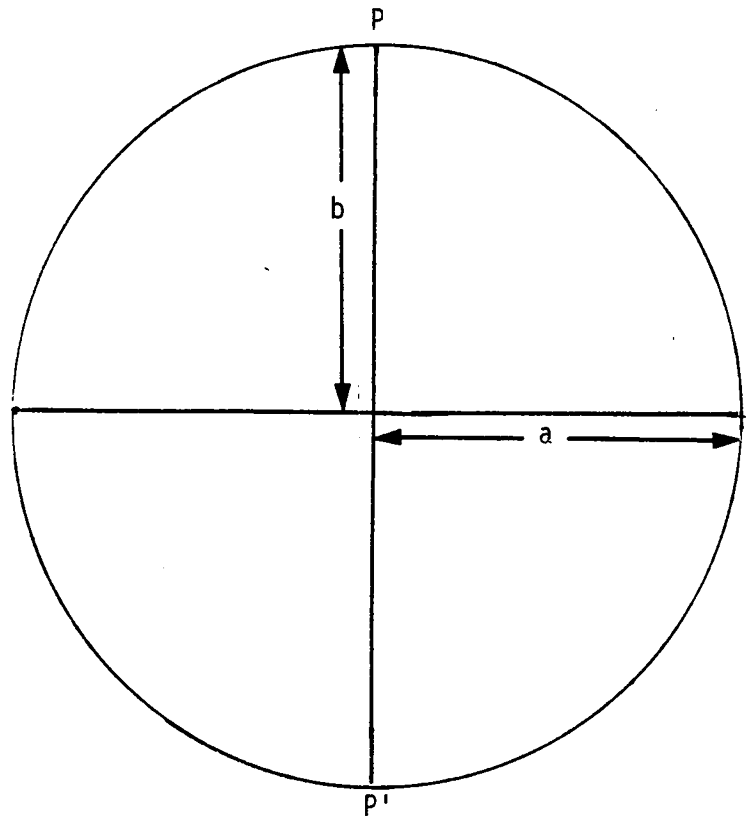


Figure A.1. The axes of the spheroid.

Note: The line, PP', indicates the axis of revolution about the Earth ellipsoid. The symbols, a and b, are equal to one-half of the major-axis and one-half of the minor-axis, respectively.

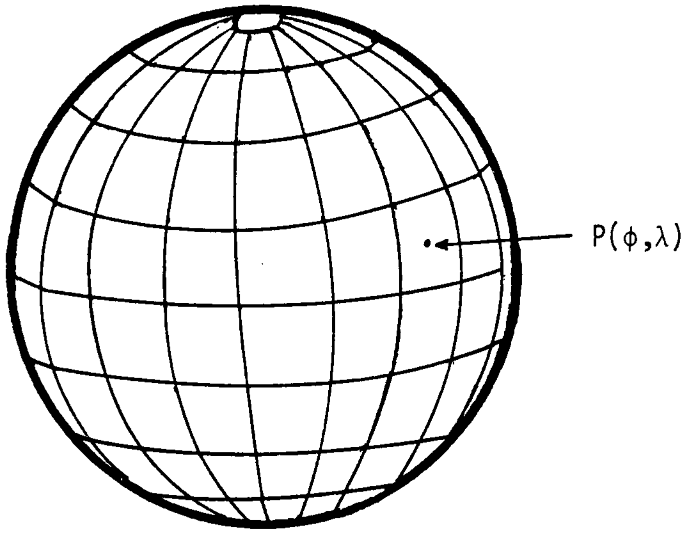


Figure A.2. Location of a point on the Earth.

Note: Lines of latitude run east to west, while longitude lines run north to south; P(ϕ,λ) is a point on the Earth having ϕ^{o} longitude and λ^{o} latitude.

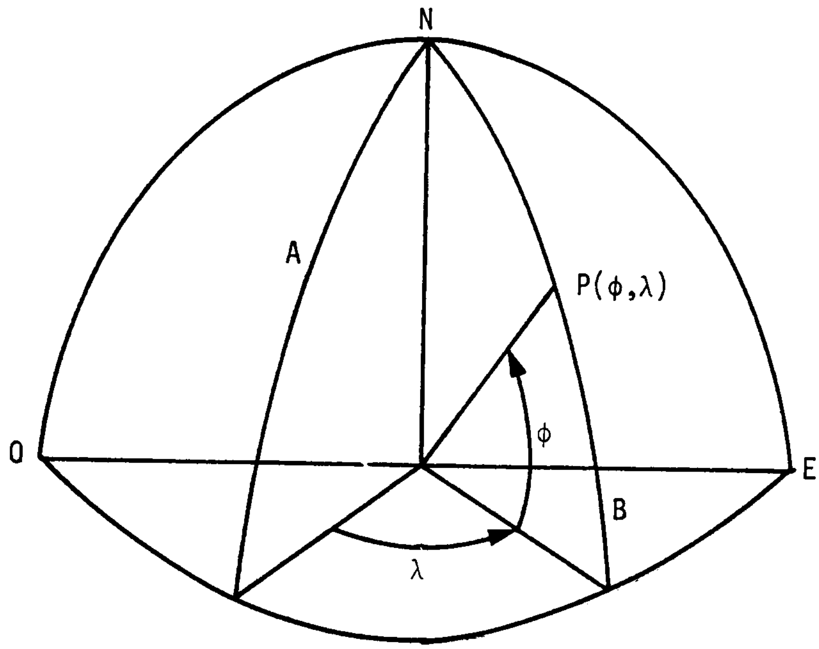


Figure A.3. Measurement of a point's latitude and longitude.

Note: Arc A is the Greenwich meridian; Arc B is the point's meridian; λ is the angle of longitude; ϕ is the angle of latitude; Arc OE is the equator; Point N represents the North Pole; Line OE is the Earth's diameter.

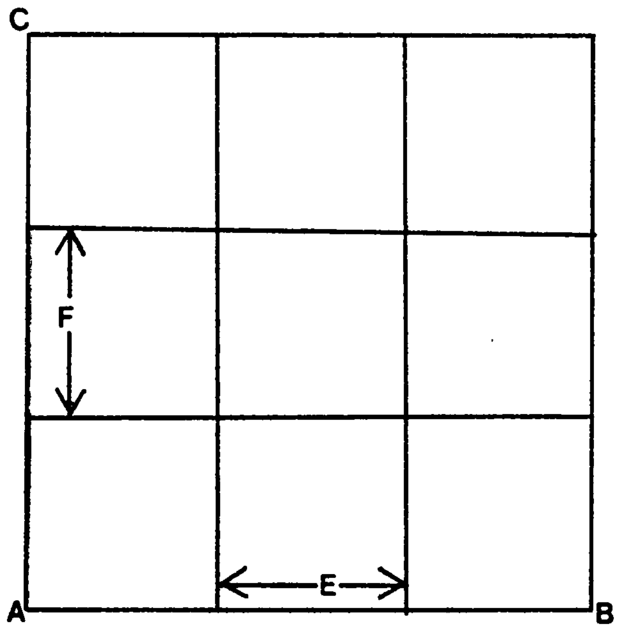


Figure A.4. Geographic map model.

Note: Point A is the southwest corner (or the origin of the map) measured in whole degrees; Point C is the maximum latitude; Segment F is the increment on the Y-axis; Point B marks the longitude end point; and Segment E is the increment on the X-axis. The increments are measured in tenths of seconds.

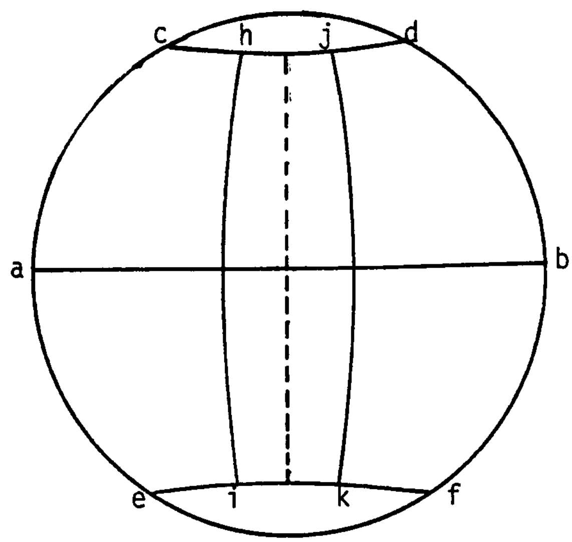


Figure A.5. UTM zone on the Earth.

Note: The zone is bounded by Arcs cd and ef (84^{0} north latitude, 80^{0} south latitude); the dashed line is the central meridian of the zone; Line ab is the equator; Lines hi and jk run the length of the globe to form the sides of the zone; the zone is 6^{0} wide.

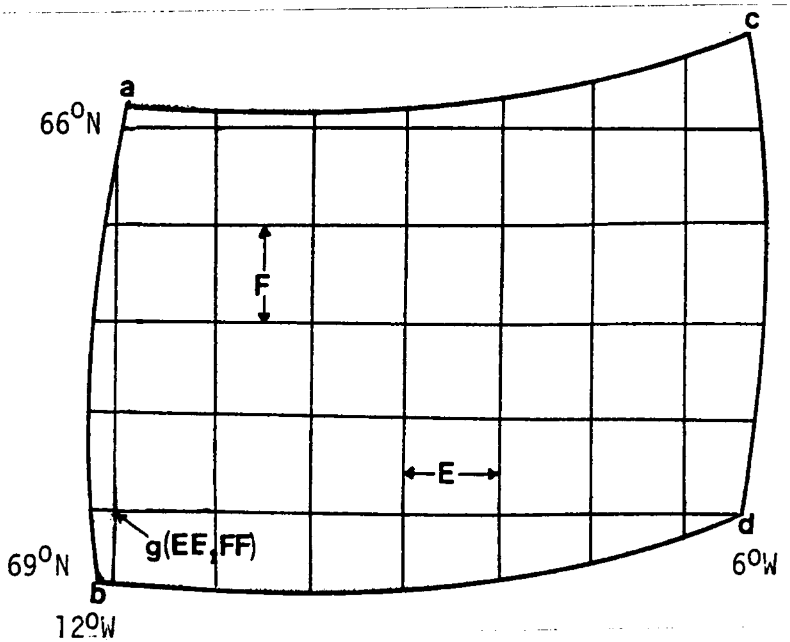
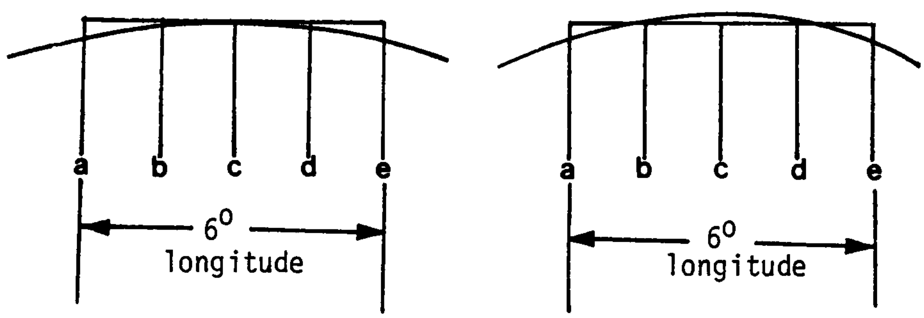


Figure A.6. The UTM map.

Note: The UTM map is rectangular in the grid system shown. The UTM zone number is 29. The curved lines are lines of longitude and latitude, respectively. Lines ab and cd are, for example, 6° west longitude and 12° west longitude. Lines bd and ac are 64° north latitude and 66° north latitude. Point g is the origin of the UTM grid, having an easting of EE meters and a northing of FF meters. The increments of the easting and the northing are E and F, respectively.



a. TM projection b. UTM projection

Figure A.7. The TM projection and the UTM projection [8].

Note: The TM projection has one standard line touching the Earth sphere. (A standard line is where the map distortion is the least.) The map scale at Point c, the standard point, is 1.0000, while at Points a and e the scale is 1.00198. The UTM projection has two standard lines because the line is secant to the Earth, instead of being tangent to the Earth. At Points b and d, the two standard line points, the map scale is 1.0000. At Points a, c, and e the scales are 1.00158, 0.99960, and 1.00158, respectively.

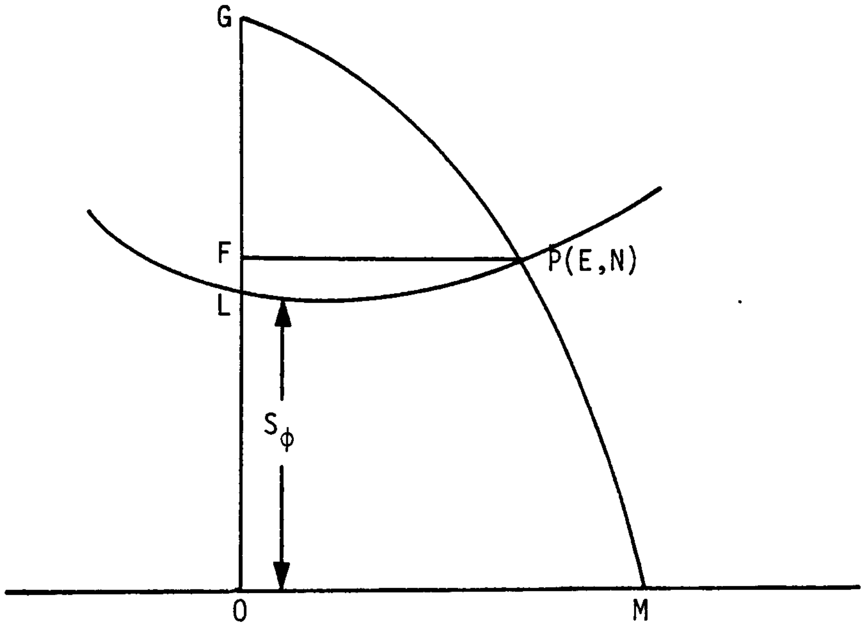
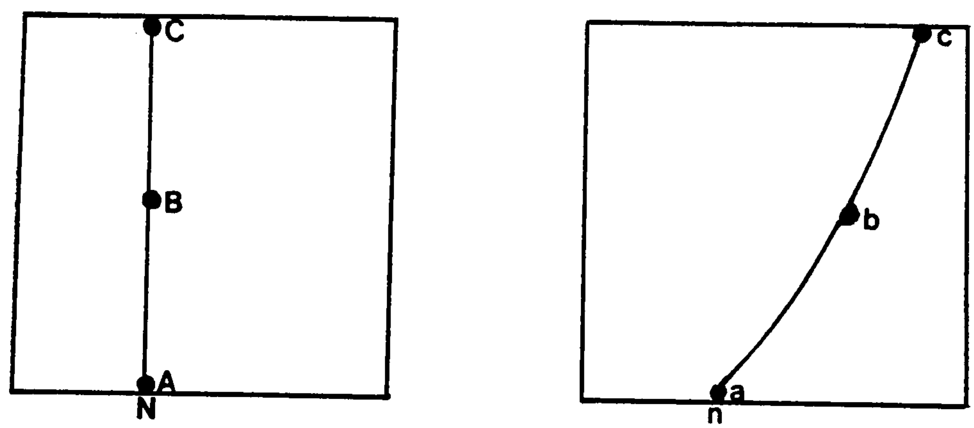


Figure A.8. The UTM grid system.

```
FUNCTIONAL TRANSFORMATION AND REGRIDDING SOFTWARE
(GSA PROJECT H-194
                    01-FEB-80      13:58:56

INPUT SPECIFICATION
     1 = DMAAC DATA TYPE
     2 = EDAS POLYNOMIAL TYPE
ENTER 1 OR 2 > 2

ENTER EDAS DISK FILE NAME AS AAAAAAA.III  >  DP:FILE1.005

******************************
TARGET SYSTEM DEFINITION
******************************
     1 = UTM
     2 = LOCAL ORTHOGONAL
ENTER 1 OR 2 >  1

*********************************
UTM DESCRIPTION OF TARGET SYSTEM
*********************************

ENTER NORTHING  ( METERS ) > 1437489.

ENTER EASTING ( METERS ) > 391553.

ENTER INCREMENT OF NORTHING  (METERS) > 500.

ENTER INCREMENT OF EASTING ( METERS ) >  500.

ENTER HEMISPHERE 0 FOR NORTH AND 1 FOR SOUTHERN > 0

ENTER UTM ZONE (1 THRU 60)  >  39.

SPECIFY SPHERICAL TYPE:
1 =FOR INTERNATIONAL
2 =FOR CLARK(1866)
3 =FOR CLARK (1880)
4 =FOR  EVEREST
5 =FOR BESSEL
6 =FOR MODEIFIED EVEREST
7 =FOR AUSTRALIAN NATIONAL
8 =FOR AIRY
9 =FOR  MODIFIED AIRY
ENTER ONE OF THE ABOVE  > 2
```

Figure A.9. Query for the target system.



a. UTM map b. Geographic map

Figure A.10. The relationship between the two maps.

Note: For the UTM map, Points A, B, and C are the three chosen points to determine the approximation equations for profile line, N. The geographic map shows transformed Points a, b, and c. The old profile line, N, off the UTM map was a straight line, but the new line made up of the 50 transformed points on the Geographic map may not be a straight line. The new line, n, could be a curve or a slanted line.

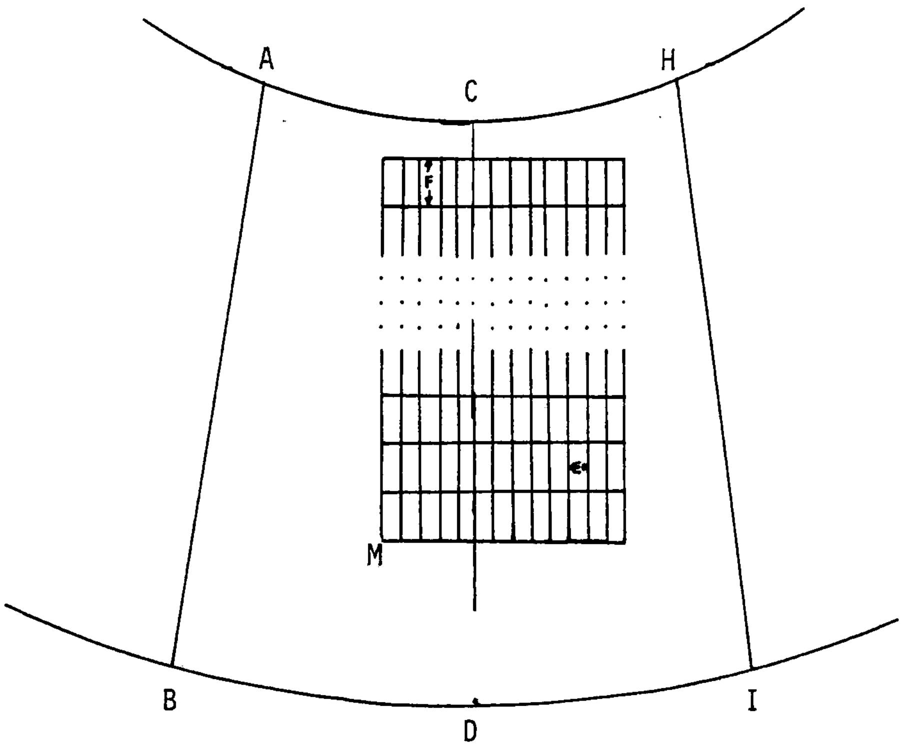


Figure A.11. Land area use in the test cases.

Note: The northern part of UTM Zone 39 is pictured. Lines AB and HI are the zone borders, being 48° east longitude and 54° east longitude. The zone central meridian, CD, is 51° east longitude. The geographic map has its origin at Point M, whose coordinates are 13° north latitude and 50° east longitude. Arc BDI lies on the equator and Arc ACH lies on the 84° north latitude line.

APPENDIX B

FLOWCHARTS AND SOFTWARE CODE LISTINGS

This appendix contains the system flowcharts (Figures B.1 through B.12) and software code listings.

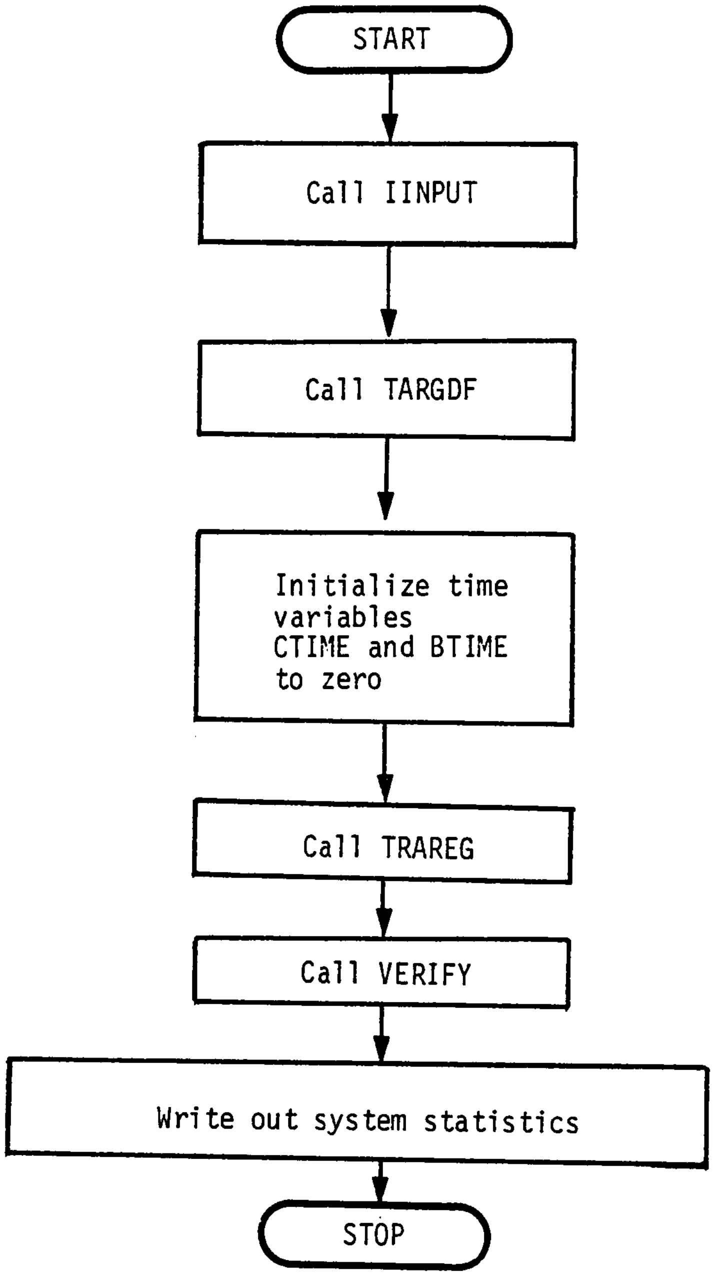


Figure B.1. Main program flow chart.

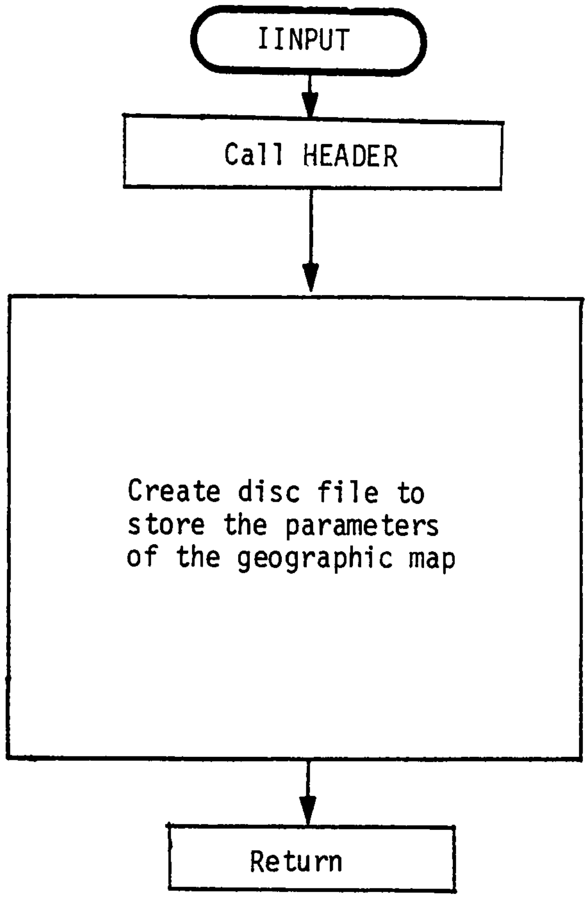


Figure B.2. Subroutine IINPUT flow chart.

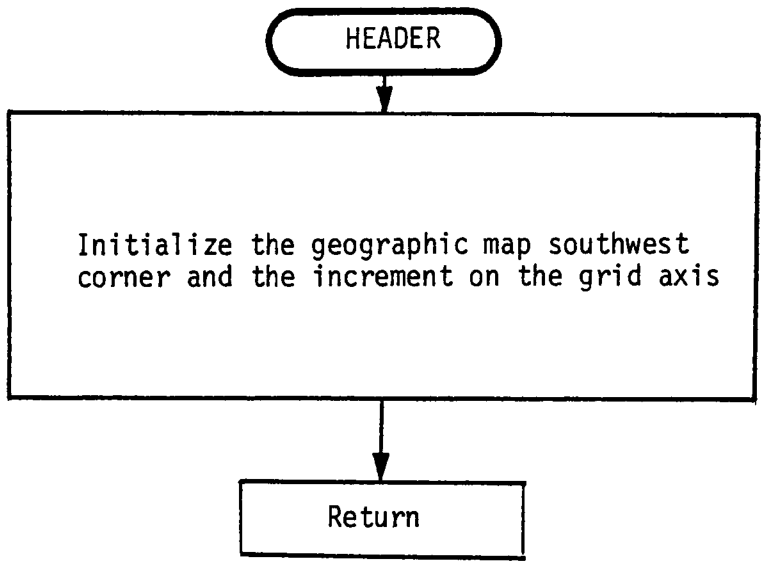


Figure B.3. Subroutine HEADER flow chart.

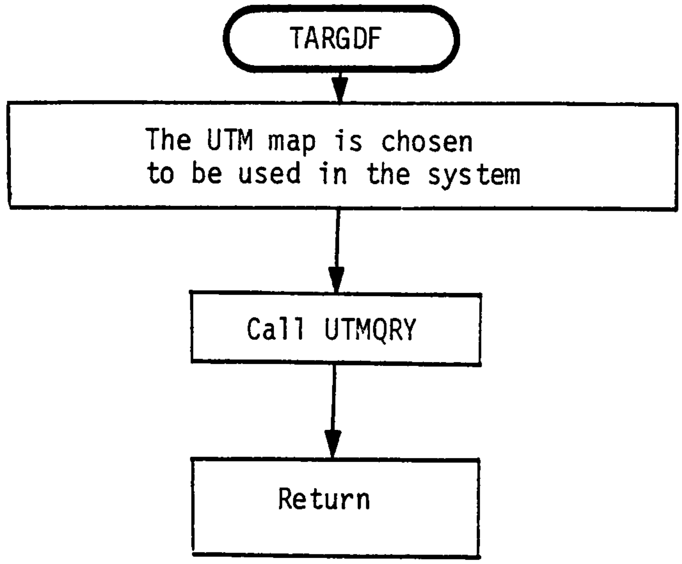


Figure B.4. Subroutine TARGDF flow chart.

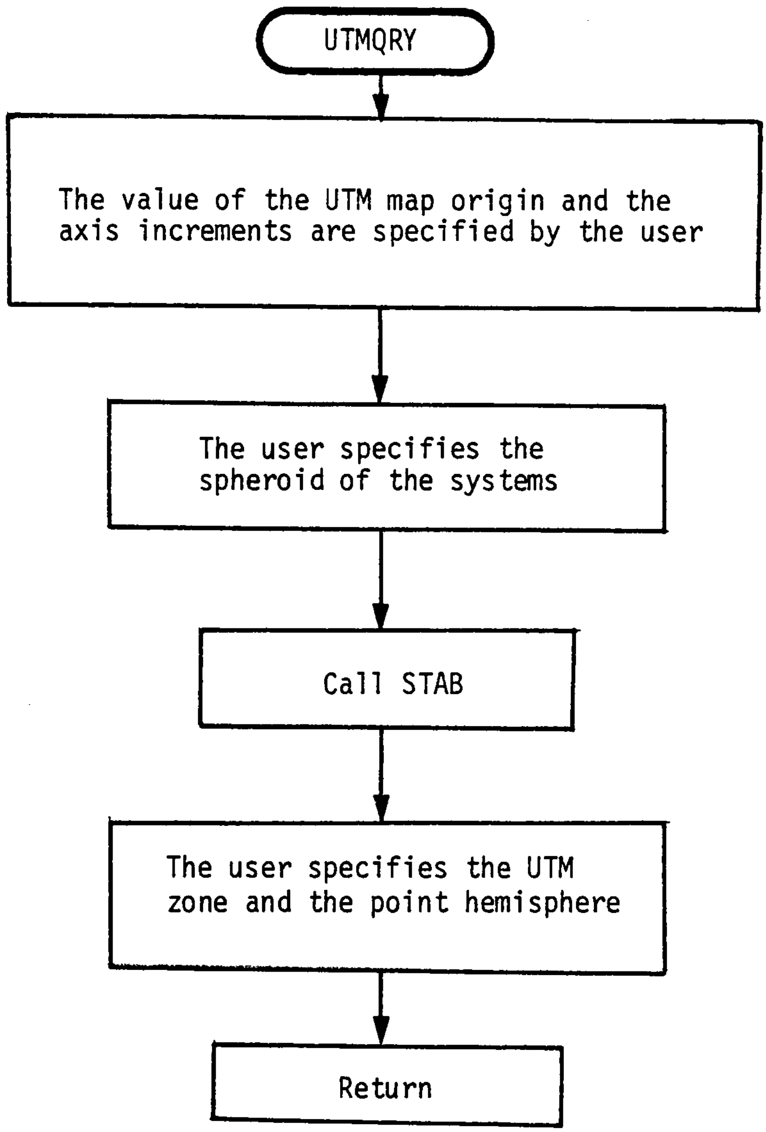


Figure B.5. Subroutine UTMQRY flow chart.

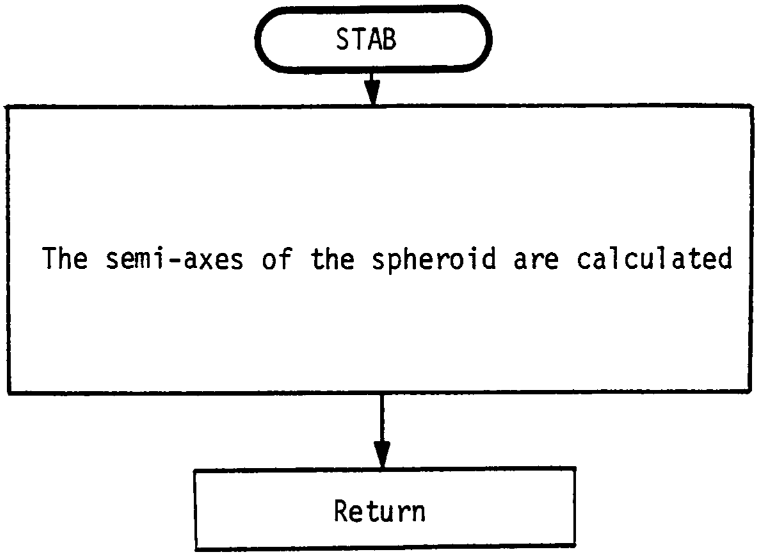


Figure B.6. Subroutine STAB flow chart.

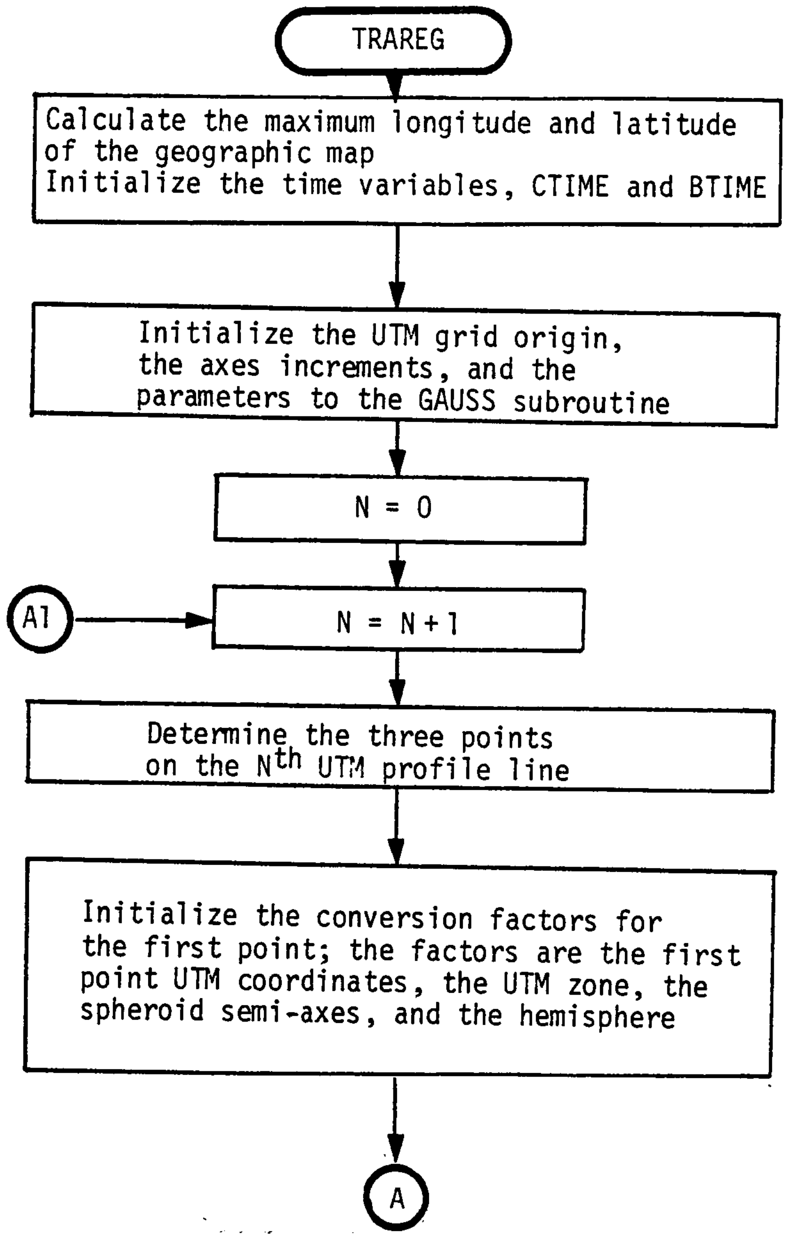


Figure B.7. Subroutine TRAREG flow chart.

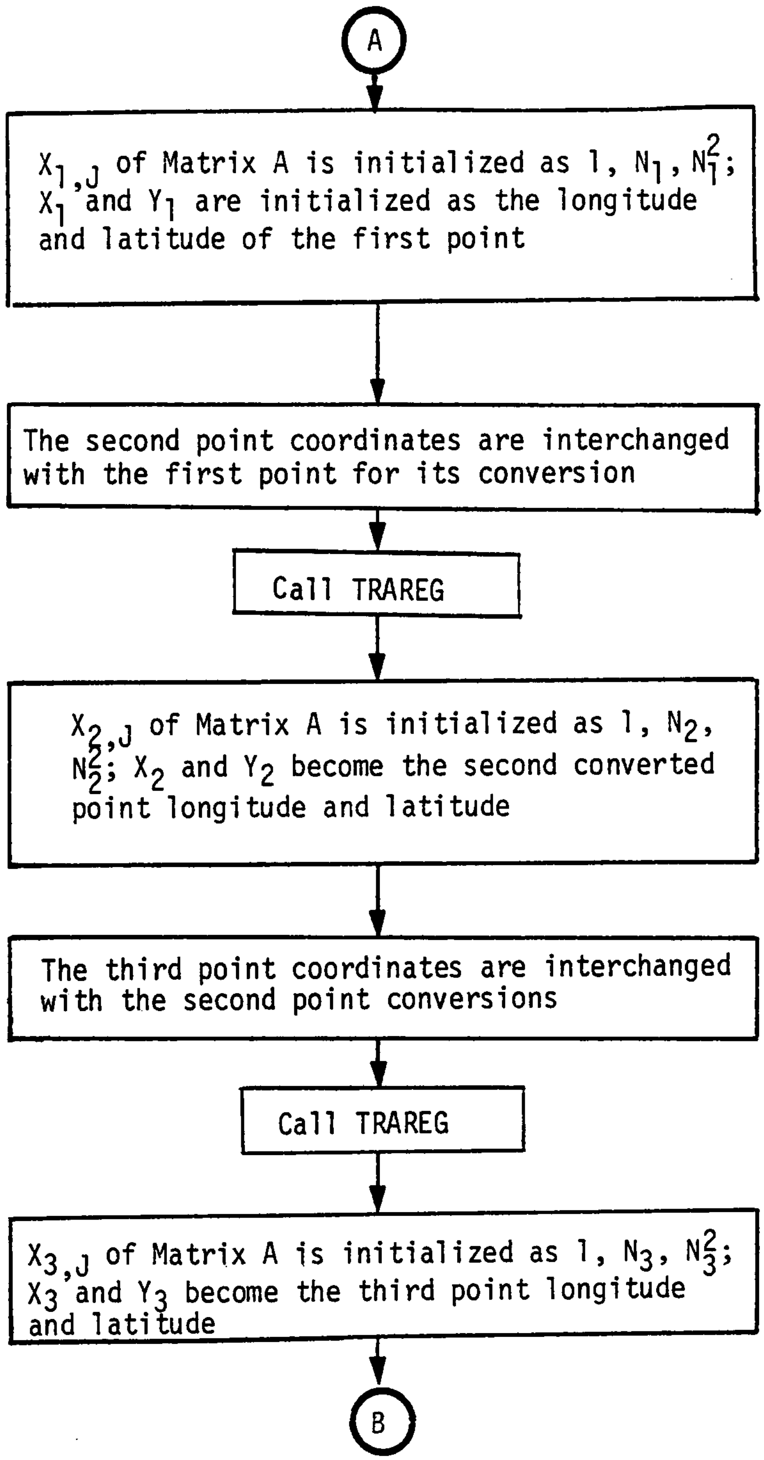


Figure B.7. (continued)

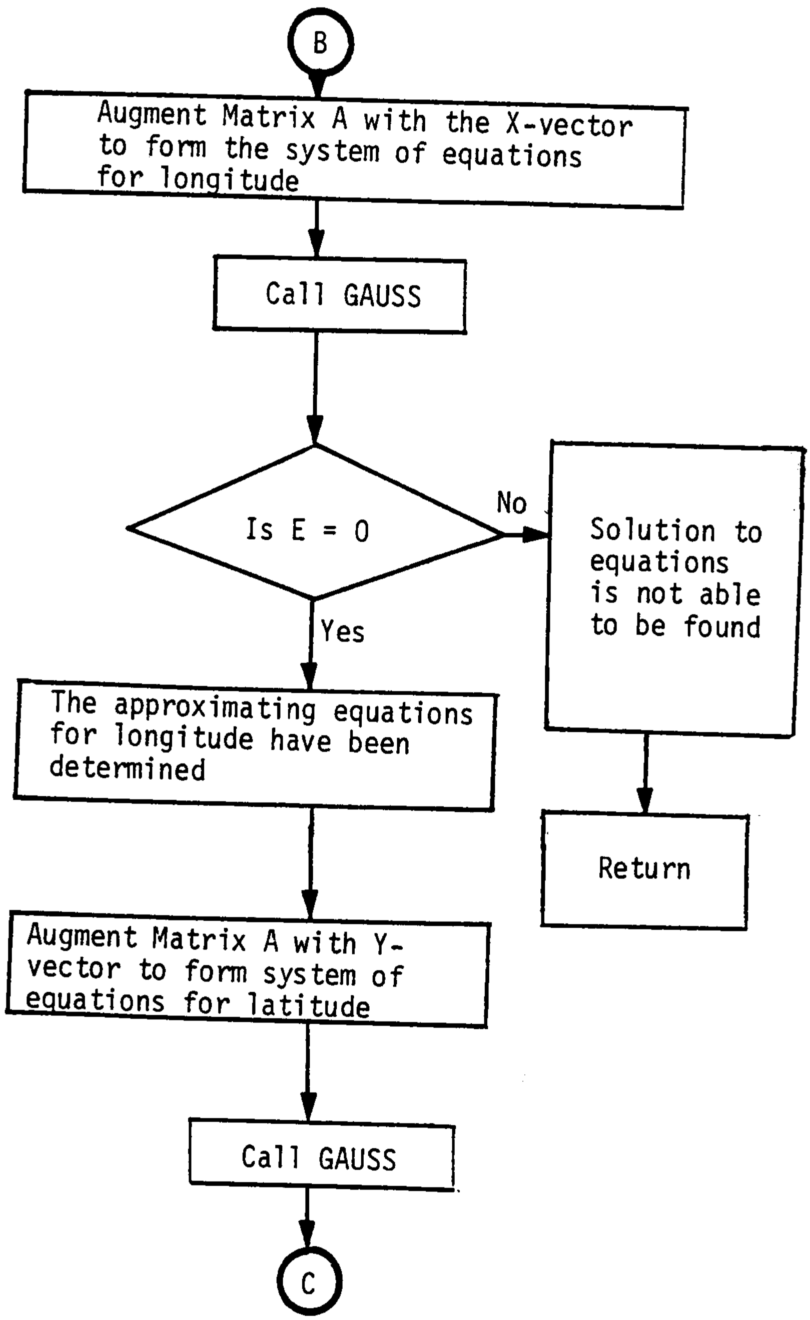


Figure B.7. (continued)

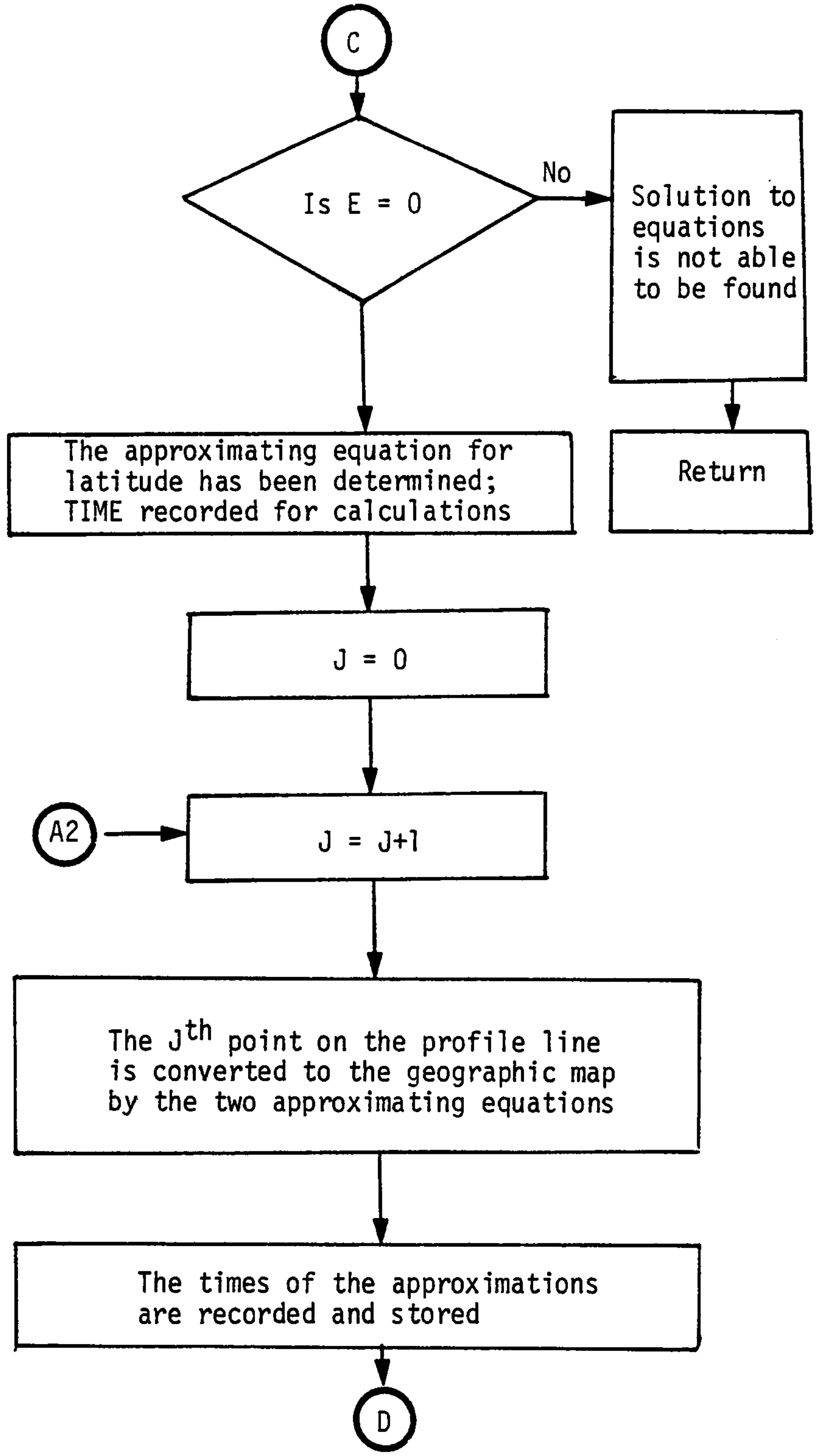


Figure B.7. (continued)

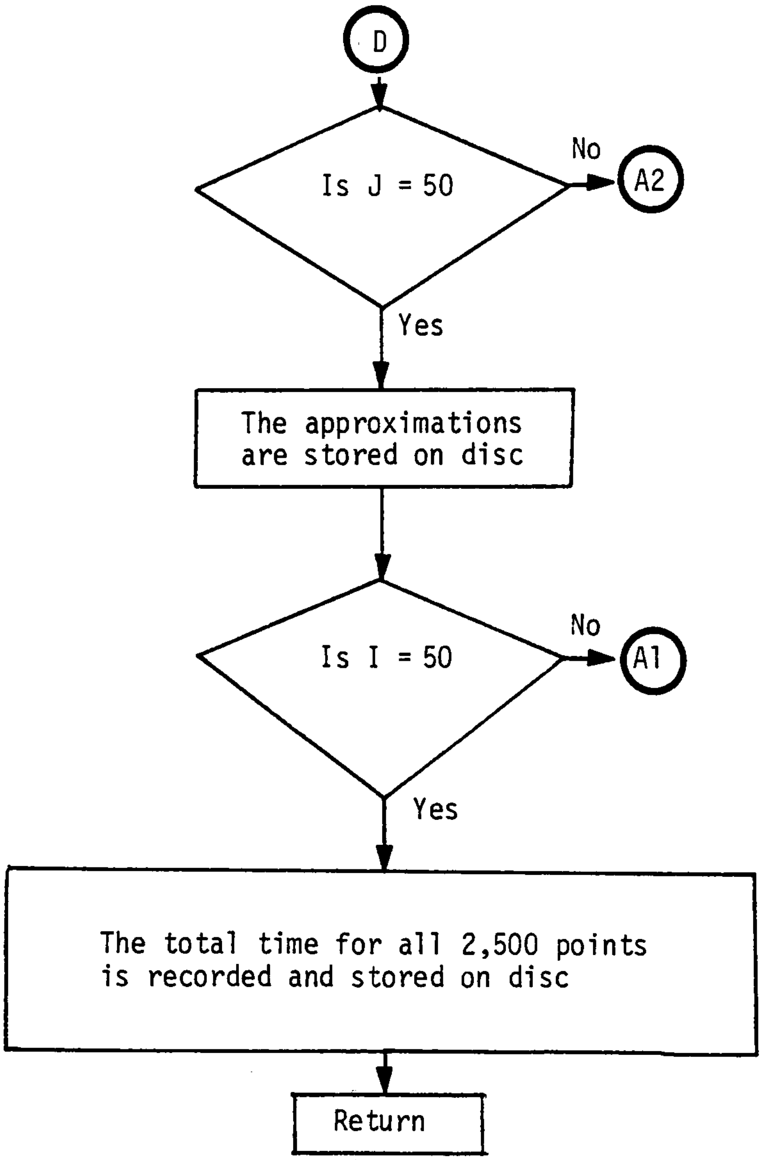


Figure B.7. (continued)

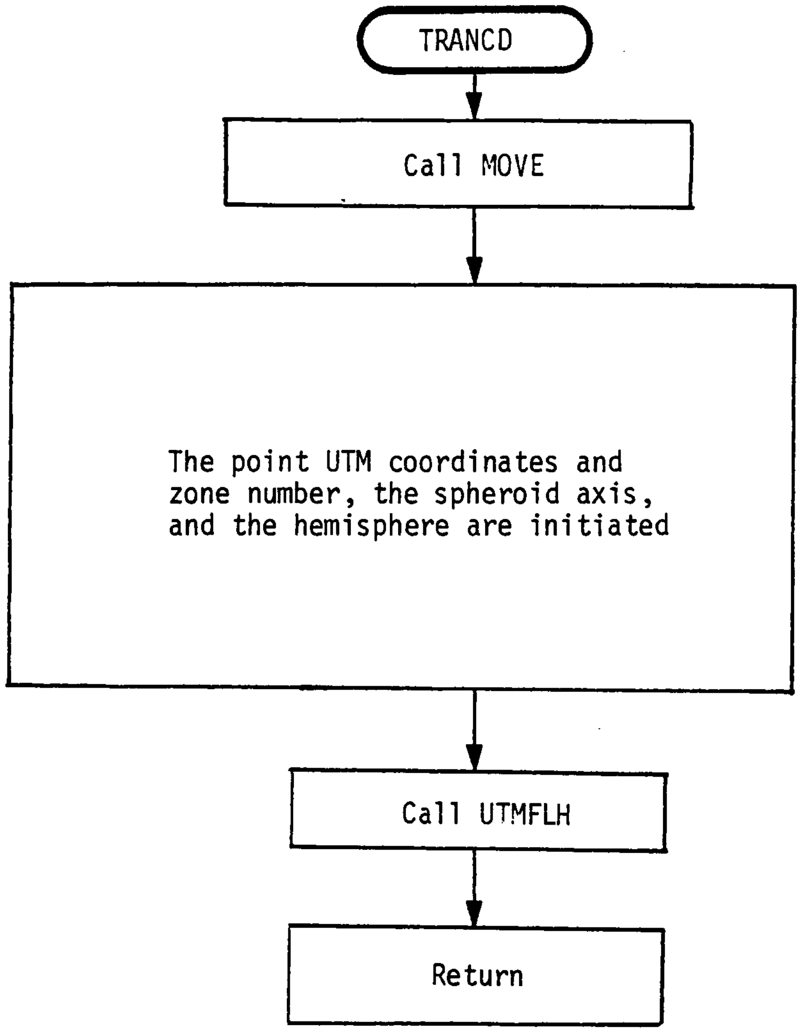


Figure B.8. Subroutine TRANCD flow chart.

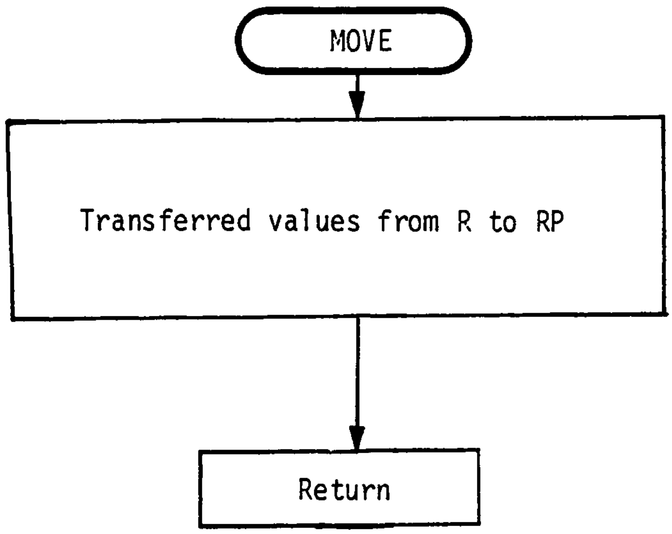


Figure B.9. Subroutine MOVE flow chart.

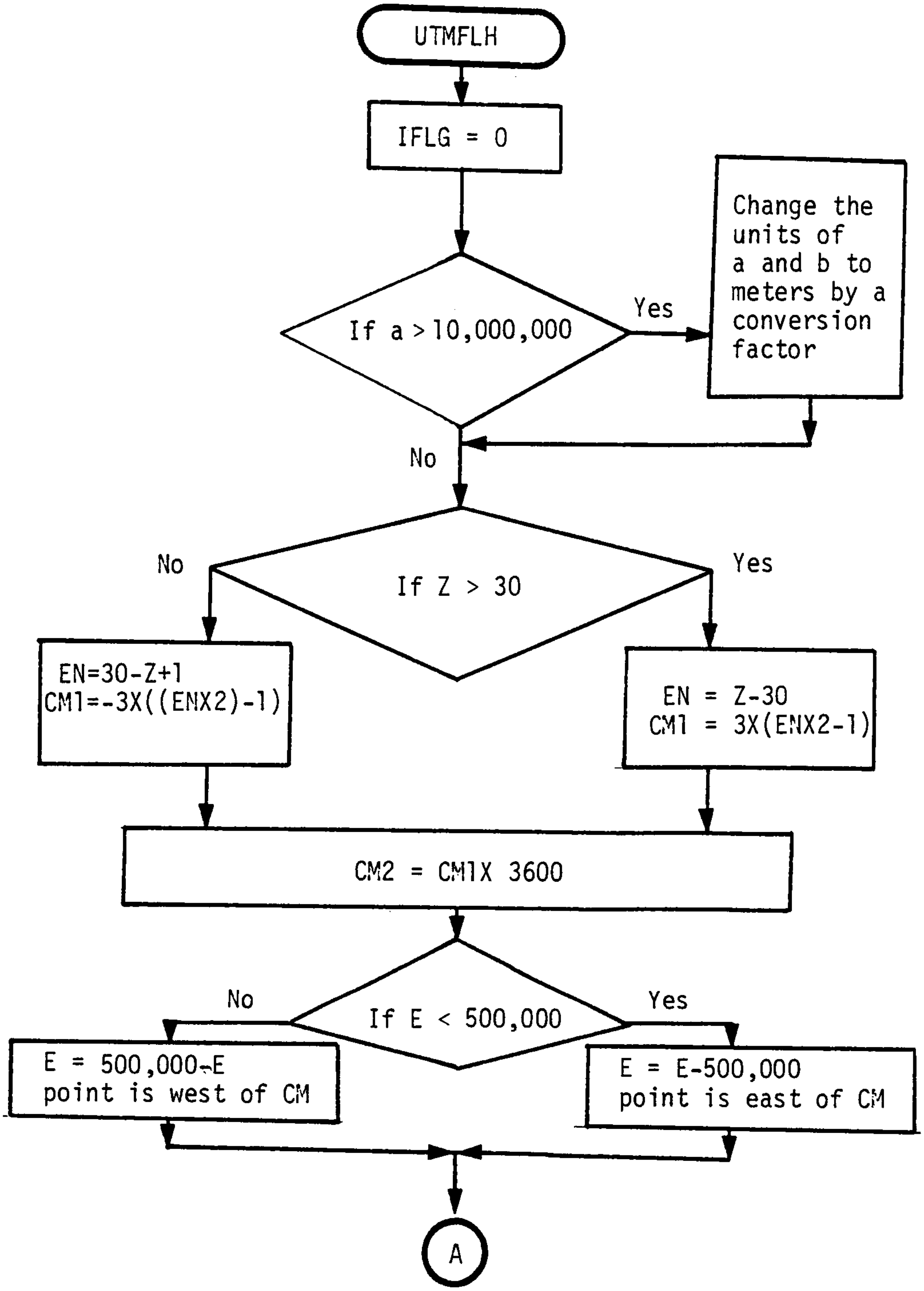


Figure B.10. Subroutine UTMFLH flow chart.

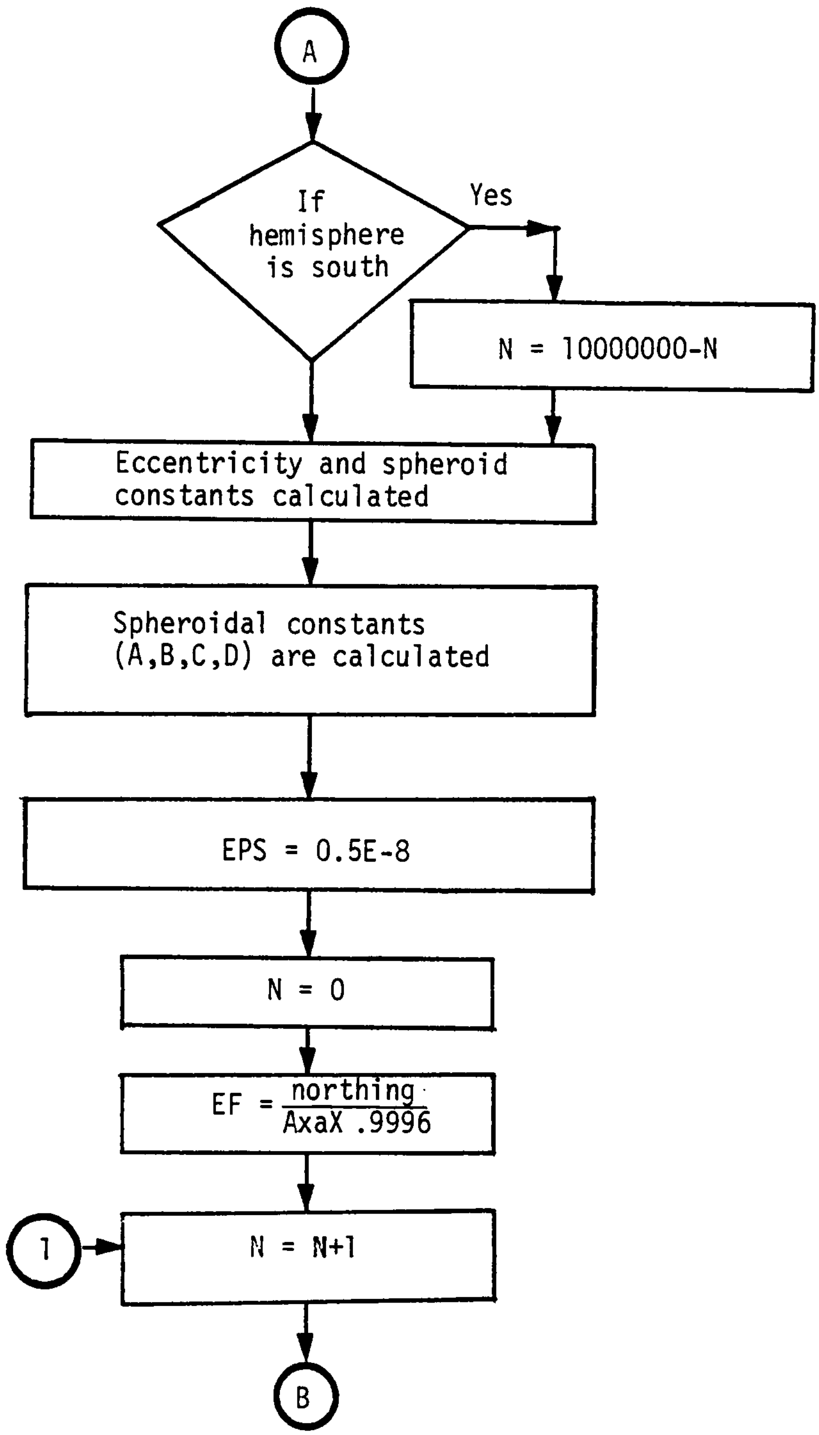


Figure B.10. (continued)

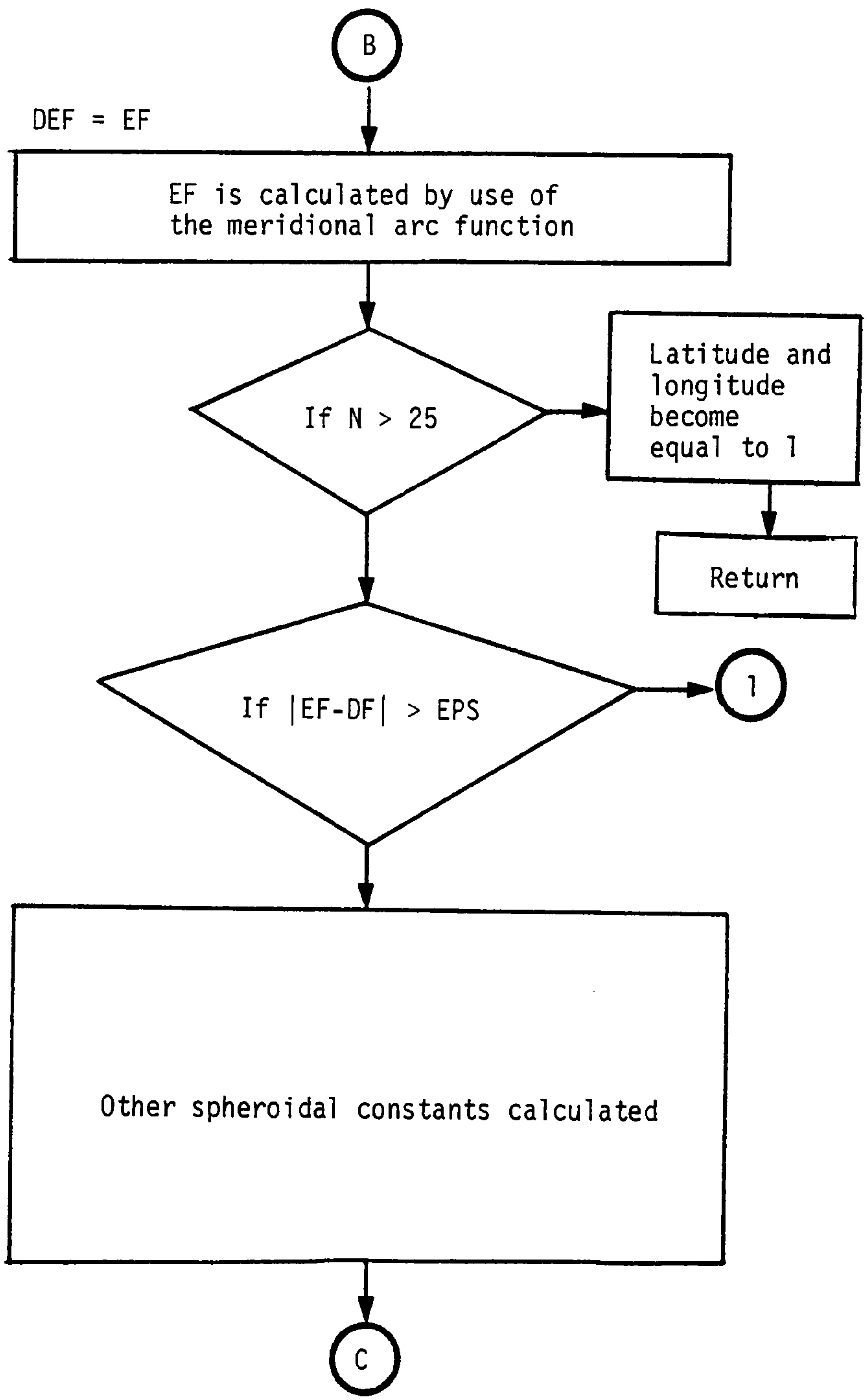


Figure B.10. (continued)

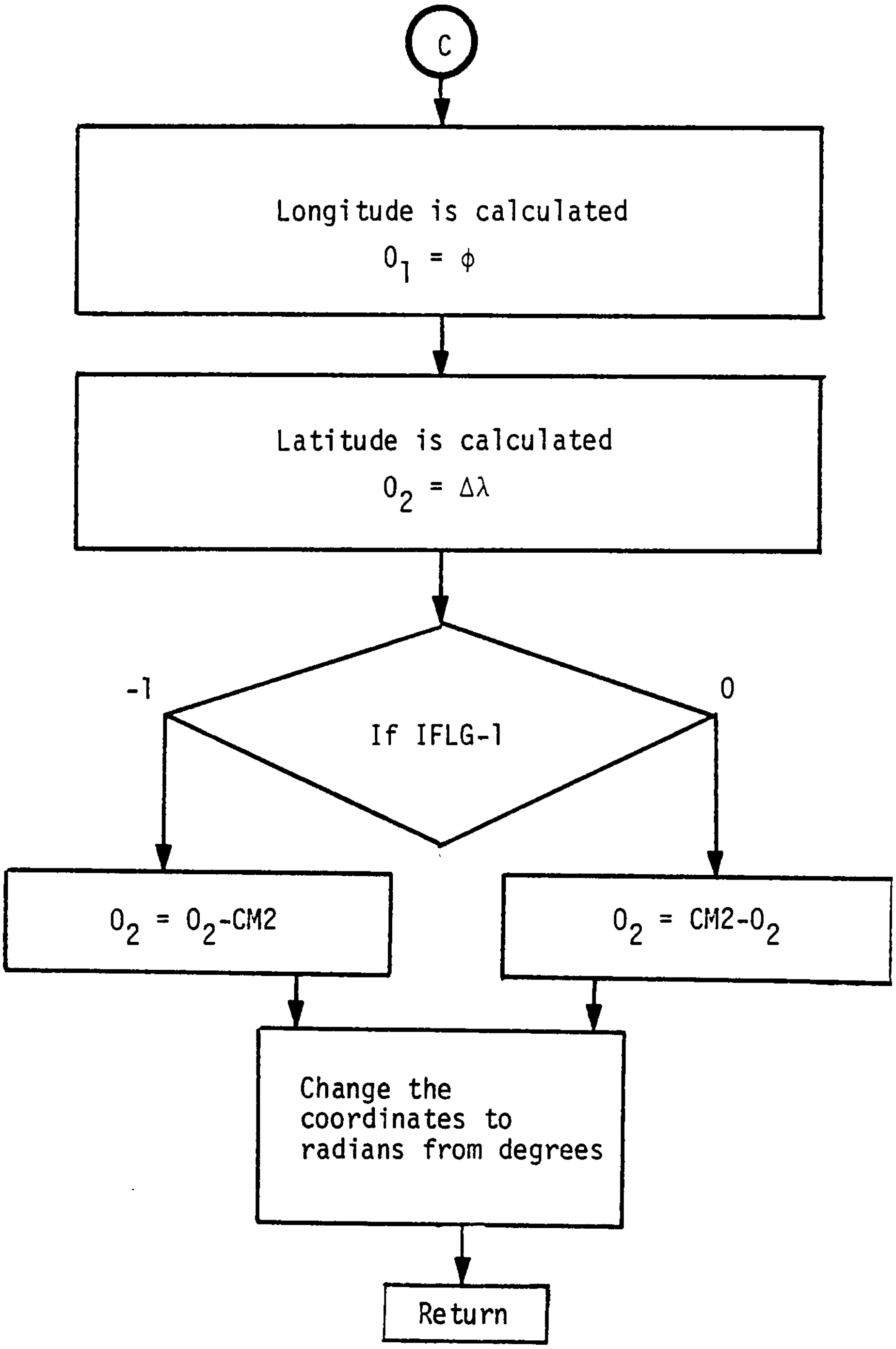


Figure B.10. (continued)

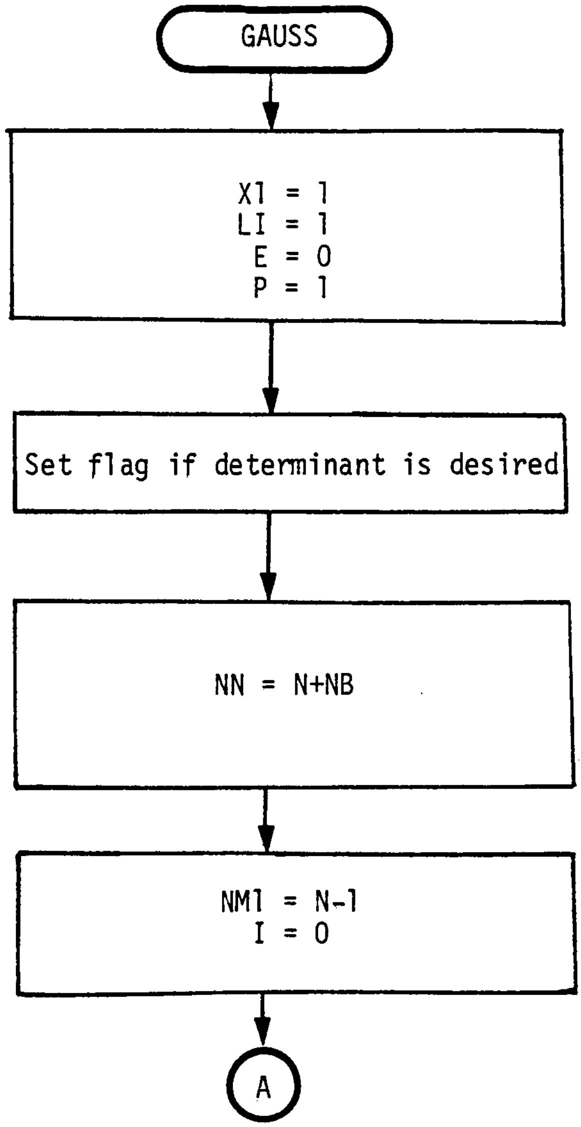


Figure B.11. Subroutine GAUSS flow chart.

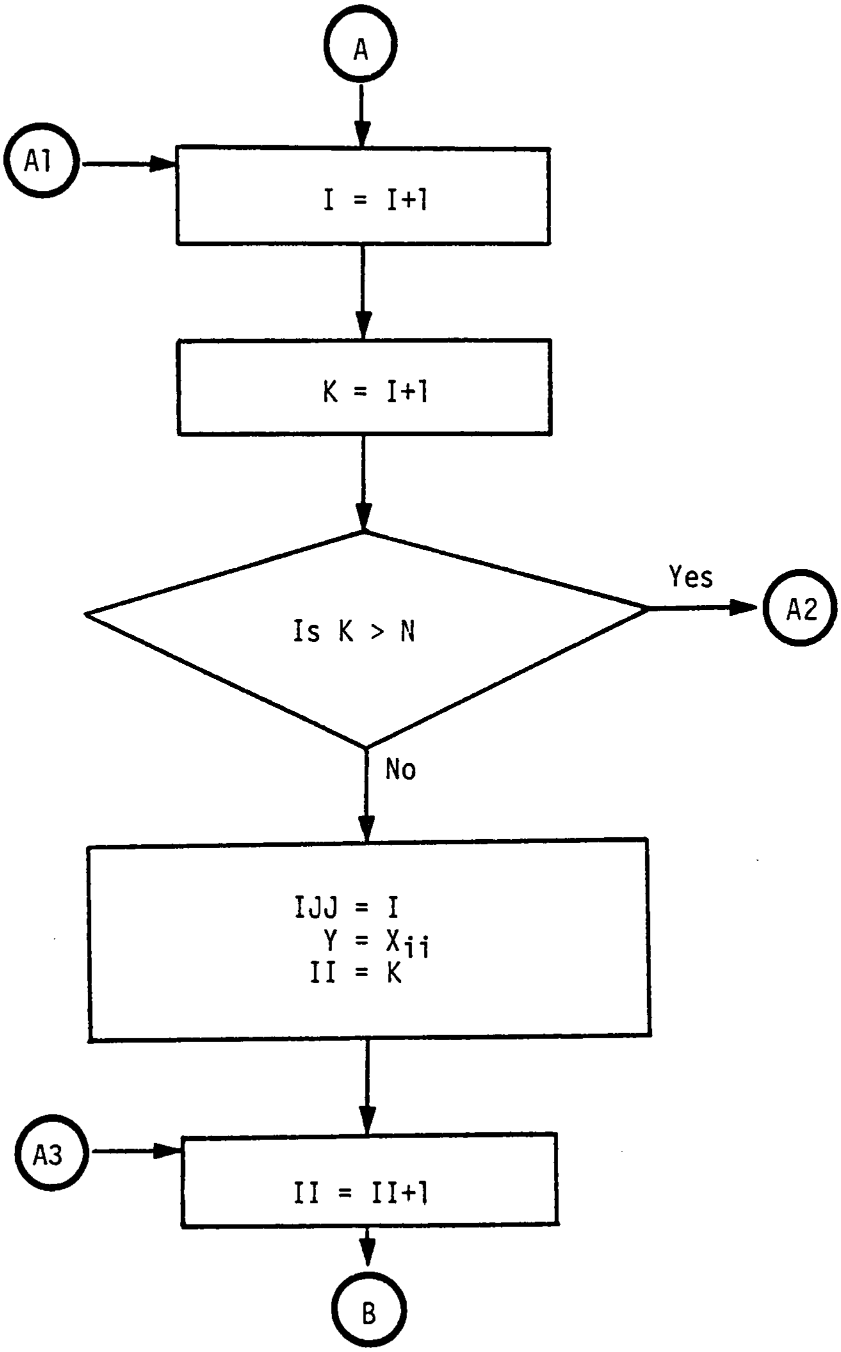


Figure B.11. (continued)

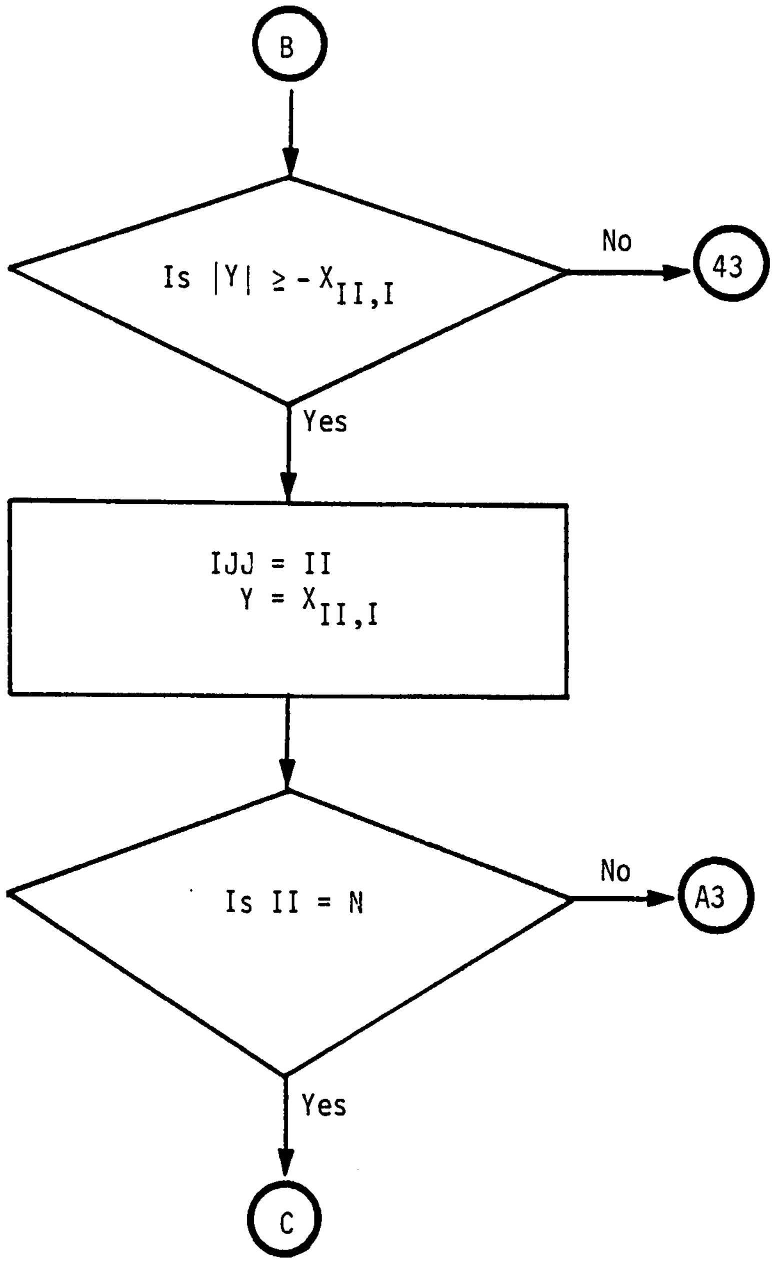


Figure B.11. (continued)

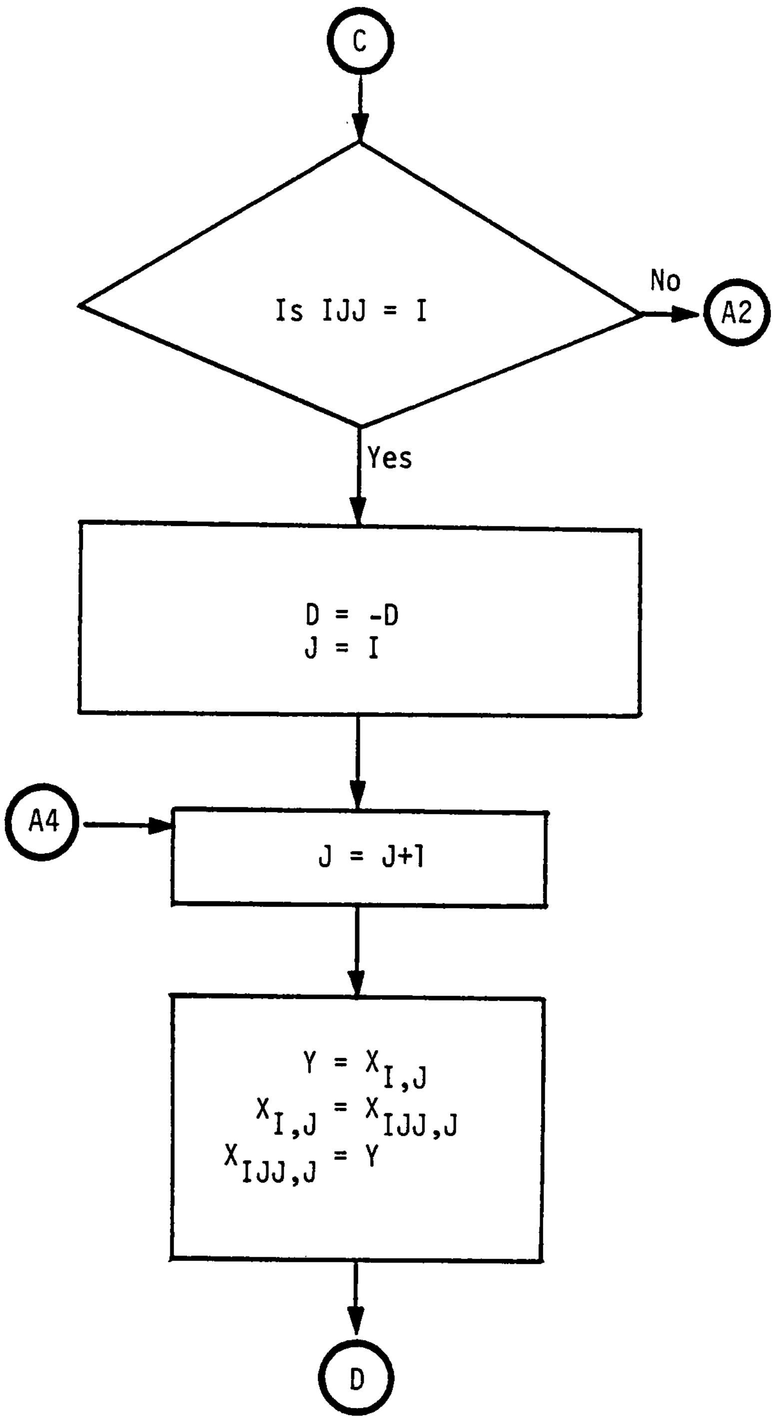


Figure B.11. (continued)

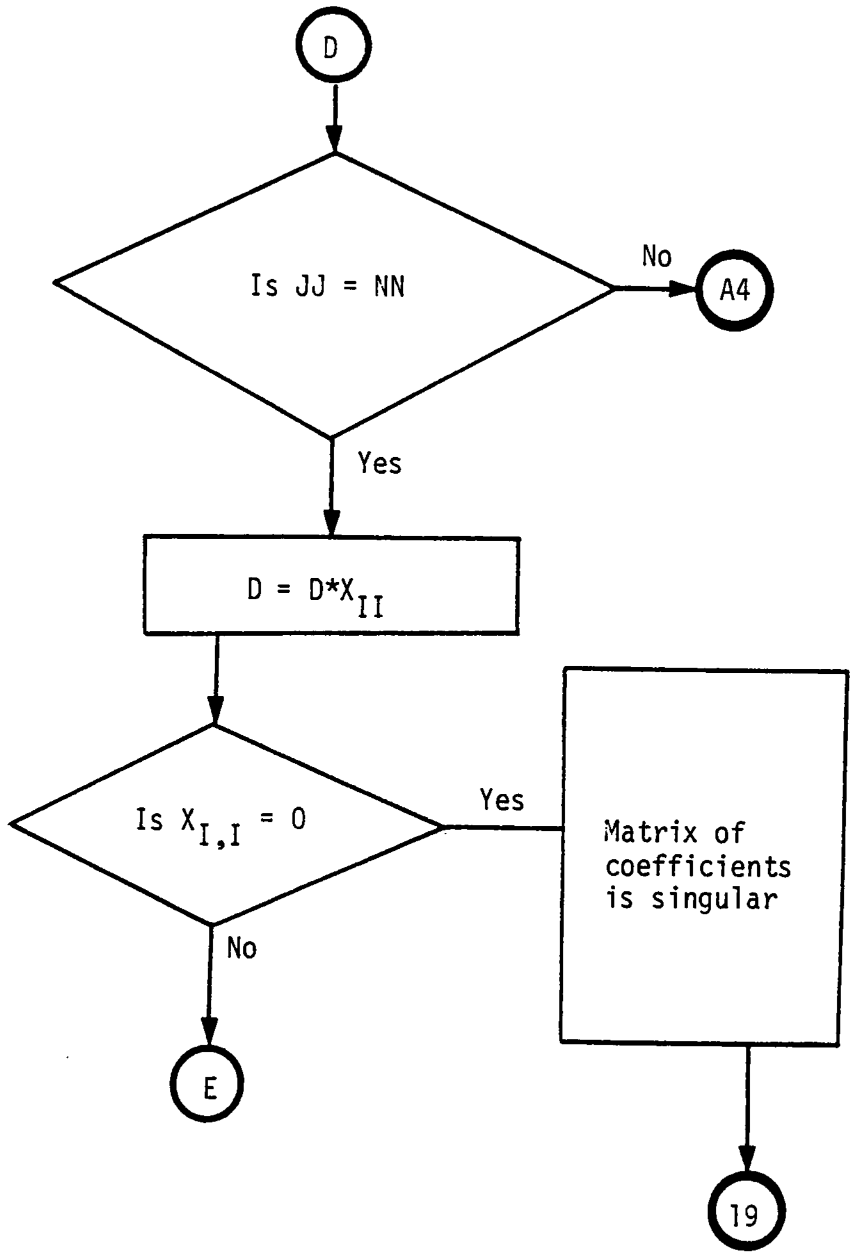


Figure B.11. (continued)

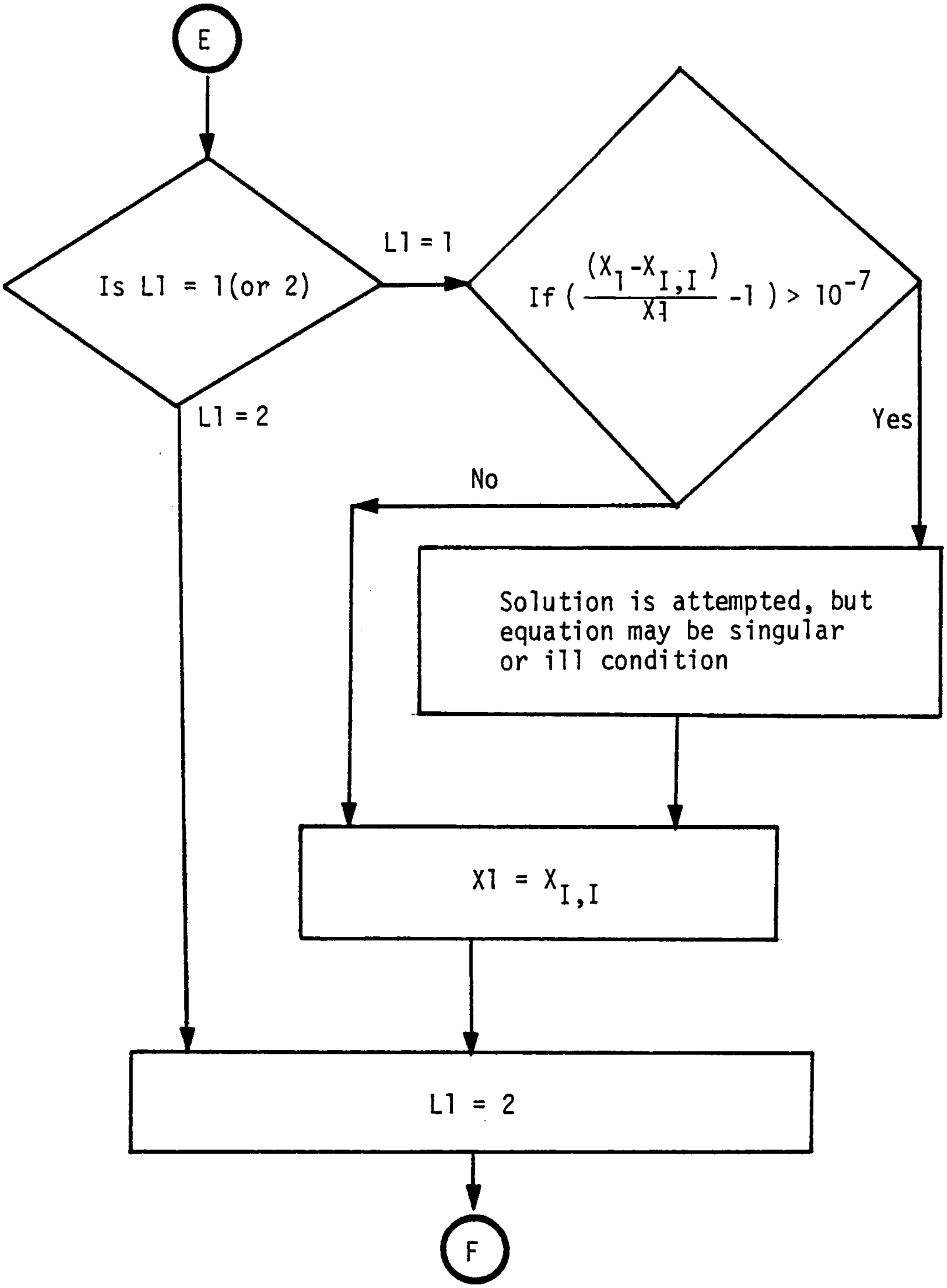


Figure B.11. (continued)

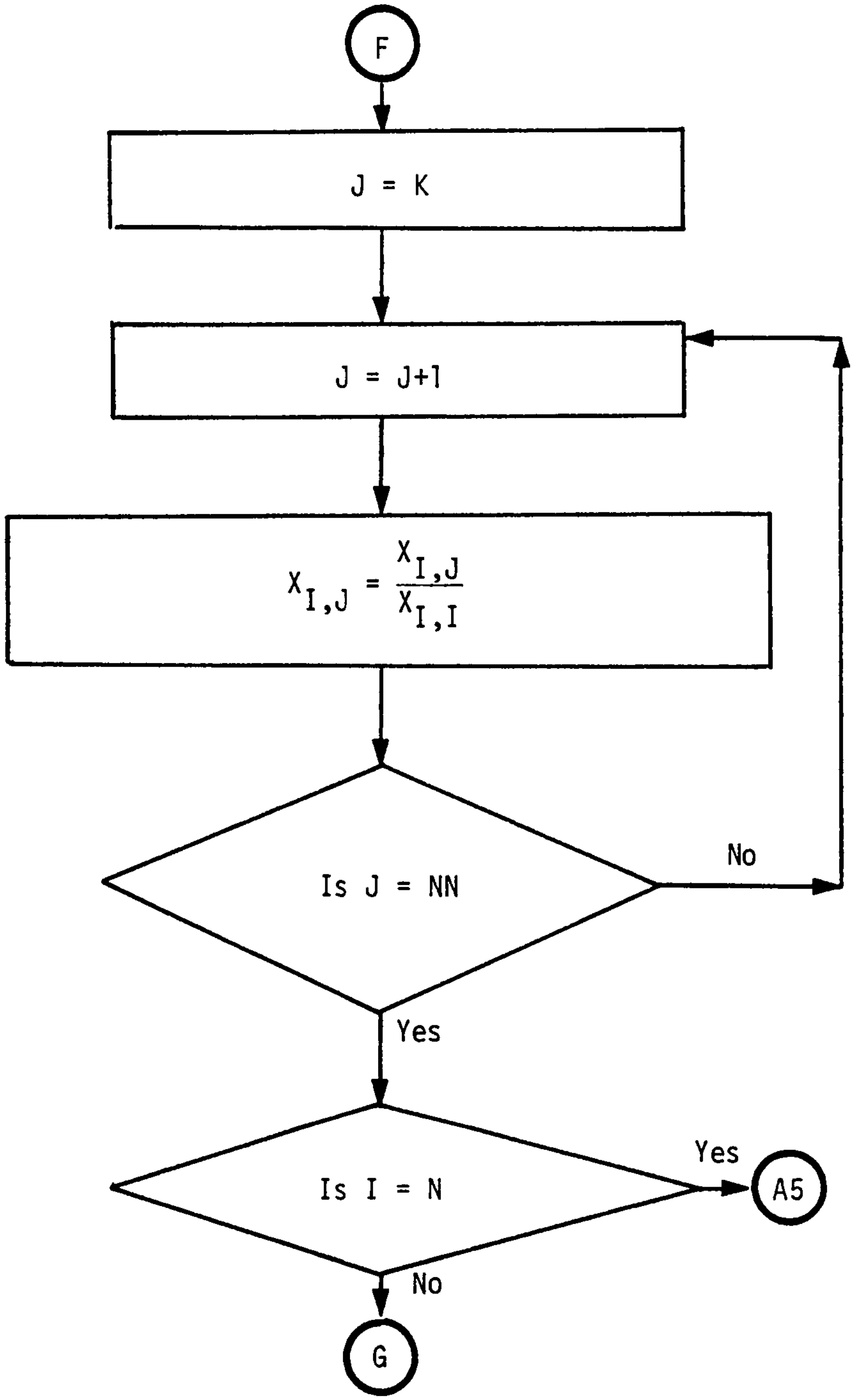


Figure B.11. (continued)

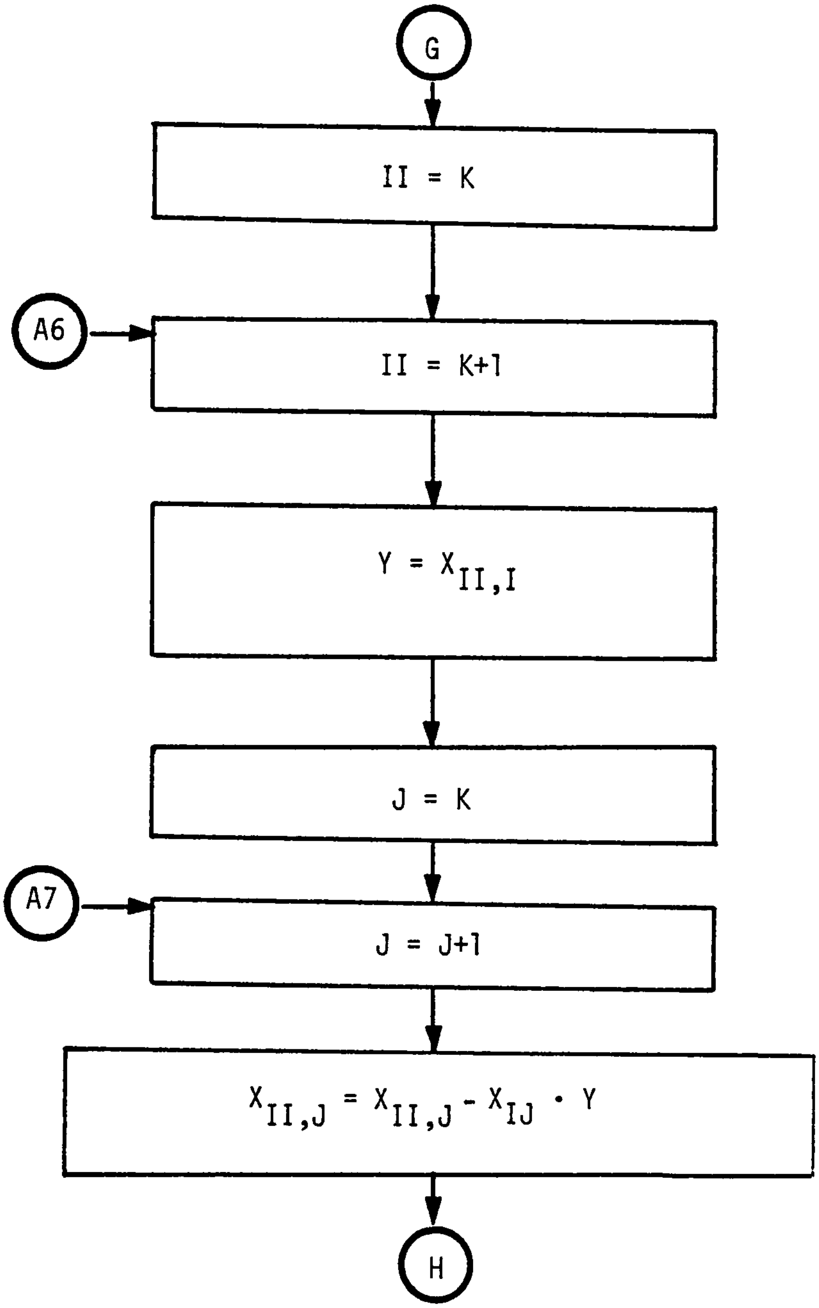


Figure B.11. (continued)

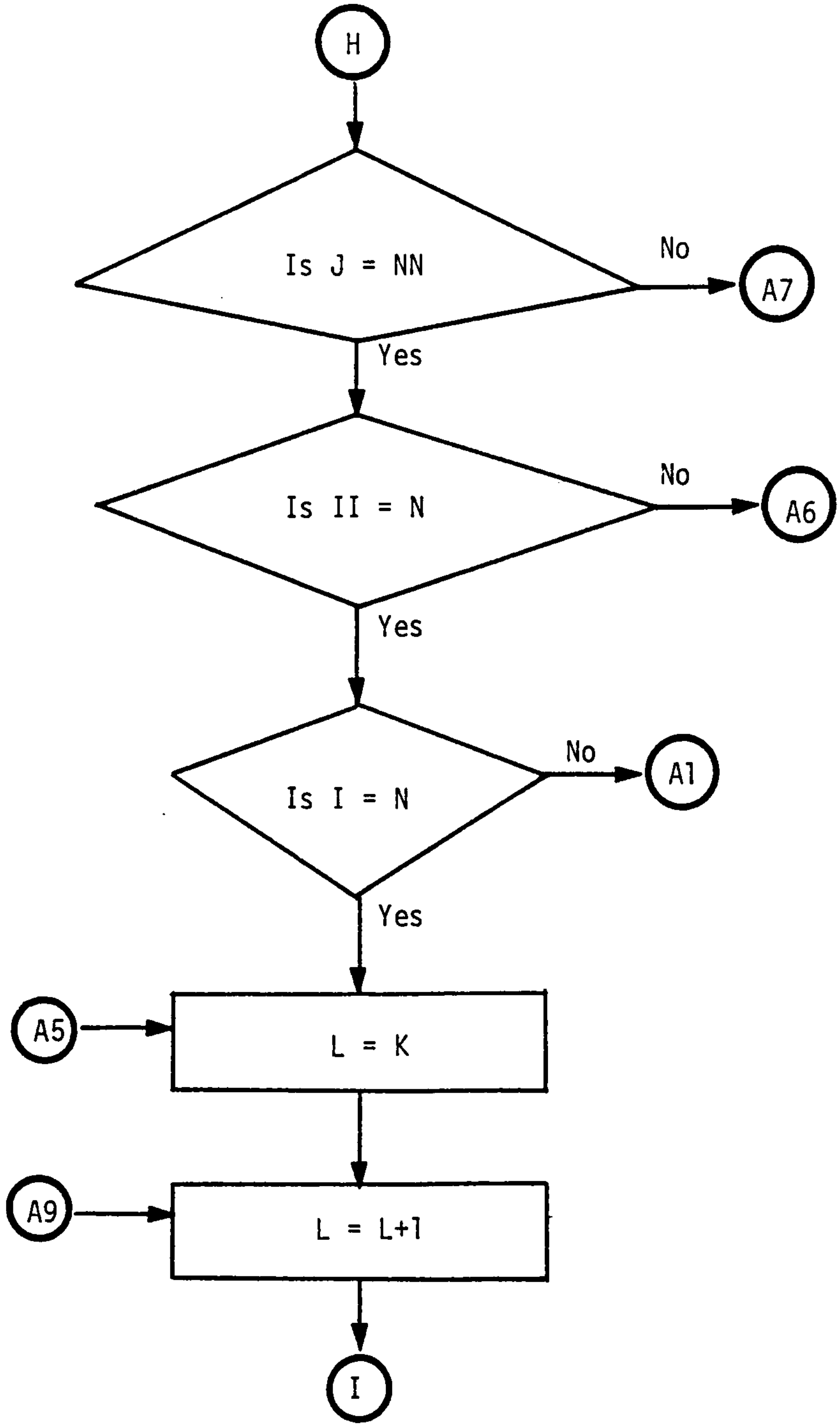


Figure B.11. (continued)

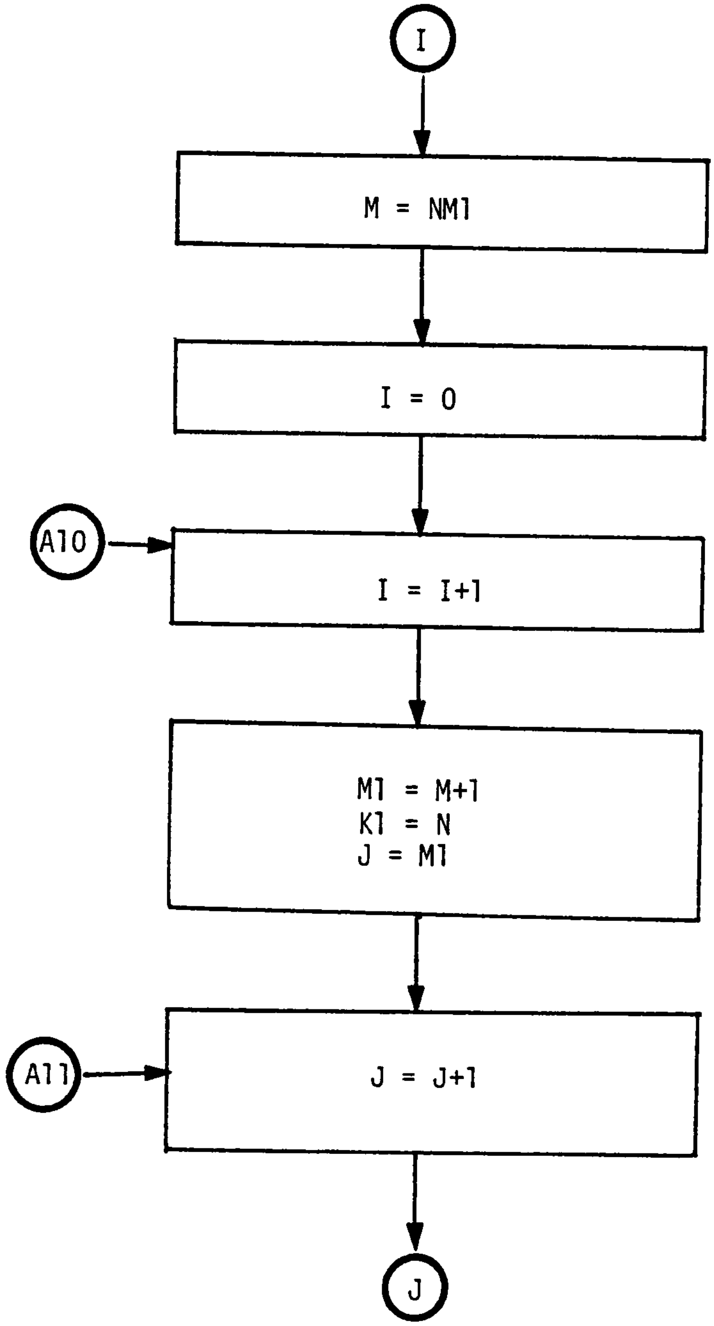


Figure B.11. (continued)

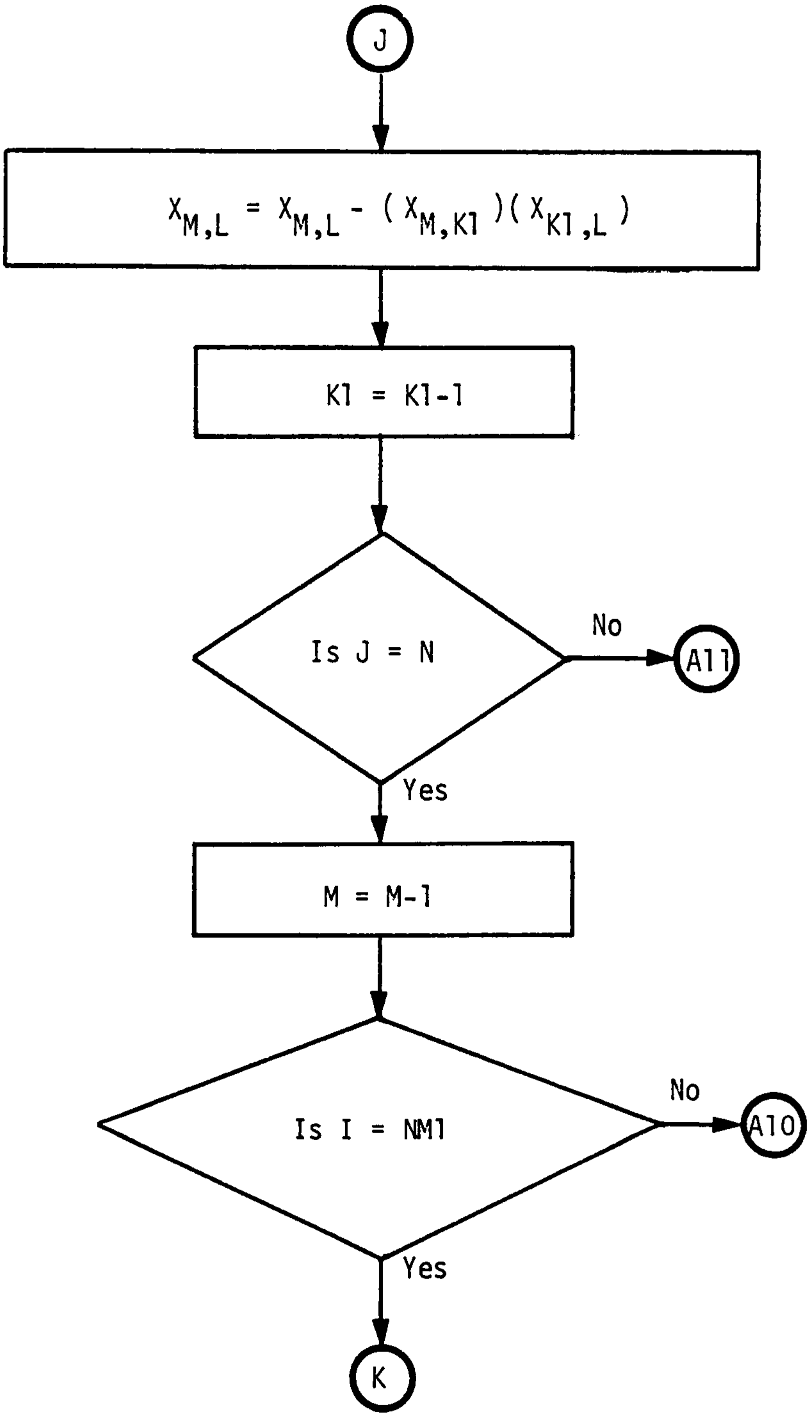


Figure B.11. (continued)

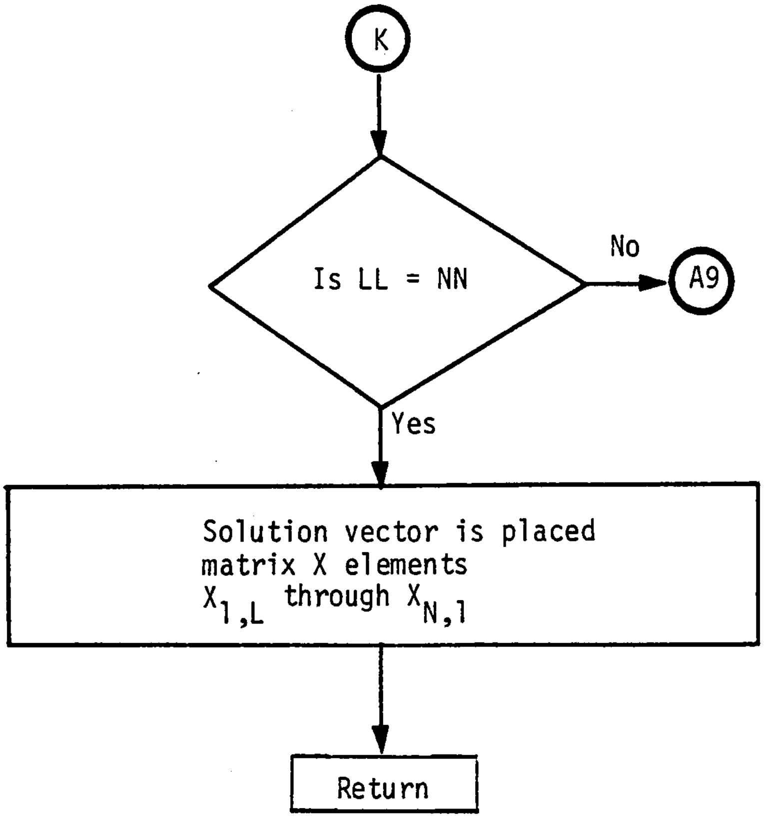


Figure B.11. (continued)

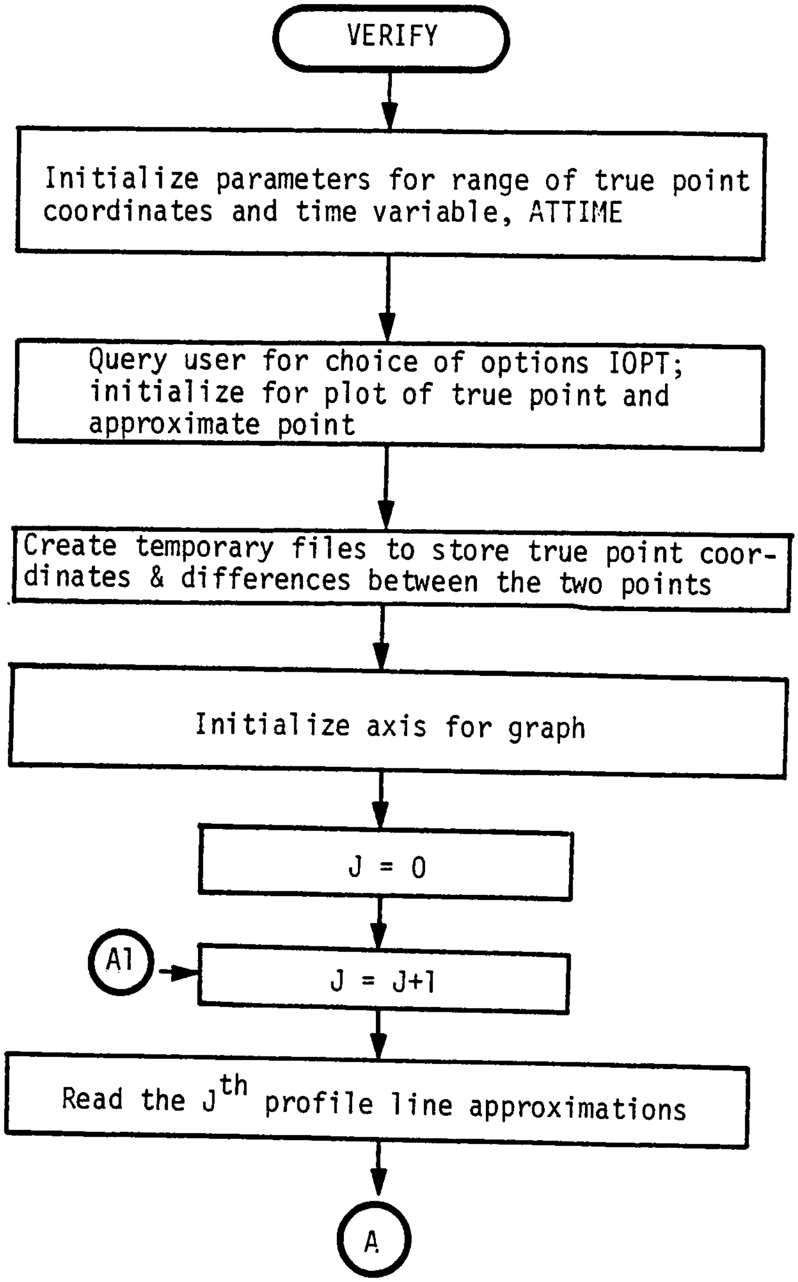


Figure B.12. Subroutine VERIFY flow chart.

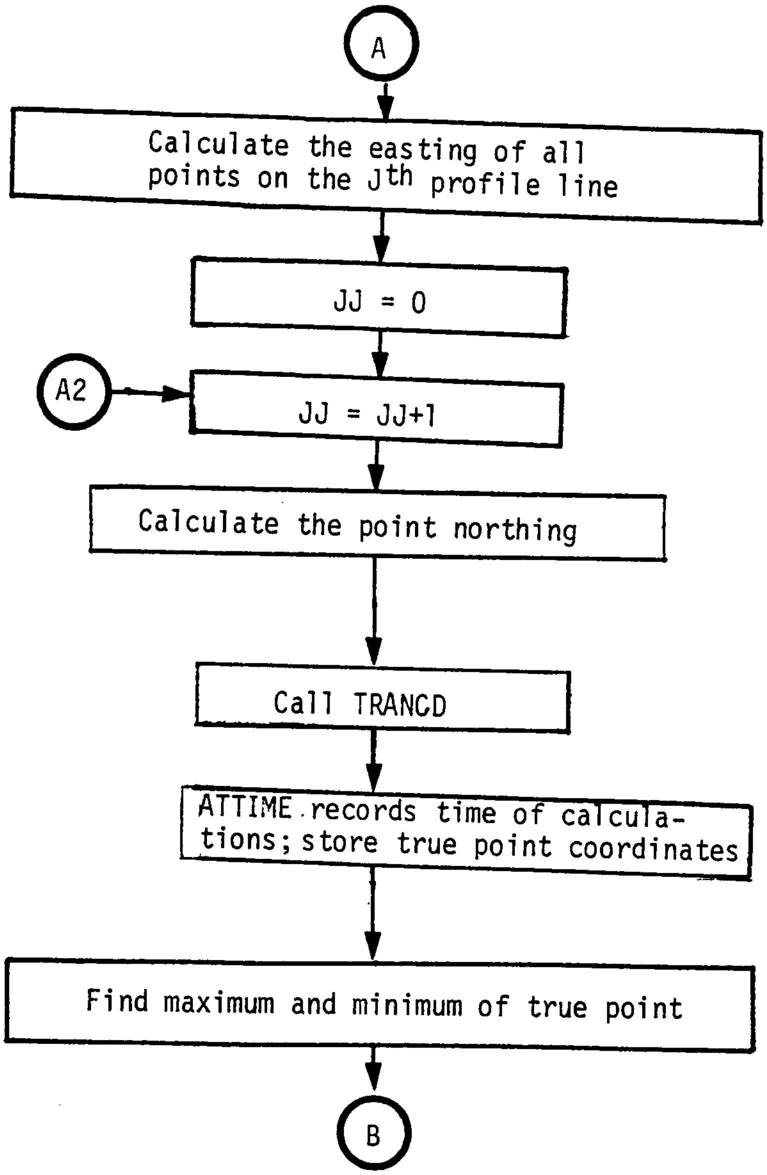


Figure B.12. (continued)

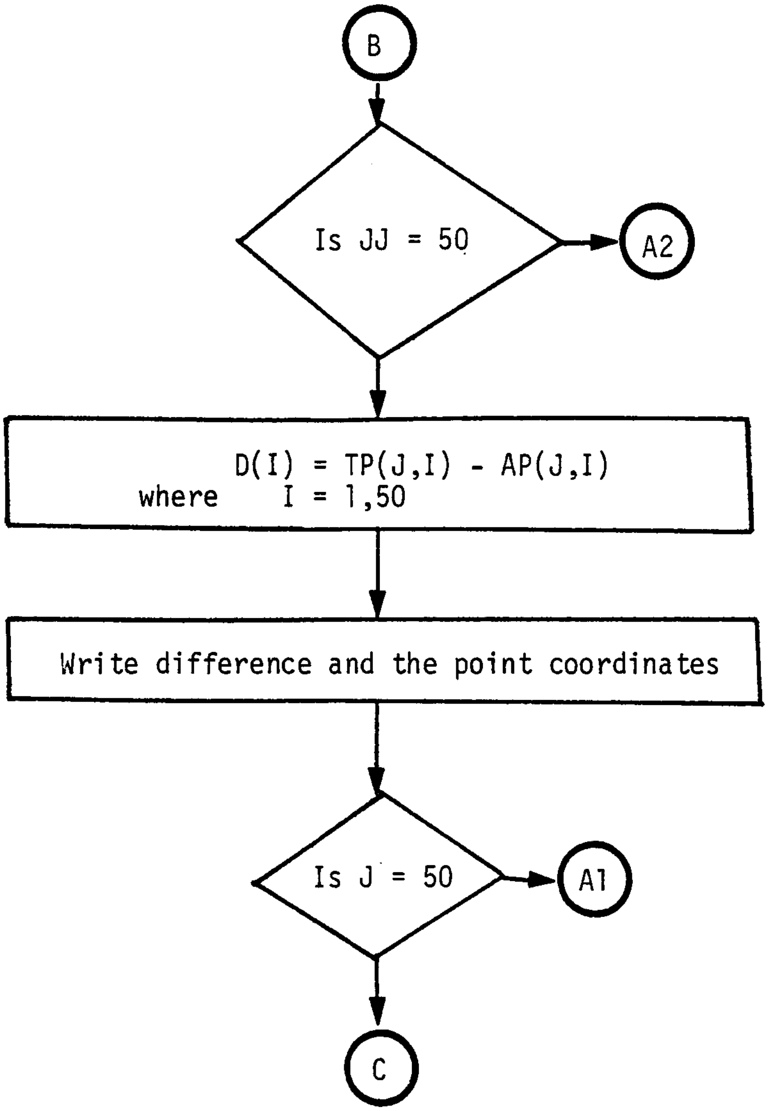


Figure B.12. (continued)

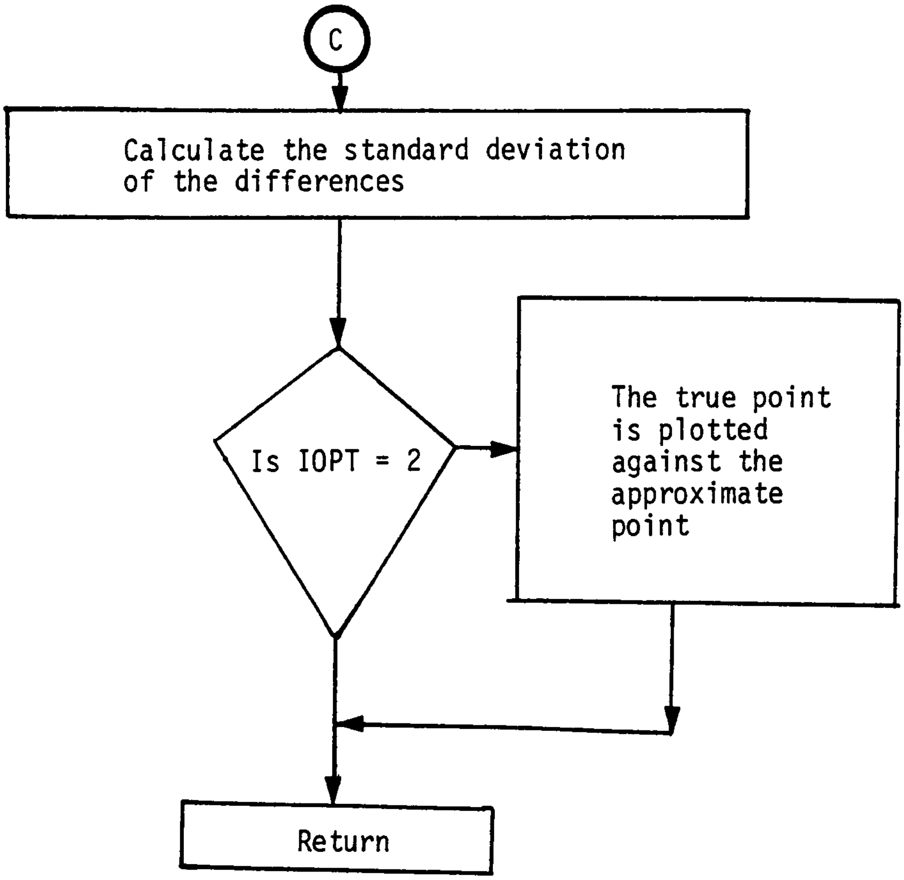


Figure B.12. (continued)

Main Program

```
        IMPLICIT REAL*8 (A-H),(O-2)
        COMMON/HEADER/LAT,LON,LATIN,LONIN,NM,MM,NFF,NFF,JCK,
     *  IREG,IO,IRIGHT,JO,ITOP
         COMMON ATIME,BTIME,CTIME
        COMMON ORLX,OLRY,PERX,PERY
        COMMON/UTM/EAST, ORTH,DEAST,DNORTH,AXIS(2),IHEMSP,ZONE,P(3)
        COMMON INTGR,ITSFLG,EPX,EPY
        BYTE FDATE(9),FTIME(8)
        REAL XYCS(300)
        TYPE 1000
  1000  FORMAT(/1X,'FUNCTIONAL TRANSFORMATION'
        ,'AND REGRIDDING SOFTWARE'/1X,'(GSA PROJECT M-194')
        CALL DATE(FDATE)
        CALL TIME(FTIME)
        TYPE 1001,(FDATE,FTIME)
  1001  FORMAT(20X,9A1,5X,8A1)
C
C       QUERY USER FOR NEEDED INFO
C
        CALL IINPUT
C
C       EDAS DISK FILE OR DMAAC TAPE FILE???????
C
        CALL TARGDF
  30    CONTINUE
         BTIME=0.0
         CTIME=0.0
C
C       EPSILON DEFAULTS(ONE MINUTE)
C
        EPX=1./60.
        EPY=1./60.
        CLOSE(UNIT=3)
C
C
        CALL TRAREG(BTIME,CTIME)
        CLOSE(UNIT=4)
        OPEN(UNIT=4,NSAME='DP:TRDATA.666',TYPE='OLD',
     *  ACCESS='DIRECT',RECORDSIZE=600)
        IYMIN=LAT
        IXMIN=LON
        IYMAX=MM
        IXMAX=NM
        CALL MONTOR(IXMIN,IYMIN,IXMAX,IYMAX)
```

```
      CLOSE(UNIT=4)
      CALL VERIFY(IXMIN,IYMIN,IXMAX,IYMAX)
      CLOSE(UNIT=9)
      CLOSE(UNIT=10)
      CLOSE(UNIT=11)
      CLOSE(UNIT=3)
      CLOSE(UNIT=4)
          TYPE 112,OLRX,OLRY,PERX,PERY,ATIME,BTIME,CTIME
112       FORMAT(1X,'X DEV,Y DEV;PERC X DEV;PERC Y DEV;'
     *       ,'ARMY RUN TIME;TIME FOR EQUATIONS CALCULATIONS;'
     *       ,';NEW APPROX TIME RUN'/,5X,7E16.8)
      STOP
      END
```

Subroutine IINPUT

```
      COMMON/RCBLEV/LENCOD,JMPMAX,RCB
      COMMON/HEADER/LAT,LOM,LATIN,LOMIN,NM,MM,MFF,MPF,JCK,
     *    IREG,IO,IRIGHT,JO,ITOP
      BYTE BMSG(SO),RCB(179),DLMARY(2)
      INTEGER WORD(3)
       DIMENSION IOTA(1000)
      INTEGER INTGR
      INTEGER ITAB(14)
      EQUIVALENCE(WORD,RCB),BMSG,RCB(17)),(JUMP,RCB(7))
      EQUIVALENCE(RCB(9),LMMSG),(RCB(177),DLMARY)
      DATA WORD/5,0,2/,JPMAX/2/
      DATA DLMARY/1H,1H,/
C
      CALL HEADR
      OPEN(UNIT=3,NAME='DP:MAPGEO.666',TYPE='NEW',ACCESS='DIRECT'
     *    FORM='UNFORMATTED',DISPOSE='SAVE',RECORDSIZE=(MM*2))
      WRITE(3'1)LAT,LON,LATIN,LONIN,MM,MM,NFF,NPF,JCK,IREG,IO,
     *    IRIGHT,JOITOP
      DO 444 MOSES=2,NM+2
      DO 333 NOAH=1,MM*5
       IOTA(NOAH)=(MOSES*10)+NOAH
  333     CONTINUE
      WRITE(3'MOSES)(IOTA(JOSE),JOSE=I,MM*4)
  444     CONTINUE
      CLOSE(UNIT=3)
      GO TO 666
  666 CONTINUE
      RETURN
      END
```

Subroutine HEADER

```
      COMMON/HEADER/LAT,LON,LATIN,LONIN,NM,MM,NFF,NPF,
     * JCK,IREG,IO,IRIGHT,JO,ITOP
      LAT=13
      LON=50
      LATIN=30
      LONIN=60
      NM=4
      MM=79
      NFF=16
      NPF=17
      JCK=1
      IREG=3091
      IO=9
      IRIGHT=594
      JO=9
      ITOP=1494
      RETURN
      END
```

Subroutine TARGDF

```
      IMPLICIT REAL*8(A-H),(O-Z)
      COMMON INTGR,ITSFLG,EPX,EPY
      COMMON/UTM/EAST, ORTH,DEAST,DNORTH,AXIS(2),IHEMSP,
       ZONE,P(3)
      COMMON/RCBLEV/LENCOD,JMPMAX,RCB
      BYTE ROB(179),BMSG(80)
      BYTE DLMARY(2),MSG(1)
      INTEGER WORD(3),BACKUP
      EQUIVALENCE(JUMP,RCB(7)),(RCB(9),LNMSG)
     *  ,(RCB(17),BMSG)
      EQUIVALENCE(WORD,RCB),(RCB(177),DLMARY)
      DATA WORD/5,0,2/,JMPMAX/2/
      DATA DLMARY/1H,1H,/
C
C
C
      JUMP=1
      GO TO 46
C
C
C**********************************************
C     USER DEFINES THE TARGET SYSTEM TO BE USED HERE
C**********************************************
 45   CALL STACK(RCB,LEVELS,2)
      IF(JUMP.NE.-1)GO TO 50
 46   TYPE 2
      TYPE 47
 47   FORMAT(1X,30('*'),/1X,'TARGET SYSTEM DEFINITION'/
     * ,1X,30('*')/5X,'1=UTM'/5X,
     * '2=LOCAL ORTHOGONAL'/1X,'ENTER 1 OR 2)'S)
      CALL GR(RCB,LEVELS,2)
      IF(JUMP.EQ.+1)GO TO BACKUP
      IF(BMSG(1).EQ.'1'.OR.BMSG(1).EQ.'2')GO TO 50
      TYPE 2
      TYPE 106
      GO TO 46
 50   DECODE(LNMSG,110,BMSG)ITSFLG
      ASSIGN 46 TO BACKUP
         GO TO (60,70)ITSFLG
C
C**********************************************
C     UTM IS THE TARGET SYSTEM
C**********************************************
C
```

```
60     CONTINUE
       TYPE 2
       TYPE 61
61     FORMAT(1X,33(1H*)/1X,'UTM DESCRIPTION OF TARGET'
    *  'SYSTEM'/1X,33(1H*))
63     CONTINUE
       CALL UTMQRY
       IF(JUMP.EQ.+1)GO TO BACKUP
       RETURN
2      FORMAT(LX,/')
100    FORMAT(1X,DATA INPUT TYPE'/1X,'ENTER 1 FOR DMAAC TAPE OR
    *  2 FOR EDAS DISK FILE > 'S)
105    FORMAT(A1)
106    FORMAT(5X,'ERROR------INVALID ENTRY,TRY AGAIN.')
110    FORMAT(I<LNMSG >)
150    FORMAT(20A1)
       END
```

Subroutine UTMQRY

```
        IMPLICIT REAL*8(A-H),(O-Z)
        COMMON/UTM/EAST, ORTH,DEAST,DNORTH,AXIS(2),IHEMSP,ZONE,P(3)
        COMMON/RCBLEV/LENCOD,JMPMAX,RCB
        BYTE BMSG(80)
        BYTE RCB(179),DLMARY(2),MSG(1)
        INTEGER WORD(3),BACKUP
        EQUIVALENCE(WORD,RCB),(BMSG,RCB(17))
        EQUIVALENCE(RCB(9),LNMSG),(RCB(177),DLMARY)
        EQUIVALENCE(JUMP,RCB(7))
        DATA WORD/5.0.2/.JMPMAX/2/2
        DATA DLMARY/1H.1H./
C
        IHEMSP=0
        DNORTH=1000
        DEAST=1000
C
C
C       GET THE NORTHING
C
  10    CALL STACK(RCB.LEVELS.2)
        IF(JUMP NE-1)GO TO 28
  21    TYPE 2
        TYPE 22
  22    FORMAT(1X.'ENTER NORTHING(METERS)>'S)
        CALL GR(RCB.LEVELS.2)
  28    IF(JUMP LE JMPMAX)RETURN
        ASSIGN 21 TO JUMPS
        DECODE(LNMSG.29.BMSG.ERR=260)ORTH
        ASSIGN 21 TO BACKUP
  29    FORMAT(F<LNMSG > 0)
C
C       GET THE EASTING
C
C
C
  30    CALL STACK(RCB.LEVELS.2)
        IF(JUMP NE-1)GO TO 37
  31    TYPE 2
        TYPE 32
  32    FORMAT(1X.'ENTER EASTING(METERS)>'S)
        CALL GR(RCB.LEVELS.2)
  37    CONTINUE
        IF(JUMP EQ.1)GO TO BACKUP
        ASSIGN 30 TO JUMPS
        DECODE(LNMSG.29.BMSG.ERR=260)EAST
        ASSIGN 31 TO BACKUP
```

```
  39    FORMAT(I<LNMSG >)
C
C       GET THE INCREMENT OF NORTHING
C
  40    CALL STACK(RCB.LEVELS.2)
        IF (JUMP.NE.-1)GO TO 48
  41    TYPE 2
        TYPE 42
  42    FORMAT(1X,'ENTER INCREMENT OF NORTHING(METERS)>'S)
        CALL GR(RCB.LEVELS.2)
  48    CONTINUE
        IF(JUMP.EQ.1)GO TO BACKUP
        IF(BMSG(1).EQ.'')GO TO 50
        ASSIGN 40 TO JUMPS
        DECODE(LNMSG,29,BMSG,ERR=260)DNORTH
        ASSIGN 41 TO BACKUP
C
C       GET THE INCREMENT OF EASTING
C
C
  50    CALL STACK(RCB.LEVELS.2)
        IF(JUMP.NE-1)GO TO 57
  51    TYPE 2
        TYPE 52
  52    FORMAT(1X.'ENTER INCREMENT OF EASTING(METERS)>'S)
        CALL GR(RCB.LEVELS.2)
  57    CONTINUE
        IF(JUMP.EQ.1)GO TO BACKUP
        IF(BMSG(1)EQ'')GO TO 60
        ASSIGN 51 TO JUMPS
        DECODE(LNMSG.29.BMSG.ERR=260)DEAST
        ASSIGN 51 TO BACKUP
C
C       GET THE HEMISPHERE OF TARGET SYSTEM
C
C
  60    CALL STACK(RCB.LEVELS.2)
        IF(JUMP.NE.-1)GO TO 67
  61    TYPE 2
        TYPE 62
  62    FORMAT(1X,'ENTER HEMISPHERE 0 FOR NORTH AND 1 FOR SOUTH>'S)
        CALL GR(RCB.LEVELS.2)
  67    CONTINUE
        IF(JUMP.EQ.1)GO TO BACKUP
        IF(BMSG(1).EQ.'')GO TO 70
        ASSIGN 60 TO JUMPS
        DECODE(LNMSG.39,BMSG,ERR=260)IHEMSP
        ASSIGN 61 TO BACKUP
```

```
C
C       GET THE HEMISPHERE OF
C       GET UTM ZONE NUMBER
C
  70    CALL STACK(RCB.LEVELS.2)
        IF(JUMP NE.-1)GO TO 77
  71    TYPE 2
        TYPE 72
  72    FORMAT(1X,'ENTER UTM ZONE(1 THRU 60)>'S)
        CALL GR(RCB.LEVELS.2)
  77    IF(JUMP.EQ.1)GO TO BACKUP
        IF(BMSG(1).GT.60.OR.BMSG(1).LT.1)GO TO 71
        ASSIGN 70 TO JUMPS
        DECODE(LNMSG.29.BMSG.ERR=260)ZONE
C
C
C       GET THE MAJOR AXIS OF SPHERE
C
C
C
  90    CALL STACK(RCB.LEVELS.2)
        IF(JUMP.NE.-1)GO TO 97
  91    TYPE 2
        TYPE 92
  92    FORMAT(1X.'SPECIFY SPHERICAL TYPE'/1X.'1=FOR INTERNATIONAL'
     *  /1X.'1=FOR CLARKE(1866)'/1X.'3=FOR CLARKE(1880)'/1X.'
         4=FOR
     *  EVEREST'
     *  /1X.'5=FOR BESSEL'/1X.'6=FOR MODIFIED EVEREST'/1X.
     *  '7=FOR AUSTRALIAN NATIONAL//1X.'8=FOR AIRY'/1X.'9=FOR
     *  MODIFIED AIRY'/1X.'ENTER ONE OF THE ABOVE>'S)
        CALL GR(RCB.LEVELS.2)
  97    CONTINUE
        IF(JUMP EQ.1)GO TO BACKUP
        ASSIGN 90 TO JUMPS
        IF(JUMP EQ JMPMAX)RETURN
        IF(BMSG(1).LT'1'.OR BMSG(1)GT'9')GO TO 91
        DECODE(LNMSG.99.BMSG.ERR=250)N
  99    FORMAT(I<LNMSG >)
        ASSIGN 91 TO BACKUP
C
C       SETUP OF SEMI AXIS OF SPHERE
C
        CALL STAB(N,A,B)
        AXIS(1)=A
        AXIS(2)=B
C
        RETURN
C
  260   GO TO JUMPS
  2     FORMAT(/)
        END
```

Subroutine STAB

```
      IMPLICIT REAL*B(A-H),(O-Z)
      GO TO(10,20,30,40,50,60,70,80,90)N
10    A=6378388.000
      F=1./297.000000
      GO TO 100
20    A=6378206.4000
      F=1./294.978698
      GO TO 100
30    A=6378249.145
      F=1./293.465000
      GO TO 100
40    A=6377276.3452
      F=1./300.801700
      GO TO 100
50    A=6377397.1550
      F=1./299.152813
      GO TO 100
60    A=6377304.053
      F=1./300.801700
      GO TO 100
70    A=6378160.000
      F=1./298.250000
      GO TO 100
80    A=6377563.396
      F=1./299.324965
      GO TO 100
90    A=6377340.189
      F=1./299.324959
      GO TO 100
100   B=A*(1-F)
      RETURN
      END
```

Subroutine TRAREG

```
      IMPLICIT REAL*8(A-H),(O-Z)
C
C     TARGET SYSTEM GRID SYSTEM DIMENSIONS
C
      PARAMETER LA=50,LB=50,LAB=300
C     LA IS THE NUMBER OF TARGET SYSTEM PROFILES
C     LB IS THE NUMBER OF TARGET SYSTEM  POINTS PER PROFILE
C
C
      COMMON INTGR,ITSFLG,EXP,EPY
      DIMENSION ITAB(1000)
      COMMON/HEADER/LAT,LON,LATIN,LONIN,NM,MM,NFF,NPF,JCK,
     *    IREG,IO,IRIGHT,JO,ITOP
      COMMON/UTM/EAST,ORTH,DEAST,DNORTH,AXIS(2),IHEMSP,ZONE,P(3)
      DIMENSION A(3.3),AA(3.4),XT(3),YT(3),AI(3.3),
     * DEE(3),WORK(3.1),IHLD(3),CEE(3),R(3),S(3),CPRIME(4)
C
      DIMENSION C(4.4)
      REAL XYCS(LAB)
C     CREATE FILE TRDATA,666(X,Y,C1,C2,C3,C4)
C     TRANSFORM X AND Y WITH ITS 4 COEFF'S OF POLY
C
      OPEN(UNIT=4,NAME='DP TRDATA.666',TYPE='NEW',ACCESS='DIRECT',
     * RECORDSIZE=600)
      OPEN (UNIT=3,NAME='DP MAPGEO.666',TYPE='OLD',ACCESS='DIRECT',
     * READONLY)
C
C     INPUT SYSTEM(GEOGRAPHY MAP MATRIX)LIMITS
C
      YMX=FLOAT(LAT)+FLOAT(MM=1))*(FLOAT(LATIN)/36000.))
      XMX=FLOAT(LON)+(FLOAT(NM-1))*(FLOAT(LONIN)/36000.))
          CTIME=0.0
          BTIME=0.0
C
C     UTM OR LOCAL ORTHOGONAL
C
      IF(ITSFLG.EQ.1)GO TO 10
      XO=P(2)
      YO=P(1)
      DX=DEAST
      DX=DNORTH
      GO TO 11
C
C     UTM OPERATIONS START HERE
C
```

```
  10    XO=EAST
        YO=ORTH
        DX=DEAST
        DY=DNORTH
C
C       SET PARAMETERS FOR SESOMI TO SOLVE SIMULTANEOUS EQUATIONS
C
C       N IS NUMBER OF SIMULTANEOUS EQUATIONS
C       NB IS NUMBER OF COLUMNS AUGMENTED TO A MATRIX
  11    CONTINUE
C
        NB=1
        NS=0
        MNI=3
        IC=0
        ID=1
        IS=0
C
C***********************************************
C       PROCESS EACH PROFILE LINE IN THE TARGET SYSTEM
C***********************************************
C
C
        DO 1000 I=1.LA
         RTIME=SECONDS(0.0)
C
C       DETERMINE THE 3 POINTS IN THE TARGET SYSTEM PROFILE LINE
C
        XT(1)=YO
        YT(2)=YO+24.0*DY
        YT(3)=YO+49.0*DY
C
C       INITIALIZE OUTPUT BUFFER
        DO 12 II=1.LAB
  12    XYCS(II)=0.0
C
        XTS=XO+FLOAT(I-1)*DX
C
C       IZONE IS UTM ZONE BETWEEN 1-60
C       AXIS IS THE MAJOR AND MINOR AXIS OF SPHEROID(A AND B)
C       HEMISPHERE IS 0 FOR NORTH AND 1 FOR SOUTH
C       R ARRAY IS LAT,LON,HEMISPHERE(0 OR 1)
C       FIRST POINT OF PROFILE LINE TRANSFORMATION
C
```

```
      R(1)=XTS
      R(2)=YT(1)
      IF(ITSFLG.EQ.2)GO TO 9
      R(3)=IHEMSP
      KFLAT=1
      HO=0
      GO TO 13
 9    CONTINUE
C     LOCAL ORTHOGONAL PARAMETERS SETUP
      R(1)=YT(1)
      R(2)=XTS
      R(3)=0
      HO=0
      IHEMSP=0
      ZONE=0
      KFLAG=2
C
 13   CONTINUE
C
      CALL TRANCD(R,S,AXIS,P,HO,IHEMSP,ZONE,KFLAG)
C
C     SET UP "A" MATRIX
C
      A(1,1)=1
      A(1,2)=R(2)
      A(1,3)=R(2)**2
      XT(1)=S(2)
      YT(1)=S(1)
C     SECOND POINT OF PROFILE LINE TRANSFORMATION
      R(2)=YT(2)
      IF(ITSFLG.EQ.1)GO TO 1
      A(1.2)=R(1)
      A(1.3)=R(1)**2
      XT(1)=S(2)
      YT(1)=S(1)
      R(1)=YT(2)
      R(2)=XTS
 1    CONTINUE
C
C
      CALL TRANCD(R,S,AXIS,P,HO,IHEMSP,ZONE,KFLAG)
C
C     SECOND ROW OF MATRIX
C
      A(2.1)=1
      A(2.2)=R(2)
      A(2.3)=R(2)**2
      XT(2)=S(2)
      YT(2)=S(1)
C     LAST POINT OF PROFILE LINE TRANSFORMATION
      R(2)=YT(3)
```

```
C
      IF(ITSFLG.EQ.1)GO TO 2
      A(2.1)=1
      A(2.2)=R(1)
      A(2.3)=R(1)**2
      XT(2)=S(2)
      YT(2)=S(1)
      R(1)=YT(3)
      R(2)=XTS
  2   CONTINUE
C
      CALL TRANCD(R,S,AXIS,P,HO,IHEMSP,ZONE,KFLAG)
C
C     THIRD ROW OF MATRIX
C
      A(3.1)=1
      A(3.2)=R(2)
      A(3.3)=R(2)**2
      XT(3)=S(2)
      YT(3)=S(1)
C
C
C
      IF(ITSFLG.EQ.1)GO TO 3
      A(3.1)=1
      A(3.2)=R(1)
      A(3.3)=R(1)**2
      XT(3)=S(2)
      YT(3)=S(1)
  3   CONTINUE
      DO 18 K=1.3
  18  CONTINUE
C
C
C     AUGMENT MATRIX "A" FOR GAUSS ELIMINATION METHOD
C     FIND THE COEFFICIENTS C(AA)FOR THE SET X EQUATIONS
C
C
C
  18  DO 30 II=1.3
      DO 20 I3=1.3
  20  AA(II,I3)=A(II.I3)
  30  AA(II.4)=XT(II)
C
C
C     X=C1+C2*Y+C3*Y**2
C
```

```
      N=3
      CALL GAUSS(AA,N,NB,NS,MNI,D,IR,E,ID)
      IF(E.NE.O.)GO TO 42
      CEE(1)=AA(1.1)
      CEE(2)=AA(2.1)
      CEE(3)=AA(3.1)
      TYPE 31,(CEE(JJ),JJ=1.3)
 31   FORMAT(/5X,'THE X APPROXIMATING EQN COEFF ARE',3(5X,E16.8))
C
CC    AUGMENT AI AGAIN FOR THE Y SET OF EQUATIONS
C     FIND THE COEFFICIENTS D(AA)FOR THE Y SET OF EQUATIONS
C
      DO 40 II=1.3
      DO 35 I3=1.3
 35   OO(II,I3)=A(II,I3)
 40   AA(II,4)=YT(II)
C
C     Y=D1+D2*Y+D3*Y**2
C
      CALL GAUSS(AA,N,NB,NS,MNI,D,IR,E,ID)
      IF(E.NE.O.)GO TO 42
      DEE(1)=AA(1.1)
      DEE(2)=AA(2.1)
      DEE(3)=AA(3.1)
      TYPE 41,(DEE(II),II=1.3)
 41   FORMAT(5X,'THE Y APPROXIMATING EQN COEFF ARE',3(5X,E16,8))
       BTIME=BTIME+SECNDS(RTIME)
      GO TO 44
C
C
 42   TYPE 43,XTS
 43   FORMAT(5X,'SOLUTION IS NOT ABLE TO BE FOUND'/
     * 5X,'NO APPROXIMATE EQUATIONS FOR PROFILE LINE',E16.5,
     * /5X,'ALL NODE VALUES(X,Y,C1,C2,C3,C4)ARE SET TO ZERO')
      GO TO 500
C
C*********************************************
C
C     REGRIDDING STARTS HERE
C
C*********************************************
C************************************************
C
C     NOW THAT THE TWO APPROXIMATE EQUATIONS ARE KNOWN EACH POINT
C     IN THE TARGET PROFILE LINE IS APPROXIMATED TO THE INPUT
C     GEOGRAPHIC SYSTEM BY THESE APPROXIMATIONS EQUATIONS
C
C*********************************************
```

```
  44    CONTINUE
        DO 525 J=1,LD
        DO 5026 JR=1.4
        CPRIME(JR)=0
        DO 5032 JT=1.4
        C(JT,JR)=0
  5032  CONTINUE
  5026  CONTINUE
C       M IS THE LOCATION IN THE OUTPUT PROFILE ARRAY XYCS WHERE
         THE POINT IS STORED
C       YTS IS THE TARGET SYSTEM POINT IN THE PROFILE LINE
        M=(J-1)*6+1
        ZTIME=SECONDS(0.0)
        YTS=YO+FLOAT(J-1)*DY
C
C       COMPUTE THE APPROXIMATE X AND APPROXIMATE Y, AND STORE IN
         XYCS ARRAY---
            CE1=CEE(1)
            CE2=CEE(2)*YTS
            CE3=CEE(3)*(YTS**2)
            AX=CE1+CE2+CE3
            DE1=DEE(1)
            DE3=DEE(3)*(YTS**2)
            DE2=DEE(2)*YTS
            AY=DE1+DE2+DE3
            CTIME=CTIME+SECONDS(ZTIME)
C
        XYCS(M)=SNGL(AX)
        XYCS(M+1)=SNGL(AY)
C
C       AFTER ALL FIFTY POINTS HAVE BEEN PROCESSED ON THIS TARGET
C       PROFILE LINE THE TABLE CONTAINING AX,AY,C1,C2,C3,C4,OF EACH
C       POINT IS PUT OUT ON DISK
  500   CONTINUE
C
  1000  CONTINUE
            CTIME=CTIME
            BTIME=BTIME
        RETURN
        END
```

Subroutine TRANCD

```
      IMPLICIT REAL*B(A-H),(O-Z)
      DIMENSION R(3),S(3),P(3),F(S),SEMI(2),RP(3)
  100 CALL MOVE(R,RP,3)
C
C     UTM TO GEOGRAPHIC
C
    1 F(1)=RP(1)
      F(2)=RP(2)
      F(3)=ZONE
      F(4)=SEMI(1)
      F(5)=SEMI(2)
      F(6)=FLAG
      CALL UTMFLH(F,S)
      S(3)=RP(3)
      RETURN
C
C
C
      END
```

Subroutine MOVE

```
      IMPLICIT REAL*B(A-H),(O-Z)
      DIMENSION A(N),B(N)
      DO 10 I=1,N
10    B(I)=A(I)
      RETURN
      END
```

Subroutine UTMFLH

```
      IMPLICIT REAL*8(A-H),(O-Z)
      DIMENSION F(6),O(2),))(2)
  7   CONTINUE
      IFLAG=0
      IF(F(4)-10000000)20,20.10
 10   FTMT=12./39.37
C
C
      F(4)=F(4)*FTMT
      F(5)=F(5)*FTMT
      IFLAG=1
 20   IF(F(3)-39.0)40,40,30
 30   EN=F(3)-30
  .   CM1=3.*(EN*2-1.)
      GO TO 50
 40   EN=30.-F(3)+1
      CM1=-3.*(EN*2).-1.)
 50   CM2=CM1*3600
      IF(F(1)-500000.)70,60,60
 60   F(1)=F(1)-500000
      IFLG=1
      GO TO 80
 70   F(1)=500000-F(1)
      IFLG=0
 80   IF(F(6)90,100,90
 90   F(2)=10000000.-F(2)
C
C     E IS ECCENTRICITY SQUARED
C
 100  E=1-(F(5)/F(4))**2
C
C     A,B,C, ARE PLYS FROM PI OF CWRIYR UP
C
C
      A=1-.25*E-046875*E**2-.01953125*E**3
      B=375*E+09375*E**2+0439453125*E**3
      C=05859375*E**2+0439453125*E**3
      D=01139322916*E**3
C
C     ITERATIVE CALCULATION FOR FOOT PT, LATITUDE
C
      EPS=5E-8
      N=0
      EF=F(2)/(A*F(4)*.9996)
```

```
110   N=N+1
      DEF=EF
      EF=(B*OSIN(2.*DEF)-C*OSIN(4.*DEF)+D*DSIN(6.*DEF)
     *    +F(2)/F(4)*.9996))/A
      IF(N-25)120.130
120   IF(DABS(EF-DEF)-EPS)140,110,110
130   O(1)=99999999
      O(2)=99999999
      GO TO 210
C
C     RHO IS DCOSECANT OF ONE SECOND
C
140   RHO=206264.8062471
C
C     T IS TAND PHI
C
C     DSINEF=DSIN(EF)
C     SQ IS ETA SQUARED
C
      SQ=E/(1.-E)*OCOSEF**2
C
C     Q IS RADIUS OF CURVATURE IN PRIME VERTICAL
C
      Q=F(4)/DSQRT(1.-E*DSINEF**2)
C
C     CONVERT EF TO SECONDS OF ARC
C
C
      EFS=EF*RHO
      TE=F(1)/(Q*.9996)
      O(1)=EFS-T*RHO*(.5*(TE**2)*(1+SQ)-(1./24)*(TE**4)*
     *    (5.+3*T**2+6*SQ-6*SQT**2-3*SQ**2-9*(SQ**2)*T**2)+
     *    (1./720)*(TE**6)*(61.+90*T**2+45.*T**4+107*SQ-
     *    162.*SQ*T**2-45*SQ*T*
     *    *2*T**2))
C
C     CALCULATE DELTA LAMBDA
C     O(2)=(RHO/DCOSEF)*)(TE-(.16666666)*(TE**3)*(1+2.*(T**2)+SQ)
     *    +(83333333E-2)*(TE**5)*(5+28*T**2+24.*T**4+6.*SQ+B.*SQ*
     *    T**2))
      IF(IFLG-1)150,160,150
```

```
  150   O(2)=CM2-O(2)
        GO TO 170
  160       O(2)=CM2+O(2)
C
C       CONVERT OUTPUT TO RADIANS
C
  170   O(1)=O(1)/RHO
  180   O(2)=O(2)/RHO
        IF(F(6))200.210.200
  200   O(1)=-O(1)
  210   CONST=57.2857795131
        OO(1)=O(1)*CDNST
        OO(2)=O(2)*CDNST
        RETURN
        END
```

Subroutine GAUSS

```
      IMPLICIT REAL*8(A-H),(O-Z)
      DIMENSION X(MN1.1)
      X1=1
      LI=1
      E=0
      D=1
      IF(ID.EQ.0)D=0
      NN=N+NB
      IF(MS.EQ.0)GO TO 21
      K=N+1
      DO 22 I=1.N
      DO 23 J=K,NN
23    X(I,J)=0
      J=I+3
22    X(I,J)=1
21    NM1=N-1
      DO 1 I=1.N
      K=I+1
      IF(K GT N)GO TO 31
      IJJ=I
      Y=X(I,I)
      DO 2 II=K,N
      IF(DABS(Y) GE DABS(X(II.I)))GO TO 2
      IJJ=II
      Y=X(II.I)
2     CONTINUE
      IF(IJJ.EQ I) GO TO 31
      D=-D
      DD 3 J=I.NN
      Y=X(I,J)
      X(I,J)=X(IJJ.J)
      X(IJJ.J)=Y
3     CONTINUE
31    D=D*X(I.I)
      IF(X(I.I)EQ.0.)GO TO 10
      GO TO(5.13)L1
13    IF(DABS((X1-X(I,I)/X1)-1.)LT.1.E-7)E=2
      X1=X(,I)
5     L1=2
      DO 4 J=K,NN
4     X(I,J)=X(I,J)/X(I,I)
      IF(I.EQ.N)GO TO 6
      DO 1 II=K,N
      Y=X(II,I)
      DO 1 J=K,NN
1     X(II,J)=X(II,J)-X(I,J)*Y
```

```
6       DO 8 L=K,NN
        M=NM1
        DO 7 I=1.NM1
        M1=M+1
        K1=N
        DO 9 J=M1,N
        X(M,L)=X(M,L)-X(M,K1)*X(K1.L)
9       K1=K-1
7       M=M-1
8       CONTINUE
        IR=K-1
        DO 12 I=1,N
        L=K
        DO 11 J=1,NB
        X(I,J)=X(I,L)
11      L=L+1
12      CONTINUE
        RETURN
10      E=1
        IR=K-1
        RETURN
        END
```

Subroutine VERIFY

```
      IMPLICIT REAL*8(A-H),(O-Z)
      PARAMETER LA=50,LB=50,LAB=300
      COMMON/BUFFRS/BYTES,INTGS(512),REALS(2)
      COMMON/RCBLEV/LENCOD,JMPMAX,RCB
      COMMON/IOPRMS/FUNC,LUN,EVTFLG,IOSB,PRMLST,DSW
          COMMON ATIME,BTIME,CTIME
        COMMON OLRX,OLRY,PERX,PERY
      COMMON/VARBLS/IVRBLS(100),RVRBLS(75)
      COMMON/HEADER/LAT,LON,LATIN,LONIN,NM,MM,NFF,NPF,
     * JCK,IREG,IO,IRIGHT,JO,ITOP
      COMMON/UTM/EAST,ORTH,DEAST,DNORTH,AXIS(2),IHEMSP,ZONE,
     * P(3)
      COMMON INTGR,ITSFLG,EPX,EPY
      DIMENSION ACCTR(100),IRSDX(50),IRSDY(50)
      REALXYCS(300)
      REAL X,Y
      BYTE BYTES(1024),BMSG(80)
      INTEGER IXT(50),IYT(50),IXF(50),IYF(50)
      INTEGER WORD(3)
      INTEGER IXTO(100),IYTO(100),IXGO(100),IYGO(100)
      BYTE RCT(179),DLMARY(2)
      INTEGER FUNC,LUN,EVTFLG,IOSB(2),PRMLST(6),DSW(2)
      DIMENSION R(3),S(3)
      EQUIVALENCE (WORD,RCB),(BMSG,RCB(17)),(LNMSG,RCB(9))
      EQUIVALENCE(JUMP,RCB(7)),(RCB(177),DLMARY)
      DATA WORD/5,0,2/,JMPMAX/2/
      DATA DLMARY/1H,1H,/
C
C     INITIALIZE PARAMETERS FOR RANGE OF TRUE POINT(X AND Y)
C
C
      XA=999999
          ATIME=0.0
      YA=999999
      YI=-XA
          OLRX=-99999999.
      TYPE 432,BTIME,CTIME
  432 FORMAT('WERE YOU HERE',2E16.8)
          OLRY=-9999999999.
          PERX=-9999999999.
          PERY=-9999999999.
      X1=-XA
      CALL NEWPAG
C
C
C     QUERY USER FOR HISTOGRAMS OR ANALOG PLOTS
C
```

```
      TYPE 10
10    FORMAT(1X,40('*'),/1X,'VERIFICATION SELECTION',/IX,40('*'))
      GO TO 24
20    CALL STACK(RCB,LEVELS,2)
      IF(JUMP.NE.-1)GO TO 50
24    TYPE 40
40    FORMAT(1X,'1=RIGOROUS AND APPROXIMATE TRANSFORMATION
       PLOTS',/1X,
     * '2=RESIDUAL HISTOGRAMS',/1X,'3=BOTH'/1X,
     * 'ENTER 1,2,3,OR BLANK FOR END>'S)
      CALL GR(RCB,LEVELS,2)
      IF(BMSG(1).EQ.'')RETURN
50    IF(BMSG(1).GE.'1'.AND.BMSG(1).LE.'3')GO TO 60
      TYPE 55
55    FORMAT(1X,'INVALID ENTRY,TRY AGAIN.')
      GO TO 24
60    DECODE(LNMSG,65,BMSG,ERR=24)IOPTN
65    FORMAT(I< LNMSG >)
C
C     OPEN TEMPORARY FILES FOR STORING TRUE POINT,
C     X RESIDUAL, AND Y RESIDUAL
C
      OPEN(UNIT=11,NAME='DP:RESIDY.567',TYPE='SCRATCH',
     * FORM='UNFORMATTED',ACCESS='SEQUENTIAL',DISPOSE='DELETE')
      OPEN(UNIT=10,NAME='DP:RESIDX.789',TYPE='SCRATCH'
     * FORM='UNFORMATTED',ACCESS='SEQUENTIAL',DISPOSE='DELETE')
      OPEN(UNIT=9,NAME='DP"TRUEPT.123',TYPE='SCRATCH'
     * FORM='UNFORMATTED',ACCESS='SEQUENTIAL',DISPOSE='DELETE')
      OPEN(UNIT=4,NAME='DP:TRDATA.666',TYPE='OLD',ACCESS='DIRECT')
C
C     INITIALIZE PARAMETERS FOR HISTOGRAMS AND ANALOG PLOTS
C
      SUMX=0.0
      IRUN=0
      XMEAN=0.0
      KLO=0
      YMEAN=0.0
      NPTS=0
      SUMY=0.0
C
C     PROCESS ALL TARGET PROFILES
C
      DO 200 J=1,LA
      FIND(4'J)
      READ(4'J) (XYCS(KICK),KICK=1,LAB)
          QTIME=SECNDS(0.0)
      XTS=EAST+(FLOAT(J-1)*DEAST)
```

```
C
C       FIND THE EXACT OR TRUE POINT IN INPUT SYSTEM BY MEANS OF
         GENCORE
C       SUBROUTINE OF ALL FIFTY
C       POINTS OF PROFILE LINE.
C
        DO 150 JJ=1.LB
        IF(ITSFLG.EQ.2)GO TO 120
        R(1)=XTS
        R(2)=ORTH+(FLOAT(JJ-1)*DNORTH)
        R(3)=IHEMSP
        KFLAG=1
        HO=0.
        GO TO 130
  120   CONTINUE
        R(1)=ORTH+(FLOAT(JJ-1)*DNORTH)
        R(2)=XTS
        HO=0.
        KFLAG=2
  130   CALL TRANCD(R,S,AXIS,P,HO,IHEMSP,ZONE,KFLAG)
        ATIME=ATIME+SECNDS(QTIME)
        ACCTR(((JJ-1)*2)+1)=(S(2))
        ACCTR(((JJ-1)*2)+2)=(S(1))
        IF(ITSFLG.EQ.1)GO TO 140
        ACCTR(((JJ-1)*2)+2)=(S(1))
        ACCTR(((JJ-1)*2)+1)=(S(2))
  140   CONTINUE
  146   CONTINUE
C
C       FIND THE MIN AND MAX OF THE TRUE POINT COORDINATE FOR THE
         ANALG PLOTS.
C
        IF(ACCTR(((JJ-1)*2)+1).LT.XA)XA=ACCTR(((JJ-1)*2)+1)
        IF(ACCTR(((JJ-1)*2)+2).LT,YA)YA=ACCTR(((JJ-1)*2)+2)
        IF(ACCTR(((JJ-1)*2)+1).GT.X1)X1=ACCTR(((JJ-1)*2)+1)
        IF(ACCTR(((JJ-1)*2)+2).GT.Y1)Y1=ACCTR(((JJ-1)*2)+2)
  150   CONTINUE
  159   CONTINUE
        IKK=0
C
C       CALCULATE THE RESIDUALS OF X AND Y
C       EXAMPLE:RESIDUAL(X)=VALUE OF X TRUE-VALUE OF X APPROXIMATE
C       THE SAME WAS DONE FOR Y RESIDUAL;IT IS JUST THE DIFFERENCE
C        BETWEEN THE TRUE POINT COORDINATES AND THE APPROXIMATE
C        COORDINATES.
C
```

```
      DO 160 KK=1,LB
      XAP=((XYCS(((KK-1)*6)+1)))
      YAP=((XYCS(((KK-1)*6)+2)))
      IF(XAP.EQ.0.AND.YAP.EQ.0)GO TO 160
      IKK=(IKK+1)
      Q1=(ACCTR(((KK-1)*2)+1))
      Q2=(ACCTR(((KK-1)*2)+2))
      THOUS=100
      WE1=(Q1-XAP)
      WE2=(Q2-YAP)
      IF((XAP-Q1),GT.OLRX)OLRX=(XAP-Q1)
      IF((YAP-Q2).GT.OLRY)OLRY=(YAP-Q2)
       IF(((XAP-Q1)/Q1)*100.GT.PERX)PERX=((XAP-Q1)/Q1)*100.
        IF(((YAP-Q2)/Q2)*100..GT.PERY)PERY=((YAP-Q2)/Q2)*100.
      IRSDX(IKK)=JIFIX(SNGL(WE1))
      ISRDY(IKK)=JIFIX(SNGL(WE2))
  160 CONTINUE
  170 CONTINUE
C
C     WRITE TEMPORARY FILES TO DISK
C
      WRITE(9)(ACCTR(JIN),JIN+L,LB*2)
      IF(IKK.EQ.0)GO TO 200
      WRITE(10)(IRSDX(IDF),IDF=1.LB)
      WRITE(11)(IRSDY(IMX),IMX=1.LB)
      IRUN=IRUN+1
  200 CONTINUE
C
C     FIX SEQUENTIAL FILES BY END FILE AND REWIND
C     FOR FUTURE USE
C
      ENDFILE 9
      ENDFILE 10
      ENDFILE 11
      REWIND 9
      REWIND 10
      REWIND 11
C
C     CALCULATE THE MEANS OF BOTH RESIDUALS
C
      DO 300 KUTY=1,IRUN
      READ(10,ERR=9999)(IRSDX(JG),JG=1,50)
      READ(11,ERR=9999)(IRSDY(JIG),JIG=1,50)
      DO 275 JOPPA=1,50
      KLO=KLO+1
      SUMX=SUMX+(FLOAT(IRSDX(JOPPA)))
      SUMY=SUMY+(FLOAT(IRSDY(JOPPA)))
  275 CONTINUE
```

```
  300   CONTINUE
        NPTS=KLO
            XMEAN=SUMX/FLOAT(NPTS)
            YMEAN=SUMY/FLOAT(NPTS)
C
C       CALCULATE THE STANDARD DEVIATIONS OF BOTH RESIDUALS
C
        STDX=0.0
        STDY=0.0
        REWIND 10
        REWIND 11
        DO 450 IU=1,IRUN
        READ(10,ERR=9999)(IRSDX(JY),JY=1,50)
        READ(11,ERR=9999)(IRSDY(JI),JI=1,50)
        DO 400 LOP=1,50
        STDX=STDX+((((FLOAT(IRSDX(LOP))))-XMEAN)**2)
        STDY=STDY+((((FLOAT(IRSDY(LOP))))-YMEAN)**2)
  400   CONTINUE
  450   CONTINUE
        STDX=STDX/FLOAT(NPTS)
        STDY=STDY/FLOAT(NPTS)
        STDX=SQRT(STDX)
        STDY=SQRT(STDY)
C
C
        IF(IOPTN.NE.2)GO TO 1000
        RETURN
C**********************************************
C
C       HISTOGRAMS OF X AND Y RESIDUALS
C       STDDEV IS THE SUBROUTINE THAT CREATES THE HISTOGRAM ON THE
         CRT SCREEN.
C
C
C**********************************************
C
C**********************************************
C       ANALOG PLOTTING OF TRUE AND
C       APPROXIMATE POINTS OF TARGET
C       SYSTEM.
C**********************************************
 1000   CONTINUE
C
C       FIX BORDERS OF CRT SCREEN AND DETERMINE RANGE OF X AND Y
         SCALES
C
        END
```

APPENDIX C

GRAPHS

This appendix contains graphs of the 50 profile lines showing the deviation in accuracy of two values (Figures C.1 through C.6). The approximate point accuracy is seen as its point lies next to the true point. These computer charts illustrate calculated error and therefore, the lines do not necessarily coincide with the grid.

Figure C.1. Profile lines showing true point versus approximate point for Case 1, where increment is 100 meters.

Note: Dashed line is the true point; solid line is the approximate point.

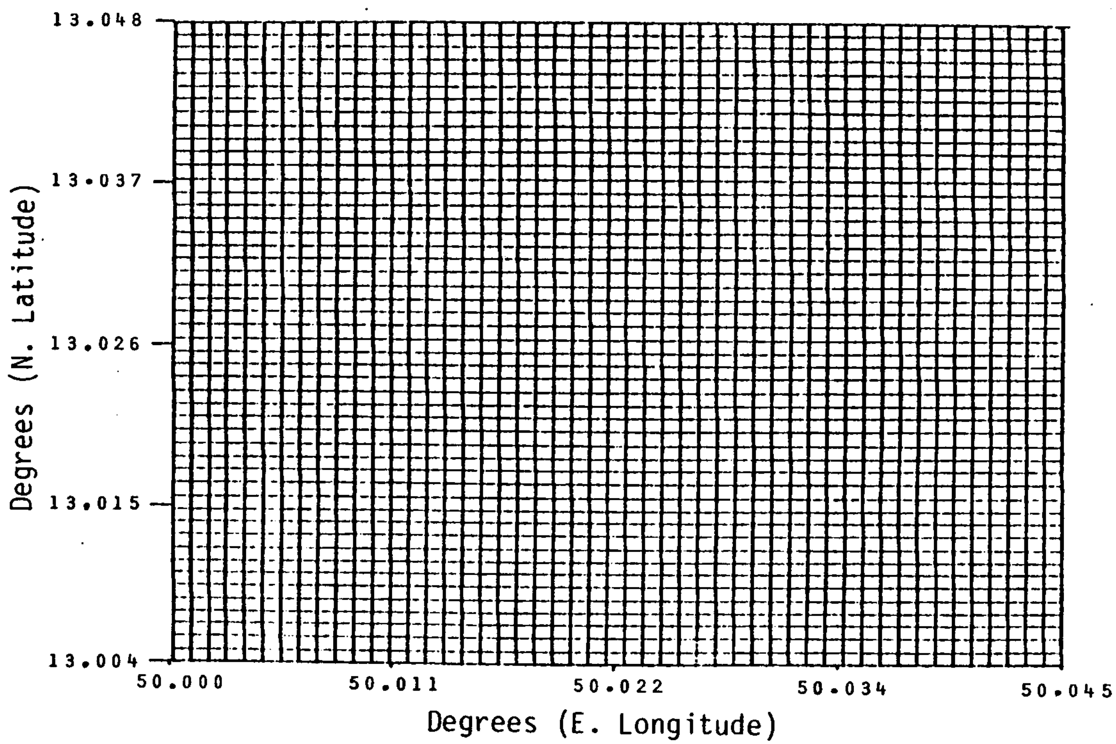

13.048
13.037
13.026
13.015
13.004
Degrees (N. Latitude)
50.000
50.011
50.022
50.034
50.045
Degrees (E. Longitude)

Figure C.1.

Figure C.2. Profile lines showing true point versus approximate point for Case 2, where increment is 1,000 meters.

Note: Dashed line is the true point; solid line is the approximate point.

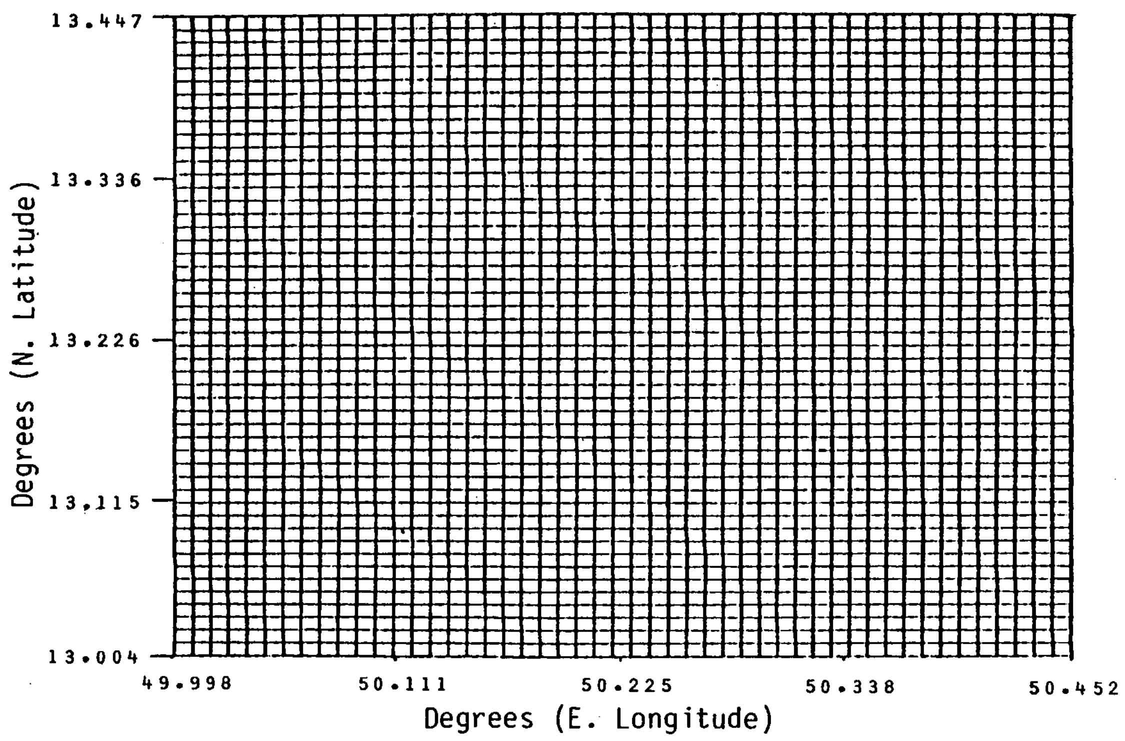

13.447
13.336
13.226
13.115
13.004
Degrees (N. Latitude)
49.998
50.111
50.225
50.338
50.452
Degrees (E. Longitude)

Figure C.2.

Figure C.3. Profile lines showing true point versus approximate point for Case 3, where increment is 10,000 meters.

Note: Dashed line is the true point; solid line is the approximate point.

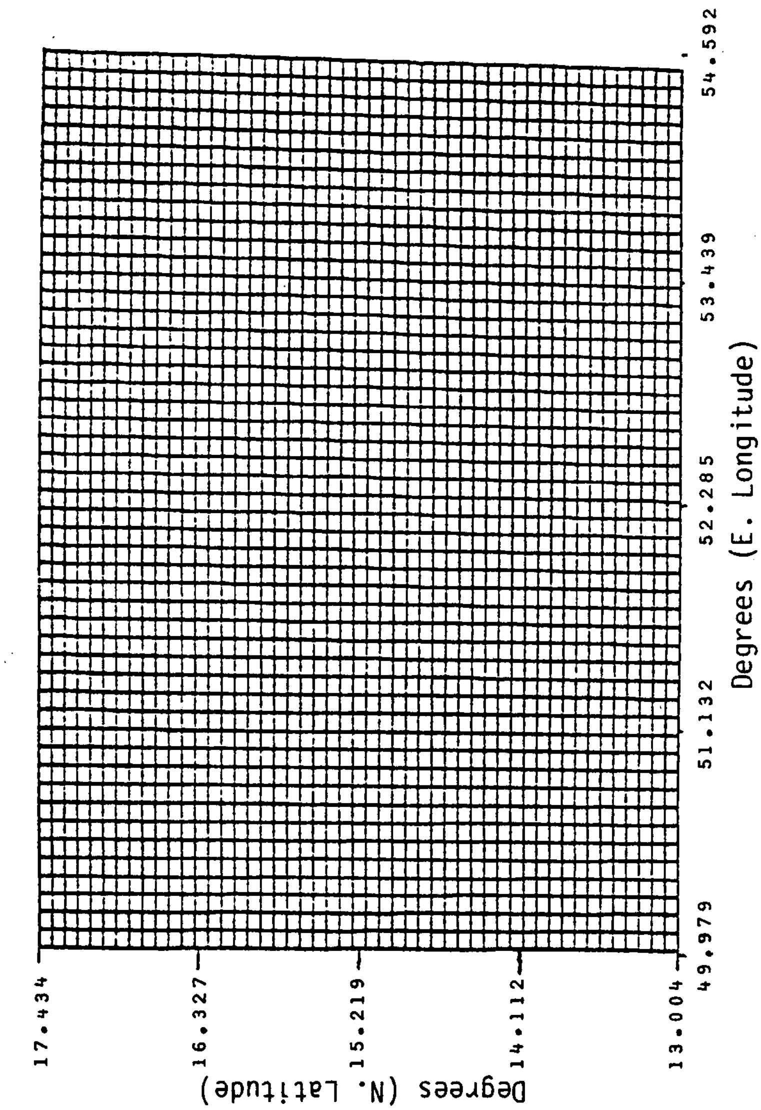

17.434
16.327
15.219
14.112
13.004
Degrees (N. Latitude)
49.979
51.132
52.285
53.439
54.592
Degrees (E. Longitude)

Figure C.3.

Figure C.4. Profile lines showing true point versus approximate point for Case 4, where increment is 37,000 meters.

Note: Dashed line is the true point; solid line is the approximate point.

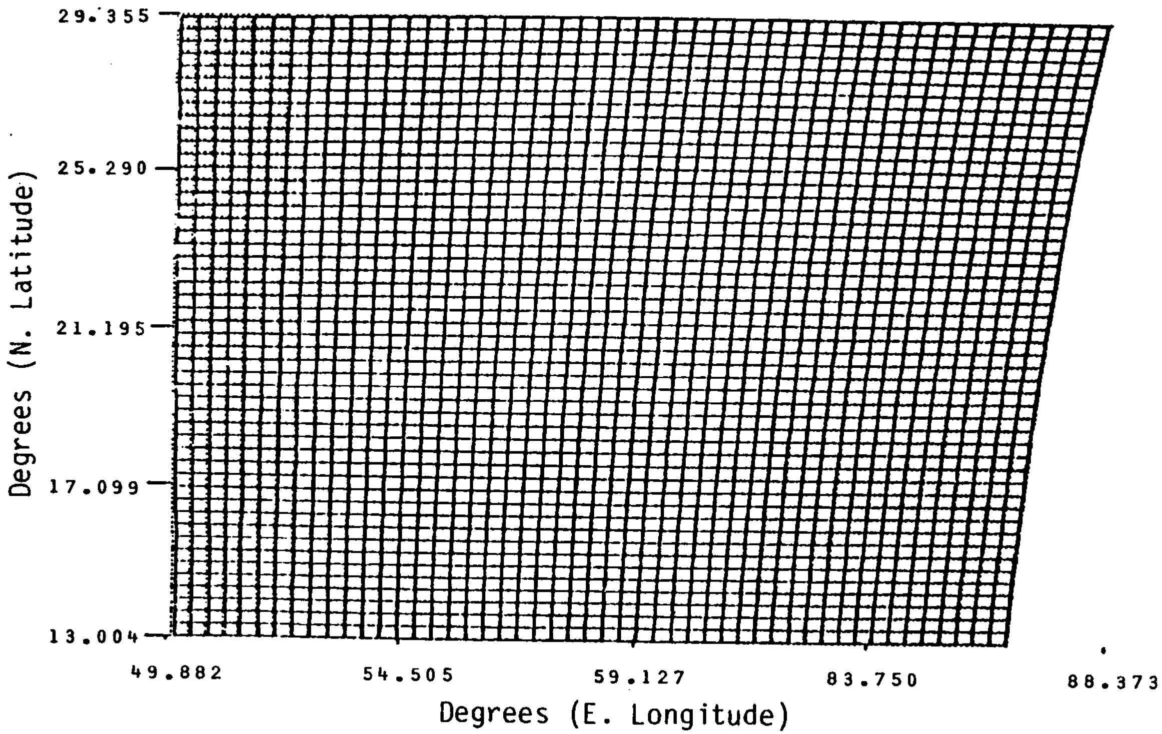

29.355
25.290
21.195
17.099
13.004
Degrees (N. Latitude)
49.882
54.505
59.127
83.750
88.373
Degrees (E. Longitude)

Figure C.4.

Figure C.5. Profile lines showing true point versus approximate point for Case 5, where increment is 63,000 meters.

Note: Dashed line is the true point; solid line is the approximate point.

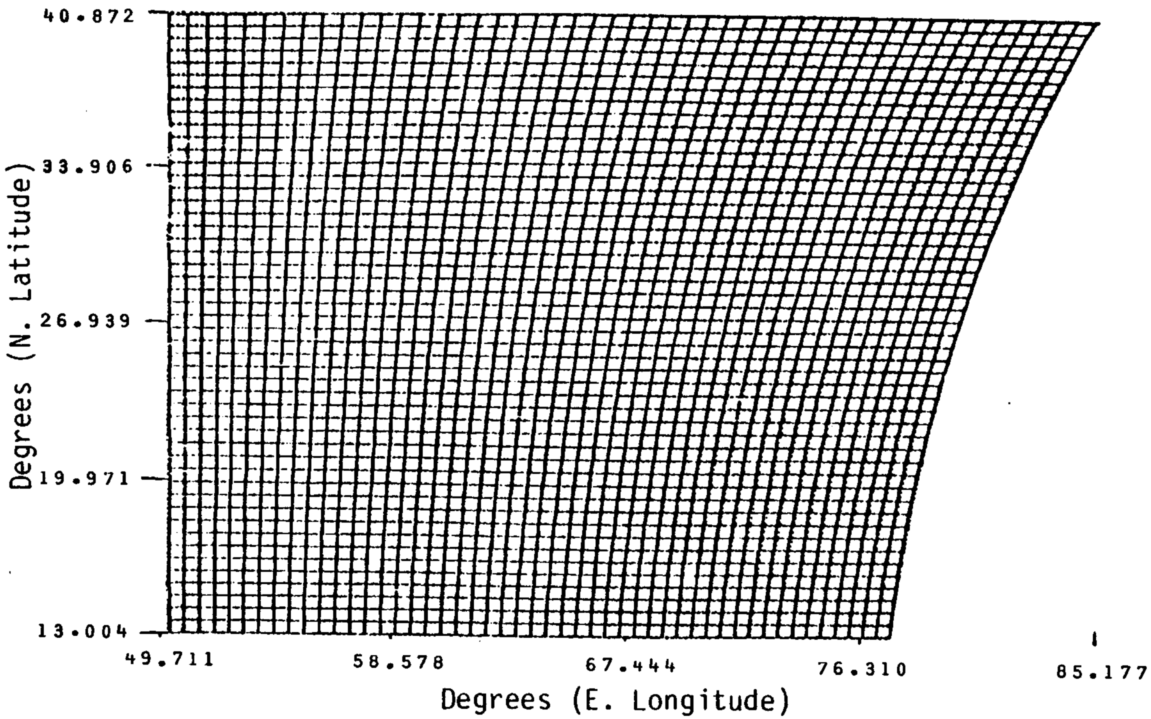

40.872
33.906
26.939
19.971
13.004
Degrees (N. Latitude)
49.711
58.578
67.444
76.310
85.177
Degrees (E. Longitude)

Figure C.5.

Figure C.6. Profile lines showing true point versus approximate point for Case 6, where increment is 150,000 meters.

Note: Dashed line is the true point; solid line is the approximate point.

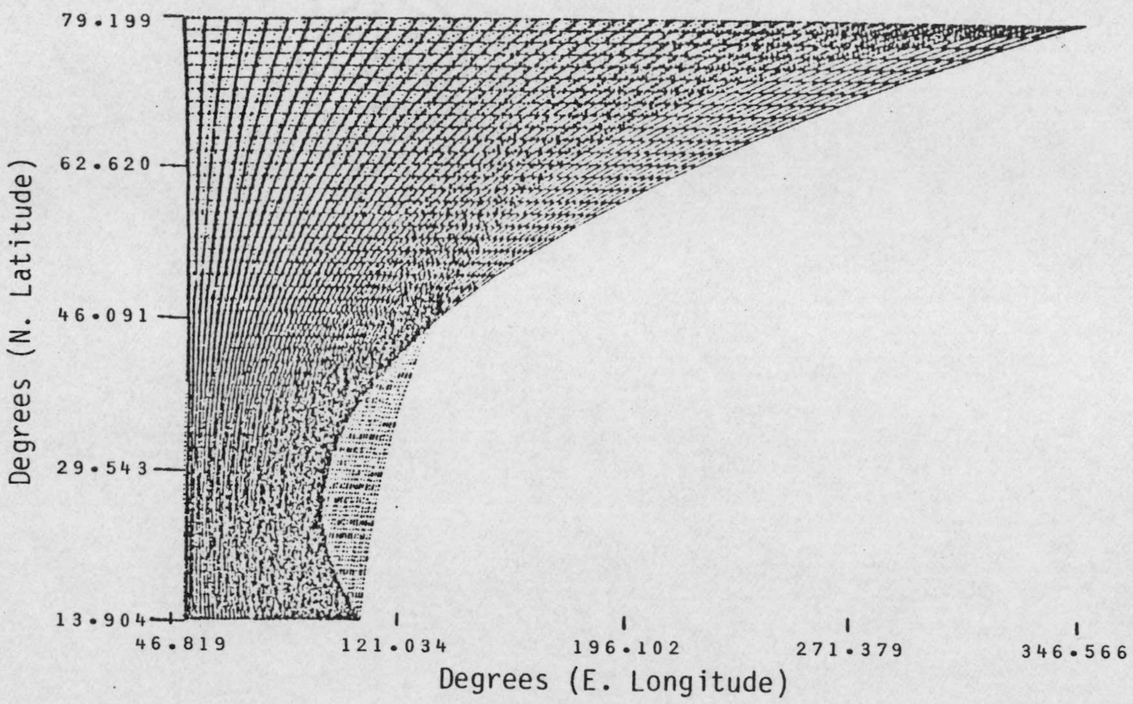

79.199
62.620
46.091
29.543
13.904
46.819
121.034
196.102
271.379
346.566
Degrees (N. Latitude)
Degrees (E. Longitude)

Figure C.6.

VITA

Joseph J. Shaw, Jr., was born in Columbia, Tennessee, on December 11, 1948. He attended elementary schools in Maury County and graduated from Central High School in June 1967.

After a four-year tour of duty with the United States Air Force, he entered Columbia State Community College in June 1971, receiving an Associate of Science degree in June 1973. In September of that same year, he entered Middle Tennessee State University, and in December 1974 he received a Bachelor of Science degree in Mathematics. In September 1975, he accepted a research assistantship at The University of Tennessee Space Institute and began study toward a Master's degree. In December 1981, he was awarded the Master of Science degree in Computer Science.

Mr. Shaw is employed at Intergraph, Inc., in Huntsville, Alabama. He is married to the former Vickie Lee Monks of Fayetteville, Tennessee.