

FG

10

ADA 026294

# EXTENDED STUDIES OF A QUADRILATERALIZED SPHERICAL CUBE EARTH DATA BASE

E.M. O'NEILL  
R.E. LAUBSCHER

## COMPUTER SCIENCES CORPORATION

SYSTEM SCIENCES DIVISION ✓

8728 Colesville Road  
Silver Spring, Maryland 20910

MAY 1976

Final Report

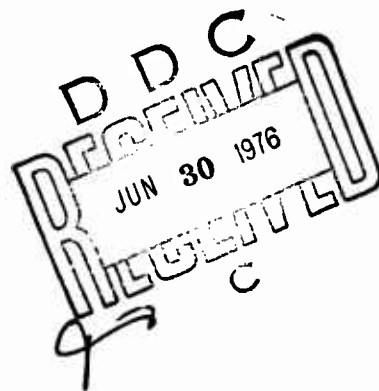
Prepared for

COMMANDER, NAVAL AIR SYSTEMS COMMAND

Department of the Navy  
Washington, D.C 20361

NAVAL ENVIRONMENTAL PREDICTION RESEARCH FACILITY

Naval Postgraduate School  
Monterey, California 93940



Copy available to DDC does not  
warrant fully legible reproduction

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (when data entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
(A8) NEPRF Technical Report 3-76 (CSC)	(19) TR-3-76	(CSC)
4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED	
(6) Extended Studies of a Quadrilateralized Spherical Cube Earth Data Base	(9) Final Report, Jul 1975 - Mar 1976	
6. AUTHOR (s)	7. PERFORMING ORG. REPORT NUMBER	
(10) E. M. O'Neill R. E. Laubscher	(14) CSC/TR-76/6008	
8. PERFORMING ORGANIZATION NAME & ADDRESS	9. CONTRACT OR GRANT NUMBER (s)	
Computer Sciences Corporation System Sciences Division 8728 Colesville Rd., Silver Spring, Md. 20910	(15) N00228-75-C-2329 NEW	
10. CONTROLLING OFFICE NAME & ADDRESS	11. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
Naval Environmental Prediction Research Facility Naval Postgraduate School Monterey, California 93940	PE 3511N, CC54, WU055:5-1	
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. REPORT DATE	
	(11) 27 May 1976	
	14. NUMBER OF PAGES	
	(12) 114 p.	
	15. SECURITY CLASS. (of this report)	
	Unclassified	
	16a. DECLASS/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this report)		
Approved for Public Release: Distribution Unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in block 20, if different from report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (continue on reverse side if necessary and identify by block number)		
Data Base, Map Projections, Constant Area Sensors, Meteorology, Satellite Data Processing		
20. ABSTRACT (continue on reverse side if necessary and identify by block number)		
This report describes the results of research into a number of aspects of a proposed global meteorological data base. Subjects covered include: improvement of formulation of map transformations; a rapid method of obtaining data base coordinates from satellite elements		



## ABSTRACT

This report presents the results of a study of mathematical formulae and algorithms necessary for the implementation of the Quadrilateralized Spherical Cube Earth Data Base preprocessor function. Topics covered include an improved formulation of the inverse equal-area transformation,  $f^*$  ; the development of a new, exact, equal-area transformation utilizing a polar coordinate system on the cube face; an improved computation of the coefficients of the equal-area transformation,  $f$  , and its inverse,  $f^*$  ; a rapid method of obtaining cube face coordinates from Keplerian elements and scanner angle; a new fast-filling algorithm; and the maintenance of multiple and variable resolution data bases.

This report addresses the properties and merits of the newly derived transformation and compares it with a previously derived form. Approximation methods for the equal-area transformation are presented. A revised, fast-filling algorithm is presented, which combines the new method of cube face coordinate computation with a recently suggested pixel sampling scheme. Applications of these methods and algorithms to an initial operational preprocessor and to refined future programs are discussed.

## TABLE OF CONTENTS

<u>Section 1 - Introduction</u> .....	1-1
<u>Section 2 - Background</u> .....	2-1
2.1 Prior Results .....	2-1
2.2 Previously Developed Transformation .....	2-1
2.3 Previously Developed Fast-Filling Algorithm .....	2-5
<u>Section 3 - New Projection and Transformation Methods</u> .....	3-1
3.1 Ground Trace Equations in Keplerian Elements .....	3-1
3.1.1 Subsatellite Point .....	3-1
3.1.2 Nonzero Scan Angle .....	3-3
3.2 Polar Formulation of the Area-Preserving Transfor- mation .....	3-6
3.3 Refinement of the Cartesian Transformation .....	3-18
3.3.1 Data Base Coordinates of the Subsatellite Point .....	3-19
3.3.2 Data Base Coordinates for Nonzero Scan Angle .....	3-20
3.4 Approximate Forms .....	3-21
3.5 Comparison of Transformations .....	3-27
<u>Section 4 - I/O System Design Considerations</u> .....	4-1
4.1 Alternative Method of Fast-Filling .....	4-1
4.2 Maintenance of Multiple- and Variable-Resolution Data Bases .....	4-4
<u>Section 5 - Geographic Orientation</u> .....	5-1
5.1 Introduction .....	5-1
5.2 Simple Orientations .....	5-1
5.3 The General Rotation Case .....	5-2
<u>Section 6 - Summary and Conclusions</u> .....	6-1
6.1 New Results .....	6-1
6.2 Recommendations for Further Study .....	6-1
<u>Appendix A - Ground Trace Equations</u>	
<u>Appendix B - Best Values for the Coefficients of <math>f^*</math></u>	

## TABLE OF CONTENTS (Cont'd)

Appendix C - Refinement of the Cartesian Transformation

Appendix D - Approximation of the Direct and Inverse Transformations

Appendix E - Listings of Test and Evaluation Programs

References

## LIST OF ILLUSTRATIONS

### Figure

1-1	Cubic Dissection of a Sphere by an Inscribed Cube . . . . .	1-2
1-2	World Map on Equal-Area Cube . . . . .	1-3
2-1	Projection From the Cartesian (a, b, c) System onto the Intermediate ( $\xi$ , $\eta$ ) System . . . . .	2-6
3-1	Keplerian Orbital Elements and Geographic Coordinates . . . .	3-2
3-2	The Scan Arc as a Central Angle . . . . .	3-4
3-3	Polar Coordinate System on Spherical Square . . . . .	3-8
3-4	Polar Coordinate System on an Inscribed Plane . . . . .	3-8
3-5	Right Spherical Triangle ABC . . . . .	3-11
3-6	Plane Right Triangle Corresponding to ABC . . . . .	3-11
3-7	Spherical Zone of One Base . . . . .	3-14
3-8	Relationship of $\phi_{\max}$ to $\theta$ . . . . .	3-16
3-9	Relationship of $\nu_{\max}$ to $\mu$ . . . . .	3-16
3-10	Grid Points Used for Interpolation . . . . .	3-23
3-11	Five-Point Solution . . . . .	3-25
3-12	Origins of Coefficient Set Domains Necessary To Define Entire Face . . . . .	3-26
3-13	Arrangement of Evaluation Points . . . . .	3-27
3-14	Distortions Arising From the Cartesian Form of the Transformation . . . . .	3-28
3-15	Distortions Arising From the Polar Form of the Transformation . . . . .	3-29
4-1	Coplanar Representation of Scanner and Data Base Grids . . .	4-2
4-2	Relationship of Scanner Input Buffer and Data Base Records . .	4-3
5-1	Orientation of the Axes of the Cube (Unrotated) . . . . .	5-2
5-2	Point $g_1$ , $g_2$ , $g_3$ in the General Reference Frame, $G_D$ , $G_2$ , $G_3$ . . . . .	5-4
5-3	Rotation of the Point ( $g_1$ , $g_2$ , $g_3$ ) About the $G_1$ Axis . . . .	5-4
5-4	Data Base Map of Earth With Cube Oriented 90 Degrees West of Greenwich . . . . .	5-6
5-5	Data Base Map of Earth With Cube Oriented 90 Degrees West of Greenwich and 30 Degrees South of the Equator . . .	5-7

## LIST OF TABLES

### Table

3-1	Rotations of $\theta$ and $u'$ for a Given $\theta^a$ . . . . .	3-18
3-2	Cube Face Identification . . . . .	3-19
3-3	Number of Coefficients Necessary for One-Pixel Accuracy as a Function of Position . . . . .	3-26
3-4	Execution Times for Transformation Forms on the IBM S/360-91 (One Ordered Pair) . . . . .	3-29



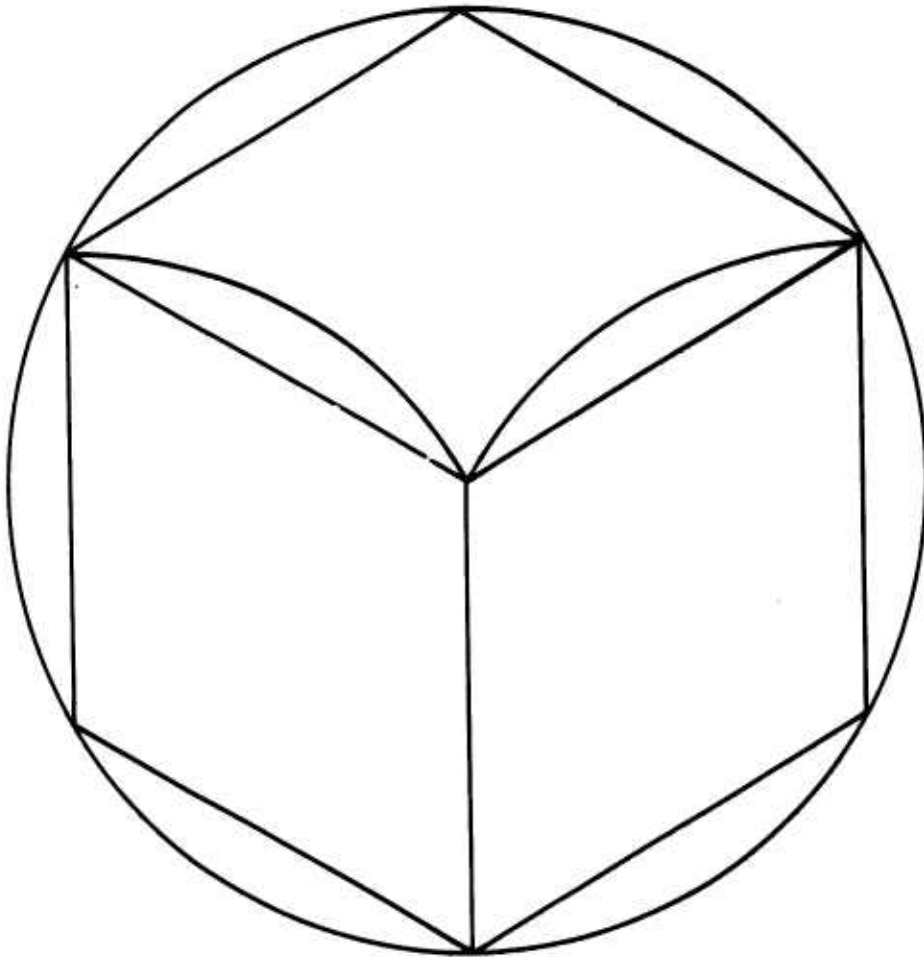
## SECTION 1 - INTRODUCTION

Previous studies (References 1 and 2) have indicated that a quadrilateralized spherical cube dissection of the Earth will provide a uniquely simple and useful set of coordinates in which to store remote sensor data on a global scale. This representation divides the globe into six square grids, allowing an extremely simple and elegant addressing scheme.

A mapping function (Reference 1) has been developed which permits an equivalent (equal area) mapping of a sphere onto the faces of a cube. This mapping provides an ideal match to constant resolution sensors and allows the simplification of map factor calculations for meteorological applications. Figure 1-1 illustrates the cubic dissection of a sphere, and Figure 1-2 shows a map of the world as projected using the equivalent mapping function.

In studying the feasibility of implementing this quadrilateralized spherical cube Earth data base system, a number of areas were identified which were deemed worthy of further study. In the course of these studies, additional details and problems were brought to light. The initial objectives of this study were

- Improvement of the formulation of the data base mapping functions
- Study of methods of representing satellite motions in terms of the data base coordinate system
- Study of data base Input/Output optimization on the CYBER/175 computer
- Production of maps illustrating various geographic orientations of the data base cube



**Figure 1-1. Cubic Dissection of a Sphere by an  
Inscribed Cube**

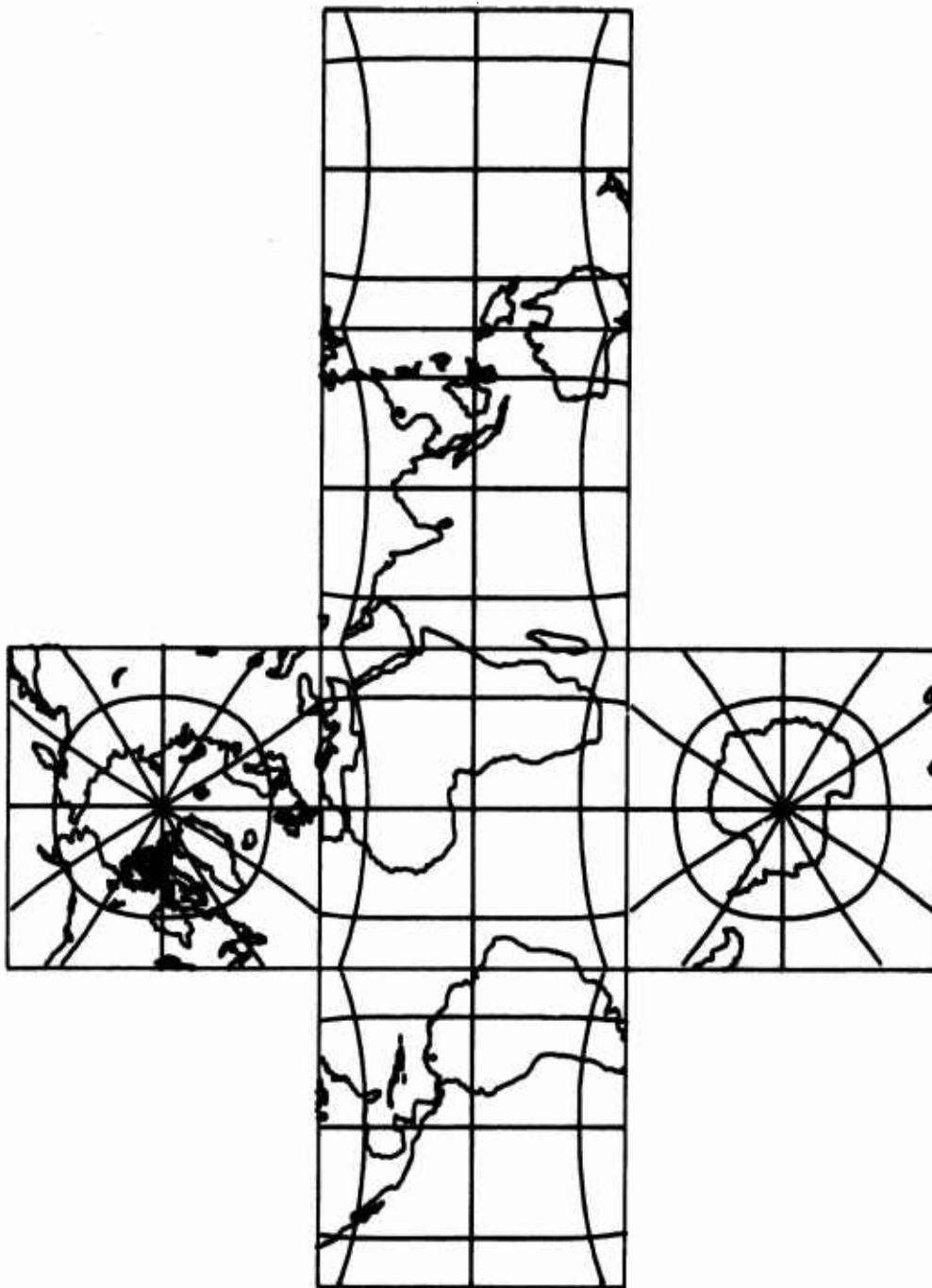


Figure 1-2. World Map on Equal-Area Cube

**Additional studies required were**

- **Study of algorithms required to implement a revised data base filling scheme**
- **Improvement in the accuracy of the data base mapping functions and the matching of direct and inverse functions**

**The general purpose in these studies has been twofold: (1) to optimize the existing data base scheme and (2) to assist in the timely development of a preliminary implementation by improving and clarifying previously developed formulae and algorithms. Most sections of this report deal with the maintenance of current data in the data base. The portion of the system which will fulfill these functions has been designated the preprocessor program.**

**This report presents several new and refined algorithms and mappings aimed at improving the speed and accuracy of an operational preprocessor program.**

## SECTION 2 - BACKGROUND

### 2.1 PRIOR RESULTS

A proposed set of algorithms for implementing the quadrilateralized spherical cube data base was presented in another report (Reference 1). That report addressed numerous problems: mapping points from the Earth's surface into the data base, determining serial addresses for data base locations, fast-filling and data retrieval, and determining the ground locations of scanner pixels. The results presented herein are derived from further study of these problems. For background information, Sections 2.2 and 2.3 present the previously derived transformation and method of fast-filling.

### 2.2 PREVIOUSLY DEVELOPED TRANSFORMATION

Section 2 and Appendix A of Reference 1 present the mathematical development of the mapping function,  $f$ , and its inverse,  $f^*$ . These functions relate the data base coordinates  $(x, y)$  to the rectangular coordinates  $(\xi, \eta)$  of the point on any cube face resulting from the central projection of a datum point on the sphere. For convenience, the mapping functions,  $f$  and  $f^*$ , are repeated here. The coordinates  $(\xi, \eta)$  are related to the data base coordinates  $(x, y)$  via

$$\begin{aligned}\xi &= f(x, y) \\ \eta &= f(y, x)\end{aligned}\tag{2-1}$$

where the function  $f$ , called the direct function, is given by

$$\begin{aligned}
 f(x, y) = & \gamma x + \frac{(1 - \gamma)}{r_o^2} x^3 \\
 & + xy^2 (r_o^2 - x^2) \left[ \delta + (r_o^2 - y^2) \sum_{i \geq 0} c_{ij} x^{2i} y^{2j} \right] \\
 & + x^3 (r_o^2 - x^2) \left[ \omega + (r_o^2 - x^2) \sum_{i \geq 0} d_i x^{2i} \right]
 \end{aligned} \tag{2-2}$$

where

$$\delta = \frac{1}{4r_o^4} \left[ -(\mu + 2\gamma) + \sqrt{(\mu^2 - 4\mu\gamma + 4\gamma^2 + 16\sqrt{2}\gamma^2)} \right] \tag{2-3a}$$

$$= 0.79048 \ 64491 \ 208$$

$$\omega = \frac{1}{2r_o^4} \left( 3 - 2\gamma - \mu - 2r_o^4 \delta \right) = -1.2254 \ 41487 \ 984 \tag{2-3b}$$

$$\gamma = \sqrt{\pi/6} \tag{2-3c}$$

$$\mu = \sqrt{\frac{\sqrt{3}\pi}{2}} \tag{2-3d}$$

$$r_o = 1/\sqrt{3} \tag{2-3e}$$

and the coefficients  $c_{ij}$  and  $d_i$  are

$$\begin{aligned} c_{00} &= -2.7217 \ 05366 \ 1814 & d_0 &= 1.4833 \ 12929 \ 4187 \\ c_{10} &= -5.5842 \ 16830 \ 5430 & d_1 &= 1.1199 \ 72606 \ 9742 \\ c_{01} &= 2.1711 \ 17480 \ 9423 & d_2 &= 6.0515 \ 38216 \ 1464 \\ c_{20} &= -3.4578 \ 62747 \ 3390 \\ c_{11} &= -6.4160 \ 15152 \ 6783 \\ c_{02} &= 1.9736 \ 26575 \ 8872 \end{aligned}$$

The inverse function,  $f^*$ , relates  $(x, y)$  to  $(\xi, \eta)$ ; that is,

$$\begin{aligned} x &= f^*(\xi, \eta) \\ y &= f^*(\eta, \xi) \end{aligned} \tag{2-4}$$

and is given by

$$\begin{aligned} f^*(\xi, \eta) &= \gamma^* \xi + \frac{(1 - \gamma^*)}{r_o^2} \xi^3 \\ &+ \xi \eta^2 (r_o^2 - \xi^2) \left[ \delta^* + \delta_1^* (r_o^2 - \xi^2) \right. \\ &+ \left. (r_o^2 - \eta^2) \sum_{\substack{i \geq 0 \\ j \geq 0}} c_{ij}^* \xi^{2i} \eta^{2j} \right] \\ &+ \xi^3 (r_o^2 - \xi^2) \left[ \omega^* + (r_o^2 - \xi^2) \sum_{i \geq 0} d_i^* \xi^{2i} \right] \end{aligned} \tag{2-5}$$

where

$$\gamma^* = 1/\gamma \quad (2-6a)$$

$$\mu^* = 1/\mu \quad (2-6b)$$

$$\gamma_1^* = 1/(\gamma + r_o^4 \delta) \quad (2-6c)$$

$$\mu_1^* = 1/(\mu + 2r_o^4 \delta) \quad (2-6d)$$

$$\delta^* = (\mu_1^* - \mu^*)/2r_o^4 \quad (2-6e)$$

$$\omega^* = \left(1/2r_o^4\right)(3 - 2\gamma^* - \mu^* - 2r_o^4 \delta^*) \quad (2-6f)$$

$$\delta_1^* = \frac{1}{r_o^2} \left[ \frac{(\gamma_1^* - \gamma^*)}{r_o^4} - \delta^* \right] \quad (2-6g)$$

and the coefficients  $c_{ij}^*$  and  $d_i^*$  are

$$c_{00}^* = 3.973 \ 89249$$

$$d_0^* = 1.811 \ 28250$$

$$c_{10}^* = 6.591 \ 19476$$

$$d_1^* = 37.635 \ 47857$$

$$c_{01}^* = -25.368 \ 92536$$

$$d_2^* = 63.000 \ 23655$$

$$c_{20}^* = -73.064 \ 97000$$

$$c_{11}^* = 77.381 \ 61133$$

$$c_{02}^* = 21.685 \ 89623$$

The use of  $f^*$  in an algorithm for obtaining the data base coordinates  $(x, y)$  of any datum point on the Earth's surface is described in Section 3.3. The algorithm presented in Section 3.3 is the result of the present study to obtain a method of implementing  $f^*$  in a computationally efficient manner.



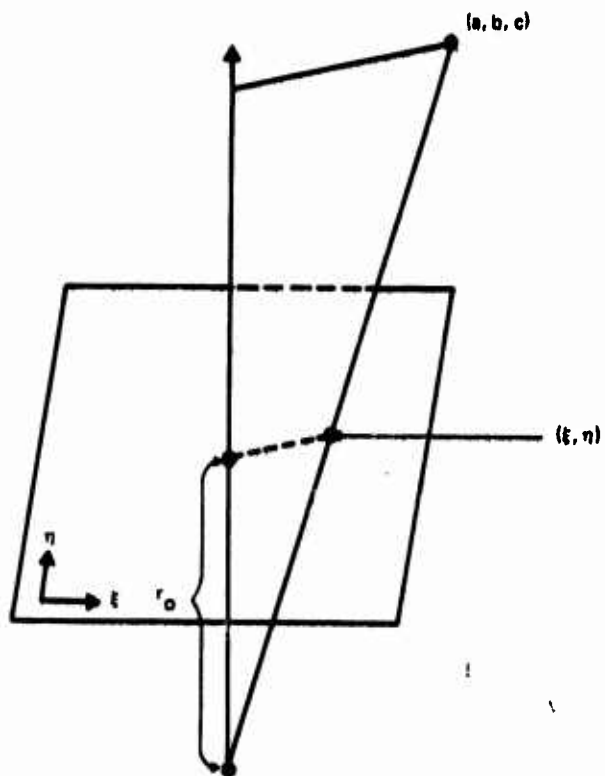
### 2.3 PREVIOUSLY DEVELOPED FAST-FILLING ALGORITHM

Sections 4 and 5 of Reference 1 present a "fast-filling" method for updating the data base. The method was developed to obtain, as directly as possible, the data base location corresponding to the location of a given scanner data element and to store the value associated with that element in the calculated data base location. Starting with a given satellite location and scanner angle, a series of five steps are necessary to calculate the corresponding data base location.

1. Determine the three-dimensional Cartesian coordinates of the point observed by the scanner.
2. Determine the face onto which the point is to be mapped
3. Project from the Cartesian (a, b, c) system into an intermediate system ( $\xi$ ,  $\eta$ ) as shown in Figure 2-1
4. Transform from ( $\xi$ ,  $\eta$ ) into two-dimensional Cartesian coordinates (x, y) on the face of the data base cube
5. Convert from face number and (x, y) to a serial location in the data base

This transformation need not be performed for every point, since both the projection onto the Earth's surface and the ( $\xi$ ,  $\eta$ ) to (x, y) transformation may be linearly approximated over a relatively long range. Thus, the complete computation is required only for widely spaced benchmark points, with intermediate points being determined by interpolation.

Input and output are handled, as described in Section 5 of Reference 1, by blocking both the input data and the data base into rectangular arrays. Methods for handling problems such as queuing data base I/O, splitting input blocks across face boundaries, and locating a vertex within an input block were also discussed in Reference 1.



**Figure 2-1. Projection From the Cartesian (a, b, c) System onto the Intermediate ( $\xi$ ,  $\eta$ ) System**

In Reference 3, this scheme was compared with an alternative scheme. The alternative scheme follows the above method through step 3; then, instead of storing from the input block directly into the data base, the following steps are performed:

- For each input data block, determine which data base blocks will be completely covered by the input data
- Determine, in the  $(\xi, \eta)$  system, a set of benchmark points corresponding to the corners of the data base blocks to be filled
- Use the above benchmarks to interpolate the  $(\xi, \eta)$  coordinates corresponding to each  $(x, y)$  data base location and store the closest point in the input data block into the corresponding serial location

This method offers the advantage of providing a completely filled data base block and eliminates data base read accesses during the update process. After a thorough examination of this method, CSC concluded that the advantages outweigh the slight additional computational cost incurred in finding the appropriate data base blocks. Consequently, some of the work of this report was oriented toward this approach.

### SECTION 3 - NEW PROJECTION AND TRANSFORMATION METHODS

#### 3.1 GROUND TRACE EQUATIONS IN KEPLERIAN ELEMENTS

This section presents general relations that yield the geocentric coordinates of the subsatellite point and of any point on the Earth's (spherical) surface lying on an arc scanned by an onboard satellite sensor. These geocentric coordinates are to be computed as an intermediate step in algorithms for computing data base coordinates (Sections 3.2 and 3.3).

The following subsections distinguish between two cases: (a) the computation of the coordinates of the subsatellite point, and (b) the computation of the coordinates of any point on the scan arc. Although the first case could be treated as a special case of the second, its development is separated in order to facilitate computer implementation. The following subsections summarize the results, which are developed in detail in Appendix A.

##### 3.1.1 Subsatellite Point

It is assumed that the satellite ephemeris is available in the form of Keplerian orbital elements. Expressions can be obtained in terms of these elements for the Earth-centered direction cosines ( $l$ ,  $m$ ,  $n$ ) of the subsatellite point in the equatorial  $x$ ,  $y$ ,  $z$  coordinate system shown in Figure 3-1.

Since the subsatellite point lies on the geocentric position vector of the satellite (see Figure 3-1),  $l$ ,  $m$ , and  $n$  will be identical with the corresponding direction cosines for the satellite's orbital position. The desired expressions for  $l$ ,  $m$ , and  $n$  are developed in Appendix A to be

$$\begin{aligned}l &= +\cos \Omega \cos (\omega + f) - \sin \Omega \sin (\omega + f) \cos i \\m &= +\sin \Omega \cos (\omega + f) + \cos \Omega \sin (\omega + f) \cos i \\n &= +\sin (\omega + f) \sin i\end{aligned}\tag{3-1}$$

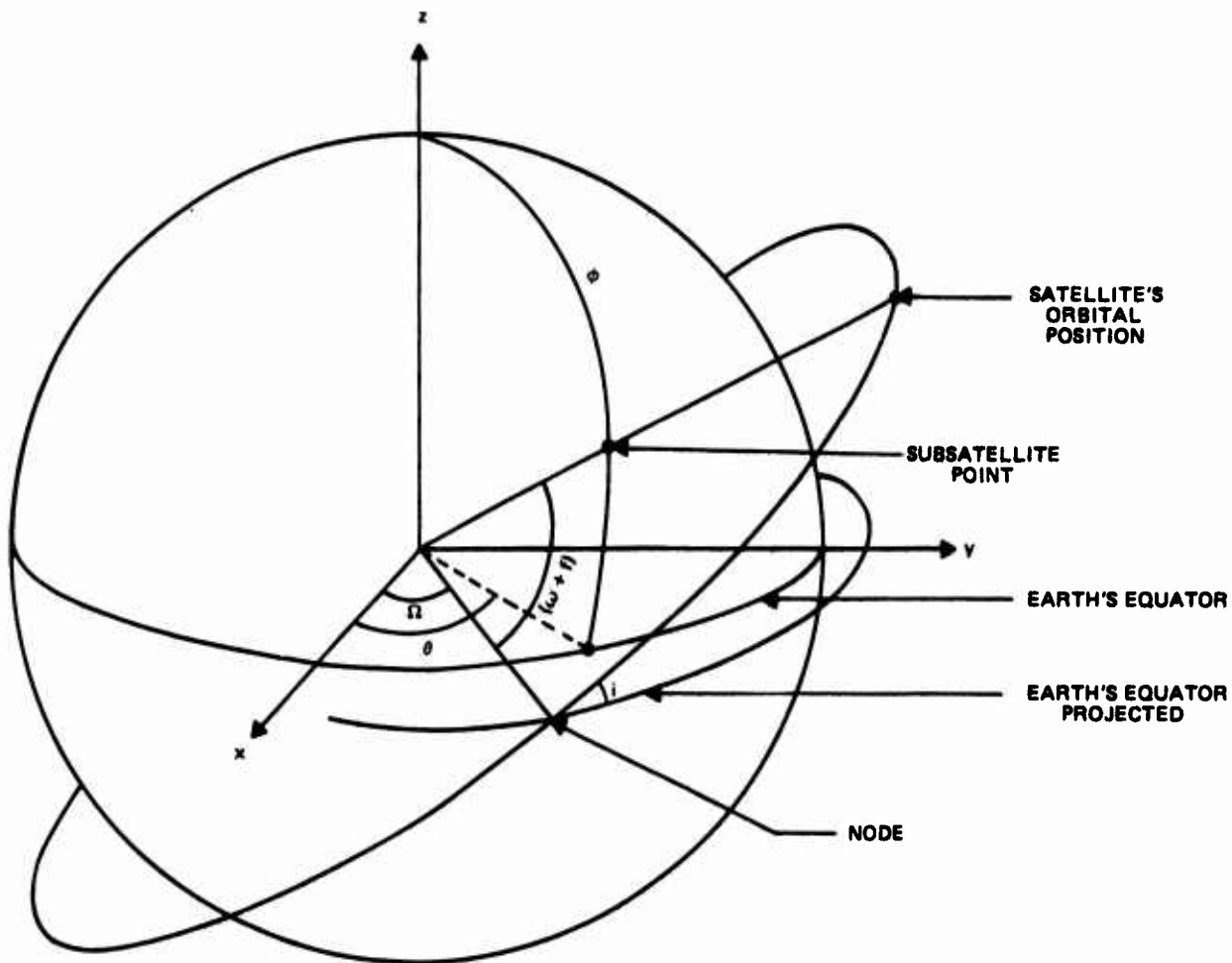


Figure 3-1. Keplerian Orbital Elements and Geographic Coordinates

The angular arguments are Keplerian elements and are referenced to the true equator and prime (Greenwich) meridian of date coordinate system. The direction cosines  $(l, m, n)$  in Equation (3-1) are obtained for the subsatellite point corresponding to any point in the orbit using the true anomaly,  $f$ , which reflects the motion of the satellite in its orbit, and the time-dependent node,  $\Omega$ , which reflects the rotation of the Earth. The time dependence of the node is given by

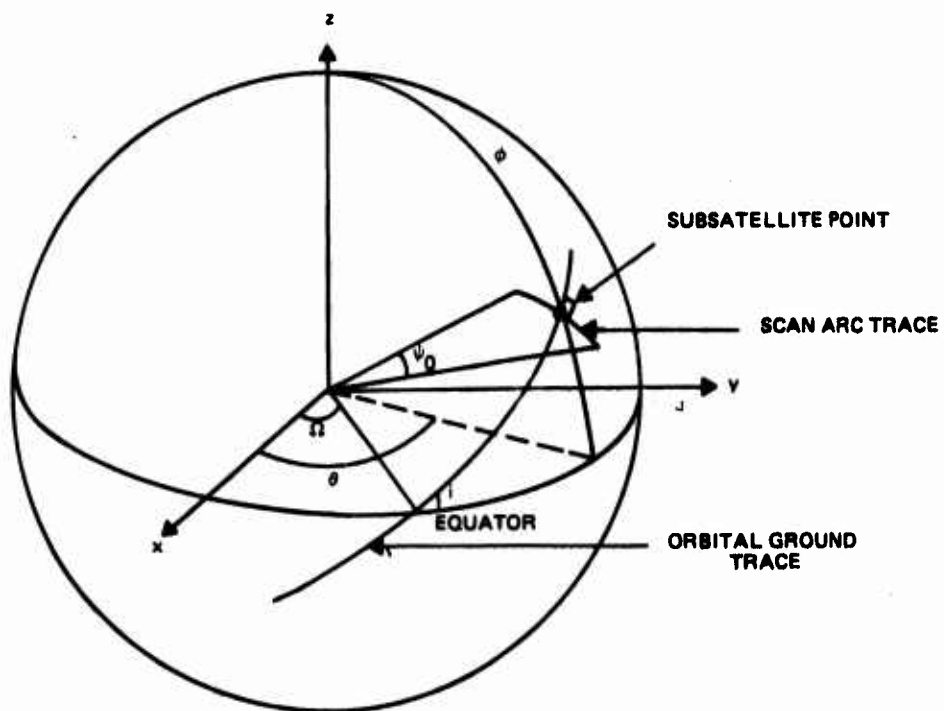
$$\Omega = \Omega_0 - s \quad (3-2)$$

where  $\Omega_0$  is the initial (constant) value of  $\Omega$  and  $s$  is the elapsed Greenwich sidereal time since the instant that  $\Omega = \Omega_0$ . For high precision,  $\Omega_0$ ,  $\omega$ ,  $i$ , and  $s$  should be corrected for the Earth's nutation, precession, and polar motion.

### 3.1.2 Nonzero Scan Angle

It is assumed that the satellite collects data through an onboard sensor which scans the surface of the Earth through an arc of fixed size in such a way that its sweep carries it through the subsatellite point in a direction at right angles to the direction of the satellite's orbital motion. The trace of the scan arc on the surface of the (spherical) Earth is a segment of a great circle subtending a fixed central angle,  $\psi_0$ , as illustrated in Figure 3-2. The introduction of the central angle,  $\psi_0$ , though generally applicable in defining the ground location of a scan element (datum point), is particularly suitable when scan elements are spaced equally along the great circle arc.

Expressions are presented below for obtaining the geocentric longitude,  $\theta'$ , and colatitude,  $\phi'$ , of any pair of points (datum points) on the scan arc which are located symmetrically with respect to the subsatellite point. Details of their development are given in Appendix A. These expressions are then



**Figure 3-2. The Scan Arc as a Central Angle**

employed in Sections 3.2 and 3.3 in describing the two alternative algorithms for obtaining the data base coordinates  $(x, y)$  of the datum points.

First, the sines and cosines of the geocentric longitude,  $\theta$ , and colatitude,  $\phi$ , of the subsatellite point are required in terms of the direction cosines available from Equation (3-1), i.e.,

$$\sin \phi \cos \theta = l \quad (3-3a)$$

$$\sin \phi \sin \theta = m \quad (3-3b)$$

$$\cos \phi = n \quad (3-3c)$$

Actually, Equation (3-3c) yields both  $\cos \phi$  and  $\sin \phi$ , since  $0 \leq \phi \leq 180$  degrees. Consequently,  $\sin \theta$  and  $\cos \theta$  can be determined from Equations (3-3a) and (3-3b).

Returning to the scan arc, it is convenient to view  $\psi/2$  as a general variable describing the positions of any pair of points along the arc. The following expressions yield two unique values for both  $\theta'$  and  $\phi'$ , depending on the algebraic sign of  $\psi/2$ . Thus,

$$\cos \phi' = \cos \phi \cos \psi/2 - \cos i \sin \psi/2 \quad (3-4)$$

where  $\cos \phi$  and  $\cos i$  are already available. Equation (3-4) gives  $\phi'$  uniquely, since  $0 \leq \phi' \leq 180$  degrees, from which  $\sin \phi'$  is then computed and used in

$$\sin (\theta' - \theta) = \frac{\cos (\omega + f) \sin i \sin \psi/2}{\sin \phi \sin \phi'} \quad (3-5)$$



$$\cos (\theta' - \theta) = \frac{\cos \psi/2 - \cos \phi \cos \phi'}{\sin \phi \sin \phi'} \quad (3-6)$$

to obtain  $\theta'$ , where all of the terms to the right of the equalities have previously been determined.

Caution is necessary when applying Equations (3-5) and (3-6) near the poles, since these relations are singular for subsatellite points or datum points near the poles. This is expected, since the geocentric longitudes  $\theta$  and  $\theta'$  are undefined at the poles. No problem need arise in computations, however, since the satellites are not expected to have exactly 90-degree inclinations and since, for the case of an equatorial orientation of the data base cube, it is a simple matter to regard the cube face coordinates of such datum points as zero within an arbitrary, but nonzero, distance from the pole.

It is emphasized that Equations (3-4), (3-5), and (3-6) yield two unique values for both  $\theta'$  and  $\phi'$ , depending on the choice of  $\pm\psi/2$ ; therefore, both halves of the scan arc are treated efficiently using the same auxiliary quantities ( $\sin i$ ,  $\sin \phi$ , etc.).

The expressions just presented will be used in Section 3.2 in an algorithm employing a newly developed transformation to obtain data base coordinates of datum points. These results are also used in Section 3.3 in an algorithm designed to employ the older transformation of Section 2.2 to compute the data base coordinates of datum points in a computationally efficient manner.

### 3.2 POLAR FORMULATION OF THE AREA-PRESERVING TRANSFORMATION

Section 2 of Reference 1 presents an area-preserving transformation from the spherical square to the inscribed plane square. That transformation utilizes a central projection onto the plane square and a transformation from the projected

coordinates  $(\xi, \eta)$  to the data base coordinates  $(x, y)$ . This transformation, designated by  $f^*$ , is given in Equation (2-4) as

$$x = f^*(\xi, \eta)$$

$$y = f^*(\eta, \xi)$$

and is based on the Jacobian determinant. It was noted in Reference 1 that the solutions obtained to differential equations based on this determinant are not fully constrained by requiring the interior and edges of the spherical square to map onto the plane square. Additionally, it was found that, despite adding further constraints, no analytic solution could be found for  $f^*$ . An approximate solution was arrived at using a numerical steepest descent method. Under the current contract, an alternative solution was studied; the results are presented below.

Two sets of polar coordinate systems are considered (Figures 3-3 and 3-4). Figure 3-3 shows a spherical square of radius  $R$ , produced by the central projection of an inscribed square upon the sphere. The polar coordinates  $(\theta, \phi)$  are defined by an axis of the sphere passing through the center of the spherical square and by a great circle arc passing through the axis and bisecting one edge of the square. The azimuthal angle,  $\theta$ , is measured from this "meridian" and the central angle,  $\phi$ , is measured from the axis. For the case where the coordinate axis and meridian coincide with those of the sphere,  $\theta$  equals the longitude and  $\phi$  is the colatitude.

Figure 3-4 shows a plane square of side  $2r_0$ . The axis in this system is normal to the plane, passes through the center of the plane, and has its origin at a distance  $r_0$  below the plane. Coordinates  $(\mu, \nu)$  are defined in the same sense as  $(\theta, \phi)$ .

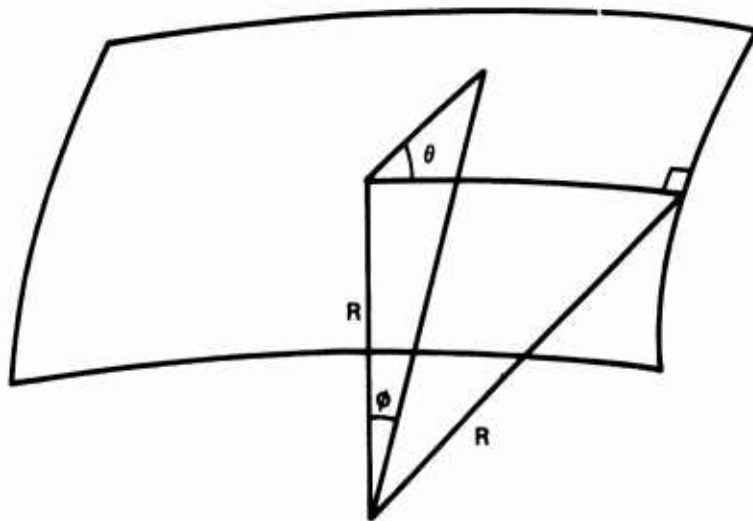


Figure 3-3. Polar Coordinate System on Spherical Square

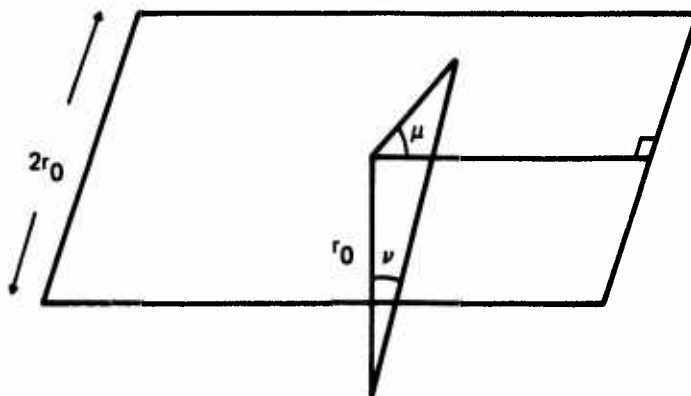


Figure 3-4. Polar Coordinate System on an Inscribed Plane

If the axes of the two systems are aligned and the plane square is inscribed within the spherical square, it can be shown that

$$r_o = R/\sqrt{3} \quad (3-7)$$

The area of the sphere is

$$A_s = 4\pi R^2 \quad (3-8)$$

and the surface area of the inscribed cube is

$$A_c = 24r_o^2 \quad (3-9)$$

The ratio of the areas of the spherical and plane squares is given by

$$\lambda = \frac{A_s}{A_p} = \frac{4\pi R^2}{24r_o^2} \quad (3-10)$$

Substituting Equation (3-7) into Equation (3-10) yields

$$\lambda = \pi/2 \quad (3-11)$$

To further constrain the mapping of the spherical square upon the plane, consider the condition  $\mu = f(\theta)$ , where  $\mu$  is independent of  $\phi$ . This condition implies the useful property that any great circle on the sphere passing through the center of the spherical square will map as a straight line passing through the center of the plane square. This function must be area preserving in the

sense that sectors of equal area on the sphere must map into equivalent areas on the plane. If the function  $f$  exists, then it is possible to relate the areas of differential elements  $d\theta$  and  $d\mu$  of the spherical and plane squares, as shown below.

To obtain the function  $\mu = f(\theta)$ , consider the right spherical triangle ABC in Figure 3-5, and let one side, B, be the meridian and A, the azimuth angle,  $\theta$ . For a unit sphere,  $R \equiv 1$  and  $b = \pi/4$ .

From the general relation for the area of a spherical triangle

$$S = R(A + B + C - \pi) \quad (3-12)$$

and the area of the right spherical triangle is

$$S_s = \theta + \beta - \pi/2 \quad (c \equiv \pi/2) \quad (3-13)$$

By Napiers' identity,  $B = \cos^{-1}(\sin A \cos b)$  and

$$S_s = \theta + \cos^{-1}(\sin \theta \cos \pi/4) - \pi/2 \quad (3-14)$$

This triangle may represent any segment of the spherical square in the range  $-\pi/4 \leq \theta \leq \pi/4$ .

The area of the corresponding right triangle on the plane (Figure 3-6) is

$$S_p = \frac{\ar_o}{2} \quad (3-15)$$

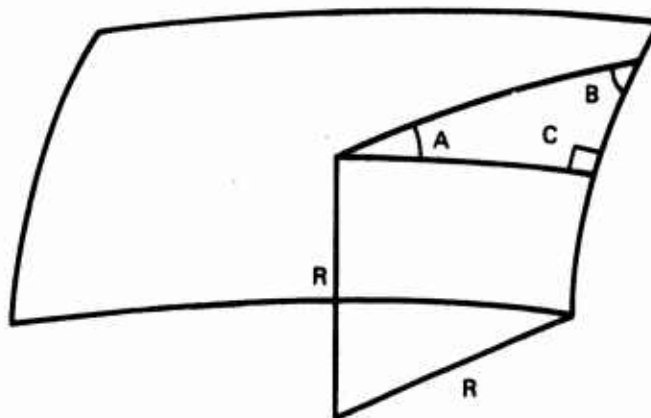


Figure 3-5. Right Spherical Triangle ABC

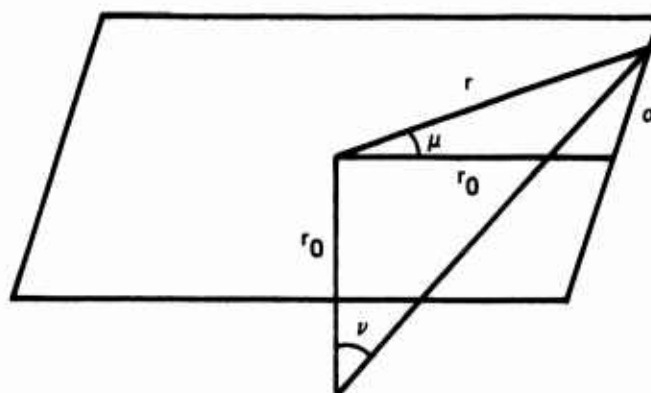


Figure 3-6. Plane Right Triangle Corresponding to ABC

or, since  $a = r_o \tan \mu$ ,

$$S_p = \frac{r_o^2}{2} \tan \mu \quad (3-16)$$

For  $\mu = f(\theta)$  to preserve area, the condition must hold that, for any  $\theta$ ,

$$S_s / S_p = \lambda \quad (3-17)$$

Thus,

$$S_s = \pi/2 S_p = \frac{r_o^2}{4} \tan \mu \quad (3-18)$$

and

$$\mu = \tan^{-1} \left( \frac{4}{\pi r_o^2} S_s \right) \quad (3-19)$$

In the case of a unit sphere,  $r_o = 1/\sqrt{3}$ , and

$$\mu = \tan^{-1} \left( \frac{12}{\pi} S_s \right) \quad (3-20)$$

$$\mu = \tan^{-1} \left\{ \frac{12}{\pi} \left[ \theta - \cos^{-1} \left( \sin \theta \cos \frac{\pi}{4} \right) - \frac{\pi}{2} \right] \right\} \quad (3-21)$$

It can be shown that  $\mu = 0$  at  $\theta = 0$  and  $\mu = \pm\pi/4$  at  $\theta = \pm\pi/4$ . The function  $f$  is defined over the quadrant  $-\pi/4 \leq (\theta, \mu) \leq \pi/4$  and, by reflection, over the entire spherical and plane squares.

To relate  $\nu$  and  $\phi$ , expressions for the differential area elements,  $\phi d\theta$  and  $\nu d\mu$ , must be obtained. Figure 3-7 illustrates a spherical zone of one base, having area

$$S_s = 2\pi Rh \quad (3-22)$$

Substituting  $h = R - R \cos \phi$  and taking  $R \equiv 1$  yields

$$S_s = 2\pi(1 - \cos \phi) \quad (3-23)$$

Since

$$S_s = \int_0^{2\pi} (1 - \cos \phi) d\theta \quad (3-24)$$

$$dS_s = (1 - \cos \phi) d\theta$$

The area of the corresponding sector on the plane is given by

$$dS_p = r^2 d\mu \quad (3-25)$$

In order to apply the area-preserving condition, Equation (3-17) is differentiated to yield

$$dS_s = \pi/2 dS_p \quad (3-26)$$

Substituting Equations (3-24) and (3-25) into Equation (3-26) gives

$$r^2 du = 2/\pi (1 - \cos \phi) d\theta \quad (3-27)$$



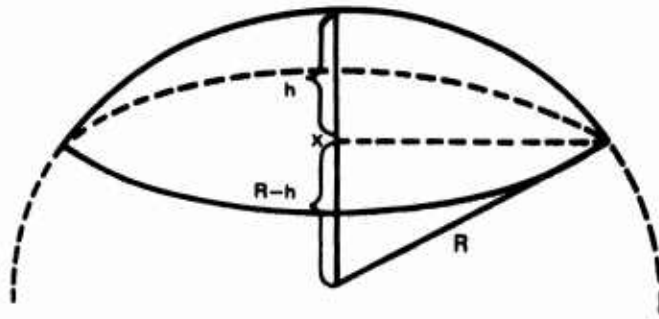


Figure 3-7. Spherical Zone of One Base

Since  $r = r_o \tan \nu$ , Equation (3-27) can be rewritten as

$$\tan^2 \nu = \frac{2}{\pi r_o^2} (1 - \cos \phi) \frac{d\theta}{du} \quad (3-28)$$

For fixed values of  $\theta$  and  $u$ , Equation (3-28) becomes

$$\tan^2 \nu = C(1 - \cos \phi) \quad (3-29)$$

where

$$C = \frac{2}{\pi r_o^2} \left. \frac{d\theta}{du} \right|_{\theta, u} \quad (3-30)$$

Clearly,  $C$  can be evaluated from boundary conditions:

1. When  $\phi = 0$ ,  $\nu = 0$  points at the center map onto themselves
2. When  $\phi = \phi_{\max}$   
 $\nu = \nu_{\max}$  points on the boundary of the spherical square must map onto points on the boundary of the plane square

To develop these boundary conditions, it is necessary to obtain  $\phi_{\max}$  and  $\nu_{\max}$  as functions of  $\theta$  and  $\mu$ , respectively. Figures 3-8 and 3-9 illustrate these angles. It should be noted that to obtain the boundary of the spherical square, a central projection to the inscribed plane is employed. This projection preserves both  $\theta$  and  $\phi$  and reduces the problem to the case of the plane square.

The loci of points describing these edges can be expressed as

$$r_s = r_o \sec \theta \quad (3-31)$$

and

$$r_p = r_o \sec u \quad (3-32)$$

and the central angles can be expressed as

$$\phi_{\max} = \tan^{-1} \sec \theta \quad (3-33)$$

and

$$\nu_{\max} = \tan^{-1} \sec u \quad (3-34)$$

Substituting  $\phi_{\max}$  and  $\nu_{\max}$  in Equation (3-27)

$$\tan^2 \nu_{\max} = C(1 - \cos \phi_{\max}) \quad (3-35)$$

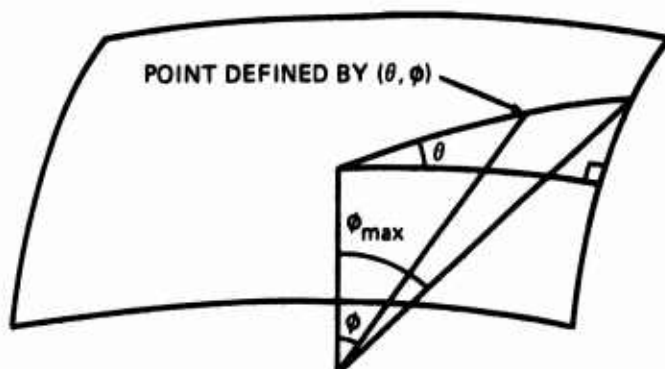


Figure 3-8. Relationship of  $\phi_{\max}$  to  $\theta$

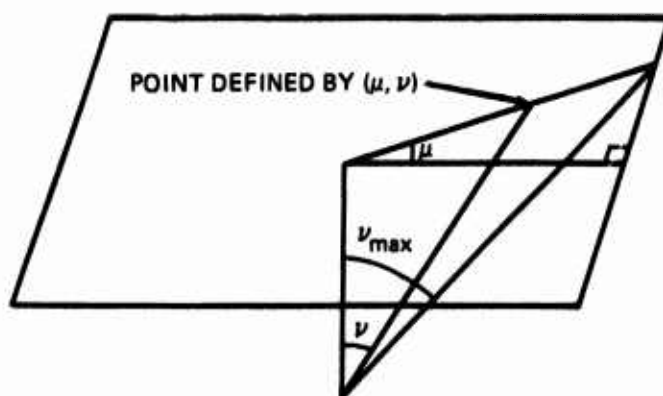


Figure 3-9. Relationship of  $\nu_{\max}$  to  $\mu$

Therefore

$$C = \frac{\tan^2 \nu_{\max}}{1 - \cos \phi_{\max}} = \frac{\tan^2 (\tan^{-1} \sec u)}{1 - \cos (\tan^{-1} \sec \theta)} = \frac{\sec^2 u}{1 - \cos (\tan^{-1} \sec \theta)} \quad (3-36)$$

Substituting C into Equation (3-35) yields

$$\tan^2 \nu = \frac{\sec^2 u}{1 - \cos \tan^{-1} \sec \theta} (1 - \cos \phi) \quad (3-37)$$

and

$$\nu = \tan^{-1} \sqrt{\frac{\sec^2 u (1 - \cos \phi)}{1 - \cos \tan^{-1} \sec \theta}} \quad (3-38)$$

This satisfies the boundary condition  $\phi = 0$ ,  $\nu = 0$ , and additionally has the property that  $\mu = 0$  at  $\phi = 0$  and  $\phi_{\max} = \nu_{\max}$ , where  $\theta = \pm\pi/4$ ,  $u = \pm\pi/4$ .

This relation holds from  $\phi = 0$  to  $\phi = \pi/2$ , and because  $\phi_{\max}$  is never greater than  $\tan^{-1} \sqrt{2} = 0.95$ , it is correct for the entire spherical square.

Thus, Equations (3-38) and

$$u = \tan^{-1} \left\{ \frac{12}{a} \left[ \theta + \cos^{-1} \left( \sin \theta \cos \frac{\pi}{4} \right) - \frac{\pi}{2} \right] \right\} \quad (3-39)$$

represent an area-preserving transformation from the spherical square to the inscribed plane square for the quadrant.

To treat the remaining quadrants, each is successively rotated into the quadrant of definition by subtracting multiples of  $\pi/2$ , as shown in Table 3-1.

Table 3-1. Rotations of  $\theta$  and  $u'$  for a Given  $\theta'$ <sup>a</sup>

Range	$\theta$	$u'$
$-\frac{\pi}{4} \leq \theta' \leq \frac{\pi}{4}$	$\theta'$	$u$
$\frac{\pi}{4} < \theta' \leq \frac{3\pi}{4}$	$\theta' - \frac{\pi}{2}$	$u + \frac{\pi}{2}$
$\frac{3\pi}{4} < \theta' < \frac{5\pi}{4}$	$\theta' - \pi$	$u + \pi$
$\frac{5\pi}{4} < \theta' < \frac{7\pi}{4}$	$\theta' - \frac{3\pi}{2}$	$u + \frac{3\pi}{2}$

<sup>a</sup>The quantities  $\theta'$  and  $u'$  are input and output, respectively; and  $\theta$  and  $u$  are as used in the computation of Equations (3-38) and (3-39).

### 3.3 REFINEMENT OF THE CARTESIAN TRANSFORMATION

Section 4.2 of Reference 1 outlines a method for obtaining the data base coordinates  $(x, y)$  of any datum point from Keplerian orbital elements, time, a scan angle, and parameters describing the attitude of the satellite using the inverse transformation,  $f^*(\xi, \eta)$ . This method has been modified in order to improve computational efficiency. The results are presented below in two parts: the first part treats the computation of the coordinates  $(x, y)$  of the subsatellite point (zero scan angle); the second part develops the necessary added formalism for datum points lying along a scan arc, as defined in Section 3.1.2. Details and derivations are included in Appendix C.

### 3.3.1 Data Base Coordinates of the Subsatellite Point

Computations for obtaining the data base coordinates (x , y) of the subsatellite point are performed as follows:

1. Evaluate Equation (3-1) for l , m , and n , and note the appropriate cube face by selecting the largest of l , m , and n and observing its sign. Table 3-2 establishes the necessary conventions. The general variables  $\xi'$  ,  $\eta'$  , and  $\rho$  are the geocentric rectangular coordinates of the subsatellite point deduced from l , m , and n by entering Table 3-2, where  $\rho$  is the largest in absolute magnitude of l , m , and n .
2. Compute the variables  $\alpha$  and  $\beta$  defined by:

$$\begin{aligned}\alpha &= \xi'/\rho \\ \beta &= \eta'/\rho\end{aligned}\tag{3-40}$$

Table 3-2. Cube Face Identification

Face Number	$\xi'$	$\eta'$	$\rho$
1	m	n	l
2	-m	n	-l
3	-l	n	m
4	l	n	-m
5	-l	-m	n
6	-l	m	-n

3. Compute  $x$  and  $y$  from

$$\begin{aligned} x &= f^*(\alpha, \beta) \\ y &= f^*(\beta, \alpha) \end{aligned} \quad (3-41)$$

where  $f^*(\alpha, \beta)$  is developed in Appendix C. For convenience, the functional relationship is repeated below.

$$\begin{aligned} f^*(\alpha, \beta) = r_o \bigg\{ & \alpha \gamma^* + \alpha^3 (1 - \gamma^*) + \alpha \beta^2 (1 - \alpha^2) \\ & \times \left[ \Gamma + (\Gamma - M) (1 - \alpha^2) + (1 - \beta^2) \sum_{\substack{i \geq 0 \\ j \geq 0}} C_{ij} \alpha^{2i} \beta^{2j} \right] \\ & + \alpha^3 (1 - \alpha^2) \left[ \Omega_1 + (1 - \alpha^2) \sum_{i \geq 0} D_i \alpha^{2i} \right] \bigg\} \end{aligned} \quad (3-42)$$

where  $\alpha$  and  $\beta$  are defined by Equations (3-40),  $r_o = 1/\sqrt{3}$ , and the remaining constants are defined in Appendix C (Equations (C-10) and (C-11)).

### 3.3.2 Data Base Coordinates for Nonzero Scan Angle

The data base coordinates  $(x, y)$  for any pair of datum points lying on a scan arc at an angular separation of  $\pm\psi/2$  from the subsatellite point are obtained by performing the following computational steps:

1. Obtain the geocentric longitude,  $\theta$ , and colatitude,  $\phi$ , of the subsatellite point using Equations (3-1) and (3-3). (If the ground trace has already been determined, its direction cosines will already have been computed from Equations (3-1).)

2. Compute the sines and cosines of the geocentric longitudes,  $\theta'$ , and colatitudes,  $\phi'$ , of the datum points, using Equations (3-4), (3-5), and (3-6), and the results of step 1.
3. Using the sines and cosines of  $\theta'$  and  $\phi'$  in Equations (3-3), obtain the two sets of direction cosines ( $l'$ ,  $m'$ , and  $n'$ ) of the datum points.
4. Identify the appropriate cube face(s) and variables  $\xi'$ ,  $\eta'$ , and  $\rho$  using Table 3-2, and compute values for  $\alpha$  and  $\beta$  using Equations (3-40) for each point.
5. Calculate the (x, y) coordinates using Equations (3-41).

### 3.4 APPROXIMATE FORMS

The transformations outlined in the previous subsections share the common disadvantage of being complex and difficult to evaluate. While studying the transformations  $f$  and  $f^*$  for the previous report, it was determined that a linear interpolation scheme was sufficient to find points between known benchmark grid points (Reference 1, Section 4.3). It is therefore evident that the transformation equation may be approximated by low-order polynomials. This subsection presents a method for accurately approximating the transformations over the entire cube face using polynomials of varying degree with locally evaluated and tabulated coefficients.

Because of symmetry properties of the transformations, it is necessary to evaluate the coefficients of the approximation polynomial over only one octant of one face. In addition, since

$$\begin{aligned} x &= f(\xi, \eta); \quad y = f(\eta, \xi) \\ \xi &= f^*(x, y); \quad \eta = f^*(y, x) \end{aligned} \tag{3-43}$$



only one set of coefficients is necessary to determine both  $x$  and  $y$ , and a second set to determine  $\xi$  and  $\eta$ .

From Table 4-2 of Reference 1, it can be seen that a double linear interpolation scheme will allow computation of intermediate points to an accuracy of 0.25 pixel over a range of at least 16 pixels (and up to 1024 pixels in the central region) on a 4096 x 4096 square. Consequently, it is reasonable to assume that a higher order polynomial would allow one-pixel accuracy over considerably greater distances. As a first approximation, it is assumed that determination and tabulation of coefficients on a grid with 64-pixel intervals will suffice. This interval is convenient, since it corresponds to the proposed standard data base record size.

To obtain suitable interpolating polynomials, the standard bivariate interpolation formulae were examined for three, four, and six points (Reference 4, Sections 25.2.65 to 25.2.67). From Figure 3-10, the formulae are

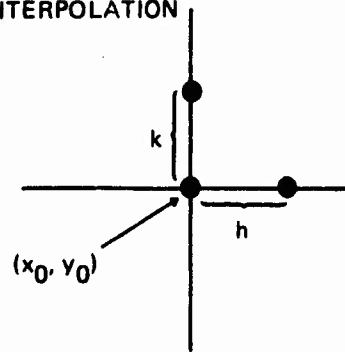
- Three Points (Linear)

$$f_a(x_0 + ph, y_0 + qk) = (1 - p - q) f_{0,0} + pf_{1,0} + qf_{0,1} + O(h^2) \quad (3-44)$$

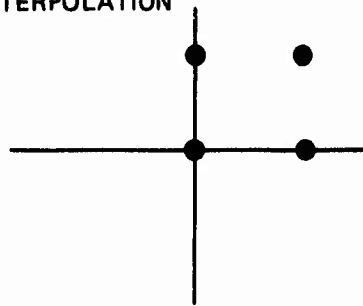
- Four Points

$$\begin{aligned} f_a(x_0 + ph, y_0 + qk) = & (1 - p)(1 - q) f_{0,0} + p(1 - q) f_{1,0} \\ & + q(1 - p) f_{0,1} + pqf_{1,1} + O(h^2) \end{aligned} \quad (3-45)$$

a. THREE POINT INTERPOLATION



b. FOUR POINT INTERPOLATION



c. SIX POINT INTERPOLATION

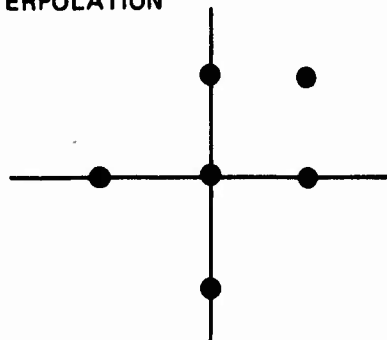


Figure 3-10. Grid Points Used for Interpolation

● Six Points

$$\begin{aligned}
 f_a(x_0 + ph, y_0 + qk) = & \frac{q(q-1)}{2} f_{0,-1} + \frac{p(p-1)}{2} f_{-1,0} \\
 & + (1 + pq - p^2 - q^2) f_{0,0} \\
 & + \frac{p(p-2q+1)}{2} f_{1,0} \\
 & + q(q-2p+1) f_{0,1} + pq f_{1,1} + O(h^3)
 \end{aligned} \tag{3-46}$$

Equations (3-44) and (3-45) can be written as follows:

$$f_a(x, y) = a_{10} x + a_{01} y + a_{00} \tag{3-47}$$

$$f_a(x, y) = a_{11} xy + a_{10} x + a_{01} y + a_{00} \tag{3-48}$$

The coefficients  $a_{00}$ ,  $a_{10}$ , ..., are evaluated by setting up a set of linear equations corresponding to the interpolation benchmark points and solving them simultaneously.

The six-point formula is less convenient to apply systematically, since it requires knowledge of points outside its range for solution. A compromise form

$$f_a(x, y) = a_{20} x^2 + a_{02} y^2 + a_{10} x + a_{01} y + a_{00} \tag{3-49}$$

can be evaluated on five points, as shown in Figure 3-11. This form eliminates the xy cross-product term and approximates the functional form with a paraboloid of revolution whose axis is perpendicular to the xy plane.

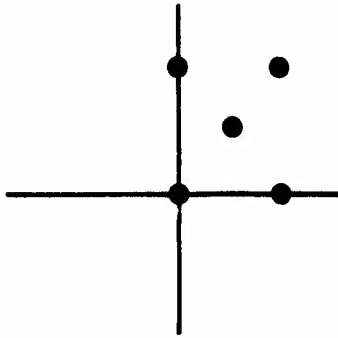


Figure 3-11. Five-Point Solution

A computer program was written to solve the system of simultaneous equations arising from each of the above approximate forms on  $64 \times 64$  pixel intervals on one octant of the cube face (Figure 3-12). A standard IBM Scientific Subroutine Package (Reference 5) routine, SIMQ<sup>1</sup>, was used for solution of the equations. To test the accuracy of each approximation, a set of evaluation points was chosen (Figure 3-13). The approximations were used successively to determine the coordinates of each evaluation point and a residual was computed for each point by solving the full transformation. Any  $64 \times 64$  region in which the largest residual exceeded one pixel was flagged and the next higher degree approximation was tried. Table 3-3 shows the number of coefficients necessary for one pixel accuracy as a function of location.

The table size required is not excessive because, as stated above, only one octant (plus diagonal elements) must be stored. For a  $64 \times 64$  grid, this would be

$$M \times (31^2 + 16) = M \times 977$$

---

<sup>1</sup> SIMQ employs the Gaussian elimination method using largest pivotal divisors.

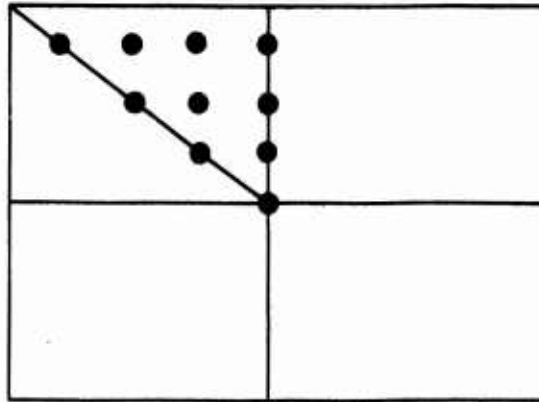


Figure 3-12. Origins of Coefficient Set Domains  
Necessary To Define Entire Face

Table 3-3. Number of Coefficients Necessary for One-Pixel  
Accuracy as a Function of Position

																4096, 4096
																4
																4 4
																3 4 4
																3 3 4 4
																3 3 3 3 4
																3 3 3 3 3 4
																3 3 3 3 3 3 3
																3 3 3 3 3 3 3 3
																3 3 3 3 3 3 3 3 3
																3 3 3 3 3 3 3 3 3 3
																3 3 3 3 3 3 3 3 3 3 3
																3 3 3 3 3 3 3 3 3 3 3 3
																3 3 3 3 3 3 3 3 3 3 3 3 3
																3 3 3 3 3 3 3 3 3 3 3 3 3 3
																3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
2048, 2048	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4096, 2048

where  $M$  is the largest number of coefficients required. For the case where  $M = 5$ , this requires 4885 words.

The program which evaluates the coefficients for the approximate direct transformation  $f_a$  is shown in Appendix E. Appendix D tabulates the coefficients derived and indicates the necessary number of coefficients for one-pixel accuracy. The use of these approximations will be discussed in Section 4.1.

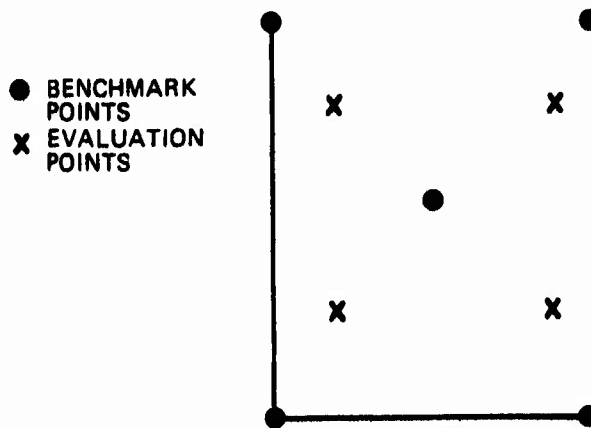


Figure 3-13. Arrangement of Evaluation Points

### 3.5 COMPARISON OF TRANSFORMATIONS

In the previous subsections, four different formulations were developed for the  $f^*$  function (inverse transformation):

1. The original Cartesian form (Equation (2-5))
2. The revised Cartesian form (Equation (3-42))
3. The polar form (Section 3.2)
4. The approximate form (Section 3.4)

In choosing between forms 1 and 2, which produce the same mapping, form 2 is clearly arithmetically simpler and has proved to produce a more accurate determination of its coefficients.

Form 4 may be used to approximate the Cartesian or the polar forms and is the most efficient form computationally. It does, however, require core storage for the coefficient tables.

Figures 3-14 and 3-15 illustrate the differences in the nature of the mappings produced by the Cartesian and polar forms. The salient difference is that the polar form exhibits nondifferentiability at crossings of the diagonals of the face. This directional discontinuity arises from the condition that all great circles pass through the center of a face map as straight lines. This problem, in conjunction with the somewhat slower computation time shown in Table 3-4, favors the Cartesian form for most applications. The polar form retains the advantage of exactness in that it produces equal area to any desired accuracy

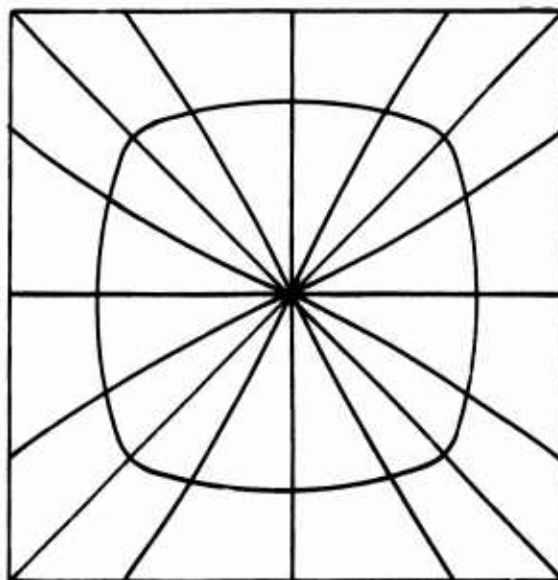
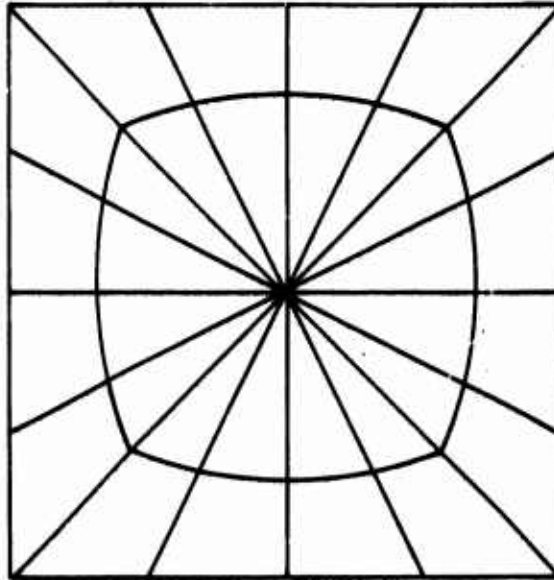


Figure 3-14. Distortions Arising From the Cartesian Form of the Transformation



**Figure 3-15. Distortions Arising From the Polar Form of the Transformation**

**Table 3-4. Execution Times for Transformation Forms on the IBM S/360-91 (One Ordered Pair)**

Transformation	Execution Time (microseconds)
Cartesian	300
Polar	1000
Approximate	40



without recomputing the coefficients. It is difficult to compute coefficients that produce Cartesian transformations which will match retransformed points below the 2-nautical-mile level.

Both the Cartesian and polar formulations have their own uses: the Cartesian system at moderate resolutions and the polar system at the highest resolutions over small areas not including a diagonal. For benchmark transformations, the speed of either transformation is adequate, given a reasonably great interpolation distance. The high computational efficiency of the approximate forms tends to favor this approach for direct computation of data base coordinates (or for computation of corresponding points in the scan line buffer). This technique could also be particularly useful in the corner regions where interpolation is not accurate over useful distances.

## SECTION 4 - I/O SYSTEM DESIGN CONSIDERATIONS

### 4.1 ALTERNATIVE METHOD OF FAST-FILLING

Section 2.3 briefly outlines a proposed alternative fast-filling scheme for the preprocessor program. This method, described in Reference 3, partially solves the problem of edge and vertex crossings and eliminates the necessity for explicit data spreading. By combining this approach with the approximation techniques discussed in this report, the overall efficiency of the preprocessor function can be increased still further.

Sections 3.1 and 3.3 present a method for obtaining data base coordinates  $(x, y)$  directly from Keplerian orbital elements and scanner angle. By omitting the last step of the process outlined in Section 3.3, positions in the  $(\alpha, \beta)$  system can be obtained, which is simply a rescaled  $(\xi, \eta)$  system  $(\alpha = \xi/r_o, \beta = \eta/r_o)$ . Using this algorithm, a benchmark grid in  $(\alpha, \beta)$  can be obtained and a procedure can then be followed which consists of a modification to that described in Reference 3. As outlined in Section 2.3 of Reference 3, filling is performed only for complete data base records (thus avoiding the necessity to read and update partially covered records). The procedure employs a new rotating Earth-centered coordinate system  $(x^R, y^R, z^R)$  in which both scanner and data base benchmarks are defined. CSC proposes modifying this procedure as follows:

1. Instead of using the  $x^R, y^R, z^R$  system, both scanner and benchmark grids in a coplanar system should be defined, projecting the scan line grid into the inscribed cube face and transforming the data base coordinates to the same  $(\alpha, \beta)$  system.
2. The data base benchmarks should be defined in terms of the projected scanner benchmarks (i.e., the  $(E, L)$  coordinates of the data base benchmarks should be defined as shown in Figure 4-1).

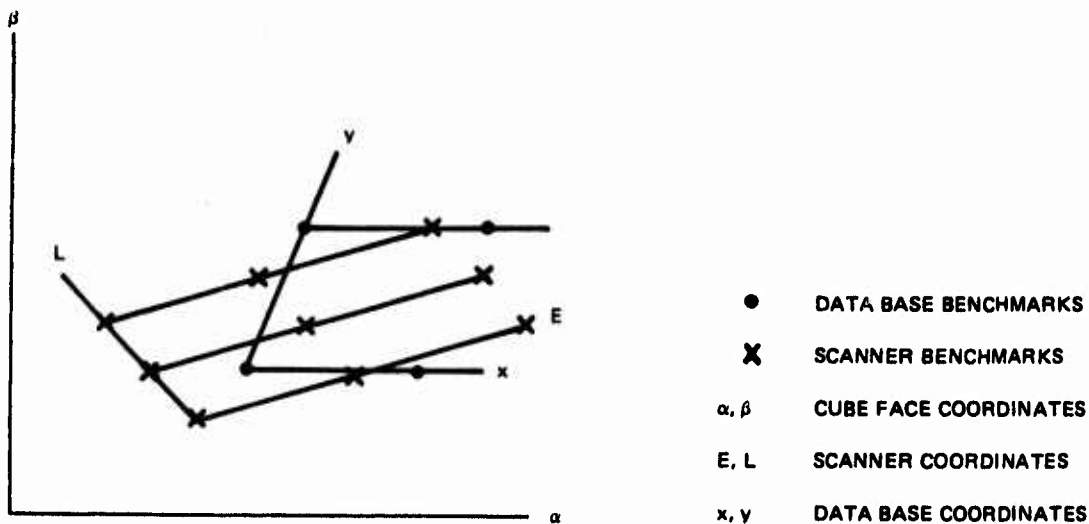


Figure 4-1. Coplanar Representation of Scanner and Data Base Grids

3. The (E, L) coordinates of each data base location should be interpolated and the nearest pixel stored in that location. (This may be done by offsetting the (E, L) system 1/2 pixel in both coordinates and obtaining E and L indices of the desired pixel by truncation.)

The objective of this modification is to avoid the necessity of searching for the nearest pixel in a three-dimensional space.

Neither this method nor the procedure in Reference 3 completely eliminates the edge and vertex crossing problems. It is necessary to predetermine, from the benchmarks of the scanner array, which data base records will be completely covered by the scanner data currently in the input buffer. This process must check for edge and vertex crossings.

Figure 4-2 illustrates the relationship between the scanner data input buffer and the data base records. The scan buffer must have sufficient length (in the along-track direction) to cover the longest diagonal of a data base record. For the 64 x 64 pixel case, this will be approximately 128 pixels in the corners of

the face. The scan line buffer must be organized as a creeping buffer, and no more lines may be added in the forward direction (and deleted in the backward direction) than the number of lines by which the buffer exceeds the minimum needed to cover the longest diagonal. A typical scanner input buffer would have a width (along the scan direction) of approximately 1600 nautical miles (about 1344 pixels), with a minimum length (in the ground track direction) of 128 pixels. This will require sufficient buffer storage for 172,032 pixels. Two methods emerge for handling this large array. The first is to pack more than one pixel per 60-bit word. This would allow the entire buffer to be maintained in central memory on the CYBER/175 but would introduce a large packing and unpacking cost. A second method would be to maintain the buffer in ECS, segmented so that it could be brought into core in blocks as required. The scheduling of these blocks, along with the ordering of output records, could be determined from

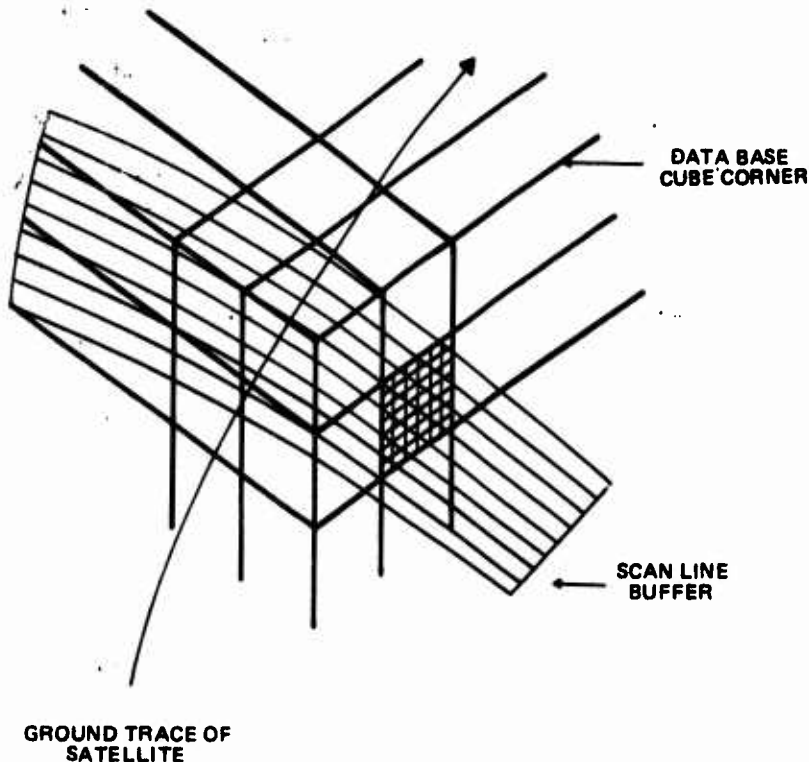


Figure 4-2. Relationship of Scanner Input Buffer and Data Base Records

the (x , y) locations of the projected and transformed scanner input benchmark grid, ordering the filling of data base records by their corresponding along-scan locations.

Because the objective of this new method is to reduce redundant stores, it could also be useful to consider windowing down the along-scan length of the input buffer as a function of latitude to reduce scan path overlap at high latitudes. Similarly, at low latitudes, it may be necessary to subdivide data base records into two or more longitude bands and partially update the records rather than fully rewriting them.

#### 4.2 MAINTENANCE OF MULTIPLE- AND VARIABLE-RESOLUTION DATA BASES

Considerable savings in computation and I/O time and in data base storage can be realized by reducing the resolution of the data base in areas of lesser interest (such as, perhaps, over land and in the southern hemisphere). It may also be desirable to be able to maintain higher resolution for other selected areas. One method for providing such variable resolution coverage consists of maintaining a single, fully global data base at the lowest desired resolution, plus separate higher resolution partial data bases. A second method consists of maintaining a single data base of variable resolution.

In either case, it will be necessary to provide a system for indicating the desired resolution as a function of global position. This can be provided for by establishing a data block properties table (for example at the 128 x 128 or 256 x 256 element level) which would contain the desired resolution described as a multiple of the finest usable resolution or by the number of bits required for the serial address. This would allow the preprocessor program to determine the desired resolution at any point.

For the multiple data base scheme, the preprocessor program would first determine which elements of the resolution table are covered by the data currently in the scan line buffer and then access the table to determine what actions must

be taken. This table would consist of entries for each of the desired resolution levels indicating

- The desired resolution
- A base address for the block of records corresponding to the table element
- The size of the data base records (in elements)

The resolution and record size for all records in a block corresponding to one table element can both be expressed as a number of bits to be added to the corresponding binary division levels of the basic global data base, i.e., if the desired resolution is twice the linear resolution of the basic data base, two more address bits must be added and, if the records are to have the same number of elements, the record/element division of the serial string must take place two bits lower.

The serial string for each resolution level would be kept separate, and the use of a base address for each record block would obviate any need for unused filler records in the data base. For maximum flexibility, each resolution level could be maintained as a separate file.

With multiple data bases, the size of a data base record would be adjustable to allow an optimum match of record size to scanner input buffer size and data item size (where one data item might be all of the simultaneously available, colocated scanner readings). To maintain this match, a separate pass through the data will be required for each resolution level.

In the variable resolution approach, only one pass through the data is required and all data records cover the same area. This means that the ratio of the scanner buffer size to the data record area remains fixed, as do the total number of I/O operations. The total number of data records remains constant; thus, a consistent record numbering and ordering can be maintained, avoiding

the necessity of a table lookup for a block address, as in the multiple data base scheme. In this case, the data block lookup table need contain only the resolution (which will also imply the record length).

Both of the above schemes have advantages and disadvantages:

- The multiple data base scheme requires redundant stores and computation
- The variable resolution scheme requires stringing together unlike data base records, thus requiring a variable record length capability
- The multiple data base scheme allows easy addition or deletion of nonstandard resolution areas for coverage of new areas of interest
- The variable resolution scheme allows more systematic control over the data base storage allocations

A possible resolution of these conflicts is to use a hybrid system in which the variable resolution format would be used for resolutions coarser than a set standard resolution level, and the multiple data base approach for higher resolution levels (and for other subdata bases required for special purposes or limited durations).

A unified block properties table would be maintained which would allow display of the current status and availability of a data record in the system. This table could be expanded to include such information as the last update time, satellite ID, dayside/nightside information, etc. The advantages of such a scheme would be that multiple passes through the data would be restricted to selected regions of highest resolution, special changes in data base content could be accomplished without restructuring the main data base as a whole, systematic control over the entire data base could be maintained, and the serial string addressing scheme would be maintained for the standard resolution data.

## SECTION 5 - GEOGRAPHIC ORIENTATION

### 5.1 INTRODUCTION

In order to maximize the usefulness and efficiency of the spherical cube data base, it may be necessary to orient the inscribed cube in some arbitrary manner with respect to the Earth. The simplest orientation of the cube aligns one Cartesian axis of the cube coordinate system with the Earth's axis of rotation and points another cube axis toward the prime meridian. A simple variation is to have one cube axis aligned with the Earth's axis of rotation and another pointing to a specified longitude. This section will discuss these orientations and will present an algorithm for producing arbitrary cube rotations.

### 5.2 SIMPLE ORIENTATIONS

In Section 4.2 and Appendix E.6 of Reference 1, algorithms are presented for the transformation of Earth-centered Cartesian coordinates (a , b , c) to data base coordinates (x , y) . For the case where the Earth-centered (a , b , c) system is aligned as shown in Figure 5-1, the Cartesian coordinates can be represented as:

$$a = \cos (\text{latitude}) \cos (\text{longitude}) \quad (5-1)$$

$$b = \cos (\text{latitude}) \sin (\text{longitude}) \quad (5-2)$$

$$c = \sin (\text{latitude}) \quad (5-3)$$

To accomplish a simple rotation about the Earth's polar axis, the following algorithms can be used:

$$a = \cos (\text{latitude}) \cos (\text{longitude} + \delta) \quad (5-4)$$



$$b = \cos (\text{latitude}) \sin (\text{longitude} + \delta) \quad (5-5)$$

$$c = \sin (\text{latitude}) \quad (5-6)$$

where  $\delta$  represents the angle of rotation.

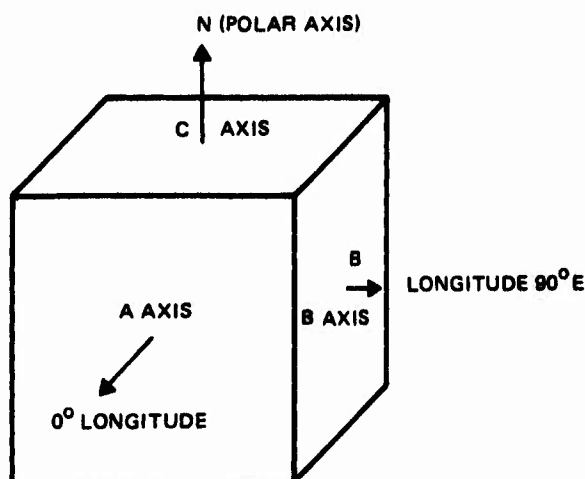


Figure 5-1. Orientation of the Axes of the Cube (Unrotated)

### 5.3 THE GENERAL ROTATION CASE

In order to conveniently generate arbitrary rotations of the data base cube with respect to the sphere, a method was adopted for making successive rotations of the sphere about each of the axes of the cube. By choosing the order in a suitable manner, this yields a conceptually simple way of arriving at a given orientation.

Taking a general set of orthogonal axes,  $G_1$ ,  $G_2$ ,  $G_3$ , as shown in Figure 5-2, the rotation about axis  $G_1$  can be expressed as follows:

Let  $\alpha$  be the desired angle of rotation about  $G_1$ . For any arbitrary point,  $g_1$ ,  $g_2$ ,  $g_3$ , the position may be expressed in terms of the following three variables:

- $g_1$ , the  $G_1$  coordinate
- $\beta$ , the angular position of the projection of  $g_1$ ,  $g_2$ ,  $g_3$  on the  $G_2G_3$  plane measure from  $G_3$  in the clockwise sense
- $\rho$ , the perpendicular distance from  $g_1$ ,  $g_2$ ,  $g_3$  to the  $G_1$  axis

For a pure rotation about  $G_1$  (Figure 5-3),  $g_1$  and  $\rho$  are constant and the final orientation,  $g'_1$ ,  $g'_2$ ,  $g'_3$ , is determined by the new orientation angle,  $\gamma$ , where

$$\gamma = \alpha + \beta \quad (5-7)$$

and since

$$\beta = \tan^{-1} (g_3/g_2) \quad (5-8)$$

then

$$\gamma = \alpha + \tan^{-1} (g_3/g_2) \quad (5-9)$$

where

$$\rho = g_2^2 + g_3^2 \quad (5-10)$$

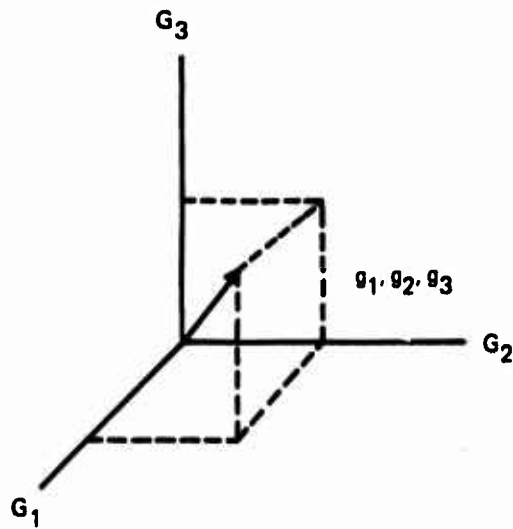


Figure 5-2. Point  $g_1, g_2, g_3$  in the General Reference Frame,  $G_D, G_2, G_3$

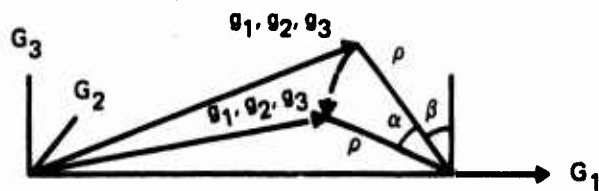


Figure 5-3. Rotation of the Point  $(g_1, g_2, g_3)$  About the  $G_1$  Axis

yielding

$$g'_1 = g_1 \quad (5-11)$$

$$g'_2 = \rho \cos \gamma \quad (5-12)$$

$$g'_3 = \rho \sin \gamma \quad (5-13)$$

Using this general approach, a rotation through an angle  $\alpha_1$  is first made about the axis coincident with the Earth's axis of rotation. This is functionally equivalent to the rotation described in Section 5.1 and results in a rotation of the  $\alpha_1$  longitude line into the center of face number 1.

The second rotation is made about the axis normal to the pole and emerging through a face normal to face number 1 (axis b in Figure 5-1). If this rotation is through an angle  $\alpha_2$ , then the point with longitude  $\alpha_1$  and latitude  $\alpha_2$  will be rotated into the center of face 1.

A third rotation about the axis emerging from the center of face 1 will produce a "position angle" change on face 1 but will not translate the point at the center of the face.

Using this method, any point whose latitude and longitude are known may be translated to the center of face 1 and the rotational aspect of that face may be altered without changing the latitude and longitude of the center. Referring to Figure 5-1, the order of these rotations is about axes c, b, and a, respectively.

Appendix E contains an algorithm for the application of the general transformation to successive axes through the interchange of components.

Figures 5-4 and 5-5 illustrate the continental outlines after the indicated rotations and are typical of maps generated by this process.

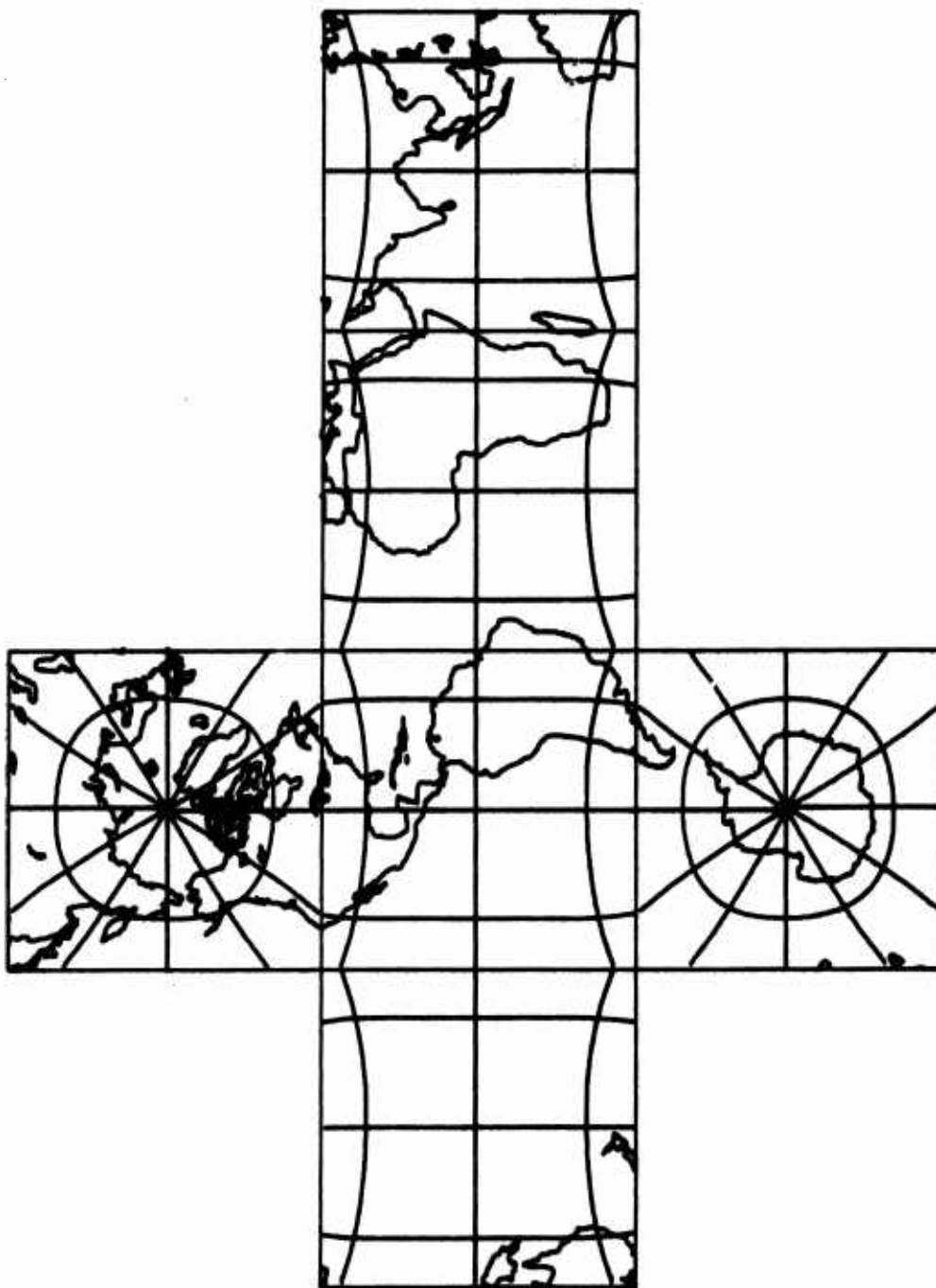
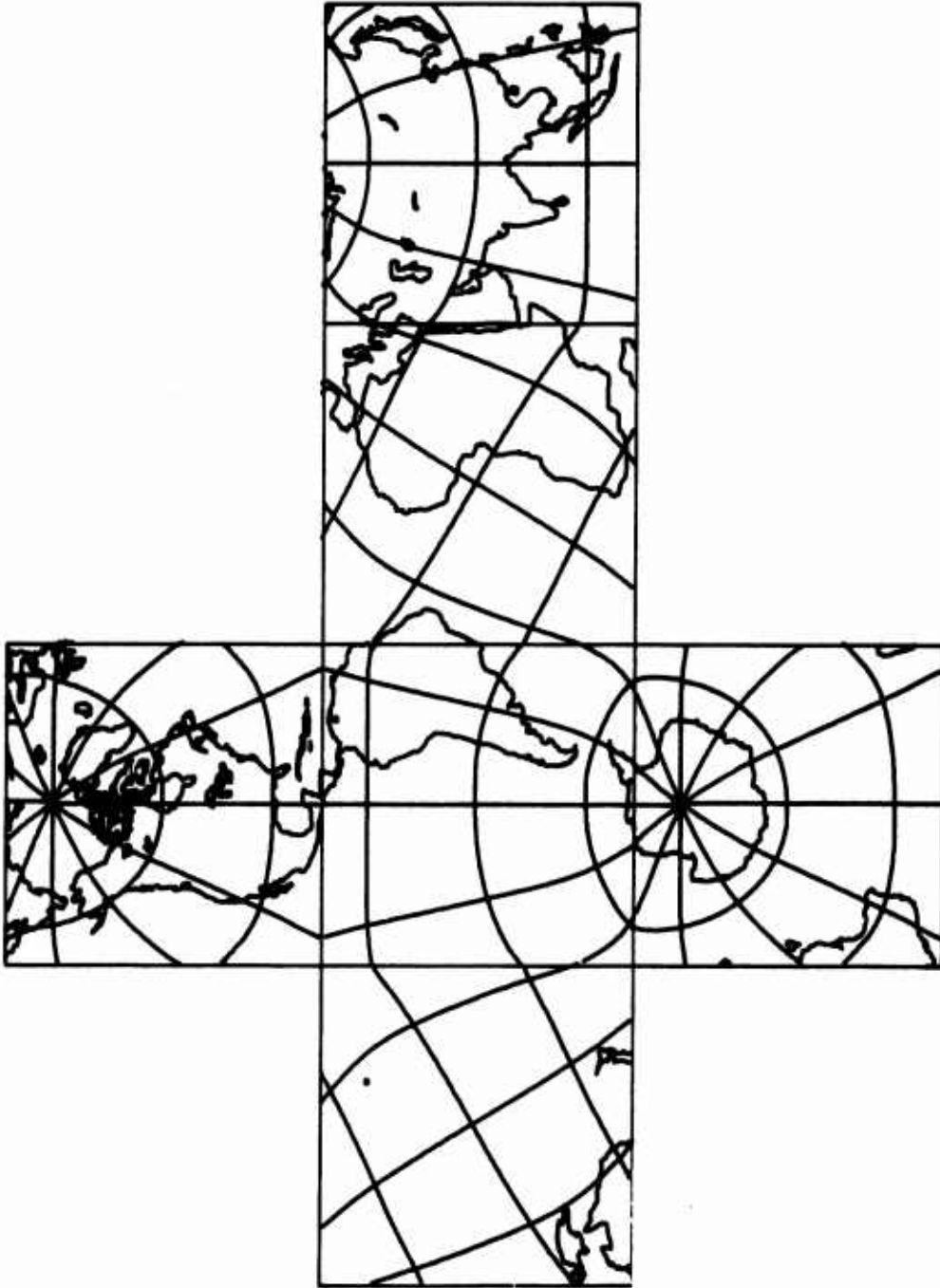


Figure 5-4. Data Base Map of Earth With Cube Oriented 90 Degrees West of Greenwich



**Figure 5-5. Data Base Map of Earth With Cube Oriented 90 Degrees West of Greenwich and 30 Degrees South of the Equator**

## SECTION 6 - SUMMARY AND CONCLUSIONS

### 6.1 NEW RESULTS

The purpose of the research reported herein is to improve the procedures and algorithms required for the efficient maintenance of data in a quadrilateralized spherical cube Earth data base. The new results obtained relate to the fast-filling of the data base. A technique was developed for rapidly finding the ground trace and scanner-sensed locations from Keplerian elements and scanner angle. The previously derived form of the inverse transformation,  $f^*$ , was improved by a change of variables. A new inverse transformation was derived using polar coordinates, and a technique was developed for approximating the transformation functions using low-order polynomials.

All of these results will contribute to improving the accuracy and speed of the preprocessor function. The general functioning of the preprocessor input/output algorithms was re-examined in light of an alternative scheme (Reference 3). This scheme and the previously developed fast-filling method (Reference 1) were compared, and a modification of the scheme described in Reference 3 is recommended.

In addition, improved world maps in data base coordinates were produced for use in determining an optimum orientation of the data base cube. Additional efforts were devoted to improving the accuracy of the transformations described in Reference 1. The revised coefficients for these transformations are included in Appendix B of this report.

### 6.2 RECOMMENDATIONS FOR FURTHER STUDY

A number of topics considered during this study are worthy of additional attention. The polar formulations of the transformation functions should be further developed using other constraints to maximize conformality of the mapping.

The role of the MAP III processor in implementing the preprocessor function was excluded from this study because documentation is not available. The role which MAP III and the high-speed communications system's processor could assume in evaluating the transformation equations, particularly the approximate transformation, should be explored further.

The various CYBER I/O access methods should be studied further, particularly to ascertain whether or not they may cause fragmentation of the data base.

Consideration should be given to producing display output in data base coordinates with a geographic grid overlay, thereby saving a major computation cost in retransforming the data.



## APPENDIX A - GROUND TRACE EQUATIONS

This appendix presents the derivation of the expressions summarized in Section 3.1 and referred to elsewhere in this document. Two problems are addressed: (1) the computation of the geocentric direction cosines  $(l, m, n)$  of the subsatellite point in terms of the Keplerian orbital elements of the satellite and (2) the computation of the geocentric longitudes and colatitudes of any pair of datum points on a scan arc in terms of previously computed quantities (e.g.,  $l, m, n$ ). A spherical Earth is assumed throughout.

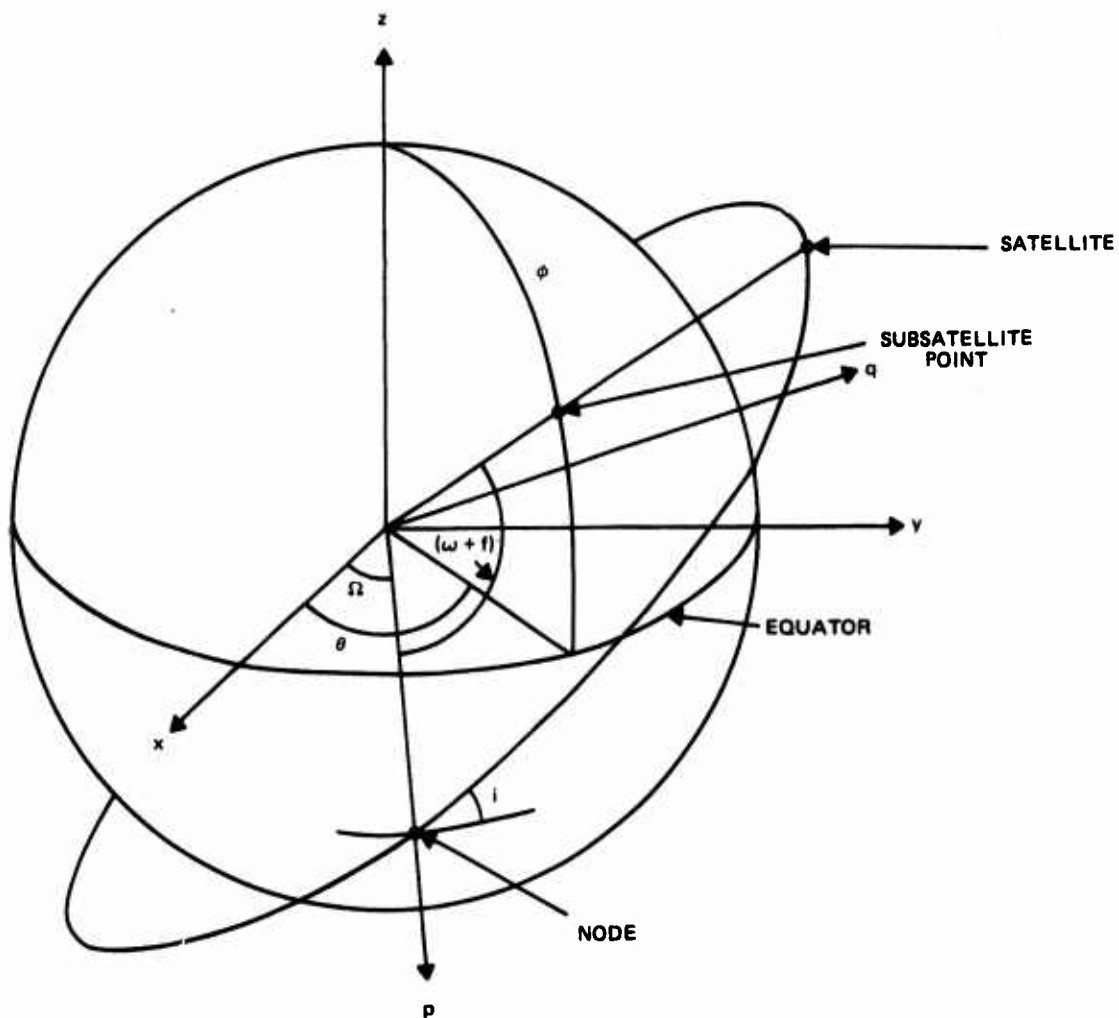
### A.1 THE SUBSATELLITE POINT

Figure A-1 shows the geometric relationship of the satellite, the satellite's orbit, and the Earth's equator to the (rotating) rectangular coordinate system  $(x, y, z)$ . The coordinate system is fixed in the Earth and, therefore, rotates at the diurnal rate. The Keplerian orbital elements whose definitions are given by the figure are the inclination,  $i$ ; the argument of perigee,  $\omega$ , shown as part of the angle  $(\omega + f)$ , where  $f$  is the satellite's true anomaly; and the node,  $\Omega$ , which is not the usual fixed quantity, but varies with the sidereal time (diurnal rate),  $s$ , as follows:

$$\Omega = \Omega_0 - s \quad (A-1)$$

where  $\Omega_0$  is the value of  $\Omega$  at an arbitrary initial time.

Figure A-1 shows the subsatellite point having geocentric coordinates longitude,  $\theta$ , and colatitude,  $\phi$ , and lying, by definition, on the line joining the satellite and the Earth's center. Therefore, the direction cosines  $(l, m, n)$  describing the direction to the satellite in the  $x, y, z$  system are identical with those for the subsatellite point. The straightforward relationship between the direction cosines and  $\theta$  and  $\phi$  will be exploited in Section A.2. First, the



**Figure A-1. Satellite Orbit Orientation and Geographic Coordinates**

expressions for  $l$ ,  $m$ , and  $n$  in terms of  $\Omega$ ,  $\omega$ ,  $i$ , and  $f$  are derived below.

Consider the geocentric position vector,  $\vec{r}$ , of the satellite in its orbit. Express  $\vec{r}$  in the two-dimensional rectangular coordinate system  $(\hat{p}, \hat{q})$  in the orbital plane, where the unit vector  $\hat{p}$  is in the direction of the node and the unit vector  $\hat{q}$  is in the direction 90 degrees in advance of  $\hat{p}$  (see Figure A-1)

$$\vec{r} = r \cos (\omega + f) \hat{p} + r \sin (\omega + f) \hat{q}$$

where  $r$  is the magnitude of  $\vec{r}$ . The coordinates  $(r_x, r_y, r_z)$  of the satellite in the geocentric equatorial rectangular coordinate system are, then,

$$\begin{aligned} r_x &= \vec{r} \cdot \hat{x} = r \cos (\omega + f) \hat{p} \cdot \hat{x} + r \sin (\omega + f) \hat{q} \cdot \hat{x} \\ r_y &= \vec{r} \cdot \hat{y} = r \cos (\omega + f) \hat{p} \cdot \hat{y} + r \sin (\omega + f) \hat{q} \cdot \hat{y} \\ r_z &= \vec{r} \cdot \hat{z} = \quad \quad \quad + r \sin (\omega + f) \hat{q} \cdot \hat{z} \end{aligned} \quad (A-2)$$

where the term in  $\hat{p} \cdot \hat{z}$  for  $r_z$  is omitted, since  $\hat{p} \cdot \hat{z} = 0$ .

Now,

$$\begin{aligned} \hat{p} \cdot \hat{x} &= \cos \Omega \\ \hat{q} \cdot \hat{x} &= \cos (90^\circ + \Omega) \cos i = -\sin \Omega \cos i \\ \hat{p} \cdot \hat{y} &= \sin \Omega \\ \hat{q} \cdot \hat{y} &= \sin (90^\circ + \Omega) \cos i = \cos \Omega \cos i \\ \hat{q} \cdot \hat{z} &= \cos (90^\circ - i) = \sin i \end{aligned} \quad (A-3)$$

Substituting Equations (A-3) into Equations (A-2),  $r_x$ ,  $r_y$ , and  $r_z$  become

$$\begin{aligned} r_x &= r \cos \Omega \cos (\omega + f) - r \sin \Omega \sin (\omega + f) \cos i \\ r_y &= r \sin \Omega \cos (\omega + f) + r \cos \Omega \sin (\omega + f) \cos i \\ r_z &= r \sin (\omega + f) \sin i \end{aligned} \quad (A-4)$$

Since the satellite position vector,  $\vec{r}$ , in the  $(\hat{x}, \hat{y}, \hat{z})$  system is

$$\vec{r} = r_x \hat{x} + r_y \hat{y} + r_z \hat{z} \quad (A-5)$$

the direction cosines  $(l, m, n)$  of its position are defined by

$$\begin{aligned} \vec{r} \cdot \hat{x} &= |\vec{r}| |\hat{x}| l \\ \vec{r} \cdot \hat{y} &= |\vec{r}| |\hat{y}| m \\ \vec{r} \cdot \hat{z} &= |\vec{r}| |\hat{z}| n \end{aligned} \quad (A-6)$$

Now, from Equation (A-5),

$$\begin{aligned} \vec{r} \cdot \hat{x} &= r_x \hat{x} \cdot \hat{x} = r_x \\ \vec{r} \cdot \hat{y} &= r_y \hat{y} \cdot \hat{y} = r_y \\ \vec{r} \cdot \hat{z} &= r_z \hat{z} \cdot \hat{z} = r_z \end{aligned}$$

which, when substituted into Equations (A-6), yields

$$\begin{aligned} l &= r_x / |\vec{r}| \quad |\hat{x}| = r_x / r \\ m &= r_y / |\vec{r}| \quad |\hat{y}| = r_y / r \\ n &= r_z / |\vec{r}| \quad |\hat{z}| = r_z / r \end{aligned} \tag{A-7}$$

since  $|\hat{x}| = |\hat{y}| = |\hat{z}| = 1$ , and  $|\vec{r}| = r$ .

Equations (A-4) and (A-7) then yield

$$\begin{aligned} l &= \cos \Omega \cos (\omega + f) - \sin \Omega \sin (\omega + f) \cos i \\ m &= \sin \Omega \cos (\omega + f) + \cos \Omega \sin (\omega + f) \cos i \\ n &= \sin (\omega + f) \sin i \end{aligned} \tag{A-8}$$

Equations (A-8) provide one way of defining the "ground trace" of the satellite, since  $l$ ,  $m$ , and  $n$  may be regarded as the "unit" geocentric coordinates of the subsatellite point. That is, assuming a spherical Earth, the actual coordinates of the subsatellite point are obtained by multiplying  $l$ ,  $m$ , and  $n$  by the Earth's radius in whatever units are required (e.g., kilometers). Equation (A-8) is given as Equation (3-1) in the text.

## A.2 THE LOCATION OF DATUM POINTS ON A SCAN ARC

The satellite's onboard sensors are assumed to sweep out a scan arc having ground trace geometric properties as defined in Section 3.1.2 and as illustrated in Figure A-2. It is convenient to regard the ground trace of the scan arc as being composed of pairs of datum points symmetrically placed on either side of the subsatellite point, and to regard  $\psi/2$  as a general variable locating the points by assigning alternate algebraic signs to  $\psi/2$  (e.g., the ends of the

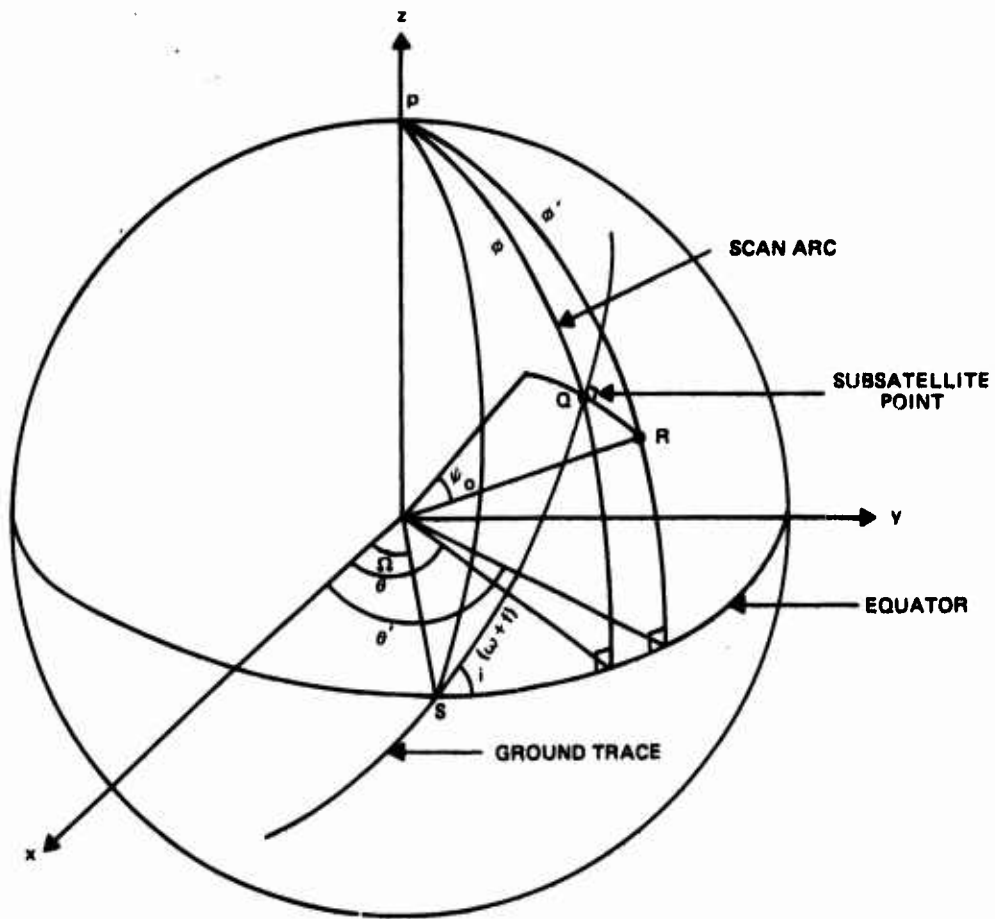


Figure A-2. The Scan Arc as a Central Angle  $\psi_0$

scan arc are located at  $\pm\psi_0/2$  from the subsatellite point). It is desired to obtain pairs of values for the geocentric longitude,  $\theta'$ , and colatitude,  $\phi'$ , of the datum points in terms of known quantities in a computationally efficient manner.

The sines and cosines of the geocentric longitude,  $\theta$ , and colatitude,  $\phi$ , of the subsatellite point are required as intermediate quantities in terms of  $l$ ,  $m$ , and  $n$  from Equations (A-8). By inspection of Figure A-1,

$$\begin{aligned}\cos \theta \sin \phi &= l \\ \sin \theta \sin \phi &= m \\ \cos \phi &= n\end{aligned}\tag{A-9}$$

Care must be taken in using Equations (A-9) to obtain  $\sin \phi$ ,  $\cos \phi$ ,  $\sin \theta$ , and  $\cos \theta$ , since the algebraic signs of these quantities must be preserved for later use. This is readily achieved by noting that  $\cos \phi$  yields  $\phi$  uniquely, since  $0 \leq \phi \leq 180$  degrees. Straightforward manipulation yields the remaining quantities. Equation (A-9) is given in the text as Equation (3-3).

To obtain the values of  $\phi'$ , the necessary relations may be derived by considering the two spherical triangles PSQ and PQR shown in Figure A-2 and having the common side PQ, which is the colatitude,  $\phi$ , of the subsatellite point. These triangles are shown in detail in Figure A-3, where it is noted that the arc PS is 90 degrees, that the scan arc intersects the ground trace at right angles, and that point R' may be regarded as any point on the scan arc at a distance of  $\psi/2$  from the subsatellite point, Q. (Figure A-2 shows only the end point of the scan arc, R, at a distance  $\psi_0/2$  from the subsatellite point.) Also shown in Figure A-3 are two auxiliary angles,  $A$  and  $\beta$ , such that  $A + \beta + 90$  degrees = 360 degrees.

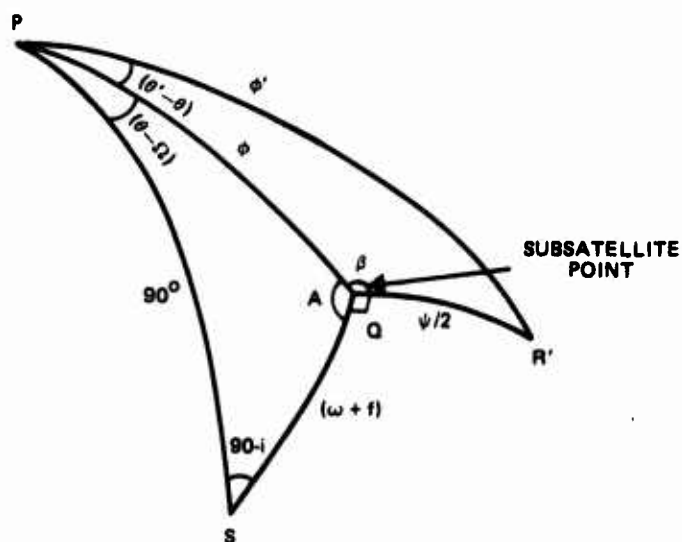


Figure A-3. Two Adjoining Spherical Triangles and the Subsattellite Point  $Q$



The cosine law applied to triangle PQR' yields

$$\cos \phi' = \cos \phi \cos \psi/2 + \sin \phi \sin \psi/2 \cos \beta \quad (\text{A-10})$$

Since  $\psi/2$  is a quantity given a priori, all of the quantities to the right of the equality in Equation (A-10) are known except  $\cos \beta$ , which is obtained as follows.

By definition

$$\beta = 270^\circ - A$$

Therefore

$$\begin{aligned} \cos \beta &= \cos (270^\circ - A) = \cos (270^\circ) \cos A + \sin (270^\circ) \sin A \\ &= -\sin A \end{aligned} \quad (\text{A-11})$$

Applying the sine law to triangle PSQ ,

$$\frac{\sin A}{\sin 90^\circ} = \frac{\sin (90^\circ - i)}{\sin \phi}$$

$$\text{or} \quad (\text{A-12})$$

$$\sin A = \cos i / \sin \phi$$

Employing Equations (A-11) and (A-12) in Equation (A-10) yields

$$\cos \phi' = \cos \phi \cos \psi/2 - \cos i \sin \psi/2 \quad (\text{A-13})$$

which is the desired relation for  $\cos \phi'$ . Equation (A-13) yields  $\phi'$  uniquely, since  $0 \leq \phi' \leq 180$  degrees, so that  $\sin \phi'$  follows immediately. By appropriately redrawing Figure A-3 and repeating the above analysis, it can be readily seen that  $\phi'$  for the datum point corresponding to  $-\psi/2$  on the other half of the scan arc is obtained from Equation (A-13) simply by replacing  $\psi/2$  with  $-\psi/2$ .

Next, to obtain the values of  $\theta'$ , Figure A-3 is again used to note that the cosine law applied in triangle PQR' yields

$$\cos \psi/2 = \cos \phi \cos \phi' + \sin \phi \sin \phi' \cos (\theta' - \theta)$$

or

(A-14)

$$\cos (\theta' - \theta) = (\cos \psi/2 - \cos \phi \cos \phi') / \sin \phi \sin \phi'$$

Since  $\theta$  and all other quantities in Equation (A-14) are previously known through Equations (A-9), it remains only to obtain an independent relation for  $\sin (\theta' - \theta)$  to uniquely specify the quadrant of  $\theta'$ . The sine law applied to triangle PQR' yields

$$\frac{\sin \beta}{\sin \phi'} = \frac{\sin (\theta' - \theta)}{\sin \psi/2}$$

or

(A-15)

$$\sin (\theta' - \theta) = \sin \beta \sin \psi/2 / \sin \phi'$$

To obtain  $\sin \beta$ , it should be noted that, by definition,

$$\beta = 270^\circ - A$$

Therefore,

$$\begin{aligned}\sin \beta &= \sin (270^\circ - A) = \sin (270^\circ) \cos A - \cos (270^\circ) \sin A \\ &= -\cos A\end{aligned}\tag{A-16}$$

Applying the cosine law for angles to triangle PQS ,

$$\begin{aligned}\cos A &= -\cos (90^\circ - i) \cos (\theta - \Omega) + \sin (90^\circ - i) \sin (\theta - \Omega) \cos (90^\circ) \\ &= -\sin i \cos (\theta - \Omega)\end{aligned}\tag{A-17}$$

However, the cosine law for sides in triangle PQS yields

$$\cos (\omega + f) = \cos (90^\circ) \cos \phi + \sin (90^\circ) \sin \phi \cos (\theta - \Omega)$$

or

$$\cos (\theta - \Omega) = \cos (\omega + f) / \sin \phi$$

Therefore, Equation (A-17) becomes

$$\cos A = -\sin i \cos (\omega + f) / \sin \phi\tag{A-18}$$

Substituting Equation (A-18) into Equation (A-16), and then substituting the result into Equation (A-15) yields

$$\sin (\theta' - \theta) = \cos (\omega + f) \sin i \sin \psi / 2 / \sin \phi \sin \phi'\tag{A-19}$$

Equations (A-14) and (A-19) are the desired expressions that uniquely determine  $\theta'$  from previously known quantities. Alternative forms for Equation (A-19) were found, but the one given here offers the simplest functions

of the quantities, all of which are previously known and is the most rapid to compute, since no additional evaluations of trigonometric functions are required.

Again, by appropriately redrawing Figure A-3 and repeating the above analysis, it can be readily seen that Equations (A-14) and (A-19) uniquely yield  $\theta'$  for the datum point corresponding to  $-\psi/2$  on the other half of the scan arc. This is accomplished by entering Equations (A-19) with  $-\psi/2$ , and entering both Equations (A-14) and (A-19) with the appropriate  $\phi'$  from Equation (A-13).

The important results just derived are Equations (A-13), (A-14) and (A-19), which yield  $\phi'$  and  $\theta'$  for pairs of datum points on any scan arc, and are cited as Equations (3-4), (3-5), (3-6), respectively, in the text. Caution is necessary when applying Equations (A-14) and (A-19) near the poles, since these relations are singular for subsatellite points or datum points at the poles.

This is expected, since the geocentric longitudes  $\theta$  and  $\theta'$  are undefined at the poles. No problem need arise in computations, however, since the satellites are not expected to have exactly 90-degree inclinations and since, for the case of an equatorial orientation of the data base cube, it is a simple matter to regard the cube face coordinates of such datum points as zero within an arbitrary, but nonzero, distance from the poles.

## APPENDIX B - BEST VALUES FOR THE COEFFICIENTS OF $f^*$

For the form of the inverse mapping function,  $f^*$ , given in Equation (3-42) the best derived coefficients are

$\gamma^* = 1.3682$	6932	2362
$\Gamma = -0.1268$	8142	8338
$M = -0.0024$	3108	1471
$\Omega_1 = -0.1583$	3565	5509
$C_{00} = 0.1014$	5485	8689
$C_{10} = 0.0160$	1398	7763
$C_{01} = -0.0904$	3332	1481
$C_{11} = 0.0574$	8768	5173
$C_{20} = -0.0272$	7606	9952
$C_{02} = -0.0342$	1969	6290
$D_0 = 0.0292$	5271	6867
$D_1 = 0.0059$	2537	5028

## APPENDIX C - REFINEMENT OF THE CARTESIAN TRANSFORMATION

Section 3.3 of the text presented a numerically efficient and compact algorithm for employing the Cartesian transformation,  $f^*$ , defined in Section 2.2 to obtain the data base coordinates  $(x, y)$  of any datum point or pair of datum points on the (spherical) Earth's surface. This Appendix discusses the rationale for the new algorithms and provides a derivation of Equation (3-42).

### C.1 OVERVIEW

The algorithm presented in Reference 1 (Section 4, Appendix B) already provides the essential information needed to implement  $f^*$ , but this study has found that the algorithm may be made more efficient computationally and better suited to computer programming by the following modifications:

- Take advantage of the nondimensional analysis presented in Appendix A that removes the need to compute the three-dimensional spatial coordinates of datum points by relying on simple direction cosines  $(l, m, n)$  of the subsatellite point and on the longitudes and colatitudes of the datum points.
- Allow for the Earth's rotation by using a varying node (see Appendix A) in computing the direction cosines instead of performing a separate transformation.
- Reduce by about half the computations required in cases of nonzero scan angles by viewing the scan arc as a central angle  $\psi_0$  (see Appendix A) and computing the data base coordinates for points in symmetrically placed pairs instead of treating each datum point independently.
- Remove a step from the algorithm, saving computations as well as yielding desirable numerical properties (see Section 3.4), by

introducing a change of variable that leads to a new formulation of  $f^*$ . The results of this step are summarized in Equations (3-40), (3-41), and (3-42) of Section 3.3.1. The mathematical details are presented in Section C.2 below.

## C.2 REFORMULATION OF $f^*$

Table 3-2 (Section 3.3.1) establishes the conventions that define variables  $\xi'$ ,  $\eta'$ , and  $\rho$ . These variables also appear in Table 4-1 of Reference 1, differing only in that  $l$ ,  $m$ , and  $n$  are dimensionless (see Appendix A), whereas  $a$ ,  $b$ , and  $c$  (of Reference 1) are spatial coordinates. Mathematically,  $\xi'$ ,  $\eta'$ , and  $\rho$  behave identically here and in Reference 1.

In Reference 1, once  $\xi'$ ,  $\eta'$ ,  $\rho$  are found in Table 4-1, they are transformed to  $(\xi, \eta)$  by

$$\begin{aligned}\xi &= \xi' P \\ \eta &= \eta' P\end{aligned}\tag{C-1}$$

where  $P = r_o/\rho$  and  $r_o = 1/\sqrt{3}$ . The new variables  $\xi$  and  $\eta$  are then used in  $f^*$  to obtain the data base coordinates  $(x, y)$  by

$$\begin{aligned}x &= f^*(\xi, \eta) \\ y &= f^*(\eta, \xi)\end{aligned}\tag{C-2}$$

The transformation given by Equations (C-1) can be eliminated by defining new variables  $\alpha$  and  $\beta$  by

$$\begin{aligned}\alpha &= \xi'/\rho \\ \beta &= \eta'/\rho\end{aligned}\tag{C-3}$$

Comparison of Equations (C-1) and (C-3) shows that the new variables  $\alpha$  and  $\beta$  eliminate the computation of the intermediate quantity  $P$  and reduce the transformation to simple ratios of previously known quantities. The introduction of  $\alpha$  and  $\beta$  will now be completed by deriving  $f^*(\alpha, \beta)$ , the transformed equivalent of  $f^*(\xi, \eta)$ , which was given in Section 2.2 and Reference 1 as

$$f^*(\xi, \eta) = \gamma^* \xi + \frac{1 - \gamma^*}{r_o^2} \xi^3 + \xi \eta^2 (r_o^2 - \xi^2) \left[ \delta^* + \delta_1^* (r_o^2 - \xi^2) \right. \\ \left. + (r_o^2 - \eta^2) \sum_{\substack{i \geq 0 \\ j \geq 0}} c_{ij}^* \xi^{2i} \eta^{2j} \right] \\ + \xi^3 (r_o^2 - \xi^2) \left[ \omega^* + (r_o^2 - \xi^2) \sum_{i \geq 0} d_i^* \xi^{2i} \right]$$

where  $r_o = 1/\sqrt{3}$  and

$$\begin{aligned} \gamma^* &= \frac{1}{\gamma} & \delta^* &= \frac{(\mu_1^* - \mu^*)}{2r_o^4} \\ \mu^* &= \frac{1}{\mu} & \omega^* &= \frac{1}{2r_o^4} (3 - 2\gamma^* - \mu^* - 2r_o^4 \delta^*) \\ \gamma_1^* &= \frac{1}{\gamma + r_o^4 \delta} & & \\ \mu_1^* &= \frac{1}{\mu + 2r_o^4 \delta} & \delta_1^* &= \frac{1}{r_o^2} \left[ \frac{(\gamma_1^* - \gamma^*)}{r_o^4} - \delta^* \right] \end{aligned} \quad (C-5)$$



and  $\gamma$ ,  $\mu$ ,  $\delta$ ,  $\omega$ , and the  $c_{ij}^*$  and  $d_i^*$  have the numerical values given in Section 2.2.

First, Equation (C-4) is expressed in terms of  $\xi'$ ,  $\eta'$ , and  $\rho$  by using Equations (C-1)

$$\begin{aligned}
 f^*(\xi', \eta', \rho) = & \gamma^* r_o \xi' / \rho + \frac{1 - \gamma^*}{r_o^2} r_o^3 (\xi' / \rho)^3 \\
 & + r_o^3 (\xi' \eta'^2 / \rho^3) \left[ r_o^2 - r_o^2 (\xi' / \rho)^2 \right] \left\{ \delta^* + \delta_1^* \left[ r_o^2 - r_o^2 (\xi' / \rho)^2 \right] \right. \\
 & + \left. \left[ r_o^2 - r_o^2 (\eta' / \rho)^2 \right] \sum_{i \geq 0} c_{ij}^* r_o^{2i+2j} (\xi' / \rho)^{2i} (\eta' / \rho)^{2j} \right\} \quad (C-6) \\
 & + r_o^3 (\xi' / \rho)^3 \left[ r_o^2 - r_o^2 (\xi' / \rho)^2 \right] \\
 & \times \left\{ \omega^* + \left[ r_o^2 - r_o^2 (\xi' / \rho)^2 \right] \sum_{i \geq 0} d_i^* r_o^{2i} (\xi' / \rho)^{2i} \right\}
 \end{aligned}$$

Next,  $\alpha$  and  $\beta$  are introduced as defined in Equations (C-3), and powers of  $r_o$  are canceled in Equation (C-6), where appropriate, to obtain

$$\begin{aligned}
 f^*(\alpha, \beta) = & r_o \gamma^* \alpha + r_o (1 - \gamma^*) \alpha^3 \\
 & + \alpha \beta^2 (1 - \alpha^2) r_o^5 \left[ \delta^* + \delta_1^* r_o^2 (1 - \alpha^2) \right. \\
 & + \left. r_o^2 (1 - \beta^2) \sum_{i \geq 0} c_{ij}^* r_o^{2i+2j} \alpha^{2i} \beta^{2j} \right] \quad (C-7) \\
 & + \alpha^3 (1 - \alpha^2) r_o^5 \left[ \omega^* + r_o^2 (1 - \alpha^2) \sum_{i \geq 0} d_i^* r_o^{2i} \alpha^{2i} \right]
 \end{aligned}$$

Equation (C-7) may be simplified ultimately by introducing the definitions (Equations (C-5)) of  $\delta^*$ ,  $\omega^*$ , and  $\delta_1^*$  to obtain:

$$\begin{aligned}
 f^*(\alpha, \beta) = & r_o \gamma^* \alpha + r_o (1 - \gamma^*) \alpha^3 + \alpha \beta^2 (1 - \alpha^2) r_o^5 \\
 & \times \left[ \frac{\mu_1^* - \mu^*}{2r_o^4} + \frac{r_o^2}{r_o^2} \left( \frac{\gamma_1^* - \gamma^*}{r_o^4} - \frac{\mu_1^* - \mu^*}{2r_o^4} \right) (1 - \alpha^2) \right. \\
 & \left. + r_o^2 (1 - \beta^2) \sum_{i \geq 0} \sum_{j \geq 0} c_{ij}^* r_o^{2i+2j} \alpha^{2i} \beta^{2j} \right] \\
 & + \alpha^3 (1 - \alpha^2) r_o^5 \left\{ \frac{3 - 2\gamma^* - \mu^* - 2r_o^4 \left[ (\mu_1^* - \mu^*)/2r_o^4 \right]}{2r_o^4} \right. \\
 & \left. + r_o^2 (1 - \alpha^2) \sum_{i \geq 0} d_i^* r_o^{2i} \alpha^{2i} \right\}
 \end{aligned} \tag{C-8}$$

Equation (C-8) is simplified by factoring the expression by  $r_o$  and carrying  $r_o^4$  through the terms in box brackets to obtain

$$\begin{aligned}
 f^*(\alpha, \beta) = & r_o \left\{ \gamma^* \alpha + (1 - \gamma^*) \alpha^3 + \alpha \beta^2 (1 - \alpha^2) \left[ \frac{1}{2} (\mu_1^* - \mu^*) \right. \right. \\
 & \left. \left. + \left( \gamma_1^* - \gamma^* - \frac{1}{2} (\mu_1^* - \mu^*) \right) (1 - \alpha^2) + (1 - \beta^2) \sum_{i \geq 0} \sum_{j \geq 0} c_{ij}^* r_o^{2(i+j+3)} \alpha^{2i} \beta^{2j} \right] \right. \\
 & \left. + \alpha^3 (1 - \alpha^2) \left[ \frac{1}{2} (3 - 2\gamma^* - \mu^* + \mu^* - \mu_1^*) + (1 - \alpha^2) \right. \right. \\
 & \left. \left. \times \sum_{i \geq 0} d_i^* r_o^{2(i+3)} \alpha^{2i} \right] \right\}
 \end{aligned} \tag{C-9}$$

Three new constants,  $M$ ,  $\Gamma$ , and  $\Omega_1$ , are introduced to replace  $\delta^*$ ,  $\omega^*$ , and  $\delta_1^*$  through the following definitions

$$M = \frac{1}{2} (\mu_1^* - \mu^*)$$

$$\Gamma = \gamma_1^* - \gamma^* \quad (C-10)$$

$$\Omega_1 = \frac{1}{2} (3 - 2\gamma^* - \mu_1^*)$$

By comparing Equation (C-10) with Equation (C-5), it is seen that these new constants are considerably simpler in form.

Noting that  $r_o$  is strictly a constant,  $r_o = 1/\sqrt{3}$ , new coefficients  $C_{ij}$  and  $D_i$  are defined as follows

$$C_{ij} = c_{ij}^* r_o^{2(i+j+3)} \quad (C-11)$$

$$D_i = d_i^* r_o^{2(i+3)}$$

Using Equations (C-10) and (C-11) in (C-9), the following is obtained

$$\begin{aligned} f^*(\alpha, \beta) = r_o & \left\{ \gamma^* \alpha + (1 - \gamma^*) \alpha^3 + \alpha \beta^2 (1 - \alpha^2) \right. \\ & \times \left[ M + (\Gamma - M) (1 - \alpha^2) + (1 - \beta^2) \sum_{i \geq 0} \sum_{j \geq 0} C_{ij} \alpha^{2i} \beta^{2j} \right] \\ & \left. + \alpha^3 (1 - \alpha^2) \left[ \Omega_1 + (1 - \alpha^2) \sum_{i \geq 0} D_i \alpha^{2i} \right] \right\} \quad (C-12) \end{aligned}$$

This completes the derivation of the transformed expression for  $f^*$ . Equation (C-12) is given as Equation (3-42) in Section 3.3.1 and its numerical advantages are discussed in Section 3.5.

## APPENDIX D - APPROXIMATION OF THE DIRECT AND INVERSE TRANSFORMATIONS

This appendix presents tables of coefficients for the approximate forms of the transformations  $f$  and  $f^*$  as presented in Section 3.4. These tables are presented for one octant of one face. Each table contains 977 sets of coefficients (ordered by column), determined by using intersection points of a  $32 \times 32$  triangular grid. Each set of coefficients represents the approximate form of the transformation within one element of the  $32 \times 32$  grid. Each of these elements corresponds to one  $64 \times 64$  element data face record at the 1.2 nautical mile scale.

Figure 3-13 illustrates the table octant (octant 1) and the seven other octants. Table D-1 is used to transform from  $(x, y)$  to  $(\xi, \eta)$  and Table D-2 is used to transform from  $(\xi, \eta)$  to  $(x, y)$ . The steps outlined below are followed to transform from  $(x, y)$  to  $(\xi, \eta)$ :

1. Add 2048.0 to  $x$  if  $x$  is less than 2048.0
2. Add 2048.0 to  $y$  if  $y$  is less than 2048.0
3. If  $y$  is greater than  $x$ , interchange  $x$  and  $y$  values
4. Truncate the  $x$  and  $y$  values obtained, divide each by 32, and add one, use these integers and IX and IY indices and look up the coefficients and number of terms in the table
5. Evaluate  $f_a(x, y)$  to obtain  $\xi$
6. Reverse steps 3, 2, and 1 to place  $\xi$  in its proper octant
7. Interchange  $x$  and  $y$  and repeat steps 2 through 7 to obtain  $\eta$

Alternatively, steps 1, 2, 3, and 7 may be avoided by constructing a full  $64 \times 64$  table containing redundant data.

COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION

Table D-1 (1 of 7)

IX	X <sub>0</sub>	Y <sub>0</sub>	NTERMS	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	RESID.
1	1.048+00	2.048+00	3	0.72479E	30-0.56604E-03	0.505307E	0.3	0.0	0.41504E-02
2	1.114+00	2.154+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.10762E-01
3	1.172+00	2.262+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.11719E-01
4	1.232+00	2.372+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.12719E-01
5	1.292+00	2.482+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.13719E-01
6	1.352+00	2.592+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.14719E-01
7	1.412+00	2.702+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.15719E-01
8	1.472+00	2.812+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.16719E-01
9	1.532+00	2.922+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.17719E-01
10	1.592+00	3.032+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.18719E-01
11	1.652+00	3.142+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.19719E-01
12	1.712+00	3.252+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.20719E-01
13	1.772+00	3.362+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.21719E-01
14	1.832+00	3.472+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.22719E-01
15	1.892+00	3.582+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.23719E-01
16	1.952+00	3.692+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.24719E-01
17	2.012+00	3.802+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.25719E-01
18	2.072+00	3.912+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.26719E-01
19	2.132+00	4.022+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.27719E-01
20	2.192+00	4.132+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.28719E-01
21	2.252+00	4.242+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.29719E-01
22	2.312+00	4.352+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.30719E-01
23	2.372+00	4.462+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.31719E-01
24	2.432+00	4.572+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.32719E-01
25	2.492+00	4.682+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.33719E-01
26	2.552+00	4.792+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.34719E-01
27	2.612+00	4.902+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.35719E-01
28	2.672+00	5.012+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.36719E-01
29	2.732+00	5.122+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.37719E-01
30	2.792+00	5.232+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.38719E-01
31	2.852+00	5.342+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.39719E-01
32	2.912+00	5.452+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.40719E-01
33	2.972+00	5.562+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.41719E-01
34	3.032+00	5.672+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.42719E-01
35	3.092+00	5.782+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.43719E-01
36	3.152+00	5.892+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.44719E-01
37	3.212+00	6.002+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.45719E-01
38	3.272+00	6.112+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.46719E-01
39	3.332+00	6.222+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.47719E-01
40	3.392+00	6.332+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.48719E-01
41	3.452+00	6.442+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.49719E-01
42	3.512+00	6.552+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.50719E-01
43	3.572+00	6.662+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.51719E-01
44	3.632+00	6.772+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.52719E-01
45	3.692+00	6.882+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.53719E-01
46	3.752+00	6.992+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.54719E-01
47	3.812+00	7.102+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.55719E-01
48	3.872+00	7.212+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.56719E-01
49	3.932+00	7.322+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.57719E-01
50	3.992+00	7.432+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.58719E-01
51	4.052+00	7.542+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.59719E-01
52	4.112+00	7.652+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.60719E-01
53	4.172+00	7.762+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.61719E-01
54	4.232+00	7.872+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.62719E-01
55	4.292+00	7.982+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.63719E-01
56	4.352+00	8.092+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.64719E-01
57	4.412+00	8.202+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.65719E-01
58	4.472+00	8.312+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.66719E-01
59	4.532+00	8.422+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.67719E-01
60	4.592+00	8.532+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.68719E-01
61	4.652+00	8.642+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.69719E-01
62	4.712+00	8.752+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.70719E-01
63	4.772+00	8.862+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.71719E-01
64	4.832+00	8.972+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.72719E-01
65	4.892+00	9.082+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.73719E-01
66	4.952+00	9.192+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.74719E-01
67	5.012+00	9.302+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.75719E-01
68	5.072+00	9.412+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.76719E-01
69	5.132+00	9.522+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.77719E-01
70	5.192+00	9.632+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.78719E-01
71	5.252+00	9.742+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.79719E-01
72	5.312+00	9.852+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.80719E-01
73	5.372+00	9.962+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.81719E-01
74	5.432+00	10.072+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.82719E-01
75	5.492+00	10.182+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.83719E-01
76	5.552+00	10.292+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.84719E-01
77	5.612+00	10.402+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.85719E-01
78	5.672+00	10.512+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.86719E-01
79	5.732+00	10.622+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.87719E-01
80	5.792+00	10.732+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.88719E-01
81	5.852+00	10.842+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.89719E-01
82	5.912+00	10.952+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.90719E-01
83	5.972+00	11.062+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.91719E-01
84	6.032+00	11.172+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.92719E-01
85	6.092+00	11.282+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.93719E-01
86	6.152+00	11.392+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.94719E-01
87	6.212+00	11.502+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.95719E-01
88	6.272+00	11.612+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.96719E-01
89	6.332+00	11.722+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.97719E-01
90	6.392+00	11.832+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.98719E-01
91	6.452+00	11.942+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	0.99719E-01
92	6.512+00	12.052+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	1.00719E-01
93	6.572+00	12.162+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	1.01719E-01
94	6.632+00	12.272+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	1.02719E-01
95	6.692+00	12.382+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	1.03719E-01
96	6.752+00	12.492+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	1.04719E-01
97	6.812+00	12.602+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	1.05719E-01
98	6.872+00	12.712+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	1.06719E-01
99	6.932+00	12.822+00	3	0.72479E	00-0.14481E-04	0.505307E	0.3	0.0	1.07719E-01
100	6.992+00	12.932+00	3	0.72479E					

[illegible]

**D-3**

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**

D-4



**Table D-1 (4 of 7)**

[illegible]

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**

Table D-1 (5 of 7)

[illegible]

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**

**D-7**

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**

IEF142I - STEP WAS EXECUTED - COND CODE 0000  
IEF205I SYSL DUMMY  
IEF285I VOL SER NOS= M2DPM1.

**Text**

Table D-2 (1 of 7)

[illegible]

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**



The image shows a document page that is severely degraded, likely due to age, damage, or poor scanning quality. The page is characterized by extreme vertical banding and horizontal streaking, which obscures most of the text. Faint, illegible markings are visible throughout the page, including what appears to be a header section at the top and a footer section at the bottom. The central area is dominated by vertical lines and horizontal noise, making the content unreadable.

**D-10**



**Table D-2 (4 of 7)**

[illegible]

**COPY AVAILABLE TO DOD DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**



Table D-2 (5 of 7)

[illegible]

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**

[illegible]

**D-14**

Table D-2 (7 of 7)

161121	1	YES	NO	NO	NO	NO	NO	161121	1	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
--------	---	-----	----	----	----	----	----	--------	---	-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**

In Tables D-1 and D-2, IX and IY (IP, IE) are the x and y ( $\xi$ ,  $\eta$ ) indices of the table,  $x_0$  and  $y_0$  ( $\xi_0$ ,  $\eta_0$ ) are the coordinates of the origin of the area of definition for the coefficient set, NTERMS is the number of terms to be evaluated,  $a_0 \dots a_5$  are the coefficients, and RESID is the largest residual found in the evaluation of the coefficients (in elements) as illustrated in Figure 3-12.

## APPENDIX E - LISTINGS OF TEST AND EVALUATION PROGRAMS

### E.1 SUMMARY

This appendix contains listings of FORTRAN programs used for testing and evaluating the transformations in this report and plotter programs used for generating the world maps.

<u>Listing</u>	<u>Figure</u>
Program used for evaluating the coefficients of $f^*(\xi, \eta)$	E-1
Program used for verifying the $f^*$ transformations	E-2
Program used for verifying the polar transformation	E-3
Program used for evaluating coefficients of approximation polynomials	E-4
Program used for generating world maps	E-5
Alternative subroutines for E-5 using the polar transformation	E-6

THIS PROGRAM EVALUATES THE COEFFICIENTS OF THE INVERSE  
TRANSFORMATION F STAR BY FORCING F STAR TO MAP POINTS  
TRANSFORMED BY THE DIRECT TRANSFORMATION BACK ONTO THEMSELVES

PROGRAMMER - E. MICHAEL D'NEILL  
COMPUTER SCIENCES CORPORATION  
8728 COLESVILLE RD. SILVER SPRING, MD.

```

IMPLICIT REAL*8 (A-H,O-Z)
EXTERNAL FUNK
COMMON RSPH,R0,R02
COMMON/ERR00/ NEMAX,NELAT,IVARY,NEXTRA
COMMON /ATLR0/ DIFF,VAR(20),VMIN(20),VMAX(20),NV(20),
1 DELMIN(20),ERR(20,20),NV,NT,MAT,MASK(20)
EQUIVALENCE (GAMMAS,VAR(12)), (AMUIS,VAR(13)),
1 (GAMIS,VAR(14)), (AMUIS,VAR(15)),
1 (FM,VAR(16)), (CGAM,VAR(17)), (CMG1,VAR(18))
DATA GAMMA,DELTA,AMU/0.723601254500,0.790486449100,
1.64945416600/
RSPH=1.000
R0=RSPH/DSORT(3.000)
R02=R0*R0
R04=R02*R02
GAMMAS=1.000/GAMMA
AMUIS=1.000/AMU
GAMIS=1.000/(GAMMA+R04*DELTA)
AMUIS=1.000/(AMU+2.000*R04*DELTA)
FM=0.500*(AMUIS-AMUS)
CGAM=GAMIS-GAMMAS
CMG1=0.500*(3.000-2.000*GAMMAS-AMUIS)
DO 10 I=1,11
VAR(I)=0.000
VMIN(I)=-2.500
VMAX(I)=2.500
NV(I)=1.00-1
DELMIN(I)=0.10-10
10 MASK(I)=0
DO 110 I=12,20
VMIN(I)=-100.000
VMAX(I)=100.000
DELMIN(I)=0.10-10
110 MASK(I)=0
MASK(1)=0
MASK(2)=0
MASK(3)=0
MASK(4)=0
MASK(5)=0
MASK(6)=0
MASK(7)=1
MASK(8)=1
MASK(9)=0
MASK(10)=0
MASK(11)=1
MASK(12)=0
MASK(13)=0
MASK(14)=0
MASK(15)=0
MASK(16)=0
MASK(17)=0
MASK(18)=0
NV(12)=0.00100*GAMMAS
NV(13)=0.00100*AMUIS
NV(14)=0.00100*GAMIS
NV(15)=0.00100*AMUIS
NV(16)=0.00100*FM
NV(17)=0.00100*CGAM
NV(18)=0.00100*CMG1
NEMAX=2000
NT=0
MAT=0
M=1
N=4
CALL STEPIT(FUNK)
CALL DISP(N)
STOP
END

```

Figure E-1 (1 of 3)

```

SUBROUTINE FUNK
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/RI1A0/ DIFF,VAR(20)
COMMON RSPW,R0,R02
XIN=R0/16.000
VIN=XIN
DIFF=0.000
X=0.000
DO 10 IX=1,17
V=0.000
DO 5 IV=1,IX
P=FMAP(X,V)
F=FMAP(V,X)
XR=FMAP2(P,F)
YR=FMAP2(F,P)
DIFF=DIFF+(X-XR)**4+(Y-YR)**4
5 V=V+VIN
10 X=X+VIN
RETURN
END
FUNCTION FMAP(A,R)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON RSPW,R0,R02
DATA GAMMA,OMEGA/0.72360125455R200,0.276398745441800/
DATA DELTA,OMEGA/0.790484449120000,-0.122544148798401/
DATA C00,C10,C01/-0.27217053601,-0.55842168301,0.21711172801/
DATA C11,C20,C02/-0.64160151501,-0.34578627401,0.14736265701/
DATA D0,D1,D2/0.14833129201,0.1199776001,0.60515382101/
A50=A*A
A3=A*A*A
A4=A50*A50
R50=R*R
R4=R50*R50
RMA=R02-A50
FMAP=GAMMA*A+OMEGA/R02*A3+
1 A3*RMA*(OMEGA+RMA*(D0+D1)*A50+D2*A4))+
2 A*R50*RMA*(DELTA+(R02-A50)*
3 (C00+C10*A50+C01*R50+C11)*A50+R50+C20*A4+C02*R4))
RETURN
END

```

Figure E-1 (2 of 3)

THIS PROGRAM VERIFIES THE CORRECTNESS OF THE INVERSE  
TRANSFORMATION BY COMPARING POSITIONS ON A TRANSFORMED AND  
BETRANSFORMED GRID

PROGRAMMER - E. MICHAEL O'NEILL  
COMPUTER SCIENCES CORPORATION  
8728 COLFESVILLE RD. SILVER SPRING, MD.

```

IMPLICIT REAL*8 (A-H,O-Z)
COMMON RCDM,RN,RDZ
RCDM=1.000
RDZ=RCDM/DSORT(3.000)
RDZ=RN*RDZ
DMAX=-1.000
X=0.000
ISKIP=1A
SKIP=ISKIP
DO 100 IX=1,2049,ISKIP
V=0.000
DO 50 IV=1,2049,ISKIP
XX=RDZ*Y/2048.000
YY=RDZ*Y/2048.000
P=FMAP(X,XX)
F=FMAP(Y,XX)
XR=(FMAP2(P,F)*2048.000/RDZ)
YR=(FMAP2(F,P)*2048.000/RDZ)
DX=X-XR
DY=Y-YR
D=DSORT(DX**2+DY**2)
IF(D,IT,DMAX) GO TO 50
WRITE(6,10) X,Y,P,F,XR,YR,DX,DY,D
10 ENOMAT(1,X,Y,P,F,XR,YR,DX,DY,D,9D12.5)
WRITE(1,15) XR,YR,D
15 ENOMAT(1,X,15.5,A)
DMAX=D
50 Y=Y+SKIP
100 X=X+SKIP
STOP
END
FUNCTION FMAP2(AA,RR)
IMPLICIT REAL*8 (A-H,O-Z)
DATA R0NT3/0.173205080801/,R0/0.57735026900/
DATA GAMMA5/0.13748484773201/,CGAM/-0.13161671474000/
DATA CMG1/-0.15959623567400/
DATA C00/0.14118963115200/,C10/0.8097012865250-1/
DATA C01/-0.28152853555700/,C11/0.15384112876700/
DATA C20/-0.17825120746600/,C02/0.10695946931400/
DATA C03/0.7591962004670-1/,C12/0.2177624906090-1/
DATA CMG4/-0.37484847732000/
A=AA*0.0001
R=RR*0.0001
A2=A*A
A3=A*A2
A4=A2*A2
R2=R*R
R4=R2*R2
DMA2=1.000-A2
FMAP2=R0*(A*GAMMA5+A3*CMGMS+A*R2*DMA2*(CGAM+A2*(EM-CGAM)+(1.00-R2)
* (C00+C10*A2+C01*R2+C11*A2*R2+C20*A4+C02*R4)
+A3*DMA2*(CMG1-DMA2*(0.0+0.1*A2)))
RETURN
END
FUNCTION FDMAP(A,R)
IMPLICIT REAL*8 (A-H,O-Z)
DATA R50/0.33333333333300/
DATA GAMMA,DMGAM/0.723601254558200,0.276398745441800/
DATA DELTA,DMEGA/0.790484460120800,-0.122544148798401/
DATA C00,C10,C01/-0.27217053601,-0.55842168301,0.21111174801/
DATA C11,C20,C02/-0.64160151501,-0.34578627401,0.19738265701/
DATA D0,D1,D2/0.14833129201,0.11199726001,0.60515382101/
A50=A*A
A2=A*A50
A4=A50*A50
R50=R*R
R4=R50*R50
DMA=A50-A50
FDMAP=GAMMA*A+DMGAM/R50*A3+
1 A3*DMA*(DMEGA+DMA*(0.0+0.1*A50+0.2*A4))+
2 A50*DMA*(DELTA+R50-R50)*
3 (C00+C10*A50+C01*R50+C11*A50+C20*A4+C02*R4)
RETURN
END

```

Figure E-2





```

C      SIMROUTINE MAP(X,Y,XS,YS)
C
C      PROGRAMMER - F. MICHAEL D'NEILL
C                  COMPUTER SCIENCES CORPORATION
C                  8728 COLESVILLE RD. SILVER SPRING, MD.
C
      IMPLICIT REAL*8 (D-H,O-Z)
      DATA PI/3.141592653589793/
      DATA PI2/0.15707963267948966/
      DATA PI4/0.7853981633974483/
      DATA R0/0.5773502691896387/
      DATA SCALE/0.3819718634205501/
      IF(Y.EQ.0.0.AND.Y.EQ.0.0) GO TO 100
      XX=X
      YY=Y
      VV=Y
      IVV=0
      IF(DARS(X).GE.DARS(Y)) GO TO 20
      IVV=1
      VV=VV
      VV=Y
20  CONTINUE
      PHI=DATAN(DSORT(X*X+Y*Y)/R0)
      THETA=DATAN(DARS(YY)/DARS(XX))
      AMU=DATAN(SCALE*(THETA+DARCOS(DSIN(THETA)*DCOS(PI4))-PI2))
      C=1.000/((1.000-DCOS(DATAN(1.000/DCOS(THETA))))*DCOS(AMU)**2)
      R=DCOS(DSORT(C*(1.000-DCOS(PHI))))
      IF(IVV.EQ.0) GO TO 30
      XS=DSIGN(R*DCOS(AMU),X)
      YS=DSIGN(R*DCOS(AMU),Y)
      RETURN
30  CONTINUE
      XS=DSIGN(R*DCOS(AMU),X)
      YS=DSIGN(R*DCOS(AMU),Y)
      RETURN
100 XS=X
      YS=Y
      RETURN
      END

```

Figure E-3 (2 of 2)

THIS PROGRAM USES THE IBM SSP ROUTINE SIMO TO SOLVE A SYSTEM  
OF SIMULTANEOUS EQUATIONS TO OBTAIN COEFFICIENTS FOR THE  
APPROXIMATE TRANSFORMATION FA

FUNCTION TMAP MAY BE MODIFIED TO CALL ANY DESIRED TRANSFORMATION  
E, F STAP, OR THE POLAR TRANSFORMATION SUBROUTINE MAP

PROGRAMMER - F. MICHAEL DINEILL  
COMPUTER SCIENCES CORPORATION  
8728 COLESVILLE RD. SILVER SPRING, MD.

```

DIMENSION RVECT(5)
DATA NOUT/6/, NPIC/512/
READ(5,10) NOUT,NPIC,MAXD
10 CONTINUE
DATA TOLER/1.0/
INTER=2048/NPIC
DY=NPIC
DX=NY
Y=2048.0
DO 100 IX=1,INTER
  Y=2048.0
  DO 50 IY=1,IX
    CALL DEG3(NC,RVECT,X,Y,DX,DY,TOLER,RESID)
    IF(NC.EQ.3.OR.MAXD.LE.3) GO TO 30
    CALL DEG2(NC,RVECT,X,Y,DX,DY,TOLER,RESID)
    IF(NC.EQ.4.OR.MAXD.LE.4) GO TO 30
    CALL DEG5(NC,RVECT,X,Y,DX,DY,TOLER,RESID)
  20 WRITE(NOUT,40) IX,IY,X,Y,NC,RVECT,RESID
  40 FORMAT(1X,2I5,2F8.2,15,2X,5F12.5,3X,F12.5)
  50 Y=Y+DY
100 Y=Y+DX
STOP
END
SUBROUTINE DEG3(NC,RVECT,X,Y,DX,DY,TOLER,RESID)
  DIMENSION RVECT(5), AVECT(25)
  DIMENSION XP(3),YP(3)
  DATA YP/0.0,1.0,0.0/, YP/0.0,0.0,1.0/
  NC=3
  DO 10 I=1,NC
    XY=Y+DX*XP(I)
    YY=Y+DY*YP(I)
    RVECT(I)=TMAP(XY,YY)
    AVECT(I)=XX
    AVECT(I+3)=YY
    AVECT(I+6)=1.0
  10 CONTINUE
  CALL SIMO(AVECT,RVECT,NC,K5)
  IF(K5.EQ.1) NC=1
  CALL DEG2(X,Y,DX,DY,RVECT,TOLER,RESID,KV)
  IF(KV.EQ.1) NC=0
  RVECT(4)=0.0
  RVECT(5)=0.0
  RETURN
END
SUBROUTINE DEG2(X,Y,DX,DY,RVECT,TOLER,RESID,KV)
  DIMENSION YP(4),XP(4),AVECT(5)
  DATA YP/0.25,0.75,0.75,0.25/, YP/0.25,0.25,0.75,0.75/
  NV=4
  KV=0
  RESID=0.0
  DO 10 I=1,NV
    XY=Y+DX*XP(I)
    YY=Y+DY*YP(I)
    XA=XY+RVECT(1)+YY*RVECT(2)+RVECT(3)
    XT=TMAP(XY,YY)
    RESIT=ABS(XA-XT)
    IF(RESIT.GT.RESID) RESID=RESIT
  10 CONTINUE
  IF(RESID.GT.TOLER) KV=1
  RETURN
END

```

Figure E-4 (1 of 3)

```

SUBROUTINE DEG4(NG,AVECT,X,Y,DX,DY,TOLER,RESID)
DIMENSION AVECT(5),AVECT(25)
DIMENSION XP(4),YP(4)
DATA XP/0.0,0.1,0.1,0.0,0.0/, YP/0.0,0.0,1.0,1.0/
NC=4
DO 10 I=1,NC
  YV=Y+DY*XP(I)
  XV=X+DX*YP(I)
  AVECT(I)=TMAP(XV,YV)
  AVECT(I+5)=XV*YV
  AVECT(I+4)=XV
  AVECT(I+8)=YV
  AVECT(I+12)=1.0
10 CONTINUE
CALL SING(AVECT,AVECT,NC,KS)
IF(KS,EO.1) NC=-1
CALL VER4(X,Y,DX,DY,AVECT,TOLER,RESID,KV)
IF(KV,EO.1) NC=0
AVECT(4)=0.0
RETURN
END
SUBROUTINE VER4(X,Y,DX,DY,AVECT,TOLER,RESID,KV)
DIMENSION XP(4),YP(4),AVECT(5)
DATA XP/0.25,0.75,0.75,0.25/, YP/0.25,0.25,0.75,0.75/
NV=4
KV=0
RESID=0.0
DO 10 I=1,NV
  YV=Y+DY*XP(I)
  XV=X+DX*YP(I)
  X=XX+YV*AVECT(1)+XV*AVECT(2)+YV*AVECT(3)+AVECT(4)
  Y=TMAP(XV,YV)
  RESIT=ARS(XA-XV)
  IF(RESIT.GT.RESID) RESID=RESIT
10 CONTINUE
IF(RESID.GT.TOLER) KV=1
RETURN
END
SUBROUTINE DEG5(NG,AVECT,X,Y,DX,DY,TOLER,RESID)
DIMENSION AVECT(5),AVECT(25)
DIMENSION XP(5),YP(5)
DATA XP/0.0,0.1,0.1,0.0,0.5/, YP/0.0,0.0,1.0,1.0,0.5/
NC=5
DO 10 I=1,NC
  YV=Y+DY*XP(I)
  XV=X+DX*YP(I)
  AVECT(I)=TMAP(XV,YV)
  AVECT(I+5)=XV*XX
  AVECT(I+5)=YV*YV
  AVECT(I+10)=XV
  AVECT(I+15)=YV
  AVECT(I+20)=1.0
10 CONTINUE
CALL SING(AVECT,AVECT,NC,KS)
IF(KS,EO.1) NC=-1
CALL VER5(X,Y,DX,DY,AVECT,TOLER,RESID,KV)
IF(KV,EO.1) NC=0
RETURN
END
SUBROUTINE VER5(X,Y,DX,DY,AVECT,TOLER,RESID,KV)
DIMENSION XP(5),YP(5),AVECT(5)
DATA XP/0.25,0.75,0.75,0.25/, YP/0.25,0.25,0.75,0.75/
NV=5
KV=0
RESID=0.0
DO 10 I=1,NV
  YV=Y+DY*XP(I)
  XV=X+DX*YP(I)
  X=XX+YV*AVECT(1)+YV*YV*AVECT(2)+XV*AVECT(3)+YV*AVECT(4)+AVECT(5)
  Y=TMAP(XV,YV)
  RESIT=ARS(XA-XV)
  IF(RESIT.GT.RESID) RESID=RESIT
10 CONTINUE
IF(RESID.GT.TOLER) KV=1
RETURN
END

```

Figure E-4 (2 of 3)

```

FUNCTION THAP(A,R)
REAL*8 ENAP, P, E
D=(A-2048.0)*0.577350/2048.0
E=(R-2048.0)*0.577350/2048.0
THAP=ENAP(P,E)
THAP=THAP*2048.0/0.577350*2048.0
RETURN
END
FUNCTION ENAP(A,R)
C THIS IS ENMAP V1
IMPLICIT REAL*8 (A-H,O-Z)
DATA PCSO/0.3333333333333333/
DATA GAMMA,OMGAM/0.723601254558200,0.276398745441800/
DATA DELTA,OMEGA/0.790486449120800,-0.122544148788401/
DATA COS,C10,C01/-0.27217053601,-0.55842168301,0.21711174801/
DATA C11,C20,C02/-0.64160151501,-0.34578627401,0.19736285701/
DATA C00,C12,C21/0.14833129201,0.1199726001,0.60415382101/
ASC=AAA
A3=AAASO
A4=ASO*ASO
RCSO=RSSO
R4=RSO*RSO
R1A=PCSO-ASC
ENAP=CA*H1A*AA+OMGAM/RSO*A3+
1 A3*R1A*(OMEGA+RMA*(C00+C11*ASO+C21*A4))+
2 A3*RSO*R1A*(DELTA+(RSO-RCSO)*
(C00+C10*ASO+C01)*RSO+C11)*ASO*ASO+C20*A4+C02*A4))
RETURN
END

```

Figure E-4 (3 of 3)

00000000

THIS PROGRAM PRODUCES WORLD MAPS IN DATA BASE COORDINATES  
USING THE WOLFPLOT PACKAGE AND CONTINENTAL OUTLINE DATASET

PROGRAMMER - E. MICHAEL O'NEILL  
COMPUTER SCIENCES CORPORATION  
8728 COLESVILLE RD. SILVER SPRING, MD.

```

DIMENSION THETA(3)
CALL CALSI7(18.0)
CALL PLOTST(2001,4)
CALL SETGRD(0.0,0.0,16.0,10.0,4)
CALL SCALE(-1.0,15.0,-0.5,9.5,0)
1 READ(5,5,FMT=999) THETA
5 FORMAT(3F10.3)
WRITE(6,10) THETA
10 FORMAT(1 ROTATION ANGLES (ORDERED 3 2 )) ' ,3F12.2)
DO 15 I=1,3
15 THETA(I)=THETA(I)/57.3
CALL WORLD(THETA)
CALL FPMADV
GO TO 1
999 CONTINUE
STOP
END
SUBROUTINE WORLD(THETA)
INTEGER*2 IX,IY
DIMENSION COORD(3)
DIMENSION THETA(3)
DIMENSION IX(48),IY(48),X(48),Y(48),PP(48),EE(48),IFACE(48)
DIMENSION XO(6),YO(6)
DIMENSION XL(12),YL(12)
DATA XI/0.0,8.0,8.0,0.0,0.0,2.0,2.0,4.0,4.0,2.0,6.0,6.0,0/
DATA YI/2.0,2.0,4.0,4.0,2.0,0.0,6.0,6.0,0.0,0.0,2.0,4.0,0/
DATA XO/3.0,7.0,5.0,1.0,3.0,3.0/
DATA YO/3.0,3.0,3.0,3.0,5.0,1.0/
DATA PP,EE /96*0.0/
READ(1) FACTOR
FACTOR=FACTOR/57.295
CALL PLOT(XL(1),YL(1),5,0)
CALL PLOT(XL(6),YL(6),5,0)
CALL PLOT(XL(11),YL(11),2,0)
1 IFACE=0
I=1
READ(1,FMT=999) N, (IY(J),IX(J),J=1,N)
DO 10 J=1,N
A=IY(J)*FACTOR
R=IX(J)*FACTOR
COORD(1)=COS(R)*COS(A)
COORD(2)=COS(R)*SIN(A)
COORD(3)=SIN(R)
CALL ROT(COORD,THETA)
CALL OTRAN(COORD,PP(J),EE(J),IFACE)
IF(IFACE.EQ.LFACE) GO TO 5
LFACE=IFACE
IF(J.EQ.1) GO TO 5
NN=J-1
CALL PLOT(PP(1),EE(1),NN,0)
I=I
5 PP(J)=PP(J)+XO(IFACE)
10 EE(J)=EE(J)+YO(IFACE)
NN=NN+1
CALL PLOT(PP(1),EE(1),NN,0)
101 GO TO 1
999 CONTINUE
REWIND 1
RETURN
END

```

Figure E-5 (1 of 2)

```

SUBROUTINE ROT(COORD,THETA)
  THIS SUBROUTINE ROTATES ONE DATUM POINT ABOUT THREE AXES
  DIMENSION COORD(3), THETA(3), JAX(3,3)
  DATA JAX/3,1,2,2,1,3,1,2,3/
  DO 20 IAXIS=1,3
    X=COORD(IAXIS,1)
    IF(X.EQ.0.0) X=1.0E-6
    Y=COORD(IAXIS,2)
    IF(Y.EQ.0.0) Y=1.0E-6
    PHI=ATAN2(Y,X)
    PHI=PHI+THETA(IAXIS)
    R=SQRT(X**2+Y**2)
    COORD(IAXIS,1)=R*COS(PHI)
    COORD(IAXIS,2)=R*SIN(PHI)
  20 CONTINUE
  RETURN
END

SUBROUTINE OTRANS(COORD,X,Y,IFACE)
  THIS SUBROUTINE TAKES AS INPUT THE CARTESIAN COORDINATE ARRAY
  (COORD) AND RETURNS DATA BASE COORDINATES (X AND Y) AND THE
  FACE NUMBER (IFACE) OF THE X,Y SYSTEM

  PROGRAMMER - E. MICHAEL O'NEILL
  COMPUTER SCIENCES CORPORATION
  8728 COLESVILLE RD. SILVER SPRING, MD.

  COMMON /TRACOM/ RS,RO,RSO,R4,GAMMA,OMGAM,DELTA,OMEGA,
  1 COO,C10,C01,C11,C20,C02,DO,D1,D2
  DIMENSION COORD(3), C(3), ROT(3,6), IROT(3,6)
  DATA ROT/1,0,1,0,1,0,-1,0,1,0,-1,0,-1,0,1,0,1,0,
  1 1,0,1,0,-1,0,1,0,-1,0,1,0,1,0,1,0,-1,0/
  DATA IROT/2,3,1,2,3,1,1,3,2,1,3,2,2,1,3,2,1,3/
  DETERMINE FACE NUMBER BY FINDING LARGEST (ABS) COMPONENT
  CMAX=ABS(COORD(1))
  J=1
  DO 10 I=2,3
    IF(CMAX.LT.(ABS(COORD(I)))) J=I
  CMAX=ABS(COORD(J))
  10 CONTINUE
  IFACE=J+2
  IF(COORD(1).GE.0.0) IFACE=IFACE-1
  INTERCHANGE CARTESIAN COMPONENTS TO GET TO FACE SYSTEM
  DO 20 I=1,3
    20 C(I)=COORD(IROT(I,IFACE))*ROT(I,IFACE)
    DO 100 I=1,3
      100 IFCT TO PSI, ETA SYSTEM
      PP=0.7/C(I)
      P=C(I)*PP
      F=C(I)*PP
      TRANSFORM TO X,Y
      X=FMAP(P,E)/RO
      Y=FMAP(F,B)/RO
    RETURN
  END
  FUNCTION FMAP(AA,RR)
  THIS IS FUNCTION FMAP2
  PROGRAMMER - E. MICHAEL O'NEILL
  COMPUTER SCIENCES CORPORATION
  8728 COLESVILLE RD. SILVER SPRING, MD.

  IMPLICIT REAL *R (A-H,O-Z)
  DATA R0NT3/0.17320508000/, R0/0.57735026900/
  DATA GAMMAS/0.13748484773201/,
  DATA FM/0.4869401981000-2/, CGAM/-0.13161671474000/
  DATA CMG1/-0.15959623547400/
  DATA COO/0.14118963115200/, C10/0.8097012865250-1/
  DATA C01/-0.28152853555700/, C11/0.15384112876700/
  DATA C20/-0.17825120746600/, C02/0.10695946931400/
  DATA DO/0.7591962004670-1/, D1/-0.2177624906990-1/
  DATA OMGAS/-0.37484847732000/
  A=AA*DONT3
  R=RR*R0NT3
  A2=A*A
  A3=A*A2
  A4=A2*A2
  R2=R*R
  R4=R2*R2
  P1A2=1.000-A2
  F1A2=R0*(A*GAMMAS+A3*OMGAS+A2*OMGA2*(CGAM+A2*(FM-CGAM)+(1.00-R2)
  * (C00+C10*A2+C01*R2+C11*A2*R2+C20*A4+C02*R4))
  * +A3*OMGA2*(CMG1-OMGA2*(DO+D1*A2)))
  RETURN
  END

```

Figure E-5 (2 of 2)





## REFERENCES

1. Computer Sciences Corporation, TR-75/6007, Feasability Study of a Quadrilateralized Spherical Cube Earth Data Base, (prepared for the Environmental Prediction Research Facility, Monterey, California), F. K. Chan, E. M. O'Neill, March 1975
2. Computer Sciences Corporation, 6002-1, Organizational Structures for Constant Resolution Earth Data Bases, (prepared for the Environmental Prediction Research Facility, Monterey, California), P. R. Beaudet, F. K. Chan, L. Goldshlak, November 1973
3. Naval Environmental Prediction Research Facility, Investigation of the Satellite Data Preprocessor Function on a CDC CYBER/175 with MAP III Special Function Processor, W. C. Ragsdale, 1975
4. National Bureau of Standards, AMS-55, Handbook of Mathematical Functions, Ed. M. Abramowitz, I. A. Stegun, May 1968
5. IBM Corporation, GH-20-0205-4, System/360 Scientific Subroutine Package, Version III, Programmers Manual, August 1970