NASA Contractor Report 163102

(NASA-CR-163102) BASELINE MATHEMATICS AND GEODETICS FOR TRACKING OPERATIONS (GMD Systems) 225 p HC A10/MF A01 CSCL 12A

N81-22453

Unclas
G3/43 42156

BASELINE MATHEMATICS AND GEODETICS FOR TRACKING OPERATIONS

Robert James

Contract NAS4-2678
April 1981

NASA

NASA Contractor Report 163102

BASELINE MATHEMATICS AND GEODETICS FOR TRACKING OPERATIONS

Robert James
GMD Systems
Lancaster, California

Prepared for
Dryden Flight Research Center
under Contract NAS4-2678

National Aeronautics and
Space Administration
1981

PREFACE

The baseline geodetic algorithms presented in this document were compiled to provide a standard for accuracy measurement and for isolation of errors in radar and optical tracking devices. In implementing the algorithms, the first emphasis was placed on accuracy and the second on speed. In all cases, the programs yield accuracies that are at least two orders of magnitude better than the LSB values associated with existing radar tracking equipment. Most of the algorithms employ mathematical techniques well known in the field of geodetics. However, in certain cases, special high-accuracy algorithms, previously developed by GMD Systems, were used because the normal solution techniques were unable to achieve the results desired. For example, the refraction correction method provided in chapter 6 uses a GMD-developed algorithm in which the refraction gradient is used to compute the amount of bending which occurs in each incremental projection of a wave front traveling through a refracting medium. The program operates on a desktop (12-digit) computer and, in a couple of hundred iterations, yields results that are almost identical to those obtained from the JSC double-precision (29-digit) Cyber program, which requires up to 50,000 iterations for low-altitude solutions.

Another GMD algorithm was used to obtain a faster closed-form method for converting off-spheroid Universal Space Rectangular coordinates to geodetic latitude, longitude, and altitude. This solution method is provided in chapter 5 (GMD Closed-Form Solution), and it is usable with any spheroid datum reference. It is not quite as fast as the two approximation methods with which it is compared, but it is faster than the other closed-form solution described in chapter 5.

Since the baseline programs will be used as a measurement standard to which data from operational systems will be compared, it was imperative that the baseline programs have provable accuracy. For this reason, considerable time was spent in developing validation techniques that could demonstrate the accuracies of the algorithms to anyone who might feel skeptical about the results. Three methods are considered valid for this purpose. The first selects special trivial cases where standard trigonometric relations can be applied to get comparison values. Since the program exercises the same algorithms for trivial as for more complex non-trivial solutions, this is considered to be a sound validation technique if correctly planned. The second validation method compares results of the baseline programs with data published by USCGS, DMAC, and other mapping and geodetic groups whose work is generally held as a standard for survey and geodetic measurement purposes. The third method compares results obtained using one mathematical approach to results obtained by other different solution methods. If two different solution techniques yield the same results, then this is accepted as an additional confidence factor.

Radar device (pedestal and antenna) corrections were eliminated from this document since they were covered in an earlier publication.

Robert James

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS

CHAPTER 4
RANGE AND ANGLE CALCULATIONS

CHAPTER 5
DETERMINATION OF GEODETIC COORDINATES FOR OFF-SPHEROID POINTS

CHAPTER 6
ATMOSPHERIC REFRACTION

CHAPTER 7
DATUM CONVERSIONS AND GENERATION OF DATA SHEETS

CHAPTER 8
RADAR CROSS SECTIONS AND ANTENNA GAIN PATTERNS

CHAPTER 9
CIRCUIT MARGIN CALCULATIONS

CHAPTER 10
SKYSCREEN PROGRAM

APPENDIX
A CLOSED-FORM QUARTIC SOLUTION

CHAPTER 1

MAPPING

The mapping routines provided in the GMD baseline geodetic programs consist of two subprograms, each with several operating modes. The subprograms and operating modes are:

A. Geodetic-Lambert Subprogram

1. Geodetic to Lambert conversion mode (general)
2. Lambert to geodetic conversion mode (general)
3. Geodetic to Lambert conversion mode (California zones)
4. Lambert to geodetic conversion mode (California zones)

B. Geodetic-Transverse Mercator Subprogram

1. Geodetic to transverse Mercator conversion mode (general)
2. Transverse Mercator to geodetic conversion mode (general)
3. Geodetic to transverse Mercator conversion mode (Nevada zones)
4. Transverse Mercator to geodetic conversion mode (Nevada zones)

This chapter presents the general mapping equations and concepts which are common to the Lambert and Mercator transformations and therefore applicable to both subprograms. The mathematical routines unique to the Lambert transformations are presented in chapter 2, and those for the Mercator transformation are given in chapter 3.

The mathematical treatments presented in this document are thought to be sufficiently complete to enable a user to grasp the basic fundamentals of the various transformation equations needed to prepare or modify mapping or geodetic programs employing Lambert conformal and transverse Mercator conformal mathematics. However, should supplemental information be needed, detailed derivations for both the Lambert and transverse Mercator transformations can be found in reference 1, and supporting data are provided in references 2 and 3.

General Mapping Theory

Types of Mapping Projections

Many types of projections are used in the science of mapping. These include spherical, conical, and cylindrical mappings, on a point-to-point basis, of the coordinates of the earth spheroid onto a new surface selected to meet a certain mapping application. Probably the simplest projection is the perspective or geometric type, in which images are mapped at the exact point on the new surface where a ray drawn from the projecting origin through a spheroid point intersects the mapping surface. Unfortunately, perspective projections introduce too much distortion to be practical in most mapping applications.

A second type consists of a general group of nonperspective projections which are not projected in the usual sense of the term, but are mathematically modified so as to suit one or more particular requirements. Although distortion is always present in a planar representation of a spheroidal surface, by mathematically forcing certain conditions to be met, nonperspective projections can be readily adapted to individual mapping needs, and they are therefore more useful than perspective projections. Thus, in many cases, it is possible to derive transformation equations such that areas transform accurately from those on the spheroid surface to those on the mapped surface. In other cases, it is possible to insure that the scale at any given point over the entire mapping projection is the same in all directions, even though the scale varies from one part of the projection to another. This type of projection preserves angles and is said to be orthomorphic or conformal.

In navigational applications, the primary concern is that heading angles measured on the map be the same as those that would be measured on the earth spheroid. Hence, navigational charts nearly always employ one of the various types of orthomorphic projections—most commonly the Lambert conformal projection or the transverse Mercator conformal projection.

Orthogonal Curvilinear Coordinate Systems

A planar curvilinear coordinate system is said to exist when two single parameter families of curves can be defined such that any single point in the region under consideration lies on one and only one curve in each of the two families of curves. For example, nearly any point on the earth spheroid can be fully defined by the intersection of unique latitude (parallel) and longitude (meridian) arcs. For this system to also be orthogonal, all of the infinite number of possible intersections between the two families of curves must occur at right angles to one another. Obviously, this condition is true on the earth spheroid, and, if angles are to be preserved, it must also be true on the two-dimensional mapping surface.

It was noted that the earth's parallels and meridians form a curvilinear coordinate system at nearly every point on the earth, but not all points. Obviously, at the two poles, the condition for uniqueness is not met by the meridians since all meridians pass through the same point, causing the azimuth measure to become indeterminate.

Conformal Isometric Projection of One Surface Onto Another

A transformation is said to be orthomorphic if the form of incremental parts of a figure retain the same shape through the transformation. However, the shape of large parts can and will change. A surface has an orthomorphic representation on another if a one-to-one correspondence can be established between points in such a manner that angles between corresponding lines on the two surfaces are equal.

Figure 1.1 represents a plane on which the set of curvilinear orthogonal coordinates are defined as τ and λ. On this plane the length of a differential arc segment is given by the relation

$$ds^2 = d\tau^2 + d\lambda^2. \tag{1.1}$$

If there exists another surface on which a differential arc could be represented by the same curvilinear coordinates and a magnification term k in the form

$$ds_1^2 = k^2(d\tau^2 + d\lambda^2) \tag{1.2}$$

then it is obviously possible to achieve a one-to-one relationship between the points on the two surfaces.

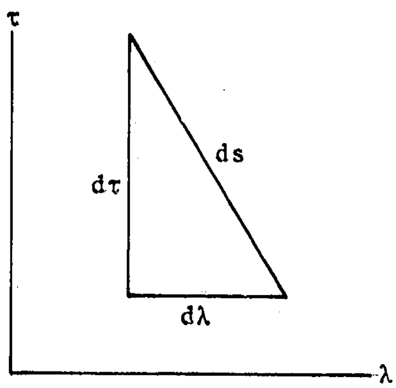

Figure 1.1.

It is important to note that for the orthomorphic property to exist, it is not necessary that k be the same value at all points over the surface, and it will not be the same in the derivations which follow. However, since at any given point the value of k magnifies both of the coordinates equally, then it is obvious that the angle between the differential segment ds and the two coordinate elements $d\tau$ and $d\lambda$ will be the same on the initial and transformed surfaces, thus preserving the shape of small (incremental) forms through the transformation. Satisfaction of equations (1.1) and (1.2) also insures that the coordinate axes τ and λ are orthogonal at all points on both surfaces.

If a relationship between two surfaces could be established such that equations (1.1) and (1.2) were satisfied, then segment angles and incremental shapes would be preserved. A final requirement of the transformation is that the transformation functions be analytic throughout the region to be mapped. A function is said to be analytic if it is continuously differentiable throughout the region of interest (that is, all higher order derivatives exist). The analytic nature of a function can be shown by application of the Cauchy-Riemann equations, which are derived later in this chapter.

The Geometry of the Spheroid

The following standard spheroidal relationships are necessary for derivation of transformation equations. They are well known and presented without proof.

A. The Spheroid: The equation of the spheroid in Cartesian form is given below. The symbol a represents the length of the semimajor axis of the spheroid, and the symbol b represents the length of the semiminor axis of the spheroid.

$$x^2/a^2 + y^2/a^2 + z^2/b^2 = 1 \tag{1.3}$$

B. Eccentricity, ε: Eccentricity, a relationship between the semimajor and semiminor axes of an ellipse, is given by

$$\varepsilon^2 = (a^2 + b^2)/a^2. \tag{1.4}$$

C. Meridional Radius of Curvature, R: The north-south (N-S) radius of curvature at any point on the spheroid is the distance measured along the normal line from the surface of the spheroid to the point which is the center of curvature of the spheroid meridian. It is given by the equation

$$R = a(1 - \varepsilon^2)/(1 - \varepsilon^2 \sin^2\mu)^{3/2} \tag{1.5}$$

where μ is the geodetic latitude of the point.

D. East-West Radius of Curvature, N: The east-west (E-W) radius of curvature at any point on the spheroid is the distance measured along the normal line from the surface of the spheroid to the semiminor axis. It is obtained from the equation

$$N = a/(1 - \varepsilon^2 \sin^2\mu)^{1/2}. \tag{1.6}$$

E. Meridional Arc, S_μ: The true length of the meridional arc from the equator to latitude μ is given by

$$S_\mu = \int_0^\mu R \, d\mu. \tag{1.7}$$

F. Geodetic latitude, μ: Geodetic latitude of a point P located on the spheroid surface is defined as the angle between the spheroid normal line at P and the spheroid equator.

G. Geocentric latitude, ξ: Geocentric latitude of a point P located on the spheroid surface is defined as the angle between the spheroid equator and the line from the spheroid origin and through the point P.

H. Geodetic to Geocentric Latitude Conversion: The conversion from geodetic latitude, μ, to geocentric latitude, ξ, is given by

$$\tan \xi = (1 - \varepsilon^2)\tan \mu, \tag{1.8}$$

Incremental Spheroid Segment

As shown in figure 1.2, the element of length on the surface of the spheroid is given by

$$ds^2 = R^2 d\mu^2 + N^2 \cos^2\mu \, d\lambda^2 \tag{1.9}$$

or

$$ds^2 = N^2\cos^2\mu \left[\frac{R^2 d\mu^2}{N^2\cos^2\mu} + d\lambda^2\right]. \tag{1.10}$$

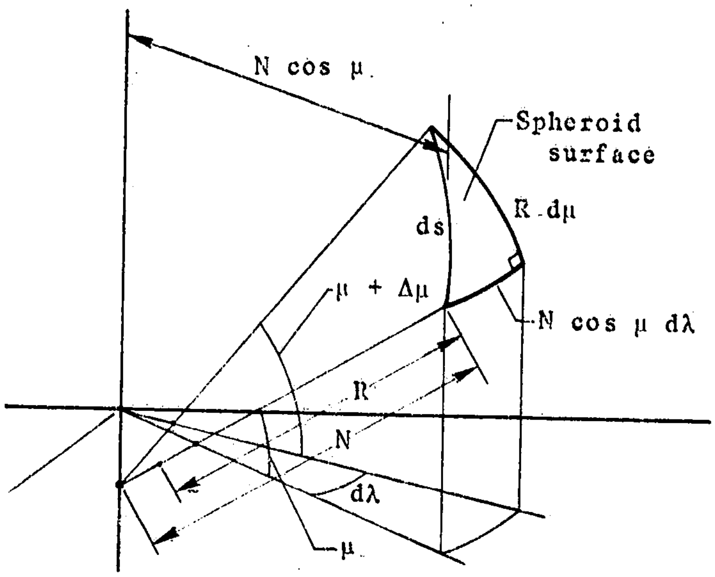

Figure 1.2.

Substituting the values for R and N given in equations (1.5) and (1.6), equation (1.10) becomes

$$ds^2 = \frac{a^2\cos^2\mu}{1 - \varepsilon^2\sin^2\mu}\left[\frac{(1 - \varepsilon^2)^2 \, d\mu^2}{\cos^2\mu(1 - \varepsilon^2\sin^2\mu)^2} + d\lambda^2\right]. \tag{1.11}$$

Now if $d\tau$ is defined as

$$d\tau = \frac{(1 - \varepsilon^2)d\mu}{\cos\mu(1 - \varepsilon^2\sin^2\mu)} \tag{1.12}$$

and k^2 as

$$k^2 = \frac{a^2 \cos^2\mu}{1 - \varepsilon^2 \sin^2\mu}, \tag{1.13}$$

ds^2 becomes

$$ds^2 = k^2(d\tau^2 + d\lambda^2), \tag{1.14}$$

which is identical to equation (1.2).

Incremental Spherical Segment

If a derivation similar to that shown in the preceding section is followed for a sphere where one wishes to obtain isometric parameters β and λ such that an incremental element of arc length maps conformally onto another surface, it has been shown that the incremental element of length must be of the form

$$ds^2 = k^2(d\tau^2 + d\lambda^2). \tag{1.15}$$

For a sphere, if we let the conformal latitude be given by β, the longitude by λ, and the radius by ρ, then the length of the incremental segment, as shown by figure 1.3, is

$$ds_1^2 = \rho^2 d\beta^2 + \rho^2 \cos^2\beta \; d\lambda^2 \tag{1.16}$$

$$= \rho^2 \cos^2\beta \left[\frac{d\beta^2}{\cos^2\beta} + d\lambda^2\right], \tag{1.17}$$

which is in the form of equation (1.15).

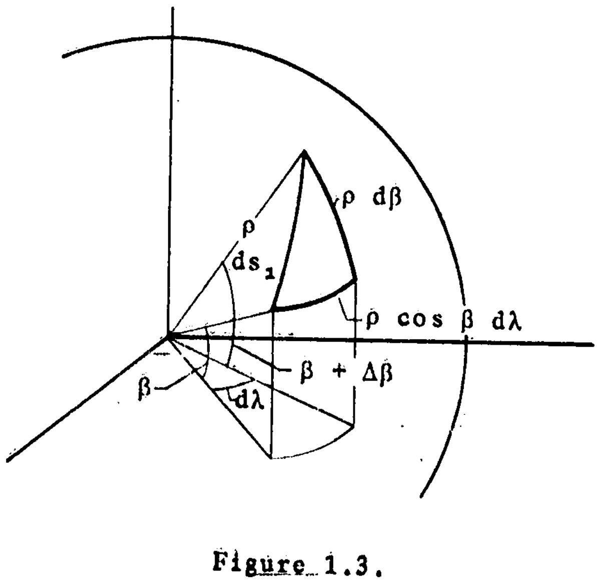

Figure 1.3.

If the substitutions

$$d\tau = d\beta/\cos\beta \quad \text{and} \quad k = \rho\cos\beta \tag{1.18}$$

are made in equation (1.17), the equation for incremental arc length on the sphere becomes identical to equation (1.15),

$$ds^2 = k^2(d\tau^2 + d\lambda^2). \tag{1.19}$$

Solving the differential equation (1.18) for τ yields

$$\tau = \int \frac{d\beta}{\cos\beta} \tag{1.20}$$

which on integration yields

$$\tau = \ln\tan\left(\frac{\pi}{4} + \frac{\beta}{2}\right). \tag{1.21}$$

Thus, the latitude and longitude coordinates on a sphere (β, λ) can be conformally mapped onto a plane in terms of τ and λ in the same manner as shown for the spheroid.

Conformal Mapping of a Spheroid Onto a Sphere

The sphere whose linear element is given by equation (1.19) is called the conformal sphere since equation (1.19) insures that the mapping of the points of the sphere onto a plane will be orthomorphic. In addition, if the incremental arc of any one surface can be shown to map onto any other surface while preserving the relations

$$ds_1^2 = k_1^2(d\tau^2 + d\lambda^2) \tag{1.22}$$

and

$$ds_2^2 = k_2^2(d\tau^2 + d\lambda^2), \tag{1.23}$$

then the mapping between those two surfaces will also be orthomorphic. It has already been shown that such relations exist for both the spheroid and the sphere (eqs. (1.10) and (1.17)). Thus it is possible to equate relations for $d\tau^2$ from the sphere and the spheroid to yield

$$d\tau^2 = \frac{d\beta^2}{\cos^2\beta} = \frac{R^2\, d\mu^2}{N^2\cos^2\mu}, \tag{1.24}$$

which makes the differential equation to be solved for the transformation

$$\frac{d\beta}{\cos\beta} = \frac{R\, d\mu}{N\cos\mu}. \tag{1.25}$$

The solution to this equation is easily obtained from the separate solutions already found for the spheroid and spherical cases,

$$\tan\left(\frac{\pi}{4} + \frac{\beta}{2}\right) = \tan\left(\frac{\pi}{4} + \frac{\mu}{2}\right)\left[\frac{1 - e \sin \mu}{1 + e \sin \mu}\right]^{e/2}. \tag{1.26}$$

Thus, in order to conformally map coordinates between the spheroid and a conformal sphere, it is necessary to convert geodetic latitude, μ, into the spherical conformal latitude, β, by means of equation (1.26). Since longitude is the same in equations (1.10) and (1.17), it follows that

$$\lambda = \lambda_1 + \lambda_2. \tag{1.27}$$

Magnification for the transformation is given by

$$k = \frac{ds_1}{ds_2} = \frac{\rho \cos \beta}{N \cos \mu}. \tag{1.28}$$

Isometric Latitude Computation

Isometric latitude, τ, was defined on the spheroid by equation (1.12). Rearrenging terms, equation (1.11) can be rewritten as

$$d\tau = \frac{d\mu}{\cos \mu} - \frac{\varepsilon^2 \cos \mu \, d\mu}{1 - \varepsilon^2 \sin^2 \mu}, \tag{1.29}$$

from which

$$\tau = \int \frac{d\mu}{\cos \mu} - \int \frac{\varepsilon^2 \cos \mu \, d\mu}{1 - \varepsilon^2 \sin^2 \mu}. \tag{1.30}$$

Integration of equation (1.30) yields

$$\tau = \ln\left[\tan\left(\frac{\pi}{4} + \frac{\mu}{2}\right)\right] + \frac{\varepsilon}{2}\left[\ln \frac{(1 - \varepsilon \sin \mu)}{(1 + \varepsilon \sin \mu)}\right] \tag{1.31}$$

or, in its more common form,

$$\tau = \ln\left[\tan\left(\frac{\pi}{4} + \frac{\mu}{2}\right)\left(\frac{1 - \varepsilon \sin \mu}{1 + \varepsilon \sin \mu}\right)^{\varepsilon/2}\right]. \tag{1.32}$$

Conformal Mapping of a Spheroid Onto a Plane

It has already been shown that for a surface to be mapped conformally onto a plane, it is necessary that a relationship of the type

$$ds_2^2 = k^2(d\tau^2 + d\lambda^2) \tag{1.33}$$

be found for the surface. For the spheroid, it has been shown in equation (1.10) that the incremental spheroid segment can be represented by

$$ds_2^2 = N^2 \cos^2\mu \left[\frac{R^2}{N^2} \sec^2\mu \, d\mu^2 + d\lambda^2\right]. \tag{1.34}$$

Thus, for the conformal mapping of a spheroid onto a plane, it is necessary that

$$d\tau = \frac{R}{N} \sec \mu \, d\mu, \quad k^2 = N^2\cos^2\mu, \quad \text{and} \quad \lambda = \lambda. \tag{1.35}$$

These equations show important properties of the conformal mapping of a spheroid onto a plane. Specifically, τ is found to be a function of μ alone, and λ is found to be a function of λ alone. Finally, by combining equations (1.32) and (1.35), the fundamental isometric relations for the conformal mapping of a spheroid onto a plane can be given as

$$\tau = \int_0^\mu \frac{R}{N} \sec \mu \, d\mu = \ln \left[\tan\left(\frac{\pi}{4} + \frac{\mu}{2}\right)\left(\frac{1 - \varepsilon^2\sin^2\mu}{1 + \varepsilon^2\sin^2\mu}\right)^{\varepsilon/2}\right] \tag{1.36}$$

and

$$\lambda = \lambda. \tag{1.37}$$

It should be noted that equations (1.36) and (1.37) are valid for all conformal mappings of a spheroid onto a plane. The form of the mapping (for example, Lambert conformal, Mercator conformal, or transverse Mercator conformal) is determined by the initial conditions used to derive specific transformation relations.

The Cauchy-Riemann Equations

Because transformation equations employ two independent parameters such as (x, y) or (τ, λ) to define a single point, the handling of mathematical operations can sometimes be simplified through the use of complex variable theory. For example, in the case of multiplication of (a, b) by (c, d), using simple algebra we have

$$(a, b)(c, d) = (ac - bd, bc + ad). \tag{1.38}$$

By using complex numbers, the mathematical relationships can be simplified. Using the term $(-1)^{1/2}$ or i, equation (1.38) can be written more conveniently as

$$\begin{aligned}(a + ib)(c + id) &= ac + iad + ibc + i^2bd \\ &= (ac - bd) + i(bc + ad).\end{aligned} \tag{1.39}$$

Likewise, the transformation of curvilinear longitudes and isometric latitudes into rectangular mapping coordinates (such as transverse Mercator coordinates) can be simplified by using complex variable theory.

Consider the set of rectangular coordinates x and y. It has been shown that curvilinear parameters τ and λ will conformally map from the spheroid onto a plane. Thus, if a function exists such that

$$x + iy = f(\lambda + i\tau) \quad \text{and} \quad x - iy = f(\lambda - i\tau), \tag{1.40}$$

then

$$dx + idy = f'(\lambda + i\tau)(d\lambda + id\tau). \tag{1.41}$$

The complex conjugate is

$$dx - idy = f'(\lambda - i\tau)(d\lambda - id\tau). \tag{1.42}$$

Multiplying equations (1.41) and (1.42) yields

$$dx^2 + dy^2 = f'(\lambda - i\tau)f'(\lambda + i\tau)(d\lambda^2 + d\tau^2). \tag{1.43}$$

Also note that

$$f'(\lambda + i\tau) = \frac{\partial x}{\partial \lambda} + i\frac{\partial y}{\partial \lambda} = \frac{\partial y}{\partial \tau} - i\frac{\partial x}{\partial \tau} \tag{1.44}$$

and

$$f'(\lambda - i\tau) = \frac{\partial x}{\partial \tau} + i\frac{\partial y}{\partial \tau} = \frac{\partial y}{\partial \lambda} - i\frac{\partial x}{\partial \lambda}. \tag{1.45}$$

Equating the real and imaginary parts of equation (1.44) or (1.45) yields

$$\frac{\partial x}{\partial \lambda} = \frac{\partial y}{\partial \tau} \quad \text{and} \quad \frac{\partial y}{\partial \lambda} = -\frac{\partial x}{\partial \tau}, \tag{1.46}$$

which are the Cauchy-Riemann equations. If the derivatives exist, and if equations (1.46) are satisfied at all points in the region, then the mapping of the two sets of parameters must be conformal since at any point the rate of change of x with respect to λ equals the rate of change of y with respect to τ, and the rate of change of x with respect to τ equals (the negative of) the rate of change of y with respect to λ. Thus, at any selected point on the mapped surface, angles must be preserved in the transformation.

CHAPTER 2

LAMBERT CONFORMAL TRANSFORMATION EQUATIONS

Lambert Theory

It has already been shown (eq. (1.22)) that the conformal mapping of a spheroid onto a plane is given by

$$\tau = \int \frac{R}{N} \sec \mu \, d\mu \quad \text{and} \quad \lambda = \lambda. \tag{2.1}$$

The requirements for a Lambert conformal conic projection are:

1. The parallels must be arcs of concentric circles with centers at the point of intersection of the meridian radials.

2. All meridians must project as radial straight lines from a central vortex point which may lie off the map.

3. All meridians and parallels must intersect each other at right angles.

4. All angles on the earth's surface must be correctly represented on the projection.

5. The scale must be true along the two selected standard parallels.

These conditions establish a conformal projection of the spheroid onto a cone that intersects the spheroid at the two standard parallels.

To satisfy condition 1 in terms of regular Cartesian coordinates, x and y must be functions of τ such that

$$x^2 + y^2 = K^2 f(\tau). \tag{2.2}$$

This condition, if satisfied, will cause the parallels to plot out as arcs of concentric circles.

To meet the second condition, x and y must be a function of λ such that

$$y = m(\lambda) x. \tag{2.3}$$

This condition, if satisfied, will cause the meridians to plot out as radials from a common origin point.

Solving equations (2.2) and (2.3) for x and y yields

$$x = K\left(\frac{[f(\tau)]}{[1 + m^2(\lambda)]}\right)^{1/2} \quad \text{and} \quad y = Km(\lambda)\left(\frac{[f(\tau)]}{[1 + m^2(\lambda)]}\right)^{1/2} . \tag{2.4}$$

If the functions x and y exist and are orthomorphic, satisfying conditions 3 and 4, the Cauchy-Riemann equations must be satisfied. Expressions for $\partial x/\partial\lambda$, $\partial x/\partial\tau$, $\partial y/\partial\lambda$, and $\partial y/\partial\tau$ are given below. Note that in equations (2.5) to (2.13), $f(\tau)$ is represented as f and $m(\lambda)$ is represented as m.

$$\frac{\partial x}{\partial\lambda} = -K\,\frac{(f)^{1/2}\,m\,m'}{(1 + m^2)^{3/2}} \tag{2.5}$$

$$\frac{\partial x}{\partial\tau} = K\,\frac{f'}{2(f)^{1/2}\,(1 + m^2)^{1/2}} \tag{2.6}$$

$$\frac{\partial y}{\partial\tau} = K\,\frac{(f)^{1/2}\,m'}{(1 + m^2)^{3/2}} \tag{2.7}$$

$$\frac{\partial y}{\partial\tau} = K\,\frac{m\,f'}{2(f)^{1/2}\,(1 + m^2)^{1/2}} \tag{2.8}$$

The Cauchy-Riemann equations (eq. (1.46)) are expressed as

$$\frac{\partial x}{\partial\lambda} = \frac{\partial y}{\partial\tau} \quad \text{and} \quad \frac{\partial x}{\partial\tau} = -\frac{\partial y}{\partial\lambda} , \tag{2.9}$$

from which, by substitution of equations (2.5) to (2.8) into equations (2.9),

$$-\frac{f'}{f} = \frac{2\,m'}{1 + m^2} . \tag{2.10}$$

Since it has been shown that f is a function of τ alone and m is a function of λ alone, equation (2.10) can be true only if both terms in the equation are equal to the same constant, which for convenience will be called 2L. Equating each term of equation (2.10) to 2L yields

$$\frac{f'}{f} = -2L \quad \text{and} \quad \frac{m'}{1 + m^2} = L , \tag{2.11}$$

or, in terms of total differentials,

$$\frac{df}{f} = -2L\,d\tau \quad \text{and} \quad \frac{dm}{1 + m^2} = L\,d\lambda . \tag{2.12}$$

The solutions to equations (2.12) are

$$\ln f(\tau) = -2L\tau \quad \text{and} \quad \tan^{-1}[m(\lambda)] = L\lambda , \tag{2.13}$$

or

$$f(\tau) = e^{-2L\tau} \quad \text{and} \quad m(\lambda) = \tan L\lambda , \tag{2.14}$$

where e is the base of the system of Napierian logarithms. Substitution of equations (2.14) in place of $f(\tau)$ and $m(\lambda)$ in equations (2.4) yields

$$x = K\,e^{-L\tau}\cos L\lambda \quad \text{and} \quad y = K\,e^{-L\tau}\sin L\lambda. \tag{2.15}$$

Since x and y may be given in terms of polar coordinates r and α as

$$x = r\cos\alpha \quad \text{and} \quad y = r\sin\alpha, \tag{2.16}$$

equations (2.15) may be rewritten as

$$r = K\,e^{-L\tau} \quad \text{and} \quad \alpha = L\lambda. \tag{2.17}$$

It was previously shown that the isometric latitude, τ, is given by the relation

$$\tau = \ln\left[\tan\left(\frac{\pi}{4}+\frac{\mu}{2}\right)\left(\frac{1-\varepsilon\sin\mu}{1+\varepsilon\sin\mu}\right)^{\varepsilon/2}\right], \tag{2.18}$$

or, in terms of conformal latitude, β,

$$\tau = \ln\tan\left(\frac{\pi}{4}+\frac{\beta}{2}\right). \tag{2.19}$$

The equation for r may now be rewritten as

$$r = K\,\frac{1}{\tan^L\left(\frac{\pi}{4}+\frac{\beta}{2}\right)}$$

$$= K\tan^L\left(\frac{\pi}{4}-\frac{\beta}{2}\right). \tag{2.20}$$

Equation (2.20) is obviously based on the use of radian measure for all angles. In degrees, equation (2.20) becomes

$$r = K\tan^L\left(\frac{90-\beta}{2}\right), \tag{2.21}$$

and, if z represents conformal colatitude defined as

$$z = 90 - \beta, \tag{2.22}$$

then the expressions for the polar coordinates r and α may be written as

$$r = K\left(\tan\frac{z}{2}\right)^L \quad \text{and} \quad \alpha = L\lambda. \tag{2.23}$$

It now remains to evaluate the constants K and L. This is accomplished by applying the final condition that the lengths of the two standard parallels be true. To satisfy this condition, the ratio of the lengths before and after the transformation must be the same, or

$$\frac{2\pi N_1 \cos \mu_1}{2\pi N_2 \cos \mu_2} = \frac{2\pi L r_1}{2\pi L r_2} \tag{2.24}$$

from which

$$\frac{N_1 \cos \mu_1}{N_2 \cos \mu_2} = \frac{r_1}{r_2} = \frac{Ke^{-L\tau_1}}{Ke^{-L\tau_2}} = \frac{e^{-L\tau_1}}{e^{-L\tau_2}}. \tag{2.25}$$

Taking the logarithms of the left and right terms of equation (2.25) yields

$$L = \frac{\ln N_1 + \ln \cos \mu_1 - \ln N_2 - \ln \cos \mu_2}{\tau_2 - \tau_1}$$

$$= \frac{\ln N_1 - \ln N_2 + \ln \cos \mu_1 - \ln \cos \mu_2}{\ln \tan \frac{z_2}{2} - \ln \tan \frac{z_1}{2}} \tag{2.26}$$

Now, by again making use of the equality in the lengths of the parallels before and after the transformation,

$$2\pi N_1 \cos \mu_1 = 2\pi L K e^{-L\tau_1} \quad \text{and} \quad 2\pi N_2 \cos \mu_2 = 2\pi L K e^{-L\tau_2}, \tag{2.27}$$

from which

$$K = \frac{N_1 \cos \mu_1}{L \tan\left(\frac{z_1}{2}\right)^L} = \frac{N_2 \cos \mu_2}{L \tan\left(\frac{z_2}{2}\right)^L}. \tag{2.28}$$

This completes the derivations of the planar coordinate equations for the Lambert conformal conic projection.

Geodetic to Lambert Computations

The steps to be followed in making the geodetic to Lambert conformal transformation are as follows:

1. Determine the geodetic latitudes for the two standard parallels, with μ_1 being the latitude of the north standard parallel and μ_2 being the latitude of the south standard parallel. Store both parameters.

2. Enter and store the geodetic latitude and longitude of the origin point selected for the particular map (μ_0, λ_0).

3. Enter and store the scale factor for the particular map being used (for example, 500,000, 2,000,000, etc.)

4. Calculate values for N_1 and N_2 from equation (1.6) using the applicable spheroid values of a and ε.

5. Calculate values of spherical conformal latitudes β_1 and β_2 corresponding to μ_1 and μ_2 using equations (2.18) and (2.19).

6. Determine values of conformal colatitude z_1 and z_2 from equation (2.22).

7. Calculate convergence, L, from equation (2.26).

8. Calculate K from equation (2.28).

9. Calculate the magnitude of the apex-to-origin position vector, r_o, from equation (2.23).

10. Enter the target point's geodetic latitude and longitude, and compute the values of r and α by the same procedures as described above (eqs. (2.18), (2.19), (2.22), and (2.23)).

11. Determine the target's position in Cartesian coordinates from the angular relationships shown in figure 2.1.

12. Add the bias term to x (if California Lambert), apply the necessary unit conversion (for example, to convert meters back to international survey (I.S.) feet), and divide by the selected scale factor.

Lambert to Geodetic Computations

In the baseline Lambert to geodetic program, the forward (geodetic to Lambert) equations are used in an iterative fashion to obtain geodetic coordinates from Lambert x and y values. The procedure is:

1. Perform steps 1 to 9 of the geodetic to Lambert sequence described in the previous section.

2. Enter scaled values of x and y in selected units.

3. Convert to meters and full scale for computations.

4. Obtain r and α directly from equations (2.16).

5. Knowing L and α, obtain the value of longitude directly from equation (2.17).

6. Knowing r, K, and L, solve equation (2.21) for β.

7. Using this value of β as the first trial value for μ, solve equation (2.18) for τ, and using that value of τ, obtain a comparison value of β from equation (2.19). The difference between the actual value of β and the computed value of β is nearly the same as the amount of error in μ. Thus, μ is adjusted by the same amount as the error in β and a second pass is made using equations (2.18) and (2.19). The error in β on the second pass will be much smaller, and it is again used to adjust μ. The iterations are continued until a value of μ is obtained for which the computed value of β equals the actual value as originally obtained from equation (2.21).

8. When the starting and ending values of β are equal, the value of μ is correct.

Four iterations are generally required before agreement is obtained to the precision limit of the present 12-digit computing system.

California Lambert

The plane coordinate system for the State of California consists of seven Lambert zones. Transformations for points in each of the zones is accomplished in the same manner as described for general Lambert transformations, except that in the California system a bias factor is added to the x coordinate so that the x values are always positive. For the first six California zones the bias factor is an even 2,000,000 I.S. feet. For zone 7 the bias is 4,186,692.58 I.S. feet.

The lines of separation between the zones run approximately east and west following county boundaries, with zone 1 being the northernmost zone and zone 6 being the southernmost zone. Zone 7 is a special zone that has been set aside for Los Angeles County. In transition areas, two zones may be specified on control data sheets. In the case of Edwards Air Force Base, this allows local control points to be referenced to the zone 5 system lying to the north and to the zone 7 system lying to the south and east.

Unlike aeronautical charts which have widely separated standard parallels and cover large areas of the country, the standard parallels for state zones are placed just one or two degrees apart and the coverage areas are limited to narrow horizontal bands that may extend over only three or four counties. This is done so that map convergence and magnification are sufficiently minimized that only minor corrections must be made to angle and distance measurements made by survey teams. In fact, much private survey work is accomplished without regard to either magnification or angular convergence, and computations are performed with flat-earth mathematics. For small parcel surveys, where extended base legs are not involved, these calculations are often accurate enough for many practical commercial applications. However, for NASA or military control nets, in which high-precision, long-range tracking systems are operated, it is extremely important that all field work and mathematical computations be carried out with the best possible accuracy and precision.

Table 2.1 lists the Lambert parameters for each of the seven California survey zones. In the baseline geodetic program, when a California zone has been selected, the program automatically initializes using the stored Lambert parameters for that particular zone.

TABLE 2.1. CALIFORNIA SURVEY ZONES

Zone	Olat	Olon	Bias	S lat	N lat
1	39 20 00.0	122 00 00.0	2,000,000	40 00 00.0	41 40 00.0
2	37 40 00.0	122 00 00.0	2,000,000	38 20 00.0	39 50 00.0
3	36 30 00.0	120 30 00.0	2,000,000	37 04 00.0	38 26 00.0
4	36 20 00.0	119 00 00.0	2,000,000	36 00 00.0	37 15 00.0
5	33 30 00.0	118 00 00.0	2,000,000	34 02 00.0	35 28 00.0
6	32 10 00.0	116 15 00.0	2,000,000	32 47 00.0	33 53 00.0
7	22 45 43.75445				
7 (continued)		118 20 00.0	4,186,692.58	33 52 00.0	34 25 00.0

Lambert Programs

The Lambert conformal programs listed here omit certain of the housekeeping functions which are present in the actual routines. However, all of the essential computational routines and subroutines are provided.

Variable Names

Name	Description
Aa	Semimajor axis of selected earth spheroid
Arg	Tangent argument in equation for isometric latitude
Bias	Bias in meters for selected California Lambert zone (0 if not in California Lambert mode)

Count	Number of iteration passes within secondary window range
Deg	Value of degrees returned from Dmstodeg subroutine
Dlon	Difference between origin and target longitudes
Dlon1	Angular difference between origin and target meridians on planar conic projection
E2	Eccentricity squared
Ee	Eccentricity of selected earth spheroid
Esinphi	E*SIN(U)
K	Lambert constant
L	Map convergence
Lx	Lambert x in meters (x bias removed)
Lx1	Scaled Lambert x in input or output units (includes x bias for California Lambert zones)
Ly	Lambert y in meters
Ly1	Scaled Lambert y in input or output units
N	E-W radius of curvature (general)
Nn	E-W radius of curvature along north standard parallel
Ns	E-W radius of curvature along south standard parallel
Olat	Origin latitude in degrees
R1	Origin-to-target (hypotenuse) direct Lambert distance
Ri	Working value of magnitude of apex-to-target vector
Ro	Magnitude of apex-to-origin vector
Rz2	Magnitude of apex-to-target vector
Scale	Map scale (for example, 500,000)
U	Geodetic latitude
Ucnv	Conversion factor from meters to selected units
Xi	Working value of conformal latitude

Zi	Working value of conformal colatitude
Zlast	Last value of target latitude in reverse iteration
Zlat	Target latitude in degrees
Zlon	Target longitude in degrees
Zn	Conformal colatitude of north standard parallel
Zo	Conformal colatitude of origin point
Zs	Conformal colatitude of south standard parallel
Zz	Conformal colatitude of target
Zz0	Conformal colatitude of target
Zz1	Current trial input value of geodetic latitude
Zz2	Current trial conformal colatitude value

Computational Algorithms

The essential algorithms for the Lambert routines were written for the System 45 computer which is programmable only in BASIC. Therefore, all algorithms are given in BASIC.

A. Lambert Initialization Routine: The values of the north and south standard parallel latitudes, the origin latitude, the origin longitude, and the map scale and bias factors are required for the initialization routines. These values are either entered manually (for normal mapping solutions) or automatically picked up from stored values (for California Lambert zone solutions). The program then computes all of the non-variable parameters required by the computational subroutines. These parameters are Nn, Ns, Zn, Zs, L, K, and Ro.

1. U=Nlat
2. GOSUB Ncalc
3. Nn=N
4. GOSUB Xicalc
5. Zn=Zi
6. U=Slat
7. GOSUB Ncalc
8. Ns=N
9. GOSUB Xicalc
10. Zs=Zi
11. U=Olat
12. GOSUB Xicalc
13. Zo=Zi
14. GOSUB Lcalc

15. GOSUB Kcalc
16. Zi=Zo
17. GOSUB Ricalc
18. Ro=Ri

B. Geodetic to Lambert Computation: The operator enters the target latitude (Zlat) and longitude (Zlon). The target's conformal colatitude (Zz) is then computed for use in subroutine Ricalc which returns the magnitude of the apex-to-target vector. The difference in longitude (Dlon) between the origin longitude (Olon) and the target longitude (Zlon) is obtained in step 5, and that difference (Dlon) is then multiplied by the map convergence factor (L) to obtain the difference in longitude (Dlon1) between the apex-to-origin and apex-to-target vectors (fig. 2.1). Step 7 resolves the polar coordinates of the apex-to-target vector (Ri and Dlon1) into Lambert Cartesian coordinates (Lx and Ly) in subroutine Lamxy. Lx is then adjusted by the amount of the California Zone bias (Bias), converted from meters into the selected output units, and scaled as necessary. For general Lambert conversions, the bias term is 0. The Ly coordinate is converted to output form in a similar manner except that there is no y-coordinate bias in the California and general Lambert systems.

1. U=Zlat
2. GOSUB Xicalc
3. Zz=Zi
4. GOSUB Ricalc
5. Dlon=(Olon-Zlon)
6. Dlon1=Dlon*L
7. GOSUB Lamxy
8. Lx1=(Lx+Bias)*Ucnv/Scale
9. Ly1=Ly*Ucnv/Scale

C. Lambert to Geodetic Computations: The operator enters the scaled Lambert x and y coordinates (Lx1 and Ly1) in the selected units. The program removes the x bias (when California Lambert has been selected), adjusts the values to full scale, and converts the distances from the selected input units to meters. The adjusted values (Lx and Ly) are then used to compute the magnitude (Rz2) of the apex-to-target vector and the angle (Dlon1) between the origin and target vectors. Dlon1 is then divided by the map's longitude convergence factor (L) to yield the true longitudinal difference (Dlon) between the origin and target points. The actual target longitude (Zlon) is then found by subtracting the longitude difference (Dlon) from the origin longitude (Olon). The target latitude (Zlat) is obtained through the iterative routine described in the Lambert to Geodetic Computations section on page 2-5.

1. Lx=Lx1/Ucnv*Scale-Bias
2. Ly=Ly1/Ucnv*Scale
3. IF Lx<>0 THEN GOTO 7
4. Dlon=0
5. Rz2=Ro-Ly
6. GOTO 10
7. Dlon1=ATN(Lx/(Ro-Ly))

```
8.  Rz2=Lx/SIN(Dlonl)
9.  Dlon=Dlonl/L
10. Zlon=Olon-Dlon
11. Zz0=Zz1=2*ATN((Rz2/K)**(1/L)
12. U=90-Zz1
13. GOSUB Xicalc
14. Zz2=Zi
15. IF ABS(Zz2-Zz0)<5E-10 THEN 18
16. Zlast=Zz1
17. Zz1=Zz0+Zz1-Zz2
18. IF Zlast=Zz1 THEN Count=Count+1
19. IF Count=3 THEN 18
20. GOTO 12
21. Zlat=U
```

D. Subroutine Ncalc: This subroutine computes the E-W radius of curvature at latitude U based on the selected spheroid parameters a (Aa) and e^2 (E2). It is a direct implementation of equation (1.6).

```
1. Sinu=SIN(U)
2. Sin2u=Sinu*Sinu
3. N=Aa/SQR(1-E2*Sin2u)
4. RETURN
```

E. Subroutine Xicalc: This subroutine computes values for conformal latitude and conformal colatitude for a point at latitude U on a spheroid whose semimajor axis Aa and whose eccentricity squared is E2. The conformal latitude (Xi) is obtained at step 3 by solving a combined form of equations (2.18) and (2.19) for β. Step 4 yields the conformal colatitude by a direct implementation of equation (2.22).

```
1. Esinphi=Ee*SIN*(U)
2. Arg=45+U/2
3. Xi=2*(ATN(TAN(Arg)*((1-Esinphi)/(1+Esinphi))**(Ee/2))-45
4. Zi=90-Xi
5. RETURN
```

F. Subroutine Lcalc: This subroutine computes convergence by a direct implementation of equation (2.26).

```
1. L=LGT(Nn)-LGT(Ns)+LGT(COS(Un)-LGT(COS(Us)))/....
     (LGT(TAN(Zs/2))-LGT(TAN(Zn/2)))
2. RETURN
```

G. Subroutine Ricalc: This subroutine computes the magnitude of the apex-to-target point vector by use of equations (2.21) and (2.22).

```
1. Ri=K*TAN(Zi/2)**L
2. RETURN
```

H. Subroutine Lamxy: This subroutine computes Lambert x-y coordinates from the angular relationships between the vectors Ro and Ri as shown in figure 2.1.

1. Lx=Ri*SIN(Dlonl)
2. Ly=Ro-Ri*COS(Dlonl)
3. RETURN

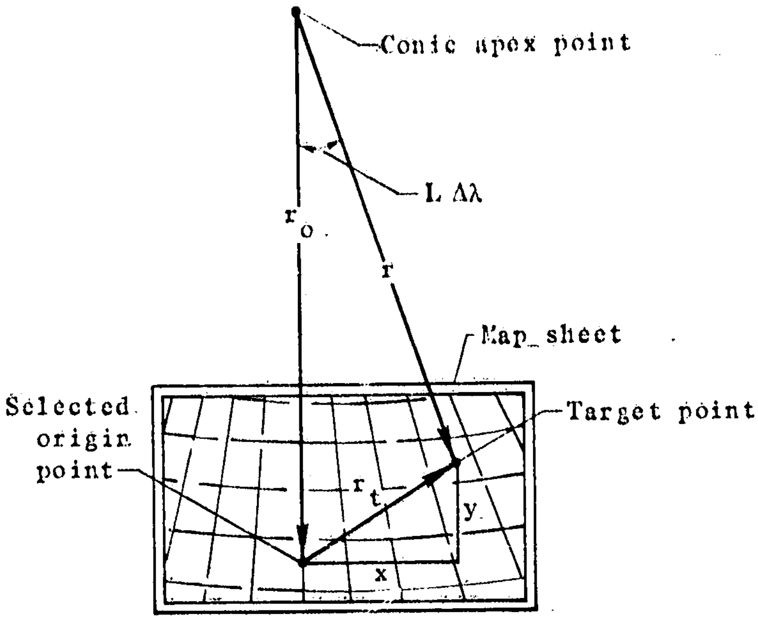

Figure 2.1.

Program Operation

The Lambert routines are part of the main program GEOD. When GEOD is run, the operator is asked to select the units and datum/spheroid reference applicable to the computations to be performed. After these selections are made, the master menu selection is displayed. One menu selection is LAMBERT TRANSFORMATIONS. The operator makes the appropriate numerical entry and the main program enters the Lambert routines. The operator is then prompted to make several simple selections. The prompting messages and program options that appear on the CRT are shown below.

A. Mode selection

SELECT MODE

0 = GEODETIC TO LAMBERT
1 = LAMBERT TO GEODETIC

B. Output device selection

SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER

C. Parameter selection

SELECT INITIALIZATION PARAMETERS

0 = MANUAL ENTRY
1 = CALIFORNIA LAMBERT
2 = CALIFORNIA LAMBERT WITH SPECIFIED ORIGIN

If the operator enters 1 or 2 and continue (CONT), the program proceeds to step D. If the operator enters CONT, the program proceeds to step E.

D. California Lambert zone selection

ENTER CALIF ZONE (1, 2, 3, 4, 5, 6, OR 7)

If 1 was selected at step C, the program uses the standard origin point for the selected California zone. If 2 was selected at step C, the program uses the stored north and south standard parallels for the selected California zone but requests an operator entry of the desired origin point. This is primarily intended for survey applications in which the N-S and E-W coordinates of a new point are to be computed from an existing survey marker.

SELECT ORIGIN

ENTER ORIGIN LATITUDE (D.MS)
ENTER ORIGIN LONGITUDE (D.MS)

E. Manual input of Lambert parameters: If California Lambert was not selected at step C, the operator must input the Lambert parameters needed for the transformation. If California Lambert was selected, step E is bypassed.

1. Selection of north standard parallel

   ENTER LAT OF N STD PARALLEL IN D.MS

2. Selection of south standard parallel

   ENTER LAT OF S STD PARALLEL IN D.MS

3. Selection of origin latitude

   ENTER LAT OF ORIGIN IN D.MS

4. Selection of origin longitude –

ENTER LON OF ORIGIN IN D.MS

F. Selection or rejection of Lambert parameter printout

TO PRINT LAMB PARAM, 1 AND CONT

If 1 and CONT are entered, the following parameters will be printed out on the selected output device.

1. Latitude of north standard parallel in dms, deg, and radian values
2. Latitude of south standard parallel in dms, deg, and radian values
3. Origin latitude in dms, deg, and radian values
4. Origin longitude in dms, deg, and radian values
5. Nn (E-W radius of curvature on north standard parallel)
6. Ns (E-W radius of curvature on south standard parallel)
7. Zn (Conformal colatitude of north standard parallel)
8. Zs (Conformal colatitude of south standard parallel)
9. Zo (Conformal colatitude of origin)
10. Ro (Magnitude of apex to origin vector)
11. K (factor)
12. L (map convergence)

G. Geodetic to Lambert computations (if menu selection was for Geodetic to Lambert mode):

1. Operator entries

ENTER IDENTIFICATION OF POINT

ENTER LATITUDE IN D.MS (EG: 35 42 33.5643 = 35.42335643)

ENTER LONGITUDE IN D.MS (EG: 117 23 45.3214 = 117.23453214)

2. Program outputs (on selected output device)

NAME OF POINT

GEODETIC LATITUDE = (Value given in dms, deg, and radians)
GEODETIC LONGITUDE = (Value given in dms, deg, and radians)

LAMBERT X = (Value given in selected units)
LAMBERT Y = (Value given in selected units)
LAMBERT R = (x-y triangle hypotenuse in selected units)

3. Program pause: Upon depressing CONT, the program returns to step G-1 for entry of the next geodetic point.

H. Lambert to geodetic computations (if menu selection was for Lambert to geodetic mode):

1. Operator entries:

   ENTER IDENTIFICATION OF POINT

   ENTER LAMBERT X VALUE IN SELECTED UNITS AND SCALE

   ENTER LAMBERT Y VALUE IN SELECTED UNITS AND SCALE

2. Program outputs (on selected output device):

   NAME OF POINT

   LAMBERT X VALUE = (Value given in selected units and scale)
   LAMBERT Y VALUE = (Value given in selected units and scale)
   LAMBERT R VALUE = (Value given in selected units and scale)

   GEODETIC LATITUDE = (Value given in dms, deg, and radians)
   GEODETIC LONGITUDE = (Value given in dms, deg, and radians)

3. Program pause: Upon depressing CONT, the program returns to step H-1 for entry of the next Lambert point.

Program Validation

The Lambert routines are validated by using U.S. Coast and Geodetic Survey horizontal control data sheets for established survey points throughout Southern California. During a validation exercise, any points can be selected and used in either the forward or reverse programs. For the comparison given in table 2.2, several first and second order survey points were selected at random. The first line of each entry gives the geodetic latitude and longitude of the point along with published USCGS Lambert coordinates. The second line provides the Lambert coordinates computed using the GMD routines. In table 2.3, the published USCGS Lambert coordinates for the same points are shown on the first line of each entry. The second line shows the geodetic coordinates calculated by the GMD routines using the USCGS Lambert values. When GMD Lambert values are input to the program, the calculated coordinate values are identical to the original geodetic coordinate values. The differences between USCGS and GMD results are equivalent to an earth spheroid distance of only about 0.01 to 0.02 foot and therefore are not significant, even for the most precise survey work. However, since the GMD Lambert routines are based on closed-form solutions of the transformation equations, any discrepancies shown must be attributed to roundoff errors in the GMD computations, to approximation or roundoff errors which may be present in the USCGS routines, or to both.

TABLE 2.1. GEODETIC TO LAMBERT VALIDATIONS

Station	Source	Latitude	Longitude	Lambert x	Lambert y
			California Zone 5		
Soledad	USCGS	34 58 57.1271	118 11 16.5426	1,943,705.88	539,573.73
	GMD			1,943,705.88	539,573.74
Willow Springs	USCGS	34 53 00.7287	118 16 31.8072	1,917,374.47	503,604.72
	GMD			1,917,374.47	503,604.73
USCGS 3293	USCGS	34 53 00.3234	118 16 31.855[illegible]	1,917,370.30	503,563.77
	GMD			1,917,370.30	503,563.77
Mint	USCGS	34 34 00.7650	118 16 41.0384	1,916,286.65	388,368.63
	GMD			1,916,286.65	388,368.63
Oban	USCGS	34 45 14.6870	118 08 43.2816	1,956,338.26	456,410.30
	GMD			1,956,338.26	456,410.31
Lope	USCGS	34 48 29.9146	118 21 33.8124	1,892,117.22	476,307.27
	GMD			1,892,117.22	476,307.28
Bajada	USCGS	34 54 00.2985	118 21 28.3580	1,892,690.93	509,704.59
	GMD			1,892,690.94	509,704.60
			California Zone 7		
Sur	USCGS	34 41 20.8412	118 19 24.5217	4,189,655.48	4,363,197.08
	GMD			4,189,655.48	4,363,197.07
Surge	USCGS	34 35 54.6055	118 27 08.8435	4,150,840.11	4,330,235.81
	GMD			4,150,840.10	4,330,235.79
Pelona	USCGS	34 33 39.2730	118 21 18.4042	4,180,134.86	4,316,533.90
	GMD			4,180,134.86	4,316,533.88
Mint	USCGS	34 34 00.7650	118 16 41.0384	4,203,332.54	4,318,710.47
	GMD			4,203,332.54	4,318,710.45

TABLE 2.2. LAMBERT TO GEODETIC VALIDATIONS

Station	Source	Lambert x	Lambert y	Latitude	Longitude
		California Zone 5			
Soledad	USCGS	1,943,705.88	539,573.73	34 58 57.1271	118 11 16.5426
	GMD			34 58 57.1270	118 11 16.5426
Willow Springs	USCGS	1,917,374.47	503,604.72	34 53 00.7287	118 16 31.9072
	GMD			34 53 00.7286	118 16 31.9072
USCGS 3293	USCGS	1,917,370.30	503,563.77	34 53 00.3234	118 16 31.8559
	GMD			34 52 00.3234	118 16 31.8559
Mint	USCGS	1,916,286.65	338,368.63	34 34 00.7650	118 16 41.0384
	GMD			34 34 00.7650	118 16 41.0384
Oban	USCGS	1,956,338.26	456,410.30	34 45 14.6870	118 08 43.2816
	GMD			34 45 14.6869	118 08 43.2816
Lope	USCGS	1,892,117.22	476,307.27	34 48 29.9146	118 21 33.8124
	GMD			34 48 29.9145	118 21 33.8124
Bajada	USCGS	1,892,690.93	509,704.59	34 54 00.2985	118 21 28.3580
	GMD			34 54 00.2984	118 21 28.3581
		California Zone 7			
Sur	USCGS	4,189,655.48	4,363,197.08	34 41 20.8412	118 19 24.5217
	GMD			34 41 20.8412	118 19 24.5217
Surge	USCGS	4,150,840.11	4,330,235.81	34 35 54.6055	118 27 08.8435
	GMD			34 35 54.6057	118 27 08.8434
Pelona	USCGS	4,180,134.86	4,316,533.90	34 33 39.2730	118 21 18.4042
	GMD			34 33 39.2732	118 21 18.4042
Mint	USCGS	4,203,332.54	4,318,710.47	34 34 00.7650	118 16 41.0384
	GMD			34 34 00.7652	118 16 41.0384

CHAPTER 3

TRANSVERSE MERCATOR TRANSFORMATION EQUATIONS

General Theory

The derivation of the transverse Mercator transformation equations is based on complex variable theory. If there exists an analytic function of λ and τ such that

$$x + iy = f(\lambda + i\tau), \tag{3.1}$$

then the transverse Mercator transformation equations can be derived by applying the following initial conditions.

1. The transformation shall be orthomorphic (conformal).

2. The scale shall be true along the central meridian.

It has been shown that the equations

$$\tau = \int_0^\mu \frac{R}{N} \sec \mu \, d\mu \quad \text{and} \quad \lambda = \lambda \tag{3.2}$$

conformally map a spheroid onto a plane. Thus, the use of these relations in the transformation equations will insure the orthomorphic requirement.

To meet the condition that the scale be true along the central meridian, where $x = 0$ and $\lambda = 0$, equation (3.3) must be satisfied.

$$iy = f(i\tau) = iS_\mu \tag{3.3}$$

As previously noted, S_μ is the true length of the meridional arc from latitude 0 to latitude μ.

It has also been shown that

$$S_\mu = \int_0^\mu R \, d\mu. \tag{3.4}$$

Differentiation of the expression for τ given in equations (3.2) yields

$$d\tau = \frac{R}{N} \sec \mu \, d\mu \tag{3.5}$$

or

$$N \cos \mu \, d\tau = R \, d\mu , \tag{3.6}$$

which when substituted for $R \, d\mu$ in equation (3.4) yields

$$f(\tau) = S_\mu = \int_0^\mu N \cos \mu \, d\tau . \tag{3.7}$$

Forward Calculations

If $x + iy = f(\lambda + i\tau)$ is expanded about a point on the central meridian $(i\tau)$ using Taylor's theorem, the following series is obtained.

$$\begin{aligned} x + iy = f(\lambda + i\tau) = f(i\tau) &\pm \lambda f^1(i\tau) + \frac{\lambda^2}{2!} f^2(i\tau) + \frac{\lambda^3}{3!} f^3(i\tau) + \\ &\frac{\lambda^4}{4!} f^4(i\tau) + \frac{\lambda^5}{5!} f^5(i\tau) + \frac{\lambda^6}{6!} f^6(i\tau) + \frac{\lambda^7}{7!} f^7(i\tau) + \\ &\frac{\lambda^8}{8!} f^8(i\tau) + \frac{\lambda^9}{9!} f^9(i\tau) + \ldots \end{aligned} \tag{3.8}$$

(Note that, as used in equation (3.8) and others to follow, the terms $f^2(i\tau)$, $f^3(i\tau)$, ... signify the second, third, ... derivatives of $f(i\tau)$.)

From equations (3.3) and (3.6), $f(i\tau) = iS_\mu = if(\tau)$. Successive differentiation of this equation and substitution of the results into equation (3.8) yields

$$\begin{aligned} x + iy = if(\tau) + \lambda f^1(\tau) - \frac{\lambda^2}{2!} if^2(\tau) - \frac{\lambda^3}{3!} f^3(\tau) + \frac{\lambda^4}{4!} if^4(\tau) + \frac{\lambda^5}{5!} f^5(\tau) + \\ \frac{\lambda^6}{6!} if^6(\tau) - \frac{\lambda^7}{7!} f^7(\tau) + \frac{\lambda^8}{8!} if^8(\tau) + \ldots . \end{aligned} \tag{3.9}$$

The successive differentiation of $f(\tau)$ is a lengthy process in which it is convenient to establish the following relationships:

$$N' = (N - R)\tan \mu, \quad R' = 3\frac{R}{N}(N - R)\tan \mu, \quad \left[\frac{N}{R}\right]' = -\frac{2(N - R)}{R}\tan \mu,$$

$$\frac{d\mu}{d\tau} = \frac{N}{R}\cos \mu, \qquad (N \cos \mu)' = -R \sin \mu, \quad \text{and}$$

$$(N \sin \mu)' = \sec \mu(N - R \sin^2\mu) = (R \cos \mu)/(1 - \varepsilon^2). \tag{3.10}$$

From equation (3.4)

$$f^1(\tau) = N \cos \mu. \tag{3.11}$$

Successive differentiation yields

$$f^2(\tau) = -\frac{N}{2} \sin 2\mu, \tag{3.12}$$

$$f^3(\tau) = -\frac{N}{4}\left[3\left(\frac{N}{R} - 1\right) \cos \mu + \left(\frac{N}{R} + 1\right) \cos 3\mu\right], \tag{3.13}$$

and so forth. Because of the very cumbersome nature of the higher order derivatives, additional terms are not given here. However, it can be shown that the function of τ given by equation (3.2) is analytic (that is, all higher order derivatives exist in the selected region, and the function can be represented by a Taylor series). Complete calculations of all derivatives through the eighth are provided in reference 1.

Substituting $t = \tan \mu$ and $\eta^2 = [\varepsilon^2/(1 - \varepsilon^2)](\cos^2\mu)$, separating the real and imaginary parts of equation (3.9), and substituting all derivatives through the eighth yields

$$\begin{aligned} x = {} & N\lambda \cos \mu + \frac{N\lambda^3\cos^3\mu}{6}(1 - t^2 + \eta^2) + \frac{N\lambda^5\cos^5\mu}{120}(5 - 18t^2 + t^4 + 14\eta^2 - \\ & 58t^2\eta^2 + 13\eta^4 - 64t^2\eta^4 + 4\eta^6 - 24t^2\eta^6) + \\ & \frac{N\lambda^7\cos^7\mu}{5040}\begin{bmatrix} 61 - 479t^2 + 179t^4 - t^6 + 331\eta^2 - 3{,}298\eta^2t^2 + 1771\eta^2t^4 + 715\eta^4 \\ - 8655t^2\eta^4 + 6080t^4\eta^4 + 769\eta^6 - 10964t^2\eta^6 + 9480t^4\eta^6 + 412\eta^8 \\ - 6760t^2\eta^8 + 6912t^4\eta^8 + 88\eta^{10} - 1632t^2\eta^{10} + 1920t^4\eta^{10} \end{bmatrix} \end{aligned}$$

and (3.14)

$$\begin{aligned} y = {} & S_\mu + \frac{N\lambda^2}{2} \sin \mu \cos \mu + \frac{N\lambda^4}{24} \sin \mu \cos^3\mu(5 - t^2 + 9\eta^2 + 4\eta^4) + \\ & \frac{N\lambda^6}{720} \sin \mu \cos^5\mu(61 - 58t^2 + t^4 + 270\eta^2 - 330t^2\eta^2 + 445\eta^4 - 680t^2\eta^4 + \\ & 324\eta^6 - 600\, t^2\eta^6 + 88\eta^8 - 192t^2\eta^8) + \\ & \frac{N\lambda^8}{40320} \sin \mu \cos^7\mu \begin{bmatrix} 1385 - 3111t^2 + 543t^4 - t^6 + 10899\eta^2 - 32802t^2\eta^2 \\ + 9219t^4\eta^2 + 34419\eta^4 - 129087t^2\eta^4 + 49644t^4\eta^4 - \\ + 56385\eta^6 - 252084t^2\eta^6 + 121800t^4\eta^6 + 50856\eta^8 \\ - 263088t^2\eta^8 + 151872t^4\eta^8 + 24048\eta^{10} - 140928t^2\eta^{10} \\ + 94080t^4\eta^{10} + 4672\eta^{12} - 30528t^2\eta^{12} + 23040t^4\eta^{12} \end{bmatrix}. \end{aligned} \tag{3.15}$$

Thus, equations (3.2) have been shown to map spheroid points, defined in terms of longitude and isometric latitude, onto a rectangular coordinate (x, y) plane.

The function is analytic at all points, and equations (3.14) and (3.15) have been derived using initial conditions set for the transverse Mercator projection. Furthermore, it was shown earlier that if an analytic function, employing equations (3.2), can be found to satisfy a selected set of initial conditions, the resulting mapping must also be orthomorphic. Hence, the general form of the transverse Mercator projection provides true measurements on the central meridian while also preserving angular measurements throughout the mapped areas.

Equations (3.14) and (3.15) are used in the geodetic program to compute transverse Mercator x and y values. Several simplified forms of equations (3.14) and (3.15) are given in references 1 to 3. Since time was not a critical factor in the baseline program, it was decided to use equations (3.14) and (3.15) without simplification so that the highest levels of accuracy could be maintained in the results. At the present time, the baseline programs are designed to operate on a 48-bit (12-decimal-digit) system. The programs may be converted at some future date to operate on a 64-bit double precision system, and in such case the additional terms will affect the results.

Inverse Calculations

It is now desirable to develop formulas for τ and λ in terms of the rectangular coordinates x and y. The inverse function may be written as

$$\lambda + i\tau = F(x + iy). \tag{3.16}$$

Again applying the initial condition that $\lambda = 0$ when $x = 0$, equation (3.16) becomes

$$F(iy) = (i\tau). \tag{3.17}$$

Using a Taylor series to expand $F(x + iy)$ about a point iy yields

$$\lambda + i\tau = F(iy) + xF^1(iy) + \frac{x^2}{2!}F^2(iy) + \frac{x^3}{3!}F^3(iy) + \frac{x^4}{4!}F^4(iy) + \frac{x^5}{5!}F^5(iy) + \frac{x^6}{6!}F^6(iy) + \frac{x^7}{7!}F^7(iy) + \frac{x^8}{8!}F^8(iy) + \dots \tag{3.18}$$

(Note that, as used in equation (3.18) and others to follow, the terms $F^2(iy)$, $F^3(iy)$, ... signify the second, third, ..., derivatives of $F(iy)$.)

Since for the initial conditions set forth $F(iy) = i\tau$, it can be shown that

$$F^1(iy) = \tau', \qquad F^2(iy) = -i\tau'', \qquad F^3(iy) = -\tau''', \qquad F^4(iy) = -\tau'''', \qquad F^5(iy) = i\tau''''', \tag{3.19}$$

and so forth. Equating the real and imaginary parts of equation (3.18) yields

$$\lambda = x\tau_1' - \frac{x^3}{3!}\tau_1''' + \frac{x^5}{5!}\tau_1''''' - \frac{x^7}{7!}\tau_1''''''' + \ldots \tag{3.20}$$

and

$$\tau = \tau_1 - \frac{x^2}{2!}\tau_1'' + \frac{x^4}{4!}\tau_1'''' - \frac{x^6}{6!}\tau_1'''''' + \frac{x^8}{8!}\tau_1'''''''' - \ldots \tag{3.21}$$

The subscript 1 in equations (3.20) and (3.21) refers to the latitude of the footpoint. The footpoint is the horizontal projection of the target point onto the central meridian. More simply stated, it is the point which would be obtained if the transverse Mercator x coordinate were zero but the y coordinate remained that of the target point. The relation of the footpoint to the target point is shown in figure 3.1.

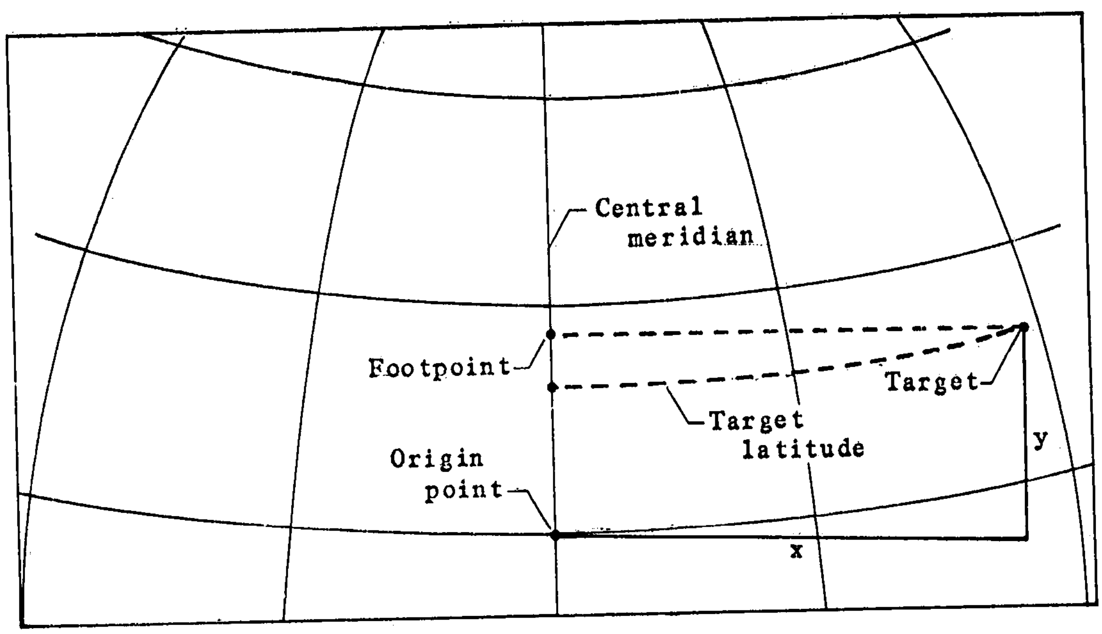

Figure 3.1.

Again, the various derivatives of τ are provided from the equation

$$\tau = \int_0^{\mu} \frac{R}{N} \sec \mu \, d\mu = \underline{S_\mu = \int_0^{\mu} R \, d\mu}, \tag{3.22}$$

from which

$$\tau' = \frac{d\tau}{dS} = \frac{1}{N \cos \mu}, \quad \tau'' = \frac{(N \cos \mu)'}{N^2 \cos^2\mu} \frac{d\mu}{dS}, \text{ and } \frac{d\mu}{dS} = \frac{1}{R}. \tag{3.23}$$

Continuing the differentiation yields

$$\begin{aligned} \tau''' &= \frac{N \cos^2\mu - 2\sin \mu (N \cos \mu)'}{N^3 \cos^3\mu} \frac{d\mu}{dS} \\ &= \frac{\frac{N}{R} \cos^2\mu + 2\sin^2\mu}{N^3 \cos^2\mu} \\ &= \frac{1}{N^3 \cos \mu} (1 + 2t^2 \pm \eta^2), \end{aligned} \tag{3.24}$$

where $t = \tan \mu$, $\eta^2 = \delta \cos^2\mu$, and $\delta = \varepsilon^2/(1-\varepsilon^2)$.

Again, the higher order derivatives become very cumbersome and are omitted from this document for the sake of brevity. For those who may wish to inquire further, complete calculations of all derivatives through the eighth order are provided in reference 1.

At this point it should be noted that since the value of longitude obtained from equation (3.20) is actually the difference in longitude between the target coordinate and the central meridian, in the equations to follow the term $\Delta\lambda$ will be substituted for λ to eliminate confusion. Substitution of the higher order derivatives into equations (3.20) and (3.21) now yields

$$\Delta\lambda = \sec \mu_1 \left[\begin{aligned} &\frac{x}{N_1} - \frac{1}{6}\left[\frac{x}{N_1}\right]^3 (1 + 2t_1^2 + \eta_1^2) \\ &+ \frac{1}{120}\left[\frac{x}{N_1}\right]^5 \left[\begin{aligned} &5 + 6\eta_1^2 + 28t_1^2 - 3\eta_1^4 + 8t_1^2\eta_1^2 \\ &+ 24t_1^4 - 4\eta_1^6 + 4t_1^2\eta_1^4 + 24t_1^2\eta_1^6 \end{aligned} \right] \\ &- \frac{1}{5040}\left[\frac{x}{N_1}\right]^7 \left[\begin{aligned} &61 + 662t_1^2 + 1320t_1^4 + 720t_1^6 + 107\eta_1^2 \\ &+ 43\eta_1^4 + 440t_1^2\eta_1^2 + 97\eta_1^6 - 234t_1^2\eta_1^4 \\ &+ 336t_1^4\eta_1^2 + 188\eta_1^8 - 772t_1^2\eta_1^6 - 192t_1^4\eta_1^4 \\ &+ 88\eta_1^{10} - 2392t_1^2\eta_1^8 + 408t_1^4\eta_1^6 \\ &+ 1536t_1^4\eta_1^8 - 1632t_1^2\eta_1^{10} + 1920t_1^4\eta_1^{10} \end{aligned} \right] \end{aligned} \right] \tag{3.25}$$

and

$$\Delta\tau = t_1 \sec \mu_1 \left[\begin{array}{l} -\frac{1}{2}\left[\frac{x}{N_1}\right]^2 + \frac{1}{24}\left[\frac{x}{N_1}\right]^4 (5 + \eta_1^2 + 6t_1^2 - 4\eta_1^4) \\ -\frac{1}{120}\left[\frac{x}{N_1}\right]^6 \left[\begin{array}{l} 61 + 46\eta_1^2 + 180t_1^2 - 3\eta_1^4 \\ + 48t_1^2\eta_1^2 + 120t_1^4 + 100\eta_1^6 \\ - 36t_1^2\eta_1^4 - 96t_1^2\eta_1^6 + 88\eta_1^8 - 192t_1^2\eta_1^8 \end{array} \right] \\ + \frac{1}{40320}\left[\frac{x}{N_1}\right]^8 \left[\begin{array}{l} 1385 + 7266t_1^2 + 1731\eta_1^2 + 10920t_1^4 \\ + 4416t_1^2\eta_1^2 - 573\eta_1^4 + 5040t_1^6 - 1830t_1^2\eta_1^4 \\ + 2688t_1^4\eta_1^2 - 2927\eta_1^6 + 5052t_1^2\eta_1^6 \\ - 1536t_1^4\eta_1^4 - 8808\eta_1^8 + 27456t_1^2\eta_1^8 \\ + 744t_1^4\eta_1^6 - 11472\eta_1^{10} + 53952t_1^2\eta_1^{10} \\ - 7872t_1^4\eta_1^8 - 4672\eta_1^{12} + 30528t_1^2\eta_1^{12} \\ - 24960t_1^4\eta_1^{10} - 23040t_1^4\eta_1^{12} \end{array} \right] \end{array} \right]. \tag{3.26}$$

A difficulty exists with equation (3.2) in that isometric latitude is obtained instead of geodetic latitude. Thus, it is necessary to expand $\Delta\mu$ with respect to $\Delta\tau$ as follows:

$$\Delta\mu = \mu - \mu_1 = \Delta\tau \frac{d\mu_1}{d\tau_1} + \frac{\Delta\tau^2}{2!}\frac{d^2\mu_1}{d\tau_1} + \frac{\Delta\tau^3}{3!}\frac{d^3\mu_1}{d\tau_1} + \frac{\Delta\tau^4}{4!}\frac{d^4\mu_1}{d\tau_1} + \ldots . \tag{3.27}$$

Starting with the relation

$$R_1 d\mu_1 = N_1 \cos \mu_1 d\tau_1 , \tag{3.28}$$

the derivatives of μ_1 with respect to τ_1 can be expressed as

$$\frac{d\mu_1}{d\tau_1} = \frac{N_1}{R_1} \cos \mu_1 = (1 + \eta_1^2) \cos \mu_1 , \tag{3.29}$$

$$\frac{d^2\mu_1}{d\tau^2} = \frac{N_1}{R_1} t_1 \cos^2\mu_1 \left[2 - 3\frac{N_1}{R_1}\right] = -(1 + \eta_1^2)(1 + 3\eta_1^2) t_1 \cos^2\mu_1 , \tag{3.30}$$

and so forth. A complete derivation of the higher derivatives of μ with respect to τ is provided in reference 1.

Substituting the higher derivatives of μ with respect to τ and the relation for $\Delta\tau$ given in equation (3.26) into equation (3.27), the various coefficient terms can be eventually reduced to yield the following equation for μ based on x and the footpoint latitude:

$$\mu = \mu_1 - \frac{t_1}{2}(1 + \eta_1^2)\left[\frac{x}{N_1}\right]^2 + \frac{t_1}{24}(1 + \eta^2)\left[\frac{x}{N_1}\right]^4 (5 + 3t^2 + \eta^2 - 4\eta^4 - 9\eta^2 t^2)$$

$$- \frac{t_1}{720}(1 + \eta_1^2)\left[\frac{x}{N_1}\right]^6 \left[\begin{array}{l} 61 + 90t_1^2 + 46\eta_1^2 + 45t_1^4 - 252t_1^2\eta_1^2 - 3\eta_1^4 + 100\eta_1^6 \\ - 66t_1^2\eta_1^4 - 90t_1^4\eta_1^2 + 88\eta_1^8 + 225t_1^4\eta_1^4 + 84t_1^2\eta_1^6 \\ - 192t_1^2\eta_1^8 \end{array}\right]$$

$$+ \frac{t_1}{40320}(1 + \eta_1^2)\left[\frac{x}{N_1}\right]^8 (1385 + 3633t_1^2 + 4095t_1^4 + 1575t_1^6). \qquad (3.31)$$

Now, recalling that the footpoint lies on the central meridian where the scale is true (that is, where meridional distance is equal to the y coordinate distance), the meridional arc of the footpoint can be easily determined from

$$y_1 = S_{\mu_1} = \int_0^{\mu} R_1 d\mu = a(1 - \varepsilon^2)\int_0^{\mu} 1/(1 - \varepsilon^2\sin^2\mu_1)^{3/2}. \qquad (3.32)$$

In expanded form,

$$y_1 = S_{\mu_1} = a(1 - \varepsilon^2) \int_0^{\mu} (1 + \frac{3}{2}\varepsilon^2\sin^2\mu_1 + \frac{15}{8}\varepsilon^4\sin^4\mu_1 + \frac{105}{48}\varepsilon^6\sin^6\mu_1$$

$$+ \frac{945}{384}\varepsilon^8\sin^8\mu_1 + \frac{2079}{576}\varepsilon^8\sin^8\mu_1 \ldots)d\mu. \qquad (3.33)$$

In the baseline program, a method was devised whereby a starting value of μ is obtained by the relation

$$\mu = y/R_o, \qquad (3.34)$$

where R_o is the north-south radius of curvature at the map's origin point. This value of μ is then used in equation (3.33) to obtain a trial value of y. The difference between the trial value and the actual value of y is then used in a corrector equation to obtain an improved value for μ.

The corrector equation used in the baseline program is

$$\mu_{n+1} = \mu_n + \left[\frac{S_o + y - S_n}{y}\right]\frac{y}{R_n}, \qquad (3.35)$$

where S_o is the length of the meridional arc from the equator to the origin latitude, S_n is the length of the meridional arc computed using the nth value of μ, R_n is the north-south radius of curvature computed using the nth value of μ, and y is the transverse Mercator y coordinate of the target point. Since the footpoint is the target point's projection on the central meridian (the y axis), it is obvious that the computed meridional arc must equal the distance of the origin point's meridional arc plus the value of the y coordinate (fig. 3.2). Thus, when a value is obtained that causes the bracketed term in equation (3.35) to go to zero, the correct latitude for the footpoint has been obtained.

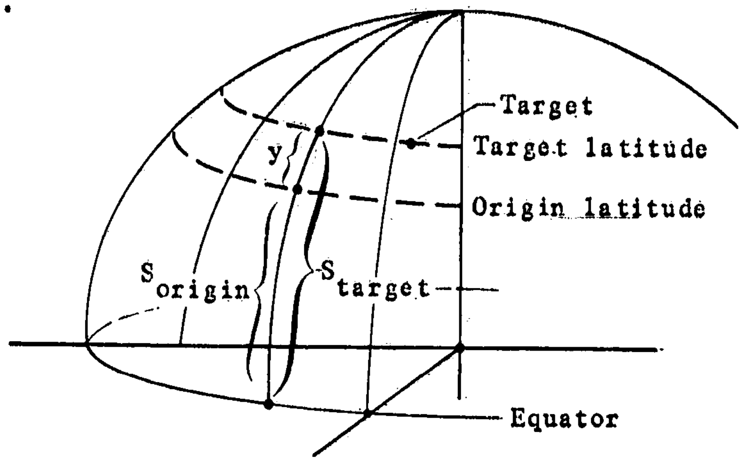

Figure 3.2.

Successive iterations and corrections are performed until the trial value of y equals the true value of y to within the precision limits of the 12-digit computational system. This generally takes three to four iterations.

Integration of equation (3.33) is accomplished by a recursive technique using the relation

$$\int \sin^n \mu \, d\mu = -\frac{\sin^{n-1} \mu \cos \mu}{n} + \frac{n-1}{n} \int \sin^{n-2} \mu \, d\mu. \tag{3.36}$$

Thus, for the inverse solution, the value of y is used to obtain the footpoint latitude. When the footpoint latitude is obtained, the program then computes the various subscripted terms needed for equations (3.25) and (3.31) and solves those equations for the true values of μ and λ. Again it is necessary to note that simplified forms of equations (3.25) and (3.31) are presented in reference 1. However, to retain the greatest possible accuracy and allow for future upgrades in the baseline computational equipment to 64-bit precision, it was decided that the complete form of equations (3.25) and (3.31) would be used in the baseline program.

Nevada Mercator

The transverse Mercator plane coordinate system for the State of Nevada consists of three zones. Transformations for points in each of the three zones are accomplished in the same manner as described for a general transverse Mercator projection except that a bias factor of 500,000 I.S. feet is added to the x coordinate in all zones, and the grid length is arbitrarily reduced by a factor of 1/10,000 to reduce overall scale error.

The reduced map scale compensates for magnification values at points off the map's central meridian. In a general transverse Mercator projection, the map scale is true along the central meridian but magnified for all points off the central meridian (fig. 3.3(a)). With grid lengths along the central meridian reduced by a factor of 1/10,000, the grid lengths are exact at about 56 miles from the central meridian and are 1/10,000 too large at about 79 miles from the central meridian (fig. 3.3(b)). Thus, for an east-west band of about 158 miles, the scale never differs from 1 by more than one part in 10,000. Obviously, without the scale reduction, field measurements would require greater adjustments at some points.

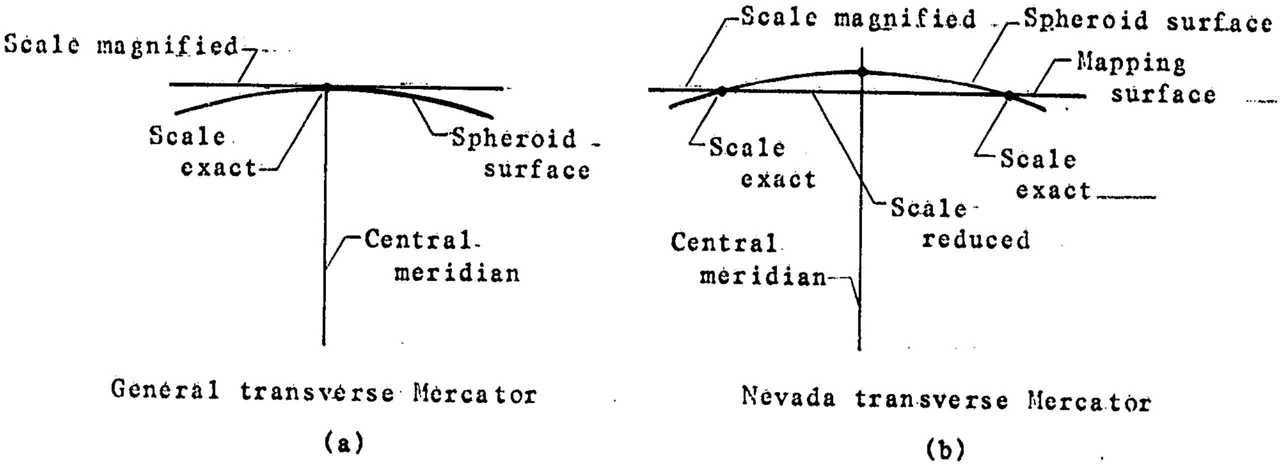

Figure 3.3.

The lines of separation between the Nevada zones extend approximately north and south, following county boundaries. The zones are designated east, central, and west, with maximum longitudinal coverage for any one zone no greater than about 3 degrees. Table 3.1 lists the basic transverse Mercator parameters used for each of the three zones.

TABLE 3.1. NEVADA SURVEY ZONES

Zone	Olat	Olon	X bias	Scale ratio
East	34 45 00.0	115 35 00.0	500,000	1:10,000
Central	34 45 00.0	116 40 00.0	500,000	1:10,000
West	34 45 00.0	118 35 00.0	500,000	1:10,000

Transverse Mercator Program

This section presents the essential computational routines used for both the forward and reverse transverse Mercator transformations.

Variable Names

Name	Description
A	Greek eta in Mercator equations where $\eta = [\varepsilon^2/(1 - \varepsilon^2)](\cos^2\mu)$
A2, A4, ...	A^2, A^4, ...
Aa	Semimajor axis of selected earth spheroid
Arc	Length of meridional arc
Arcsave	Length of meridional arc from latitude 0 to the origin point
Cor	Scale factor correction (0.9999 if Nevada)
Cor1	Bias factor in meters
Cos	Cosine of target latitude ($\cos \mu$)
Cos2, Cos4, ...	$\cos^2\mu$, $\cos^4\mu$, ...
D	Greek delta in Mercator equations where $\delta = \varepsilon^2/(1 - \varepsilon^2)$
Den	Denominator in meridional arc integration term
E2, E4, ...	ε^2, ε^4, ...
Ee	Eccentricity of selected earth spheroid, ε
Exp	Exponent in meridional arc integration term
Flg1	Program flag indicating first pass through meridional arc computation
I	Subscript of ith meridional arc integration term
Imult	(2*I-1)/(2*I)
Integ1	Accumulated value of integral
Integ2	Final value of integral on exit from subroutine

Iterm	Part of ith term in numerical integration
L	Difference between target and origin longitude, in radians
L2, L4, ...	L^2, L^4, ...
Lat	Target latitude in degrees
Lat0	Origin latitude in degrees
Lon	Target longitude in degrees
Lon0	Origin longitude in degrees
Mult	Multiplier in meridional arc integration term
N	East-west radius of curvature
Nuarc	Intermediate value of meridional arc
Num	Numerator in meridional arc integration term
P	Target latitude in radians
Pcor	Iterative correction to P in footpoint latitude calculations
Psave	Saved value of origin latitude in radians
R	North-south radius of curvature
Radius	Meridional radius
Sin	Sine of target latitude (sin μ)
Sin2, Sin4, ...	$\sin^2\mu$, $\sin^4\mu$, ...
T	Tangent of target latitude (tan μ)
Term1, 2, 3, ...	Parts of infinite series terms in the forward calculations
Terma, b, c, ...	Parts of infinite series terms in the reverse calculations
Ucnv	Unit conversion factor
X	Transverse Mercator x in meters and without bias
X2, X4, ...	X^2, X^4, ...

Xa	Transverse Mercator x in selected units
Y	Transverse Mercator y in meters
Y0	Length of meridional arc from latitude 0 to the origin point
Y2, Y4, ...	Y^2, Y^4, ...
Ya	Transverse Mercator y in selected units

Computational Algorithms

A. Transverse mercator initialization: In the manual mode, the operator enters the origin latitude and longitude (Lat0 and Lon0). In the Nevada mode, the program automatically uses stored values of origin latitude and longitude and proceeds with the following initialization routine. In this routine, the radian (RAD) mode is set and the scale factor (Cor) and bias factor (Cor1) terms are set. Note that if a general transverse Mercator solution is to be obtained, Cor is set to 1. If a Nevada transverse Mercator solution is desired, Cor is set to 0.9999 to provide the 1/10,000 scale reduction shown in figure 3.3. Cor1 is set to 0 for general transverse Mercator solutions and to 500,000 for Nevada transverse Mercator solutions. This adds an X bias of 500,000 I.S. feet (table 3.1) when Nevada survey solutions are desired. Since all computations in the baseline program are carried out in meters, Cor1 is multiplied by a factor of 1200/3937 to convert the bias term to meters. In step 7 the origin latitude (Lat0) is converted to radian measure (P) for use in the meridional arc calculation performed by subroutine Arc. Subroutine Arc implements the recursive equation (3.36) to compute the meridional arc length from the equator to the latitude P. Since the origin lies on the central meridian, the origin's X coordinate (X0) and the longitude difference (L) are both set to 0. Finally, recalling that the scale is true along the central meridian of a transverse mercator projection, the length of the meridional arc from the equator to the origin must be exactly equal to the Mercator y coordinate (Y0) measured along the central meridian from the equator to the origin point (step 11).

```
1.  RAD
2.  Cor=1
3.  Cor1=0
4.  IF Nev=0 THEN 7
5.  Cor=.9999
6.  Cor1=500000*1200/3937
7.  P=Lat0*2*PI/360
8.  L=0
9.  GOSUB Arc
10. X0=0
11. Y0=Arc
```

B. Geodetic to transverse mercator routine: The operator enters the target latitude and longitude (Lat and Lon). Again, the radian mode is set and target latitude and longitude are converted to radian measure (steps 3 and 4). Next, subroutine Param is called to compute the various trigonometric and power terms required by equations (3.14) and (3.15). Subroutine Xycalc then implements equations (3.14) and (3.15) to yield values for the target's transverse Mercator X and Y coordinates. Since the Y coordinate is computed from the equator, the equator-to-origin distance must be subtracted from the equator-to-target distance (step 8) to obtain the value (Y) of the origin-to-target Y coordinate. Finally, the scale and bias corrections are applied (if a Nevada Mercator solution is to be obtained), and the conversion factor (Ucnv) is applied to both coordinates to convert them to the selected output units of length.

1. L=Lon0-Lon
2. RAD
3. P=Lat*2*PI/360
4. L=L*2*PI/360
5. GOSUB Param
6. GOSUB Arc
7. GOSUB Xycalc
8. Y=Y-Y0
9. X1=(X*Cor+Cor1)*Ucnv
10. Y1=(Y*Cor)*Ucnv
11. GOTO Gtmprint

C. Transverse Mercator to geodetic routine: The operator inputs scaled transverse Mercator coordinates in selected units (X1, Y1). These units are converted to meters by application of the conversion factor (Ucnv). In the same steps, the scale and bias corrections (Cor and Cor1) are also applied when Nevada survey solutions are to be obtained. Otherwise, Cor is set to 1 and Cor1 is set to 0. Flag 1 (Flg1) is also set to 1 to signal the first pass through the recursive solution so that an approximate starting value of the footpoint latitude will be computed in step 10. The origin's latitude and Y coordinate are saved (Psave and Arcsave) so that the same transfer variables (P and Arc) can be used for the target coordinates. The N-S radius of curvature of the footpoint is first approximated using the N-S radius of curvature (R) of the origin point. The computed value of R and the target's Y coordinate are used in step 10, which implements equation (3.34). After the first approximation of latitude is obtained, flag 1 (Flg1) is reset to 0 so that step 10 will be bypassed on subsequent corrector passes. Subroutine Arc is then called and returns a computed meridional arc length for the first approximation of footpoint latitude derived in step 10. Step 11 implements equation (3.35). In this step, the returned equator-to-target arc length is compared with the sum of the target's true Y coordinate (measured from the origin point) and the equator-to-origin distance. Since the footpoint lies on the central meridian where the scale is true, it is apparent that when a correct value for μ is obtained, the equator-to-origin meridional distance plus the origin-to-footpoint meridional distance must equal the equator-to-footpoint meridional distance. So long as this is not true, step 11 (which implements equation (3.35)) will generate a correction (Pcor) to be applied to the current value of μ (P). After the correction factor is applied, the program returns to step 6 to recompute R based on the corrected value of P. The new value of P is then used by subroutine Arc, which computes a more accurate value of the meridional arc distance to the footpoint. Again, this value is used in step 11 to generate a second correction for μ (P). This process is repeated until the numerator within the bracketed term in equation (3.35) is essentially equal to 0. When this occurs, P will be at the value of the correct footpoint latitude. Knowing the correct footpoint latitude, the various trigonometric and power terms required by equations (3.25) and (3.26) can be computed by subroutine Param. Subroutine Gcalc is then called to implement equations (3.25) and (3.31).

```
 1.  X=(X1/Ucnv-Cor1)/Cor
 2.  Y=Y1/(Ucnv*Cor)
 3.  Flg1=1
 4.  Psave=P
 5.  Arcsave=Y0
 6.  Sin=SIN(P)
 7.  Sin2=Sin*Sin
 8.  R=Aa*(1-E2)/SQR(1-E2*Sin2)**3
 9.  IF Flg1=0 THEN 12
10.  P=Y/R+Psave
11.  Flg1=0
12.  GOSUB Arc
13.  IF ABS(Arcsave+Y-Arc)<.00002 THEN 17
14.  Pcor=(Arcsave+Y-Arc)/Y*(Y/R)
15.  P=P+Pcor
16.  GOTO 6
17.  GOSUB Param
18.  GOSUB Gcalc
19.  GOTO Mtgprint
```

D. Subroutine Arc: This subroutine computes the length of a meridional arc from latitude 0 to latitude μ. Arc lengths are computed by use of the Taylor expansion given in equation (3.33) with the integrations being performed using the recursive integration formula of equation (3.36). The recursive calculations performed in the loop between step 6 and step 18 add one additional integration term of equation (3.33) on each pass. Since the Taylor series is convergent, each succeeding term in equation (3.33) is smaller than the preceeding term. Thus, when the currently computed term is zero, the overall result is accurate to the precision limit of the system. As shown in step 17 below, the baseline program makes as many passes as necessary to reach the precision limit of the computer.

In step 1, the term $a(1 - \varepsilon^2)$ is computed. Next, the first integral term, $\int d\mu$, is evaluated from 0 to μ, simply yielding μ (P) which is then multiplied by $a(1 - \varepsilon^2)$ at step 2 to yield the total contribution to the arc length arising from the first term of the Taylor series. Num and Den (step 3) are used to form the numeric fractions shown as multipliers in each succeeding term in the Taylor expansion. For I = 1 (step 5), the numerator (Num) is computed to be 3 and the denominator (Den) is computed to be 2, the proper numeric values for the multiplier of the second integral term. The numeric values are combined with the proper ε power term in step 8 which, for I = 1, yields the second-term multiplier, $\frac{3}{2}\varepsilon^2$, denoted Mult. Next, the integral of $\sin^2\mu \, d\mu$ is obtained through the use of the recursive relations provided in equation (3.36). In this case, the power (n) equals 2. Imult is used to compute the numeric multiplier of the integral to the right of the equal sign in equation (3.36), which for the n = 2 case equals 1/2. Exp is the exponent (n - 1) of the sine term immediately after the equal sign. In this case, when n equals 2, the exponent of the sine is equal to 1. Integ2 is the integral in the right-hand part of equation (3.36). Recall that on the first pass, Integ was set equal to P. In solving for the second term of equation (3.33), where n takes on the value of 2, the sine exponent is n - 2, reducing the integral $\int \sin^{n-2}\mu \, d\mu$ to nothing more than $\int d\mu$, which is identical to the first integral in the equation (3.33), which was stored as Integ2 in step 4. Thus, for the case where n = 2, step 11 yields the value of the right-hand integral term of equation (3.33), and the complete value of equation (3.33) is obtained in step 12. The contribution to the total arc length from the second term is obtained in step 13 by multiplying the solution of the integral of $\sin^2\mu$ just obtained by Mult ($3\varepsilon^2/2$) and $a(1 - \varepsilon^2)$. The total arc length (Arc) is then obtained by adding the value from the first term (Arc) to the contribution from the second term (Nuarc). Finally, at step 15, the variable Integ2 is set equal to the value just computed for $\int \sin^2\mu \, d\mu$, I is indexed by 1, and the value of Nuarc is tested. If Nuarc is found to be zero, the precision limit of the system has been reached and the program returns to the calling point. Otherwise, the program remains in the Arc loop and solves for the next term in equation (3.33). It is important to note that, in each succeeding pass, Integ2 is precisely the value of the integral term on the right side in equation (3.36), meaning that only the first term and simple numeric multipliers must be computed. Thus, when I = 2, step 8 yields a value for Mult of $\frac{15}{8}\varepsilon^4$, step 12 yields a value for $\int \sin^4\mu \, d\mu$, and step 14 updates the value of the meridional arc (Arc) to include the contributions from the first three terms of the Taylor series. The subroutine continues to compute additional terms until the value of Nuarc becomes 0, at which time the program returns to the calling point.

```
1.  Radius=Aa*(1-E2)
2.  Arc=P*Radius
3.  Num=Den=1
4.  Integ2=P
5.  I=1
6.  Num=Num*(I*2+1)
7.  Den=Den*(I*2)
8.  Mult=Num/Den*E2**I
9.  Exp=2*I-1
10. Imult=(2*I-1)/(2*I)
11. Iterm=Imult*Integ2
12. Integ1=-SIN(P)**Exp*COS(P)/(2*I))+Iterm
```

13. Nuarc=Mult*Integ1*Aa*(1-E2)
14. Arc=Arc+Nuarc
15. Integ2=Integ1
16. I=I+1
17. IF Nuarc=0 THEN RETURN
18. GOTO 6

E. Subroutine Param: This subroutine computes the various trigonometric values required by both the forward and reverse transverse Mercator routines. Additional definitions of these terms may be found in the variables list.

1. Sin=SIN(P)
2. Sin2=Sin*Sin
3. Cos=COS(P)
4. Cos2=Cos*Cos
5. Cos3=Cos*Cos2
6. Cos5=Cos3*Cos2
7. Cos7=Cos5*Cos2
8. L2=L*L
9. L3=L2*L
10. L4=L2*L2
11. L5=L4*L
12. L6=L4*L2
13. L7=L6*L
14. L8=L4*L4
15. N=Aa/SQR(1-E2*Sin2)
16. R=Aa*(1-E2)/SQR(1-E2*Sin2)**3
17. D=E2/(1-E2)
18. A2=D*Cos2
19. A4=A2*A2
20. A6=A4*A2
21. A8=A4*A4
22. A10=A8*A2
23. A12=A10*A2
24. T=TAN(P)
25. T2=T*T
26. T4=T2*T2
27. T6=T4*T2
28. RETURN

F. Subroutine Xycalc: This subroutine compiles all the infinite series terms required for the computation of transverse Mercator x and y coordinates from geodetic coordinates entered by the operator. Term1, Term2, and Term3 are the multiplying terms of the third, fifth, and seventh power functions of N, λ, and μ in equation (3.14). Term4, Term5, and Term6 are the multiplying terms of the fourth, sixth, and eighth power functions of N, λ, and μ in equation (3.15). Step 7 is a direct implementation of equation (3.14), and Step 16 is a direct implementation of equation (3.15).

1. Term1=1-T2+A2
2. Term2=5-18*T2+T4+14*A2-58*T2*A2+13*A4-64*T2*A4+4*A6-24*T2*A6

3. Term3a=61-479*T2+179*T4-T6+331*A2-3298*A2*T2+1771*A2*T4
   +715*A4-8655*T2*A4
4. Term3b=6080*T4*A4+769*A6-10964*T2*A6+9480*T4*A6=412*A8-6760*T2*A8
5. Term3c=6912*T4*A8+88*A10-1632*T2*A10+1920*T4*A10
6. Term3=Term3a+Term3b+Term3c
7. X=N*(L*Cos+L3*Cos3/6*Term1+L5*Cos5/120*Term2±L7*Cos7/5040*Term3)
8. Term4=5-T2+9*A2+4*A4
9. Term5=61-58*T2+T4+270*A2-330*T2*A2+445*A4-680*T2*A4
   +324*A6-600*T2*A6+88*A8-192*T2*A8
10. Term6a=1385-3111*T2+543*T4-T6+10899*A2-32802*T2*A2
11. Term6b=9219*T4*A2+34419*A4-129087*T2*A4+49644*T4*A4
12. Term6c=56385*A6-252084*T2*A6+121800*T4*A6+50856*A8
13. Term6d=-263088*T2*A8+151872*T4*A8+24048*A10-140928*T2*A10
14. Term6e=94080*T4*A10+4672*A12-30528*T2*A12+23040*T4*A12
15. Term6=Term6a+Term6b+Term6c+Term6d+Term6e
16. Y=Arc+N*(L2*Sin*Cos/2+L4*Sin*Cos3/24*Term4
    +L6*Sin*Cos5/720*Term5+L8*Sin*Cos7/40320*Term6)
17. RETURN

G. Subroutine Gcalc: This subroutine compiles the infinite series terms needed to compute the target's geodetic position from transverse Mercator coordinates entered by the operator. Terma, Termb, and Termc are the parentheses-enclosed t and η values multiplying the fourth, sixth, and eighth power terms of equation (3.31). Termd, Terme, and Termf are the same types of multiplying terms found in equation (3.25). Step 13 is an implementation of equation (3.31) with the substitution N/R being made for the $(1 - \eta^2)$ terms (eq. (3.29)). Step 14 is a direct implementation of equation (3.25).

1. Terma=5+3*T2+A2-4*A4-9*A2*T2
2. Termb1=61+90*T2+46*A2+45*T4-252*T2*A2-3*A4+100*A6-66*T2*A4
3. Termb2=-90*T4*A2+88*A8+225*T4*A4+84*T2*A6-192*T2*A8
4. Termb=Termb1+Termb2
5. Termc=1385+3633*T2+4095*T4+1574*T6
6. Termd=1+2*T2+A2
7. Terme=5+6*A2+28*T2-3*A4+8*T2*A2+24*T4-4*A6+4*T2*A4+24*T2*A6
8. Termf=61+662*T2+1320*T4+720*T6
9. X2=X*X
10. X4=X2*X2
11. X6=X4*X2
12. X8=X4*X4
13. Lat=P+T*(-X2/(2*R*N)+X4/(24*R*N**3)*Terma
    -X6/(720*R*N**5)*Termb+X8/(40320*R*N**7)*Termc)
14. Dlon=1/Cos*(X/N-1/6*(X/N)**3*Termd+1/120*(X/N)**5*Terme
    -1/5040*(X/N)**7*Termf)
15. Lat=Lat*360/(2*PI)
16. Lon=Lon0-Dlon*360/(2*PI)
17. RETURN

Program Operation

The transverse Mercator routines are a subprogram to GEOD. When GEOD is run, the operator is asked to select the units and datum/spheroid reference applicable to the computations to be performed. After these selections have been made, the master menu is displayed. One selection is for TRANSVERSE MERCATOR transformations. The operator selects the appropriate numerical entry and the main program enters the transverse Mercator routines. The operator is then instructed to make the following simple selections:

A. Output device selection, which is displayed as:

SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER

B. Forward or reverse transformation, which is displayed as:

SELECT MODE

1 = GEODETIC TO TRANSVERSE MERCATOR
2 = TRANSVERSE MERCATOR TO GEODETIC

C. Transformation parameter selection, which is displayed as:

SELECT MERCATOR PARAMETERS

0 = MANUAL ENTRY OF MERCATOR PARAMETERS
1 = NEVADA EAST ZONE
2 = NEVADA CENTRAL ZONE
3 = NEVADA WEST ZONE

If the operator selects any of the Nevada zones, the program automatically uses the stored transformation parameters for the zone selected and proceeds to step E or F as appropriate.

D. Manual input of transformation parameters, which is displayed as:

ENTER ORIGIN LATITUDE IN D.MS

ENTER ORIGIN LONGITUDE IN D.MS

E. Geodetic to transverse Mercator routine, for which the operator is prompted as follows:

ENTER LATITUDE IN D.MS

ENTER LONGITUDE IN D.MS

The program computes the transverse Mercator coordinates of the point and prints the results on the selected output device. The results are also displayed on the CRT, regardless of output selection. Computed values are displayed in the output units selected, and the format is as shown below:

GEODETIC LATITUDE = 39 00 00.0000
GEODETIC LONGITUDE = 115 00 00.0000

TRANSVERSE MERCATOR X = 665,775.57 I.S. FEET
TRANSVERSE MERCATOR Y = 1,547,730.51 I.S. FEET

When CONT is depressed, the program asks for the geodetic coordinates of the next point.

F. Transverse Mercator to geodetic routine: The operator is prompted to enter transverse Mercator x and y values in the units previously selected.

ENTER VALUES OF TRANSVERSE MERCATOR X AND Y

ENTER X IN I.S. FEET

ENTER Y IN I.S. FEET

The program computes the geodetic coordinates of the point and prints the results. The results appear on both the CRT and on the selected output device unless the CRT was selected for output, in which case the results are displayed only on the CRT. Printout format is as indicated below:

TRANSVERSE MERCATOR X = 665,775.57 I.S. FEET
TRANSVERSE MERCATOR Y = 1,547,730.51 I.S. FEET

GEODETIC LATITUDE = 39 00 00.0000
GEODETIC LONGITUDE = 115 00 00.0000

When CONT is depressed, the program asks for the transverse Mercator coordinates of the next point.

Program Validation

The transverse Mercator routines are validated by comparing results with those in the U.S. Coast and Geodetic Survey plane coordinate intersection tables for the State of Nevada (ref. 4). During a validation exercise, any point may be selected and used in either the forward or reverse programs. Table 3.2 compares several test coordinates from reference 4 with results obtained using the GMD algorithms. The first line of each entry gives the geodetic latitude and longitude of the point along with the published USCGS transverse Mercator coordinates. The second line provides the transverse Mercator coordinates returned for the same point by the GMD routines.

For table 3.3, the published transverse Mercator coordinates for the points as given in table 3.2 were used as input values to the reverse transformation program. The first line of each entry shows the published geodetic latitude and longitude corresponding to the input value. The second line shows the geodetic coordinates obtained by the GMD routines. The third line shows the coordinates that would be obtained using the GMD values obtained from the forward transformation for each point.

Since entirely different algorithms were used for the forward and reverse GMD transformations, since the GMD routines include higher order terms which are usually neglected, and since the values obtained by forward GMD routines return precisely the correct geodetic coordinates when entered into the reverse routines, it is felt that the GMD transverse Mercator algorithms provide accuracies that are equivalent to those obtained from the closed-form Lambert solutions.

TABLE 3.2. GEODETIC TO TRANSVERSE MERCATOR VALIDATIONS

Source	Latitude	Longitude	X coordinate	Y coordinate
		Nevada East Zone		
USCGS	35 00 00.0000	116 00 00.0000	375,217.01	91,241.17
GMD			375,217.02	91,241.16
USCGS	37 00 00.0000	115 00 00.0000	670,340.20	819,487.76
GMD			670,340.20	819,487.75
USCGS	40 00 00.0000	115 30 00.0000	523,345.20	1,911,421.77
GMD			523,345.20	1,911,421.78
		Nevada Central Zone		
USCGS	37 00 00.0000	116 00 00.0000	694,674.80	819,647.52
GMD			694,674.80	819,647.51
USCGS	38 00 00.0000	117 00 00.0000	403,952.51	1,183,223.29
GMD			403,952.52	1,183,223.29
USCGS	41 00 00.0000	116 30 00.0000	546,002.23	2,275,729.94
GMD			546,002.23	2,275,729.93
		Nevada West Zone		
USCGS	38 00 00.0000	117 30 00.0000	812,158.43	1,184,868.37
GMD			812,158.44	1,184,868.28
USCGS	40 00 00.0000	118 00 00.0000	663,416.87	1,911,945.60
GMD			663,416.87	1,911,945.60
USCGS	42 00 00.0000	118 30 00.0000	522,649.99	2,640,036.34
GMD			522,649.99	2,640,036.33

TABLE 3.3. TRANSVERSE MERCATOR TO GEODETIC VALIDATIONS

Source	X coordinate	Y coordinate	Latitude	Longitude
		Nevada East Zone		
USCGS	375,217.01	91,241.17	35 00 00.0000	116 00 00.0000
USCGS/GMD	375,217.01	91,241.17	35 00 00.0001	116 00 00.0001
GMD	375,217.02	91,241.16	35 00 00.0000	116 00 00.0000
USCGS	670,340.20	819,487.76	37 00 00.0000	115 00 00.0000
USCGS/GMD	670,340.20	819,487.76	37 00 00.0001	115 00 00.0000
GMD	670,340.20	819,487.75	37 00 00.0000	115 00 00.0000
USCGS	523,345.20	1,911,421.77	40 00 00.0000	115 30 00.0000
USCGS/GMD	523,345.20	1,911,421.77	39 59 59.9999	115 30 00.0000
GMD	523,345.20	1,911,421.78	40 00 00.0000	115 30 00.0000
		Nevada Central Zone		
USCGS	694,674.80	819,647.52	37 00 00.0000	116 00 00.0000
USCGS/GMD	694,674.80	819,647.52	37 00 00.0001	116 00 00.0000
GMD	694,674.80	819,647.51	37 00 00.0000	116 00 00.0000
USCGS	403,952.51	1,183,223.29	38 00 00.0000	117 00 00.0000
USCGS/GMD	403,952.51	1,183,223.29	38 00 00.0000	117 00 00.0001
GMD	403,952.52	1,183,223.29	38 00 00.0000	117 00 00.0000
USCGS	546,002.23	2,275,729.94	41 00 00.0000	116 30 00.0000
USCGS/GMD	546,002.23	2,275,729.94	41 00 00.0001	116 30 00.0000
GMD	546,002.23	2,275,729.93	41 00 00.0000	116 30 00.0000
		Nevada West Zone		
USCGS	812,158.43	1,184,868.37	38 00 00.0000	117 30 00.0000
USCGS/GMD	812,158.43	1,184,868.37	38 00 00.0009	117 30 00.0001
GMD	812,158.44	1,184,868.28	38 00 00.0000	117 30 00.0000
USCGS	663,416.87	1,911,945.60	40 00 00.0000	118 00 00.0000
USCGS/GMD	663,416.87	1,911,945.60	40 00 00.0000	118 00 00.0000
GMD	663,416.87	1,911,945.60	40 00 00.0000	118 00 00.0000
USCGS	522,649.99	2,640,036.34	42 00 00.0000	118 30 00.0000
USCGS/GMD	522,649.99	2,640,036.34	42 00 00.0001	118 30 00.0000
GMD	522,649.99	2,640,036.33	42 00 00.0000	118 30 00.0000

CHAPTER 4

RANGE AND ANGLE CALCULATIONS

This chapter provides general theory related to the calculation of angles and distances between points on or off the earth's surface. To make such calculations, it is necessary to relate real-world points to corresponding points on one or more representative spheroid models on which a uniform gridwork of parallels and meridians can form the basis for a suitable coordinate reference frame.

Geoid and Spheroid Definitions

Confusion arises from time to time regarding various range and angle measurements made in regard to earth surface, geoid, spheroid, and airborne or spaceborne points. Some of the confusion is the result of ambiguities in the definition of terms, even in the most authoritative references dealing with the subject of geodesy. To attach specific meaning to terms used in this document, a few brief definitions are provided below. Additional information and mathematical derivations can be found in references 1 to 3, but caution should be exercised since, even though the names of certain terms may be the same, the physical definitions may differ slightly.

Geoid

The geoid is generally considered to be the true mean sea-level surface of the earth. It is an equipotential surface that arises as a combination of rotational and gravitational forces but neglects tidal forces caused by the moon or sun. Although the geoid surface closely approximates the shape of a spheroid, irregularities of mass distribution and density within the earth cause the geoid surface to depart slightly from a true spheroidal shape. Departures of up to 80 meters or more can be measured in mountainous regions and areas of heavy mass concentrations, but normal separations are usually much less.

Although the geoid is a physical reality, it is not suitable for use as a reference for locating points on, above, or below the earth's surface. Because of its irregular shape, a uniform grid system of parallels and meridians can not be constructed on its surface. That is, the separation between adjacent parallels and meridians would vary because of undulations in the geoid surface.

Reference Spheroid

The reference spheroid is a pure geometric shape with its center situated at a point which is taken as approximating the center of gravity of the earth. The reference spheroid is defined by the lengths of the semimajor and semiminor axes. Other parameters sometimes associated with the spheroid (but more often associated with a selected datum) are the position of the spheroid center with respect to a standard coordinate reference and the alignment of the spheroid with selected reference points on the surface of the earth. (Note: As used in this document, reference spheroid refers only to the dimensions and eccentricity of the figure.)

Obviously the selection of reference spheroid parameters affects the amount of departure between the spheroid and geoid surfaces. In recent years, due to satellite observations, the ability to make accurate earth measurements has improved considerably, and agreement between modern spheroid models is generally very good (within 10 to 20 meters). Efforts toward further improvements are probably not practical since an exact fit to the slightly irregular geoid surface can never be achieved, regardless of how accurate the measurement systems become.

Datum

As used in this document, datum specifies the location of the origin point of the reference spheroid and its alignment with one or more specified earth reference points.

Normal Line

The normal line is defined as a line constructed perpendicular to the spheroid surface at any point.

Fundamental Plane

The fundamental plane for a spheroid point is defined as the tangent plane to the spheroid at that point. The fundamental plane is used to measure elevation angles for points on, above, or below the reference spheroid. In off-spheroid cases, the fundamental plane is translated to the point of concern along a normal line drawn from the spheroid surface through the point.

Vertical Line

A vertical line is a line constructed perpendicular to the geoid surface at any point. Since the geoid is an equipotential gravitational surface, the vertical line also represents the direction of the gravitational vector at the point where it passes through the geoid surface. At points off the geoid surface, the vertical line follows the direction of the gravitational vector.

Geodetic Position

The geodetic position of a target above, below, or on the earth's surface is defined relative to the position on the reference spheroid at which a normal line would pass through the target point. Geodetic latitude is the angle between the normal line and the spheroid equator. Geodetic longitude is the angle between the meridian plane passing through the spheroid point and another meridian plane passing through an arbitrary zero-longitude reference point.

Separation of Geoid

The separation of geoid at any point is the distance between the geoid and the reference spheroid measured along a vertical line passing through the point. The value is different for different reference spheroids and datums since the shape of the spheroid and the location of the origin affect the position of the spheroid surface with respect to the geoid surface. A negative value for the separation of geoid implies that the geoid surface lies below that of the reference spheroid. For example, in relation to NAD-27, the separation of geoid at the NASA AN/FPS-16 radar site is -24.4 meters.

It should be noted that the sea-level or geoid elevation of a point is a measure of the distance along the curve followed by a gravitational vector going through the point to the surface of the geoid, whereas the spheroid elevation is the distance between the spheroid and the same point measured along the spheroid normal line passing through the point. This means that spheroid elevation and geoid elevation are actually measured along different paths. However, the practice of obtaining spheroid elevation as the sum of sea-level elevation and geoid separation causes no difficulty since the error introduced by this approximation is far less than the uncertainty in the position of the geoid surface. Published geoid separations are normally obtained through the use of a spherical harmonic equation which approximates the gravitational potential field of the earth. The constants used in these equations are obtained through measurements at several hundred reference points. Values at other points are then computed from the approximating equation. Thus, the published values of geoid separation are probably accurate only to about 1 to 2 meters at most locations.

Geodetic Azimuth

Geodetic azimuth is a directional measurement between two spheroid points. For example, the geodetic azimuth from point A to point B is defined as the angle between the meridional plane passing through the point A and a plane containing both the normal line at A and the point B. This is also known as spheroid azimuth.

True Azimuth

True azimuth is a directional measurement between two points on, above, or below the earth's surface. If these two points are C and D, the true azimuth from C to D is defined as the angle between the meridional plane passing through C and the plane containing the normal line at C as well as the point D. If C and D were earth surface points whose spheroid coordinates were given by A and B in the preceding section, then the true azimuth would differ from the geodetic azimuth because the normal line drawn from B through D is not parallel to the normal line drawn from A through C.

Geodesic

The geodesic is defined as the curve of minimum length between two points lying on the surface of a spheroid.

Astronomical Position

Astronomical positions are determined by star observations. The angles are measured with respect to a plumb-bob vertical placed at the point of observation. While it might first appear that this would align the system to the underlying (or overlying) geoid surface, the fact that the observation points are at some specified elevation above or below sea level means that the direction of the gravitational vector is not influenced by the same relative mass geometry as the equivalent geoid point. Therefore, the direction of the vector will be slightly different from the direction it takes at the point where it cuts through the geoid surface.

Astronomical position is also subject to change due to the precession and nutation of the earth's axis of rotation. This phenomenon is caused by three influencing factors whose net effect is that, over a 7-year period, the axis of rotation moves about a mean position with an amplitude from approximately 0 to 0.3 arc second.

The common definition of astronomical latitude is the inclination of the local vertical measurement above the equatorial plane. Astronomical longitude is the angle between one plane containing the axis of rotation and the meridional vertical at the observation point, and another containing the axis of rotation and the Greenwich meridian.

Considering the variations in the position of the rotational axis, the deflection of the gravitational vector from its direction at the underlying (or overlying) point where it cuts through the geoid surface, and the fact that the plane of the local vertical is not exactly parallel to the axis of rotation, it is obvious that some ambiguity is present in the common definition of the astronomical coordinates. Because of this, it seems desirable to correct the ground-level observations to those for the geoid surface. However, this is often impractical because the mass distribution of underlying material is generally not known, and, even if known, the effect of complex mass distributions on the gravitational vector is difficult to compute. Therefore, either the corrections are neglected or standard correction factors are applied. Since these corrections relate to field work which is not the subject of this document, they will not be presented here. However, additional information about astronomical corrections can be found in reference 3. Computations used in the baseline geodetic programs assume that the astronomical coordinates have been corrected to the proper geoid-level values.

Astronomical Azimuth

The astronomical azimuth from point A to point B is defined as the angle between two planes, each of which contains the vertical at point A, but one of which contains the celestial north pole and the other of which passes through point B.

Deflection of Vertical

The deflection of the vertical at any point on the earth's surface is taken as the angular difference between the spheroid normal passing through the point and the plumb-bob vertical at the same point. It is measured in terms of its meridional component and prime vertical components, which are represented by

$$\mu_a - \mu_g \quad \text{and} \quad \lambda_a - \lambda_g, \tag{4.1}$$

respectively, where μ_a represents astronomical latitude, μ_g represents geodetic latitude, λ_a represents astronomical longitude, and λ_g represents geodetic longitude.

Astronomical and geodetic azimuths can also be related by

$$A_a - A_g = -(\lambda_a - \lambda_g)\sin \mu_g \tag{4.2}$$

where A_a is the astronomical azimuth and A_g is the geodetic azimuth.

Sea-Level Elevation

The sea-level elevation of a point is the height of the point above or below the geoid as measured along the gravitational vector passing through the point. Sea-level elevation is generally given with all the various types of coordinate references since the position of the geoid with respect to the point in question is the same regardless of the reference spheroid selected.

Spheroid Elevation

The spheroid elevation of a point is the height of the point above or below the reference spheroid as measured along the normal line passing through the point.

Coordinate Systems

Several coordinate systems are used to perform geodetic computations in the baseline programs. These coordinate systems are explained in the following paragraphs.

Geodetic (Spheroid) Coordinates

Geodetic coordinates are given in terms of latitude, μ, and longitude, λ. As noted above, elevation is generally given as the height above mean sea level (height above the geoid). Thus, in all spheroid calculations, the sea-level elevation must be corrected by the amount of the geoid separation. Spheroid coordinates are generally given with respect to the North American Datum (1927). When coordinates are given with respect to one or more datums, the datum designator (such as NAD-27 or WGS-72) is generally shown alongside each coordinate.

Astronomical Coordinates

Astronomical coordinates are also given in terms of latitude, μ, longitude, λ, and sea-level elevation. When the possibility of confusion exists, astronomical coordinates are labeled as astronomical.

Geocentric Coordinates

Geocentric coordinates are given in terms of geocentric latitude, ξ, and geocentric longitude, λ. Geocentric altitude is the distance from the surface of the spheroid to the target point as measured along the geocentric position vector. Note that, for identical points, the values of geocentric longitude and geodetic longitude are the same.

Universal Space Rectangular Coordinates

In tracking system geodetics, the universal space rectangular coordinate (E-F-G) frame of reference is a right-handed Cartesian system whose origin is at the center of the reference spheroid and whose G axis is coincident with the mean axis of rotation of the reference spheroid. The E axis passes through the Greenwich meridian, and the E and F axes define the equatorial plane. The location of the origin point for the E-F-G system varies from one datum to another. In this document, the WGS-72 datum is taken as the standard, and the centers of all other datums are given in terms of ΔE, ΔF, and ΔG measurements from the WGS-72 origin. Since most tracking system geodetics are referenced to a fixed earth, it is easy to overlook the fact that the E-F-G coordinates apply to a rotating system whose rotation rate is the same as that of the earth.

Local East-North-Vertical Coordinates

The local east-north-vertical coordinate (X-Y-Z) frame of reference is a right-handed Cartesian system that is centered at any point on, above, or below the reference spheroid. The X coordinate radiates in the local east direction measured at the point, the Y coordinate radiates in the north direction along the meridional plane passing through the point, and the Z coordinate radiates upward along the outward normal that passes through the point. The X and Y coordinates lie in the fundamental plane passing through the origin point.

Local Space Rectangular Coordinates

The local space rectangular coordinate (E'-F'-G') frame of reference is a right-handed Cartesian system whose origin is centered at any point on, above, or below the reference spheroid. The E', F', and G' coordinates are parallel to but spatially offset from the E, F, and G coordinates described earlier.

ΔE-ΔF-ΔG Coordinates

The ΔE-ΔF-ΔG coordinates are used to describe the positions of E'-F'-G' origin points in the E-F-G system. They are used for translational purposes.

Local Range-Azimuth-Elevation Coordinates

The local range-azimuth-elevation (R-A-E) coordinate frame is the spherical equivalent of the X-Y-Z Cartesian coordinate system. In this system, azimuth, A, is the angular component of the target position lying in the fundamental plane and measured clockwise from true north. Elevation, E, is the inclination of the target above the fundamental plane. Range, R, is the slant range of the target from the coordinate origin. For radar and optical trackers, the origin of both the R-A-E and X-Y-Z coordinate systems is taken as the intersection of the horizontal and vertical (H-V) axes of the antenna pedestal. Note that the angle measurements made by a tracking system must be corrected by a minor rotation from plumb-bob vertical into alignment with the fundamental plane and spheroid normal. This is accomplished as part of the mislevel corrections.

Subroutines Common to Range and Angle Programs

The range and angle calculations used in the baseline software are based on spheroidal relationships and on a simple application of vector and matrix mathematics. The more common spheroidal relationships were given without proof in equations (1.3) to (1.8) in chapter 1. In the baseline software, standard subroutines are often shared by many of the subprograms. Each of these common subroutine algorithms is provided below and is not repeated in later sections. Also note that the terms U, W, and H used in the subroutines which follow are generalized transfer variables. Individual calling programs will set specific program variables, let us say Lat, Lon, and Elev, equal to U, W, and H prior to calling a specific subroutine.

A. Subroutine Ncalc: Subroutine Ncalc returns a value of the east-west radius of curvature at a specified latitude on the reference spheroid. This subroutine was also presented in chapter 2, but it is repeated here for convenience. Ncalc is a direct implementation of equation (1.6).

1. Sinu=SIN(U)
2. Sin2u=Sinu*Sinu
3. N=Aa/SQR(1-E2*Sin2u)
4. RETURN

In this subroutine, U represents the geodetic latitude of the point in the selected spheroid/datum system, Aa and E2 are the length of the semimajor axis and the square of the eccentricity of the reference spheroid, and N is the east-west radius of curvature.

B. Subroutine Rcalc: Subroutine Rcalc returns a value of the north-south radius of curvature at a specified latitude on the reference spheroid. Rcalc is a direct implementation of equation (1.5).

1. Sinu=SIN(U)
2. Sin2u=Sinu*Sinu
3. R=Aa*(1-E2)/(1-E2*Sin2u)**1.5
4. RETURN

In this subroutine the variables U, Aa, and E2 are the same as described previously. The variable R represents the north-south radius of curvature.

C. Subroutine Efgcalc: Subroutine Efgcalc returns the E-F-G coordinates of a point above, below, or on the reference spheroid.

1. Sinu=SIN(U)
2. Cosu=COS(U)
3. Sinw=SIN(W)
4. Cosw=COS(W)
5. E=(N+H)*Cosu*Cosw
6. F=-(N+H)*Cosu*Sinw
7. G=(N*(1-E2)+H)*Sinu
8. RETURN

In this subroutine, U is geodetic latitude and W is geodetic longitude. E, F, and G are the Cartesian coordinates of the point given in the universal space rectangular coordinate system. N is the east-west radius of curvature and H is the height of the target above the reference spheroid. H is measured along the normal line at the surface point defined by U and W.

D. Subroutine Xyzrae: Subroutine Xyzrae converts radar-centered east, north, and vertical (X-Y-Z) Cartesian coordinates into radar-centered range, azimuth, and elevation spherical coordinates. The computational algorithm is:

1. Hyp=SQR(X**2+Y**2)
2. Rng=SQR(Hyp**2+Z**2)
3. Abs1=ABS(X/Hyp)
4. IF Abs1>.707 THEN 7
5. Az=ASN(Abs1)
6. GOTO 9
7. Az=ACS(Hyp/Rng)

```
8.  Abs2=ABS(Z/Rng)
9.  IF Abs2>.707 THEN 12
10. El=ASN(Abs2)
11. GOTO 14
12. El=ACS(Hyp/Rng)
13. Quadx=34
14. IF X>=0 THEN Quadx=12
15. Quady=23
16. IF Y>=0 THEN Quady=41
17. Quadt=Quadx+Quady
18. IF Quadt=53 THEN Quad=1
19. IF Quadt=35 THEN Quad=2
20. IF Quadt=57 THEN Quad=3
21. IF Quadt=75 THEN Quad=4
22. IF Quad=2 THEN Az=180-Az
23. IF Quad=3 THEN Az=180+Az
24. IF Quad=4 THEN Az=360-Az
25. IF Z<0 THEN El=-El
26. RETURN
```

As used in this subroutine, Hyp is the hypotenuse of a triangle formed by the target's X and Y coordinates lying in the fundamental plane passing through the tracking site. Rng is the spherical radius from the origin to the target point. It is found as the hypotenuse of the triangle formed by Hyp and the Z coordinate. Abs1 is the absolute value of X/Hyp and Abs2 is the absolute value of Z/Rng. Abs1 and Abs2 are used to determine the most accurate computational method for determining the raw azimuth and elevation angles. When Abs1 is greater than 0.707, the slope of the arccosine function is steeper and this yields more accuracy in the calculation of the azimuth angle. Conversely, when Abs1 is less than 0.707, the slope of the arcsine function is steeper and greater accuracy is obtained by using the arcsine solution. A similar comparison is made using Abs2 to optimize the accuracy of the elevation angle computation.

Program steps 13 to 25 are a GMD-derived method for resolving the azimuth quadrant. If the value of X is positive, Quadx is given the value 12, meaning that X lies in either quadrant 1 or quadrant 2. If X is less than 0, Quadx is given the value 34, indicating that X lies in either quadrant 3 or quadrant 4. Similarly, if the value of Y is positive, Quady is given the value 41, indicating that Y lies in either quadrant 4 or quadrant 1. If Y is negative, Quady is given the value of 23, indicating that Y lies in either quadrant 2 or quadrant 3. The sum of Quadx and Quady is Quadt (step 17). The value of Quadt can only be 53, 35, 57, or 75. If the value is 53, then Quad is set equal to 1, indicating that the point lies in the first quadrant. Similarly, if Quadt equals 35, 57, or 75, then the point lies in quadrant 2, 3, or 4, respectively, and Quad is set equal to 2, 3, or 4 as the case may be. When the point lies on one of the coordinate axes, the coordinate value is assumed to be positive. However, the subroutine provides an identical solution for on-axis points if they are assumed to be negative.

Obviously, for quadrant 1, the azimuth value is equal to Az. For the second quadrant the azimuth value is equal to 180 - Az, for the third it is equal to 180 + Az, and for the fourth it is equal to 360 - Az.

In step 25, the sign attached to the elevation angle is determined by noting that El is negative when Z is negative and positive when Z is positive.

E. Subroutine Raexyz: Subroutine Raexyz converts target position in spherical range, azimuth, and elevation coordinates into Cartesian east-north-vertical (X-Y-Z) form.

1. Cosel=COS(El)
2. X=Rng*SIN(Az)*Cosel
3. Y=Rng*COS(Az)*Cosel
4. Z=Rng*SIN(El)
5. RETURN

In this subroutine, El is target elevation, Az is azimuth, and Rng is range. As in the previous subroutine, Az is measured clockwise from true north. El is 0 when horizontal (lying in the fundamental plane), +90 degrees at the zenith point, and -90 degrees when pointing vertically downward.

F. Subroutine Xyzefg: Subroutine Xyzefg is designed to rotate local Cartesian coordinates between an X-Y-Z (east-north-vertical) reference and a local system aligned with the earth-centered E-F-G (universal space rectangular) coordinate frame. When the subroutine is entered, if Matflg equals 1, the subroutine will rotate X-Y-Z coordinates into E-F-G alignment. If Matflg equals 0, the subroutine will rotate local E-F-G-aligned Cartesian coordinates to X-Y-Z alignment. The values stored in arrays A and B are the direction cosines for standard three-dimensional forward (X to E) and reverse (E to X) rotations. Mathematically, the forward transformation may be expressed as:

$$\begin{bmatrix} E \\ F \\ G \end{bmatrix} = \begin{bmatrix} \sin W & -\cos W \sin U & \cos W \cos U \\ \cos W & \sin W \sin U & -\sin W \cos U \\ 0 & \cos U & \sin U \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix},$$

or in program variables

$$\begin{bmatrix} E(1) \\ E(2) \\ E(3) \end{bmatrix} = \begin{bmatrix} A(1,1) & A(1,2) & A(1,3) \\ A(2,1) & A(2,2) & A(2,3) \\ A(3,1) & A(3,2) & A(3,3) \end{bmatrix} \begin{bmatrix} X(1) \\ X(2) \\ X(3) \end{bmatrix}.$$

Similarly, the reverse transformation is expressed mathematically as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \sin W & \cos W & 0 \\ -\cos W \sin U & \sin W \sin U & \cos U \\ \cos W \cos U & -\sin W \cos U & \sin U \end{bmatrix} \begin{bmatrix} E \\ F \\ G \end{bmatrix},$$

or in program variables

$$\begin{bmatrix} X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} B(1,1) & B(1,2) & B(1,3) \\ B(2,1) & B(2,2) & B(2,3) \\ B(3,1) & B(3,2) & B(3,3) \end{bmatrix} \begin{bmatrix} E(1) \\ E(2) \\ E(3) \end{bmatrix}.$$

The subroutine algorithm is:

```
1.  Sinu=SIN(U)
2.  Cosu=COS(U)
3.  Sinw=SIN(W)
4.  Cosw=COS(W)
5.  B(1,1)=A(1,1)=Sinw
6.  B(2,1)=A(1,2)=-Cosw*Sinu
7.  B(3,1)=A(1,3)=Cosw*Cosu
8.  B(1,2)=A(2,1)=Cosw
9.  B(2,2)=A(2,2)=Sinw*Sinu
10. B(3,2)=A(2,3)=-Sinw*Cosu
11. B(1,3)=A(3,1)=0
12. B(2,3)=A(3,2)=Cosu
13. B(3,3)=A(3,3)=Sinu
14. IF Matflg=1 THEN MAT E=A*X
15. IF Matflg=0 THEN MAT X=B*E
16. Matflg=0
17. RETURN
```

In this subroutine, U is geodetic latitude and W is geodetic longitude. The A array develops the forward direction cosines used for the X-Y-Z to E-F-G rotation, and the B array develops the inverse direction cosines for the E-F-G to X-Y-Z rotation. The symbol E in steps 14 and 15 represents a three-dimensional array whose elements E(1), E(2), and E(3) contain the current values of E, F, and G. The symbol X in the same program steps represents a three-dimensional array whose elements X(1), X(2), and X(3) contain the current values of X, Y, and Z. The BASIC command MAT E=A*X causes the 9 elements of matrix A to multiply the 3 elements of matrix array X. Similarly, MAT X=B*E causes the 9 elements of matrix array B to multiply the three elements of matrix array E.

Program to Calculate Forward and Reverse Azimuth and Elevation Angles and True Slant Range

The forward and reverse azimuth and elevation angles and the slant range between any two points (for example, A and B) are calculated by first converting the position coordinates of each point from geodetic form to the corresponding earth-centered E-F-G elements. Using simple vector subtraction, the difference vector (ΔE-ΔF-ΔG) is obtained. At the point A, the elements of the difference vector are first rotated into local east, north, and vertical alignment and then converted to spherical range, azimuth, and elevation form. This yields the forward azimuth and elevation angles along with the true slant range. Similarly, at the point B, the negative values of the elements of the difference vector are first rotated into local east, north, and vertical alignment and then converted to spherical form to yield values for the reverse azimuth and elevation parameters. This sequence is shown graphically in figure 4.1.

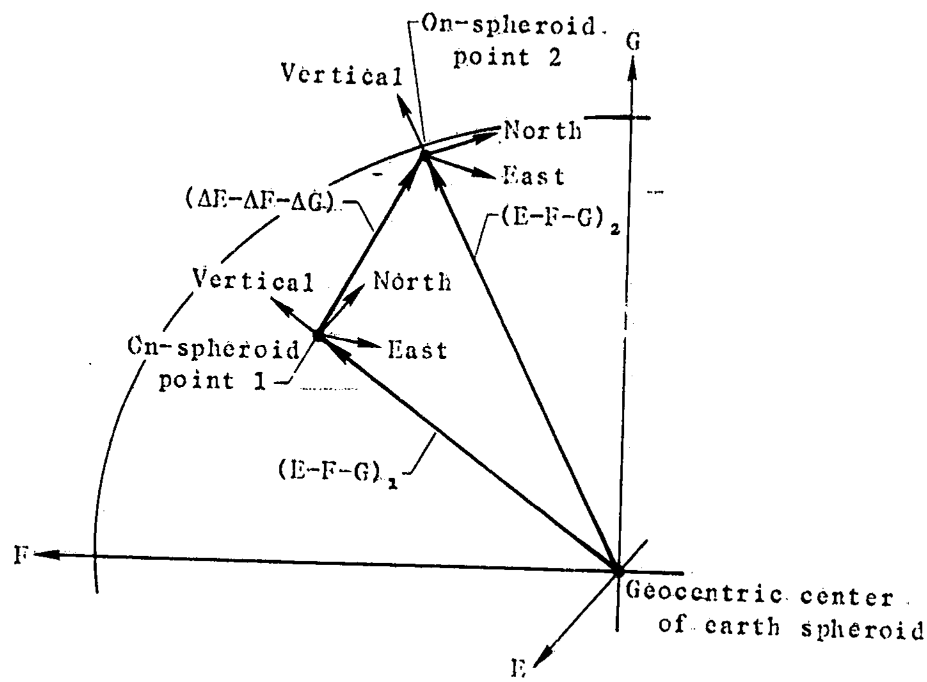

Figure 4.1.

Variable Names

Name	Description
Alt1	Spheroid elevation of point 1
Alt2	Spheroid elevation of point 2
Az1	Forward azimuth in degrees

Az2	Reverse azimuth in degrees
De, Df, Dg	Elements of difference vector
E	Designator of three-dimensional E(*) array in BASIC array multiplications (different from symbol E shown below)
E, F, G	Earth-centered universal rectangular coordinate system elements
E(1), E(2), E(3)	Elements of three-dimensional E(*) array
E01, F01, G01	Earth-centered E, F, and G coordinates of point 1
E02, F02, G02	Earth-centered E, F, and G coordinates of point 2
E11	Forward elevation in degrees
E12	Reverse elevation in degrees
Elev1	Sea-level elevation of point 1
Elev2	Sea-level elevation of point 2
Geosep1	Geoid separation at point 1
Geosep2	Geoid separation at point 2
Lat1	Latitude of point 1 in degrees
Lat2	Latitude of point 2 in degrees
Lon1	Longitude of point 1 in degrees
Lon2	Longitude of point 2 in degrees
Matflg	Flag to select E'-F'-G' to X-Y-Z or X-Y-Z to E'-F'-G' rotation in Xyzefg subroutine
Rng	Slant range in selected units
X	Designator of three-dimensional X(*) array in BASIC array multiplications (different from X symbol shown below)
X, Y, Z	Elements of east, north, vertical coordinate frame of reference
X(1), X(2), X(3)	Elements of three-dimensional X(*) array

Computational Algorithms

The essential algorithms for the forward and reverse azimuth and slant range calculations are given in the following paragraphs.

A. Gzentry subroutine: This subroutine allows the operator to enter geodetic coordinates, sea-level elevation, and geoid separation for the two coordinate points. The input variables for point 1 are Lat1, Lon1, Elev1, and Geosep1. The input variables for point 2 are Lat2, Lon2, Elev2, and Geosep2. The subroutine is called by the main subprogram Gz and returns to Gz after the appropriate entries have been made.

B. Gz subroutine: The Gz subroutine computes the slant range and the forward and reverse azimuth and elevation angles between point 1 and point 2. Upon entering Gz, the program calls Osel which requests the operator selection of an output device. When the output device selection has been made, the program returns to Gz. Next, Gz calls Gzentry which requests operator inputs for points 1 and 2. These consist of the latitide, longitude, sea-level elevation, and geoid separation for each point. At steps 5 and 6, the program converts each of the sea-level (geoid) elevations to spheroid elevations by adding the geoid separation for each point to its sea-level elevation. In the same steps, the spheroid elevations in the selected input units are converted to meters by applying the appropriate conversion factor (Ucnv). The spheroid elevations (in meters) are labeled Alt1 and Alt2.

Next, the program assigns the transfer variables U, W, and H the values of Lat1, Lon1, and Alt1 and calls subroutine Ncalc which computes the east-west radius of curvature (N) at point 1. The same values of N, U, W, and H are then used by subroutine Efgcalc to compute the E-F-G coordinates of point 1, which are stored as E01, F01, and G01. The same procedure is then followed for point 2. U, W, and H are assigned values of Lat2, Lon2, and Alt2, and the value of N at point 2 is computed for use in subroutine Efgcalc. Efgcalc returns the E-F-G coordinates of point 2, which are stored as E02, F02, and G02. The ΔE-ΔF-ΔG elements from point 1 to point 2 are computed in steps 23 to 25. At step 26, Matflg is set equal to 2, and the three E-F-G elements and the latitude and longitude of point 1 are used by subroutine Xyzefg to compute the coordinates of point 2 in a Cartesian E-N-V frame of reference centered at point 1. At step 37, the Cartesian E-N-V elements of the point 2 position vector are converted to R-A-E form. The computed Az and El values are stored as Az1 and El1. Next, the negative values of the ΔE-ΔF-ΔG vector are assigned to the MAT E array (E(1), E(2), and E(3)) and U and W are assigned the values of latitude and longitude for point 2. Matflg is set equal to 2 for a reverse (E-F-G to X-Y-Z) rotation, and subroutine Xyzefg computes the E-N-V Cartesian elements of the vector from point 2 to point 1. These values, which are returned as X(1), X(2), and X(3), are assigned to variables X, Y, and Z, the transfer variables for the Xyzrae subroutine. Xyzrae computes the azimuth and elevation of point 1 from point 2 along with the range between the two points. The reverse azimuth and elevation values are stored as Az2 and El2. The range between the points is stored as R12. Subroutine Gz then calls Gzprint, which converts the range value from meters back to the operator-selected units, then prints the forward and reverse azimuths and elevations and the range between the two points. After the printout has been completed, the program returns to Gzent to await new operator entries for point 1 and point 2.

```
Gz:!
GOSUB Osel
Gzent:!
GOSUB Gzentry
Alt1=(Elev1+Geosep1)/Ucnv
Alt2=(Elev2+Geosep2)/Ucnv
U=Lat1
W=Lon1
H=Alt1
GOSUB Ncalc
GOSUB Efgcalc
E01=E
F01=F
G01=G
U=Lat2
W=Lon2
H=Alt2
GOSUB Ncalc
GOSUB Efgcalc
E02=E
F02=F
G02=G
De=E02-E01
Df=F02-F01
Dg=G02-G01
Matflg=2
E(1)=De
E(2)=Df
E(3)=Dg
U=Lat1
W=Lon1
MAT X=ZER
GOSUB Xyzefg
X=X(1)
Y=X(2)
Z=X(3)
GOSUB Xyzrae
Az1=Az
El1=El
U=Lat2
W=Lon2
E(1)=-De
E(2)=-Df
E(3)=-Dg
MAT X=ZER
Matflg=2
GOSUB Xyzefg
X=X(1)
```

49. Y=X(2)
50. Z=X(3)
51. GOSUB Xyzrae
52. R12=Rng
53. Az2=Az
54. E12=E1
55. GOSUB Gzprint
56. PAUSE
57. GOTO Gzent

C. Subroutine Gzprint: This subroutine prints both the input coordinates and the computed values for slant range and forward and reverse azimuth and elevation.

Program Operation

The slant range and forward and reverse azimuth and elevation algorithms are a subroutine to GEOD. When GEOD is run, after the operator selects the desired units and spheroid/datum reference, the master menu is displayed. One selection on the menu is TRUE GEODETIC FWD AND REV AZ AND EL AND SPHEROID DIST. The operator selects the appropriate numerical entry and the main program enters the distance and angle calculation subroutines.

After entry into the subroutine, the operator is asked to make the following simple entries:

A. Output device selection, which is displayed as:

SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER

B. Parameter entries, which are displayed sequentially as:

ENTER GEODETIC LATITUDE OF POINT 1 IN D.MS

ENTER GEODETIC LONGITUDE OF POINT 1 IN D.MS

ENTER SEA-LEVEL ELEVATION OF POINT 1 IN (selected units)

ENTER GEOID SEPARATION AT POINT 1 IN (selected units)

ENTER GEODETIC LATITUDE OF POINT 2 IN D.MS

ENTER GEODETIC LONGITUDE OF POINT 2 IN D.MS

ENTER SEA-LEVEL ELEVATION OF POINT 2 IN (selected units)

ENTER GEOID SEPARATION AT POINT 2 IN (selected units)

If the operator does not know the value of geoid separation at either point, a value of 0 should be entered for both geoid separations. This will cause the calculations to be performed as if the sea-level elevation at both points were the same as the spheroid elevation. For most applications, the error will be insignificant (probably less than the least significant digit displayed). However, it should be noted that the value of geoid separation must be known and entered for both points, or zero must be entered for both points. If the true value for geoid separation is entered for one point for which it is known and zero is entered for the other point because the true value is unknown, there will be a measurable error in all the calculations.

The program then performs the computations and, when completed, displays the results in the following typical form. Angle data are provided as degree-minute-second, degree, and radian values.

```
POINT 1 GEODETIC LAT =  35 00 00.0000  ( 35.000000000)   ( 0.610865238)
POINT 1 GEODETIC LON = 118 00 00.0000  (118.000000000)   ( 2.059488517)
POINT 1 SEA LEV ELEV =         500.000 METERS
POINT 1 GEOID SEP    =          25.000 METERS

POINT 2 GEODETIC LAT =  36 00 00.0000  ( 36.000000000)   ( 0.628318531)
POINT 2 GEODETIC LON = 119 00 00.0000  (119.000000000)   ( 2.076941810)
POINT 2 SEA LEV ELEV =         250.000 METERS
POINT 2 GEOID SEP    =          15.000 METERS

FORWARD AZ (1 TO 2)  = 321 00 47.7143  (321.013253980)   ( 5.602738225)
FORWARD EL (1 TO 2)  = - 0 44 55.2557  ( -0.748682135)   (-0.013066968)
REVERSE AZ (2 TO 1)  = 140 25 57.0875  (140.432524308)   ( 2.451009926)
REVERSE EL (2 TO 1)  = - 0 32 26.8292  ( -0.540785893)   (-0.009438494)

SLANT RANGE          =      143326.771 METERS
```

When CONT is pressed, the program returns to the input steps and prompts the operator for entry of the second set of coordinate points. In cases where one point remains the same but the other point varies, the operator need only repeatedly press CONT as the redundant point inputs are requested. This will cause the program to use the values last entered for those points. Once the output device selection has been made, it will remain in effect so long as the program remains in the range and angle calculation subroutine.

Program Validation

Two methods are available to validate the range and angle calculations. The first uses data points which allow a trivial solution that can be checked on a desk calculator using simple trigonometry. Two such points can be located on the equator but at different longitudes so that the result is simply the chord of a known circle. Or, the points may be at the same longitude but with one point at the equator and the other at the north pole, forming a triangle whose base is the semimajor axis of the reference spheroid and whose leg is the semiminor axis of the reference spheroid. The second validation technique compares calculated angle data with data published in USCGS or DMAC documents.

This can provide a rough check, but when a difference is found, there is no way to know whether the error is in the published data, the calculated data, or both. In addition, the published USCGS and DMAC values are not generally given to the precision that is needed to accurately compare results. Therefore, the trivial geometric solutions are considered to be a more accurate validation method since, even though a trivial solution is used for comparison purposes, the values obtained from the program are obtained using all the computational algorithms used for the more complex solutions.

In this document, the first method is employed, and the results are shown in table 4.1. The first comparison is made using two points situated on the equator but with a difference in longitude of 180 degrees, the second comparison is made using two points situated on the equator but with a difference in longitude of 90 degrees, and the third comparison is made using the first point on the equator at the Greenwich meridian and the second point at the north pole. All calculations were made using the Clarke spheroid and a spheroid elevation of 0.

TABLE 4.1. TRIGONOMETRIC VALIDATION OF ANGLE DATA

Parameter	GMD baseline routines	Trigonometric solution
Pt 1 latitude	0 00 00.0000	0 00 00.0000
Pt 1 longitude	0 00 00.0000	0 00 00.0000
Pt 2 latitude	0 00 00.0000	0 00 00.0000
Pt 2 longitude	180 00 00.0000	180 00 00.0000
Forward azimuth	Indeterminant	Indeterminant
Reverse azimuth	Indeterminant	Indeterminant
Forward elevation	- 90.000000000	- 90.000000000
Reverse elevation	- 90.000000000	- 90.000000000
Slant range	12,756,412.800 m	12,756,412.800 m
Pt 1 latitude	0 00 00.0000	0 00 00.0000
Pt 1 longitude	0 00 00.0000	0 00 00.0000
Pt 2 latitude	0 00 00.0000	0 00 00.0000
Pt 2 longitude	90 00 00.0000	90 00 00.0000
Forward azimuth	90 00 00.0000	90 00 00.0000
Reverse azimuth	270 00 00.0000	270 00 00.0000
Forward elevation	- 45.000000000	- 45.000000000
Reverse elevation	- 45.000000000	- 45.000000000
Slant range	9,020,145.994 m	9,020,145.994 m
Pt 1 latitude	0 00 00.0000	0 00 00.0000
Pt 1 longitude	0 00 00.0000	0 00 00.0000
Pt 2 latitude	90 00 00.0000	90 00 00.0000
Pt 2 longitude	0 00 00.0000	0 00 00.0000
Forward azimuth	0.000000000	0.000000000
Reverse azimuth	180.000000000	180.000000000
Forward elevation	- 45.097283309	- 45.097283309
Reverse elevation	- 44.902716691	- 44.902716691
Slant range	9,004,869.488 m	9,004,869.488 m

The above results show agreement to at least nine decimal places in angle and at least three decimal places in range (the accuracy limit of the desk computer system). It should be noted that the program results are all based on closed-form solutions whose accuracies are affected only by the precision limit of the computational equipment.

Program to Calculate Spheroid Forward and Reverse Azimuth and Spheroid Distance

To determine the course and distance between two points A and B, a pilot draws a straight line between the two points on a Lambert conformal aeronautical chart. Neglecting the effects of wind, the magnetic heading to be flown from A to B is simply the true course (measured as the angle between the course line and the meridian passing through the starting point A) plus or minus the magnetic variation (plus for west variation and minus for east variation). In flight, as the airplane proceeds along this path, the magnetic heading must be slowly adjusted because the angle between the course line and each succeeding meridian en route is slightly different due to meridian convergence. If the earth were a true sphere, the path taken between the two points would be the arc of a great circle representing the shortest distance between the two points. In the real world, because of the flattening of the spheroid, the Lambert course does not represent the shortest distance between the two points. However, because the flattening is very slight, the difference in distance between the geodesic and the Lambert course is also very slight for normal flight legs, even those of reasonably long distance (1000 to 2000 nautical miles).

Subroutine Lx in the baseline program is used to compute the Lambert forward and reverse azimuth angles and the spheroid distance between any two chart points. In this subroutine the geodetic coordinates (latitude and longitude) of two map points are entered. The program then computes the forward and reverse azimuth angles using the Lambert conformal transformation, which is mathematically constrained to preserve angle measurements.

Distance calculations made on a Lambert map obviously require some adjustment since the scale is true only on the two standard parallels. A method has been devised to obtain accurate distance measurements by computing a correction factor that can be applied to the Lambert hypotenuse (which represents the length of the projection of an on-spheroid line drawn between the two points). This method is used in subroutine Lx. First, to reduce initial scale error, the north and south standard parallels used in the transformation are selected so as to pass directly through the two points between which the measurements are to be made. Next, to correct for scale error due to magnification between the two standard parallels, the program computes the Lambert distance between the northern and southern parallels and compares this distance with the spheroid distance between the two parallels as computed from the meridional arc algorithm used in the Mercator subprogram. The ratio of these two distances provides a factor that is used to correct the Lambert distance measurement. Results of the on-spheroid distance computations have been verified to be well within 1 meter of accuracy for distances up to 1000 nautical miles or more.

Variable Names

Name	Description
Aa	Length of semimajor axis of reference spheroid

Az1	Forward spheroid azimuth in degrees
Az2	Reverse spheroid azimuth in degrees
Azflg	Flag indicating azimuth value as indeterminant
Bb	Length of semiminor axis of reference spheroid
Dlon	Difference in longitude between point 1 and point 2
E2	Square of ellipticity value for reference spheroid
Ee	Ellipticity value for reference spheroid.
Factor	Factor used to correct Lambert distance computation
L11	Transfer variable used to send latitude of north standard parallel to meridional arc subroutine
L12	Transfer variable used to send longitude of south standard parallel to meridional arc subroutine and to return computed arc length
Lat1	Latitude of point 1 in degrees
Lat2	Latitude of point 2 in degrees
Lm	Flag sent to subprogram Mercator to indicate that only meridional arc computation is to be performed
Lon1	Longitude of point 1 in degrees
Lon2	Longitude of point 2 in degrees
Lx	Lambert X value returned from Lamxy subroutine
Ly	Lambert Y value returned from Lamxy subroutine
Lycheck	Check value used in the development of the Lambert correction factor
Nlat	Latitude of north standard parallel in degrees
Olat	Latitude of Lambert origin point (point 1) in degrees
Olon	Longitude of Lambert origin point (point 2) in degrees
Rng	Lambert distance computed in Xyzrae subroutine
Slat	Latitude of south standard parallel in degrees
X	Lambert X

Y Lambert Y

Zlat Target latitude (point 2) in degrees

Zlon Target longitude (point 2) in degrees

Computational Algorithms

A. Gzentry subroutine: This subroutine allows the operator to enter geodetic coordinates of the first and second spheroid points. The same subroutine is used to enter data for the angle and slant range program previously described. When used for entry of on-spheroid points, a mode flag causes the requests for sea-level elevation and geoid separation to be bypassed.

B. Lz subroutine: The Lz subroutine computes the spheroid distance and the Lambert forward and reverse azimuth angles between point 1 and point 2. Certain housekeeping functions and simple degree conversion calls are omitted.

The Osel subroutine (step 2) is entered only on initial entry into the Lz subroutine and allows the operator to select the output device (CRT, thermal printer, or line printer). Gzentry is called at step 3 to accept the latitudes and longitudes of points 1 and 2, after which the main computations are performed.

The Lz subroutine uses the generalized Lambert algorithms to compute forward and reverse spheroid azimuth angles and Lambert distances. It is important to recall that for this calculation, Lambert parameters are selected such that the north standard parallel passes through the northernmost point and the southern standard parallel passes through the southernmost point. To do this, the conventional Lambert origin point is assigned the coordinates of point 1 and the conventional Lambert target point is assigned the coordinates of point 2 (steps 4 and 5) in order to first compute the forward azimuth from point 1 to point 2. A check is made at step 8 to determine whether both points lie at the same latitude. In such case, the transformation reduces to that of a single standard parallel passing through both points. Since, when this happens, the course runs very near the standard parallel where the distance is true, it suffices to simply set the correction factor to 1 and bypass the meridional arc calculations which determine the true arc distance between the standard parallels. If the two points are not at the same latitude, flag Lm is set to 1 and the Mercator subprogram is called. Flag Lm causes the program to bypass all but the meridional arc calculations in the Mercator subprogram, thus eliminating the need for a separate meridional arc subroutine. In the calling argument (step 10), the dummy variables L11 and L12 pass the latitude values of point 1 and point 2 to the Mercator subprogram. The meridional arc length between north-south points whose latitudes are the same as those of points 1 and 2 but whose longitudes are zero is then computed. The arc length obtained represents the separation of the parallels passing through points 1 and 2, and it is returned as dummy variable L12. Parameter Ycheck is then set equal to the absolute value of the distance separation between the standard parallels passing through the two points (step 12).

Steps 15 to 18 are used to recognize the indeterminant situation where both points lie at the pole. In such case, range is set to 0 and Azflg is set to 1 to cause the word INDETERMINANT to print out for the azimuth value. Step 19 recognizes the condition where 1 point lies at the pole and the other lies at any point on the spheroid. In this case one azimuth angle is always 0 degrees and the other is always 180 degrees. The logic in steps 20 to 24 determines which of the two azimuth values is 0 and which is 180 degrees. In this case, it is also obvious that the spheroid distance between the two points is exactly the same as the meridional arc distance, and Rng is set equal to the absolute value returned from the meridional arc calculation (steps 12 and 25).

Next, the Lamb subroutine is called. This subroutine causes the program to enter the Lambert routines previously described. These routines initialize using the point 1 latitude as the north standard parallel, the point 2 latitude as the south standard parallel, the coordinates of point 1 as the origin, and the coordinates of point 2 as the target. Lamb returns values Lx and Ly, representing the Lambert coordinates of point 2 with point 1 as the origin. The values of the two position coordinates of point 2 with respect to point 1 are stored as X and Y.

In step 30, the difference in longitude between the two points is set to 0 (Dlon=0), and Lamxy is called. This bypasses the Lambert initialization routines which are calculated when Lamb is called, but using the same initialization parameters, computes the Lambert distance between two comeridional points lying on each of the two standard parallels. This value is stored as Lycheck at step 32. At step 35, a factor is developed which represents the ratio of the true meridional arc distance between the two parallels containing points 1 and 2, and the Lambert distance between the same two parallels. Since the Lambert value obtained by this procedure represents the length of the Lambert projection of a meridional arc between the two parallels, the same scale factor correction should be reasonably accurate for correcting the total distance between two non-comeridional points lying on the same two parallels.

At step 36, Subroutine Xyzrae is called to compute the Lambert distance between the two points. Since only X and Y have been assigned values (steps 28 and 29), the subroutine returns only Az (azimuth) and Rng (range). In a Lambert transformation, angles are preserved. This insures that the correct azimuth is obtained. Rng, in this case, is the Lambert hypotenuse between the two points. The value of the forward azimuth is stored as Az1 (step 37).

To find the reverse on-spheroid azimuth, the point 1 coordinates are assigned as target coordinates Zlat and Zlon, and the point 2 coordinates are assigned as the origin coordinates Olat and Olon (steps 38 to 43). Subroutine Lam1 is again called to initialize the Lambert transformation using the same two standard parallels, but with point 2 now serving as the origin point. The returned Lambert X and Y values are again resolved into polar form in subroutine Xyzrae which returns the value of the reverse (point 2 to point 1) azimuth. This value is stored as Az2 in step 48.

Finally, at step 49, the Lambert distance between the two points (Rng) is corrected by the value of Factor (Rng=Rng*Factor) to obtain the approximate spheroid distance between the two points. After the range adjustment, the program calls the printout subroutine Gzprint, which displays or prints the results.

The program sequence used to accomplish these calculations is:

```
1.  Lz:1
2.  GOSUB Osc1
3.  GOSUB Gzentry
4.  L11=Olat=Lat1
5.  Olon=Lon1
6.  L12=Zlat=Lat2
7.  Zlon=Lon2
8.  IF Olat=Zlat THEN 13
9.  Lm=1
10. CALL Mercator(Aa,Bb,Ee,E2,Ucnv,L11,L12)
11. Lm=0
12. Ycheck=ABS(L12)
13. Nlat=MAX(Olat,Zlat)
14. Slat=MIN(Olat,Zlat)
15. IF (Nlat=90) AND (Slat=90) THEN Azflg=1
16. IF (Nlat=90) AND (Slat=90) THEN Rng=0
17. IF (Nlat=90) AND (Slat=90) THEN GOTO Gzprint
18. IF (Lat1=90) OR (Lat2=90) THEN 21
19. GOTO 27
20. Az1=Az1r=0
21. Az2=180
22. IF Lat2=90 THEN 26
23. Az1=180
24. Az2=Az2r=0
25. Y=Rng=Ycheck
26. GOTO Gzprint
27. GOSUB Lam1
28. X=Lx
29. Y=Ly
30. Dlon=0
31. GOSUB Lamxy
32. Lycheck=Ly
33. IF Ly>1000 THEN 36
34. IF FRACT(Lycheck*1000)=0 THEN Ycheck=Ly
35. Factor=Ycheck/Lycheck
36. GOSUB Xyzrae
37. Az1=Az
38. Lat=Olat
39. Lon=Olon
40. Olat=Zlat
41. Olon=Zlon
```

42. Zlat=Lat
43. Zlon=Lon
44. GOSUB Lam1
45. X=Lx
46. Y=Ly
47. GOSUB Xyzrae
48. Az2=Az
49. Rng=Rng*Factor
50. GOTO Gzprint

C. Subroutine Lam1: Subroutine Lam1 performs the Lambert initialization based on the values for the north and south standard parallels (Nlat and Slat) and the values of the origin latitude and longitude (Olat and Olon). After the initialization the program proceeds into Lamxy which computes the Lambert X and Y values for the points previously entered by the operator. A complete description of the Lambert routines is provided in chapter 2 of this document.

D. Subroutine Lamxy: Subroutine Lamxy computes the Lambert X and Y coordinates (Lx and Ly) using the initialization values from subroutine Lam1 and the target's geodetic coordinates (Zlat and Zlon). Subroutine Lamxy is fully described in chapter 2 of this document.

E. Subroutine Xyzrae: Subroutine Xyzrae converts local east, north, and vertical Cartesian coordinates into spherical range, azimuth, and elevation form. For on-spheroid calculations, the elevation computations are not used. Subroutine Xyzrae is given earlier in this chapter (Subroutines Common to Range and Angle Programs).

Program Operation

The spheroid angle and distance algorithms are a subroutine of GEOD. When GEOD is run, the operator is prompted to select the units and the datum/spheroid reference applicable to the computations to be performed. After these selections have been made, the master menu selection is displayed. One selection is SPHEROID FWD AND REV AZIMUTH AND SPHEROID DISTANCE. The operator selects the appropriate numerical entry and the main program enters the spheroid angle and distance subroutines.

After entry into the subroutine, the operator is asked to make the following simple entries:

A. Output device selection, which is displayed as:

SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER

B. Parameter entries, which are displayed sequentially as:

ENTER GEODETIC LATITUDE OF POINT 1 IN D.MS

ENTER GEODETIC LONGITUDE OF POINT 1 IN D.MS

ENTER GEODETIC LATITUDE OF POINT 2 IN D.MS

ENTER GEODETIC LONGITUDE OF POINT 1 IN D.MS

The program then enters its computational mode and, when completed, displays the following:

POINT 1 GEODETIC LATITUDE	= 35 00 00.0000	(35.000000000)	(0.610865238)
POINT 1 GEODETIC LONGITUDE	= 118 00.00.0000	(118.000000000)	(2.059488517)
POINT 2 GEODETIC LATITUDE	= 36 00 00.0000	(36.000000000)	(0.628318531)
POINT 2 GEODETIC LONGITUDE	= 119 00 00.0000	(119.000000000)	(2.076941810)
FORWARD AZIMUTH (1 TO 2)	= 321 00 51.9530	(321.014431394)	(5.602758774)
REVERSE AZIMUTH (2 TO 1)	= 140 26 01.3948	(140.433720789)	(2.451030809)
SPHEROID DISTANCE	= 143320.67 METERS		

In this display, angles are first shown in degree, minute, second, and decimal second format, followed by the same value in degrees and in radians. Distance units are as selected by the operator upon initial entry into GEOD.

When CONT is pressed, the program returns to the entry point and requests coordinates for the next two points. If either point is the same as for the previous run, the operator need only press CONT and the coordinates for the previous run will be used for that particular entry parameter.

Program Validation

The spheroidal azimuth and distance routines compute the angles and distance measurements that would be obtained by a pilot or flight planner when he draws a straight course line between two points on a Lambert conformal chart. On a Lambert conformal chart, the angles provide what would be a great circle route if the earth were perfectly round. Because the earth is a spheroid and not a sphere, the chart course approximates a great circle path.

Validation of the program is first accomplished by computing angles and distances between a selected origin and a pattern of points surrounding the origin. The points are kept sufficiently close to the origin that the spheroid angles and distances returned by this routine are approximately equal to the true azimuth and slant range distances computed by the previous program. For this comparison the Clarke/NAD references were used, and the results are shown in table 4.2.

CHAPTER 5

DETERMINATION OF GEODETIC COORDINATES FOR OFF-SPHEROID POINTS

This chapter describes four methods for the determining the geodetic latitude, longitude, and altitude of a target from known universal space rectangular coordinates. These methods are:

1. the Lagrange (Hedgley) closed-form method
2. the Purcell and Cowan approximation method
3. the Bowring approximation method
4. the GMD (James) closed-form method

The Lagrange multiplier method is a closed-form solution which provides results whose accuracies are valid to the precision limit of the computational system. While in theory a closed-form solution should be best for use in a baseline system, the solution requires finding the roots of a quartic equation, and, with normal computational systems, this degrades the accuracy because of the need to work with fourth-order terms. The algorithms used for solving the quartic equation are time consuming, but the solution concept is straightforward and easily implemented.

The Purcell and Cowan approximation method offers sufficient speed for real-time applications. It is not quite as accurate in the latitude determination as the other solution methods, but the altitude calculations are extremely accurate. For real-time programs, the Purcell and Cowan approximation usually achieves results that are as good as or better than the least-significant-bit precision of the tracking systems.

The Bowring approximation method also has sufficient speed for real-time applications. In the Bowring method a rough approximation of altitude is computed and then refined. Even if the first approximation is in error by a significant amount, the equations will still return a surprisingly accurate solution. Generally, a one-pass solution provides sufficient accuracy for most tracking applications. However, the Bowring method has the advantage that, by using the output of one solution as input for a succeeding pass, the errors can be reduced to any specified limit in reasonable computational time.

The GMD method was developed during an attempt to find a closed-form solution with speed comparable to that of the approximation methods. The technique is straightforward and does not suffer from singularities that are sometimes present in other solutions. Unlike the Lagrange method, which solves a quartic equation for the value of the Lagrange multiplier, the GMD solution solves directly for the coordinates of the target point. Unfortunately, the method also requires a quartic solution, which is time consuming. At altitudes under 1,000,000 meters, accuracies are approximately the same as for the Lagrange solution. However, at very high altitudes (up to 1,000,000,000 meters), the GMD solution seems to yield slightly better accuracies.

A comparison of computational time requirements for the four methods shows the Purcell and Cowan solution to be the fastest with the Bowring method taking about twice as long, the GMD method taking about three times as long, and the Lagrange method taking about four times as long.

The Lagrange Multiplier Method

The application of the Lagrange multiplier method to the conversion of universal space rectangular coordinates to geodetic values was proposed by D. R. Hedgley, Jr. (ref. 5). The uniqueness of the Hedgley method lies in the fact that the solution yields the coordinates of the surface point for which the square of the surface-to-target distance is at a minimum value. By minimizing the square of the distance instead of the distance itself, a variables separable condition exists which permits a direct application of the Lagrange solution.

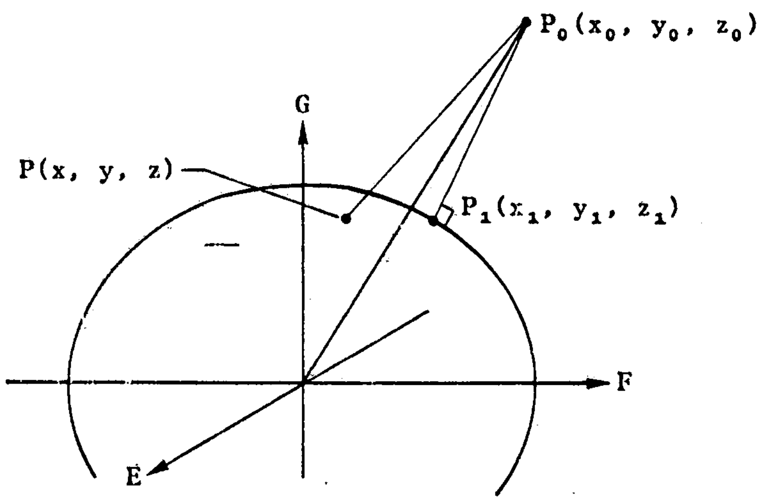

Figure 5.1.

Consider a target situated at a point P_0 that lies off the earth spheroid, as shown in figure 5.1. Let P be any point on the surface of the spheroid and let P_1 be the spheroid point whose normal line passes directly through the target point. Using universal space rectangular coordinates, the distance h between the surface point and the target point is given by

$$h = [(E_0 - E)^2 + (F_0 - F)^2 + (G_0 - G)^2]^{1/2} \tag{5.1}$$

and

$$h^2 = (E_0 - E)^2 + (F_0 - F)^2 + (G_0 - G)^2. \tag{5.2}$$

Obviously, the values of both h and h^2 reach minimums when $E = E_1$, $F = F_1$, and $G = G_1$, and it will suffice to minimize h^2 instead of h if this will simplify the computations. This simplification becomes apparent when the partial derivatives of h^2 are taken with respect to E, F, and G, yielding

$$\frac{\partial h^2}{\partial E} = 2(E_0 - E), \quad \frac{\partial h^2}{\partial F} = 2(F_0 - F), \quad \text{and} \quad \frac{\partial h^2}{\partial G} = 2(G_0 - G) \tag{5.3}$$

where the partial of h^2 with respect to E is a function of E alone, the partial of h^2 with respect to F is a function of F alone, and the partial of h^2 with respect to G is a function of G alone.

The equation for the spheroid in universal space rectangular coordinates is

$$\frac{E^2}{a^2} + \frac{F^2}{a^2} + \frac{G^2}{b^2} = 1 \tag{5.4}$$

which, by rearranging terms, can be written in a zero form as

$$\frac{E^2}{a^2} + \frac{F^2}{a^2} + \frac{G^2}{b^2} - 1 = 0. \tag{5.5}$$

Since equation (5.5) is equal to 0, it will not change the value of equation (5.2) if it is rewritten as

$$H^2 = (E_0 - E)^2 + (F_0 - F)^2 + (G_0 - G)^2 - \alpha\left[\frac{E^2}{a^2} + \frac{F^2}{a^2} + \frac{G^2}{b^2} - 1\right]. \tag{5.6}$$

In equation (5.6), the value of h^2, a function of E, F, and G, has been replaced by H^2, a function of E, F, G, and α, where α is known as the Lagrange multiplier. The added right-hand members of equation (5.6) are nothing more than equation (5.5) multiplied by α. Since equation (5.5) is equal to 0, the product of α and equation (5.5) must also be equal to 0. Therefore, H^2 is numerically equal to h^2 for all values of E, F, G, and α.

To obtain the maxima and minima of H^2, the four partial derivatives are set equal to zero.

$$\frac{\partial H^2}{\partial E} = 2(E_0 - E) - \frac{2\alpha E}{a^2} = 0 \tag{5.7}$$

$$\frac{\partial H^2}{\partial F} = 2(F_0 - F) - \frac{2\alpha F}{a^2} = 0 \tag{5.8}$$

$$\frac{\partial H^2}{\partial G} = 2(G_0 - G) - \frac{2\alpha G}{b^2} = 0 \tag{5.9}$$

$$\frac{\partial H^2}{\partial \alpha} = -\frac{E^2}{a^2} - \frac{F^2}{a^2} - \frac{G^2}{b^2} + 1 = 0 \tag{5.10}$$

The means by which the Lagrange multiplier method yields constrained maxima and minima is readily apparent from equations (5.7) to (5.10). When the partial derivatives of H^2 with respect to each of the three coordinate variables E, F, and G are set equal to 0, the values obtained will be those for zero-slope points, either maxima or minima. At the same time, the derivative of H^2 with respect to the Lagrange multiplier α yields nothing less than the zero-form equation for the spheroid. Because of this, the only values of E, F, and G for which equations (5.7) to (5.10) can possibly be satisfied are those for the set of coordinate points lying on the surface of the spheroid.

By intuition one can deduce that there are only two points at which the maximum or minimum distance between the surface of a spheroid and an off-spheroid point will occur. One is the surface point whose outward normal passes through the target point. The other is the surface point that is most distant from the target point. However, because h^2, and not h, was used in the application of the Lagrange multiplier method, two additional points are introduced, both of which are imaginary. Thus, to determine the surface geodetic coordinates of the target, it is necessary to find the real values for E, F, and G for which h or h^2 is minimized (eq. (5.1) or (5.2)).

This is accomplished by solving equations (5.7) to (5.9) for E, F, and G, respectively. The values obtained are then substituted into equation (5.10), yielding

$$-\frac{(E_0)^2}{a^2\left(1+\frac{\alpha}{a^2}\right)^2}-\frac{(F_0)^2}{a^2\left(1+\frac{\alpha}{a^2}\right)^2}-\frac{(G_0)^2}{b^2\left(1+\frac{\alpha}{b^2}\right)^2}+1=0. \qquad (5.11)$$

Equation (5.11) may be rearranged into

$$\begin{aligned}&\alpha^4+(2a^2+2b^2)\alpha^3+(a^4+b^4-E_0^2a^2-F_0^2a^2-G_0^2b^2+4a^2b^2)\alpha^2\\&+(2a^2b^4+2a^4b^2-2E_0^2a^2b^2-2F_0^2a^2b^2-2G_0^2a^2b^2)\alpha\\&+(a^4b^4-E_0^2a^2b^4-F_0^2a^2b^4-G_0^2a^4b^2)=0\end{aligned} \qquad (5.12)$$

which is in the form of a quartic equation

$$A\alpha^4+B\alpha^3+C\alpha^2+D\alpha+E=0. \qquad (5.13)$$

In the baseline program, the quartic equation is solved by the method of Descartes, which is provided in detail in the appendix.

Of the two real roots obtained by the quartic root solution, the one that yields values of E, F, and G which minimize the value of h in equation (5.1) will be the proper root, and the values of E, F, and G thus obtained will be the universal space rectangular coordinates of the surface point directly beneath the target.

To obtain the geodetic coordinates of the point, it is first necessary to convert the E-F-G coordinates into geocentric spherical coordinates and thence into geodetic coordinates. The geocentric coordinates of the point can be obtained from

$$R = (E^2 + F^2 + G^2)^{1/2} \tag{5.14}$$

and

$$\xi = \arccos \frac{(E_1^2 + F_1^2)^{1/2}}{R_1} \tag{5.15}$$

where R_1 is the length of the geocentric position vector from the origin to the point P_1, and ξ_1 is the geocentric latitude of the point P_1.

Using equation (1.8), geocentric latitude ξ_1 may be converted to geodetic latitude μ_1 by the relation

$$\mu_1 = \arctan [\tan \xi_1/(1 - \varepsilon^2)]. \tag{5.16}$$

The geodetic longitude is obviously the same as the geocentric longitude of the target point and can be found from

$$\lambda = \arctan (F_1/E_1) = \arctan (F_0/E_0). \tag{5.17}$$

Knowing the geodetic latitude and longitude of the surface point as well as the target E_1, F_1, and G_1 coordinates, the value of h can be obtained from

$$h = \frac{E_0}{\cos \mu_1 \sin \lambda} - \frac{a}{(1 - \varepsilon^2 \sin^2 \mu_1)^{1/2}}, \tag{5.18}$$

$$h = \frac{F_0}{\cos \mu_1 \sin \lambda} - \frac{a}{(1 - \varepsilon^2 \sin^2 \mu_1)^{1/2}}, \tag{5.19}$$

or

$$h = \frac{G_0}{\sin \mu_1} - \frac{a(1 - \varepsilon^2)}{(1 - \varepsilon^2 \sin^2 \mu_1)^{1/2}}. \tag{5.20}$$

Equations (5.18) to (5.20) are used instead of equation (5.1) to obtain the final value for h because a solution with equation (5.1) would involve finding the square root of a very large term which would reduce the accuracy of the result.

Variable Names

Name	Description
A2	a^2 term in equation (5.12)
A2b2	a^2b^2 term in equation (5.12)
A2b4	a^2b^4 term in equation (5.12)
A4b2	a^4b^2 term in equation (5.12)
Aa	Length of semimajor axis of reference spheroid
Alpha	Lagrange multiplier
Alt	Altitude of target above sea level
An	Normalized length of semimajor axis (An = 1)
B2	b^2 term in equation (5.12)
B4	b^4 term in equation (5.12)
Bb	Length of semiminor axis of reference spheroid
Bn	Normalized length of semiminor axis (Bb/Aa)
Coslat	Cosine of target geodetic latitude
E1	E coordinate in input units
E2n	En^2
En	Normalized E coordinate
F1	F coordinate in input units
F2n	Fn^2
Fn	Normalized F coordinate
G1	G coordinate in input units
G2n	Gn^2
Geoidsep	Separation of geoid in meters
Geosep	Separation of geoid in selected units
Gn	Normalized G coordinate

Lat	Geodetic latitude of target on spheroid
Lon	Longitude of target on spheroid
Sinlat	Sine of target geodetic latitude
Sin2lat	$Sinlat^2$
Terma	Multiplier of fourth-power term in equation (5.12)
Termb	Multiplier of third-power term in equation (5.12)
Termc	Multiplier of second-power term in equation (5.12)
Termc1	Constant portion of Termc
Termc2	Variable portion of Termc
Termd	Multiplier of first-power term in equation (5.12)
Termd1	Constant portion of Termd
Termd2	Variable portion of Termd
Terme	Last bracketed term in equation (5.12)
Terme1	Constant portion of Terme
Terme2	Variable portion of Terme
Ucnv	Conversion factor for selected units

Computational Algorithms

The essential algorithms used for the Lagrange multiplier determination of geodetic latitude, longitude, and altitude from known universal space rectangular coordinates are as follows:

A. Lagrange subroutine: The Lagrange subroutine computes the Lagrange terms needed by the quartic solution subprogram and, upon receipt of the real roots from the quartic solution, computes geodetic latitude, longitude, and altitude.

Upon initial entry into the Lagrange subroutine, the program calls subroutine Osel (step 2) which allows the operator to select the output device to be used throughout the computations. The program then calls subroutine Efgentry (step 6) to permit the operator to enter the E-F-G coordinates of the target point. These coordinates are entered in the units selected by the operator at program initialization. The input coordinates are assigned simple variable names of E1, F1 and G1. In steps 7 to 9 the program normalizes each of the input coordinates through division by the appropriate metric conversion value (Ucnv) and the length of the semimajor axis (Aa). In steps 12 to 17 the program forms the multipliers of the power terms in equations (5.12) and (5.13). Note that Terma through Terme correspond to the parameters A through E in equation (5.13). At step 18 the program calls subroutine Quartic which implements the Descartes technique to solve for the roots of a generalized quartic equation. A full description of subroutine Quartic is provided in the appendix. When Quartic is called from subroutine Lagrange, a mode flag causes the subroutine logic to test for the root that provides the minimum value for h. The selected root is stored as simple variable Alpha. Steps 20 through 22 implement equations (5.7) through (5.9), which have been rearranged to solve for the parameters E, F, and G respectively. Note that the computed coordinates are restored from normalized to engineering unit values in the same steps. At step 18, the Efgtolat subroutine is called to compute the surface point latitude coordinate from the values of E, F, and G. At step 24, the longitude of the surface point (which is identical to the longitude of the target point) is computed by simple trigonometry. In step 29, the computed latitude and longitude values of the surface point are used to determine the correct value for h (Alt). Step 29 is a direct implementation of equation (5.18).

```
 1.  Lagrange:!
 2.  GOSUB Osel
 3.  PRINT PAGE
 4.  GOSUB Lagrangeterm
 5.  Lagrange1:!
 6.  GOSUB Efgentry
 7.  En=E1/Ucnv/Aa
 8.  Fn=F1/Ucnv/Aa
 9.  Gn=G1/Ucnv/Aa
10.  E2n=En*En
11.  F2n=Fn*Fn
12.  Termc2=A2n*(E2n+F2n)+B2n*G2n
13.  Termd2=2*A2b2*(E2n+F2n+G2n)
14.  Terme2=A2b4*(E2n+F2n)+A4b2*G2n
15.  Termc=Termc1-Termc2
16.  Termd=Termd1-Termd2
17.  Terme=Terme1-Terme2
18.  GOSUB Quartic
19.  Coordinates:!
20.  E=En*Aa/(1+Alph)
21.  F=Fn*Aa/(1+Alph)
22.  G=Gn*Aa/(1+Alph/B2n)
23.  GOSUB Efgtolat
24.  Lon=180-ATN(Fn/En)
25.  Sinlat=SIN(Lat)
26.  Sin2lat=Sinlat*Sinlat
27.  Coslat=COS(Lat)
28.  Coslon=COS(Lon)
29.  Alt=En*Aa/(Coslat*Coslon)-Aa/SQR(1-E2*Sin2lat)
30.  GOSUB Coordprint
31.  GOTO Lagrange1
```

B. Lagrangeterm subroutine: The Lagrangeterm subroutine computes the spheroid/datum dependent terms A through E in quartic equation (5.13). These terms remain the same for all solutions using the same spheroid/datum reference, and therefore, the calculations are made only upon initial entry into the Lagrange subroutine. Note that spheroid length parameters (a and b) are normalized to prevent a real precision overflow that would otherwise occur in the fourth-power terms.

```
1.  An=A2n=A4n=1
2.  Bn=Bb/Aa
3.  B2n=Bn*Bn
4.  B4n=B2n*B2n
5.  A2b2=B2n
6.  A4b2=B2n
7.  A2b4=B4n
8.  Terma=1
9.  Termb=2*(A2n+B2n)
10. Termc1=A4n+B4n+4*A2b2
11. Termd1=2*(A2b4+A4b2)
12. Terme1=A4n*B4n
```

C. Efgentry subroutine: The Efgentry subroutine is used to receive operator inputs of the E, F, and G coordinates of the target point. Entries are made in units selected by the operator at program initialization and stored as E1, F1, and G1.

D. Quartic subroutine: The Quartic subroutine is a generalized subroutine that will find the roots of any quartic equation. Input to the subroutine are the values for A, B, C, D, and E as indicated in equation (5.13). In the Lagrange program, these terms are labeled as Terma, Termb, Termc, Termd, and Terme. When the entry to Quartic is made from the Lagrange program, the two real roots are tested to select the one which provides the minimum value for h (Alt). This root is stored as simple variable Alpha for subsequent use in the Lagrange subroutine. A full description of subroutine Quartic is provided in the appendix.

E. Efgtolat subroutine: The Efgtolat subroutine is a generalized algorithm that converts E-F-G coordinate values into geodetic latitude. At step 1 the value of $E^2 + F^2$ is computed. In step 2 the length of the geocentric radius vector R1 is computed as the square root of $E^2 + F^2 + G^2$. Geocentric latitude (Latgeocen) is computed at step 3 through simple trigonometry. In step 4, this value is converted to geodetic form by a direct implementation of equation (1.8).

```
1.  Eterm1=E*E+F*F
2.  R1=SQR(Eterm+G*G)
3.  Latgeocen=ACS(SQR(Eterm1)/R1)
4.  Lat=ATN(TAN(Latgeocen)/(1-E2)
5.  RETURN
```

Program Operation

The Lagrange multiplier solution is a subroutine of GEOD. When GEOD is run, the operator is asked to select the input/output units and spheroid/datum reference to be used in the computations. The program then enters the Lagrange subroutine, requests the output selections, and initializes all the spheroid/datum dependent terms of the quartic equation. After the initialization has been completed, the program prompts the operator to make the following entries.

A. Parameter entries. Parameter entries are sequentially displayed as:

ENTER GEOCENTRIC E VALUE IN (selected units) ____

ENTER GEOCENTRIC F VALUE IN (selected units)

ENTER GEOCENTRIC G VALUE IN (selected units)

ENTER GEOID SEPARATION IN (selected units)

The program then forms the quartic equation and enters subroutine Quartic which returns the proper root of the quartic equation from which the geocentric coordinates of the surface point may be determined. The geocentric coordinates are converted to geodetic coordinates, and the altitude of the target above sea level is computed.

B. Program output. The entered values and computed values are displayed as:

```
DATUM:  NORTH AMERICAN (NAD)    EARTH MODEL:  CLARKE

GEOCENTRIC 'E' COORDINATE = -2459439.14 METERS
GEOCENTRIC 'F' COORDINATE = -4625532.17 METERS
GEOCENTRIC 'G' COORDINATE =  3643414.76 METERS

GEODETIC LATITUDE  =  35 00 00.0000  (35.00000000)  (0.610865237)
GEODETIC LONGITUDE = 118 00 00.0000 (118.00000000)  (2.059488518)
SPHEROID ELEVATION =     10000.00 METERS
GEOID ELEVATION    =     10023.50 METERS
```

The latitude and longitude are given in degrees, minutes, seconds, and decimal seconds, followed by the same values in degrees and radians. The output units (METERS shown) are those selected by the operator for input and output.

Program Validation

The Lagrange multiplier solutions provide coordinate and altitude information for off-spheroid targets. While the calculation of the universal space rectangular coordinates for any off-spheroid target point is mathematically simple, the reverse calculation is considerably more difficult. Because the forward calculations involve only simple trigonometric relationships, extremely accurate E-F-G coordinates can be obtained for any off-spheroid target point.

These values can then be used in the reverse solution, which employs a completely different mathematical concept. If the values returned from the reverse solution match the starting values used in the forward solution, then a successful validation is obtained.

For the validation cases shown in table 5.1, target altitudes of 0, 1000, 10,000, 100,000, 1,000,000, and 10,000,000 meters were used. 35 00 00.0000 N and 118 00 00.0000 W were used as the geodetic coordinates of the test point. As shown in the table, the reverse calculations returned position coordinates that were always within one-half centimeter of the starting point until the altitude exceeded 1,000,000 meters. Above this altitude the precision of the 12-digit computational system begins to limit the accuracy that can be attained. However, it is important to remember that the Lagrange method yields a closed-form solution whose accuracy is limited only by the precision of the computational system.

TABLE 5.1. LAGRANGE SOLUTION VALIDATIONS

Parameter	Actual value	Computed value
For E = −2455593.45 m, F = −4618299.59 m, and G = 3637679.00 m		
Geodetic latitude	35 00 00.0000	34 59 59.9999
Longitude	118 00 00.0000	118 00 00.0000
Altitude	0.00 m	0.00 m
For E = −2455978.02 m, F = −4619022.86 m, and G = 3638252.58 m		
Geodetic latitude	35 00 00.0000	35 00 00.0001
Longitude	118 00 00.0000	118 00 00.0000
Altitude	1000.00 m	1000.00 m
For E = −2459439.14 m, F = −4625532.27 m, and G = 3643414.76 m		
Geodetic latitude	35 00 00.0000	34 59 59.9998
Longitude	118 00 00.0000	118 00 00.0002
Altitude	10000.00 m	10000.00 m
For E = −2494050.31 m, F = −4690626.42 m, and G = 3695036.64 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	118 00 00.0001
Altitude	100000.00 m	100000.00 m
For E = −2840162.04 m, F = −5341567.92 m, and G = 4211255.44 m		
Geodetic latitude	35 00 00.0000	34 59 59.9999
Longitude	118 00 00.0000	117 59 59.9999
Altitude	1000000.00 m	1000000.00 m
For E = −6301279.35 m, F = −11850982.85 m, and G = 9373443.36 m		
Geodetic latitude	35 00 00.0000	35 00 00.0005
Longitude	118 00 00.0000	117 59 59.9999
Altitude	10000000.00 m	10000000.03 m

The method of Purcell and Cowan calculates geodetic latitude and longitude from universal space rectangular coordinates by the use of numerous small angle approximations that enhance computational speed but still retain a reasonable degree of computational accuracy.

Figure 5.2 represents the first quadrant of a meridian ellipse from an oblate earth of eccentricity, e. The position of the target is denoted by the point P, which is at an altitude h above the surface of the reference spheroid. The target's geodetic latitude is denoted by μ, its geocentric latitude is denoted by μ_1, and the geocentric latitude of the surface point whose outward normal passes through the target is denoted by μ_2. The magnitude of the geocentric position vector of the target is denoted by R, the geocentric radius of the spheroid at the point T is denoted by R_1, the geocentric radius of the spheroid at the point Q is denoted by R_2, and the geodetic radius at the surface normal point is denoted by N. The symbol a denotes the length of the semimajor axis of the spheroid, b denotes the length of the semiminor axis of the spheroid, d denotes the distance along the X axis from the origin to the intersection of the normal line, and c denotes the distance measured along the normal line from the point of intersection with the X axis to its intersection with the surface of the spheroid. The difference between geodetic and geocentric latitude is denoted by α at the point Q, by α_1 at the point T, and by β at the point P.

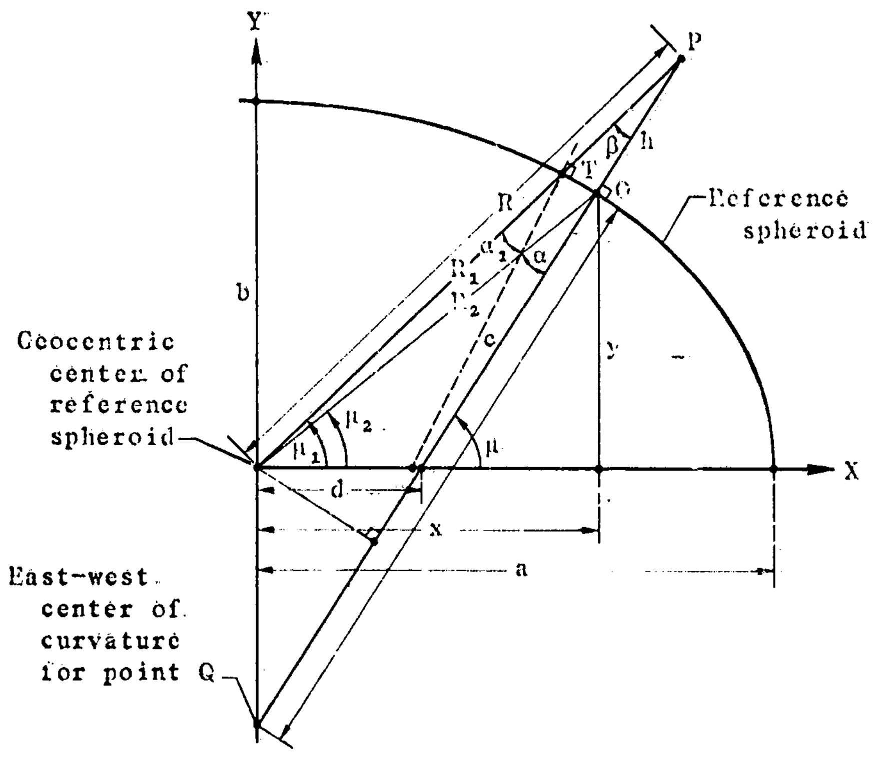

Figure 5.2.

It has been shown that the relation between geodetic and geocentric latitude for on-spheroid points is given by

$$\tan \mu_2 = (1 - \varepsilon^2)\tan \mu. \tag{5.21}$$

Using double-angle trigonometric relations, it is possible to write

$$\tan(\mu - \mu_2) = \frac{\tan \mu - \tan \mu_2}{1 + \tan \mu \tan \mu_2} \tag{5.22}$$

in which the tan μ_2 terms can be eliminated by the use of equation (5.21) and α can be substituted for $\mu - \mu_2$. Applying familiar trigonometric identities, equation (5.22) can then be reduced to

$$\tan \alpha = \frac{\varepsilon^2 - \sin \mu \cos \mu}{1 - \varepsilon^2 \sin^2\mu}, \tag{5.23}$$

or

$$\alpha = \arctan \frac{\varepsilon^2 - \sin \mu \cos \mu}{1 - \varepsilon^2 \sin^2\mu}. \tag{5.24}$$

A similar solution for α may be obtained in terms of μ_2 by eliminating the tan μ terms in equation (5.22) in the same manner as used above to eliminate the tan μ_2 terms. This yields

$$\alpha = \arctan \frac{\varepsilon^2 - \sin \mu_2 \cos \mu_2}{1 - \varepsilon^2 \sin^2\mu}. \tag{5.25}$$

The difference between geodetic and geocentric latitude at point T may be determined by the same method and may be expressed as

$$\alpha_1 = \arctan \frac{\varepsilon^2 - \sin \mu_1 \cos \mu_1}{1 - \varepsilon^2 \sin^2\mu}. \tag{5.26}$$

Again considering the geometry of the meridional ellipse shown in figure 5.2, the angles α and α_1 appear very nearly equal. Although figure 5.2 greatly exaggerates the eccentricity of the earth spheroid, the actual difference between α and α_1 can be shown never to exceed 0.0000001 radian. Because of this, the following convenient approximation can be made.

$$\alpha = \arctan \frac{\varepsilon^2 \sin \mu_1 \cos \mu_1}{1 - \varepsilon^2 \cos^2\mu} \tag{5.27}$$

It should be noted that, since α is very small, it is generally the practice to make a small-angle tangent approximation in solving for α. However, in the baseline program it was found that this substitution decreased the accuracy of the latitude calculation from about 0.0002 arc second to 0.01 arc second. This is quite a significant difference when comparing geodetic coordinates as given on the standard USCGS horizontal control sheets. Thus, since speed was not a problem in the baseline program, the usual small angle approximation was not made for α so that accuracies comparable with those of the USCGS data could be maintained.

Next, from the polar form of an equation for an ellipse, R_1 may be expressed as

$$R_1 = \frac{b}{(1 - \varepsilon^2 \cos^2\mu_1)^{1/2}} \,. \tag{5.28}$$

Applying the law of sines to triangle OPL provides the relation

$$d = R \sin\beta/\sin\mu = R_2 \sin\alpha/\sin\mu,$$

from which

$$\sin\beta = \frac{R_2 \sin\alpha}{R}. \tag{5.29}$$

Again, since the differences between α and α_1 and R_1 and R_2 are negligible, equation (5.29) can be rewritten as

$$\sin\beta \doteq \frac{R_1 \sin\alpha}{R} \tag{5.30}$$

without significant loss of accuracy. Finally, since both α_1 and β are very small, equation (5.30) can be simplified to

$$\beta = R_1\alpha/R. \tag{5.31}$$

The final step in the calculation of geodetic latitude from the target's geocentric latitude is accomplished by noting in figure 10 that

$$\mu = \mu_1 + \beta. \tag{5.32}$$

Thus, it has been shown that the transformation of geocentric to geodetic latitude can be approximated by:

1. Calculation of the angle α using eccentricity, ε, and geocentric latitude, μ_1 (eq. (5.27))
2. Calculation of R_1 using eccentricity, ε, and geocentric latitude, μ_1 (eq. (5.28))
3. Calculation of β using equation (5.31)
4. Calculation of geodetic latitude, μ, from geocentric latitude, μ_1, and β (eq. (5.32))

C - 2

It now remains to calculate the altitude of the target above the reference spheroid. Although the difference between R_1 and R_2 does not have an appreciable effect on the calculation of α, it can introduce measurable error in the calculation of h. Therefore, it is desirable to obtain a more accurate solution for h than would be possible by direct substitution. This is accomplished by obtaining a relationship between h and the known parameters R, R_1, α_1, and β.

From the polar equation for an ellipse, the ratio between R_2 and R_1 can be expressed as

$$\frac{R_2}{R_1} = \frac{(1 - \varepsilon^2 \cos^2\mu_1)^{1/2}}{(1 - \varepsilon^2 \cos^2\mu_2)^{1/2}}, \tag{5.33}$$

which upon rearranging may be written as

$$\frac{R_2}{R_1} = \left[1 + \frac{\varepsilon^2(\sin\mu_2 \cos\mu_1 + \cos\mu_2 \sin\mu_1)}{1 - \varepsilon^2 \cos^2\mu_2} \sin(\mu_1 - \mu_2)\right]^{1/2}. \tag{5.34}$$

It is apparent (fig. 5.2) that $\mu_1 - \mu_2$ equals $\beta - \alpha_1$, a very small angle. This allows another convenient approximation to be made. That is, $\sin\mu_1$ is approximately equal to $\sin\mu_2$, and $\cos\mu_1$ is approximately equal to $\cos\mu_2$. Thus, by substituting $\beta - \alpha_1$ for $\mu_1 - \mu_2$, using the sine and cosine approximations, and combining equations (5.27) and (5.28), the R_2/R_1 ratio may now be written as:

$$\frac{R_2}{R_1} = [1 + 2\alpha(\alpha - \beta)]^{1/2}. \tag{5.35}$$

Substituting the expression for β obtained from equation (5.31) yields

$$\frac{R_2}{R_1} = 1 + 2\alpha_1(1 - R_2/R)^{1/2}. \tag{5.36}$$

Applying the first two terms of a binomial expansion to equation (5.36) yields

$$\frac{R_2}{R_1} = 1 + \frac{\alpha_1^2}{R}(R - R_2). \tag{5.37}$$

Noting that $R - R_2$ very nearly equals h, and making the substitution

$$R_2 - R = \alpha_1\beta h \tag{5.38}$$

in equation (5.37), R_2 can be expressed as

$$R_2 = R_1 + \alpha_1\beta h. \qquad (5.39)$$

Applying trigonometric relations to triangles OPS and OQS in figure 5.2 yields

$$h = R\cos\beta - R_2\cos\alpha_1. \qquad (5.40)$$

Approximating $\cos\alpha_1$ and $\cos\beta$ using the first two terms of a cosine series, equation (5.40) becomes

$$h = R(1 - \beta^2/2) - R_2(1 - \alpha_1^2/2), \qquad (5.41)$$

or

$$h = (R - R_2) - \frac{1}{2}(R\beta^2 - R_2\alpha_1^2). \qquad (5.42)$$

Noting that $\alpha_1 \doteq \beta(R/R_2)$ and $\beta = \alpha_1(R_2/R)$, equation (5.42) can be rewritten as

$$h = (R - R_2)(1 + \alpha_1\beta/2). \qquad (5.43)$$

It now remains to combine equations (5.39) and (5.43) to obtain the following expression for h in terms of r, R_1, α_1, and β.

$$h = [R - (\alpha_1\beta h + R_1)](1 + \alpha_1\beta/2) \qquad (5.44)$$

Multiplying the two right-hand terms and rearranging, equation (5.44) becomes

$$h = \frac{(R - R_1)(1 + \alpha_1\beta/2)}{1 + \alpha_1\beta + \alpha_1^2\beta^2/2}. \qquad (5.45)$$

Since the second term in the numerator is the first two terms of the expansion of $1/(1 - \alpha_1\beta/2)$, and since the denominator is nearly the value of the first three terms of the expansion of $1/(1 - \alpha_1\beta/2)^2$, equation (5.45) can be approximated by

$$h = \frac{(R - R_1)(1 - \alpha_1\beta/2)^2}{(1 - \alpha_1\beta/2)} \qquad (5.46)$$

or

$$h = (R - R_1)(1 - \alpha_1\beta/2). \qquad (5.47)$$

Thus, it has been shown that by the use of small angle approximations and the first two or three terms of binomial expansions, it is possible to arrive at approximation solutions for the spheroid geodetic coordinates and the altitude above spheroid of a tracked target. While the numerous approximations involved in this solution method might tend to destroy confidence in the accuracy of the solution, the Program Validation section shows that the approximation delivers surprisingly accurate results that are well within the least-significant-bit accuracy of most modern radar tracking systems.

Variable Names

Name	Description
Aa	Length of semimajor axis of reference spheroid in meters
Alt	Target altitude above sea level in meters
Bb	Length of semiminor axis of reference spheroid in meters
Cost	Cosine of target geocentric latitude, μ_1
Cost2	Cos μ_1^2
Alpha	α_1 term in equation (5.26)
E	E coordinate in meters
E1	E coordinate in input units
E2	Eccentricity squared
Base	Square root of Base2
Base2	$E^2 + F^2$
Beta	β term in equation (5.31)
F	F coordinate in meters
F1	F coordinate in input units
G	G coordinate in meters
G1	G coordinate in input units
Radius	Length of target geocentric position vector in meters
R1	R_1 term in equation (5.28) in meters
Sint	Sine of target geocentric latitude μ_1
Ucnv	Conversion factor

Computational Algorithms

The essential algorithms used for the Purcell and Cowan approximation solution are as follows:

A. Subroutine Purcell: Subroutine Purcell contains the primary computational algorithms needed for the Purcell and Cowan solution. Upon initial entry into Purcell, the subroutine calls Osel (step 2) to allow the operator to make the output device selection. After the output selection has been made, the program calls Efgentry, which prompts the operator to enter the universal space rectangular coordinates of the target in the selected input/output units (E1-F1-G1). Subroutine Efgentry also requests the appropriate value of geoid separation in the selected units. If this value is not known, zero should be entered. The program then returns to subroutine Purcell where it first converts the input (E1-F1-G1) coordinates and separation of geoid value into meters (steps 5 to 7). Next the square of the base length (Base2) of the geocentric triangle is computed as $E^2 + F^2$ (step 8). At step 9 the square root of Base2 is taken to yield the value of the base leg. The geocentric radius (Radius) of the target point is next computed in step 10, and step 12 computes the geocentric latitude of the target point. Intermediate steps (13 to 16) simply compute the sine and cosine terms needed in steps 17 and 18. In step 17 a value for Alpha is computed from equation (5.27), in step 18 a value for R1 (figure 5.2) is computed from equation (5.28), in step 19 a value for Beta is computed from equation (5.31), and in step 20 a value for geodetic latitude is computed from equation (5.32). In step 20 the geodetic latitude in radians is converted to degrees by multiplying by the factor $180/\pi$. The degree mode is restored at step 22 and at step 23 the longitude of the surface point is computed using the usual trigonometric relations. After the computations are completed, the program calls subroutine Coordprint, which prints out and displays the results. If a value for the separation of geoid was entered in the Efgentry subroutine, then the target altitude is adjusted by this amount prior to printout. If additional entries are desired, the operator presses CONT, and the program returns to the Efgentry subroutine to receive the next set of E1-F1-G1 coordinates and the geoid separation value.

```
1.  Purcell:!
2.  GOSUB Osel
3.  Purcell1:!
4.  GOSUB Efgentry
5.  E=E1/Ucnv
6.  F=F1/Ucnv
7.  G=G1/Ucnv
8.  Base2=E*E+F*F
9.  Base=SQR(Base2)
10  Radius=SQR(Base2+G*G)
11. DEFAULT ON
12. RAD
13. U1=ATN(G/Base)
14. Cost=Base/Radius
15. Sint=G/Radius
```

```
Cost2=Cost*Cost
Alpha=ATN(E2*Sint*Cost/(1-E2*Cost2))
R1=Bb/SQR(1-E2*Cost2)
Beta=R1*Alpha/Radius
Lat=(U1+Beta)*180/PI
Alt=(Radius-R1)*(1-Alpha*Beta/2)
DEG
Lon=180-ATN(F/E)
DEFAULT OFF
GOSUB Coordprint
PAUSE
GOTO Purcell1
```

B. Efgentry subroutine: The Efgentry subroutine is used to receive operator inputs of the universal space rectangular coordinates and geoid separation in the selected input/output units. These values are used by the Purcell subroutine.

C. Coordprint subroutine: The Coordprint subroutine is common to all the off-spheroid coordinate determination programs. It prints both the input values (E1-F1-G1) and the output values of target latitude, longitude, spheroid elevation, and geoid elevation.

Program Operation

The Purcell and Cowan approximation solution is computed by the subroutines previously described. At the start of the main program, the operator is asked to select the desired input/output units and the spheroid/datum reference to be used in the calculations. Once these selections have been made, the main menu is displayed and the operator makes the EFG TO LAT, LON, AND ALT selection. The program then displays:

```
SELECT METHOD

0 = PURCELL AND COWAN
1 = LAGRANGE (CLOSED FORM)
2 = BOWRING
3 = GMD (CLOSED FORM)
```

The operator selects 0 and presses CONT to proceed to the Purcell and Cowan solution.

A. Parameter entries. Parameter entries are requested sequentially as:

```
ENTER GEOCENTRIC 'E' COORDINATE IN (selected units)

ENTER GEOCENTRIC 'F' COORDINATE IN (selected units)

ENTER GEOCENTRIC 'G' COORDINATE IN (selected units)

ENTER GEOID SEPARATION IN (selected units)
```

B. Program output. The entry parameters and computed values are output on the selected output device as:

```
GEOCENTRIC 'E' COORDINATE = -2459439.14 METERS
GEOCENTRIC 'F' COORDINATE = -4625532.17 METERS
GEOCENTRIC 'G' COORDINATE =  3643414.76 METERS

GEODETIC LATITUDE  =  35 00 00.0023  ( 35.00000065)  (0.610865250)
GEODETIC LONGITUDE = 118 00 00.0002  (118.00000005)  (2.059488518)
SPHEROID ELEVATION =     10000.00 METERS
GEOID ELEVATION    =     10014.57 METERS
```

As previously indicated, the unbracketed angle term is the angle value in degrees, minutes, and seconds. The first bracketed angle term is the angle value in degrees, and the second bracketed angle term is the angle value in radians.

Program Validation

Validation of the Purcell and Cowan routines is performed in the same manner as described for the other off-spheroid coordinate solutions, and the same entry values are used to provide a cross-check of computational accuracies between the programs. With the Purcell and Cowan solution method, the greatest degradation in accuracy occurs in the latitude calculation, which is in error by 11 meters when the target altitude is 10,000,000 meters. The altitude calculations return the same accuracies as do the closed-form solutions.

TABLE 5.2. PURCELL AND COWAN SOLUTION VALIDATIONS

Parameter	Actual value	Computed value
For E = -2455593.45 m, F = -4618299.59 m, and G = -3637679.00 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	118 00 00.0000
Altitude	0.00 m	0.00 m
For E = -2455978.02 m, F = -4619022.86 m, and G = 3638252.58 m		
Geodetic latitude	35 00 00.0000	35 00 00.0003
Longitude	118 00 00.0000	118 00 00.0000
Altitude	1000.00 m	1000.00 m
For E = -2459439.14 m, F = -4625532.27 m, and G = 3643414.76 m		
Geodetic latitude	35 00 00.0000	35 00 00.0023
Longitude	118 00 00.0000	118 00 00.0002
Altitude	10000.00 m	10000.00 m
For E = -2494050.31 m, F = -4690626.42 m, and G = 3695036.64 m		
Geodetic latitude	35 00 00.0000	35 00 00.0234
Longitude	118 00 00.0000	118 00 00.0001
Altitude	100000.00 m	100000.00 m
For E = -2840162.04 m, F = -5341567.92 m, and G = 4211255.44 m		
Geodetic latitude	35 00 00.0000	35 00 00.1801
Longitude	118 00 00.0000	117 59 59.9999
Altitude	1000000.00 m	1000000.00 m
For E = -6301279.35 m, F = -11850982.85 m, and G = 9373443.36 m		
Geodetic latitude	35 00 00.0000	35 00 00.3634
Longitude	118 00 00.0000	117 59 59.9999
Altitude	10000000.00 m	9999999.97 m

The Bowring Approximation Method

The Bowring method calculates the geodetic latitude and altitude of a target from universal space rectangular coordinates by the use of successive approximations. The program requires rough calculations of geodetic latitude and target altitude as inputs. These calculations are refined during one or more passes through the corrector equations until a specified level of accuracy is achieved. In practice it has been found that a single pass through the equations yields results that are comparable to those of the closed-form solutions. For 12-digit computational equipment, one-pass solutions for target points under approximately 1,000,000 meters are close to the precision limit of the system. For higher altitude points, the precision limit of the system is reached on the second pass, so additional passes are unnecessary and do not improve the results.

The theory presented in this section reflects findings contained in reference 7, which was prepared by the Ohio State University Mapping and Charting Research Laboratory. However, emphasis in the present document is placed on the derivation of the approximating equations and on a comparison of the results of the Bowring method with results from the other solution techniques.

Figure 5.3 shows the first quadrant of a meridional ellipse. N is the normal line QP_0, the extension of which passes through an off-spheroid point P. G_0 is the G coordinate value of the surface point P_0, and G is the G coordinate value of the off-spheroid point P. The distance of the point P_0 from the rotational (G) axis is given by r_0, and the distance of the point P from the G axis is given by r. Obviously, $r_0^2 = E_0^2 + F_0^2$ and $r^2 = E^2 + F^2$. Other important geometric relationships from figure 5.3 are:

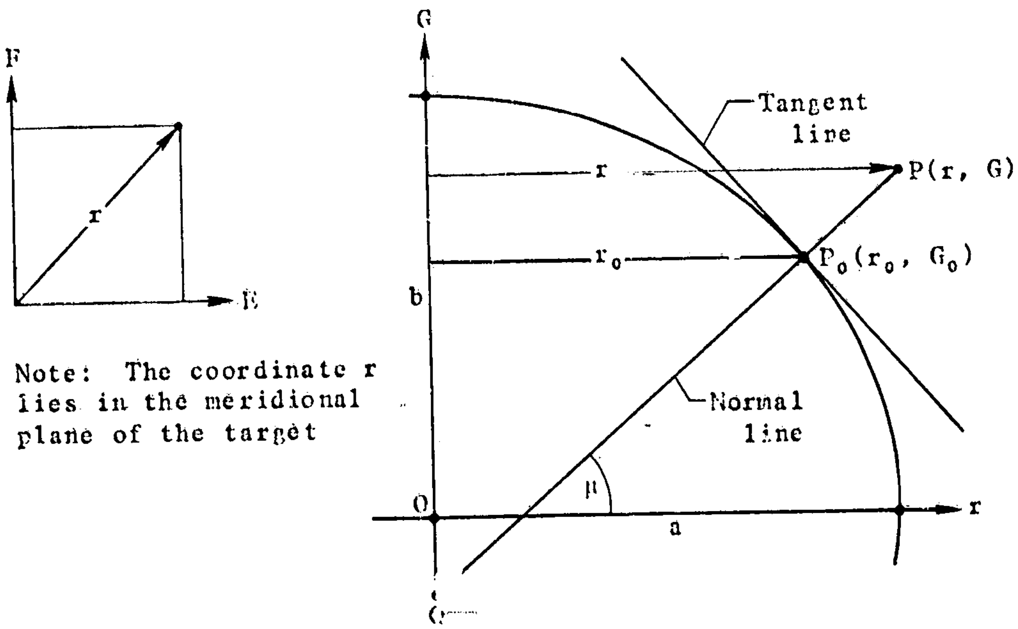

Figure 5.3.

$$r_0 = r - h \cos \mu = N \cos \mu, \tag{5.48}$$

and

$$G_0 = G - h \sin \mu = [(1 - \varepsilon^2)N] \sin \mu. \tag{5.49}$$

The equation of the meridional ellipse shown in figure 5.3 is given by

$$\frac{r^2}{a^2} + \frac{G^2}{b^2} = 1. \tag{5.50}$$

The slope of a tangent line to the ellipse is obtained by implicit differentiation of equation (5.50):

$$\frac{2r}{a^2} + \frac{2G}{b^2}\frac{dG}{dr} = 0 \tag{5.51}$$

or

$$\frac{dG}{dr} = -\frac{b^2}{a^2}\frac{r}{G}. \tag{5.52}$$

Obviously, the slope of the normal line is the negative reciprocal of the slope of the tangent line, or

$$-\frac{dr}{dG} = \frac{a^2}{b^2}\frac{G}{r}. \tag{5.53}$$

The equation for points lying on the tangent line to the meridional ellipse at point P_0 is

$$G_t - G_0 = -\frac{b^2}{a^2}\frac{r_0}{G_0}(r_t - r_0), \tag{5.54}$$

where r_t and G_t define the set of points lying on the tangent line.

Recalling that $b^2/a^2 = 1 - \varepsilon^2$ and grouping the r and G terms yields

$$a^2 = r_0 r_t + \frac{1}{1 - \varepsilon^2} G_0 G_t. \tag{5.55}$$

Equation (5.55) must be satisfied for all points lying on the line tangent to the meridional ellipse at the point P_0.

Similarly, the equation for a line normal to the meridional ellipse at the point P_0 is.

$$G_n - G_0 = \frac{a^2}{b^2}\frac{G_0}{r_0}(r_n - r_0). \tag{5.56}$$

In equation (5.56), r_n and G_n define the set of points on a line normal to the meridional ellipse at the point (r_0, G_0).

Again, by substituting $a^2/b^2 = 1/(1 - \varepsilon^2)$ and grouping like terms, equation (5.56) may be rewritten as

$$\frac{r_n}{r_0} - \frac{G_n}{G_0}(1 - \varepsilon^2) = \varepsilon^2. \tag{5.57}$$

If the relations for r_0 and G_0 given in equations (5.48) are substituted and the terms rearranged, equation (5.57) may be rewritten for the line passing through P_0 and P as:

$$(1 - \varepsilon^2)\, r \sin\mu - G\cos\mu + \varepsilon^2 h \sin\mu\cos\mu = 0. \tag{5.58}$$

Since the specific coordinates of the point P have been defined as r and G, the subscript n has been dropped in equation (5.58).

Substituting $k = (a^2/b^2) - 1$ and rearranging equation (5.58) further yields

$$\tan\mu = \frac{(1 - k)G}{r} - \frac{k(h \sin\mu)}{r}. \tag{5.59}$$

To obtain starting values for h and sin μ, a spherical solution may be used. Thus,

$$\mu = \arctan(G/r) \text{ and } h = (r^2 + G^2)^{1/2} - (a + b)/2. \tag{5.60}$$

These values are then substituted into the h and sin μ terms in equation (5.59) to obtain a more accurate approximation of μ.

Because k is always less than 0.007 and the value of h sin μ is generally much smaller than the values of r and G, errors in the rough calculation of h sin μ should not greatly affect the first-pass approximation of tan μ obtained from equation (5.59). (Note that equation (5.59) is an exact equation, but the result is approximate because both h and sin μ are only roughly known at this point.)

The next step makes use of the equation for the tangent line at the point P_0 (eq. (5.55)). Since both the coordinates (r, G) of the point P are known exactly, and since the value of μ has been approximated with reasonable accuracy using equation (5.59), it is now possible to apply the general formula for the distance between a point and a line to compute a more accurate value of h.

From analytic geometry, the distance d between a point $P(x_1, y_1)$ and a line whose equation is in the form $Ax + By + C = 0$ is given by the formula

$$d = \frac{Ax_1 + By_1 + C}{(A^2 + B^2)^{1/2}}. \tag{5.61}$$

In this case, the coordinates of the point P are (r, G) and the equation for the tangent line in the r and G coordinate system was given by equation (5.55). Substituting values for r_0 and G_0 from equations (5.49) and rearranging the terms of equation (5.55) to the form $Ax + By + C = 0$ yields

$$r_t \cos \mu + G_t \sin \mu - a^2/N = 0, \tag{5.62}$$

where r_t and G_t again define the set of points lying on the tangent line. If the values from equation (5.62) are now substituted into formula (5.61), a more accurate value of h can be obtained as

$$h = r \cos \mu + G \sin \mu - a^2/N, \tag{5.63}$$

where

$$N = a/(1 - \varepsilon^2 \sin^2\mu)^{1/2}. \tag{5.64}$$

If the first-pass value of tan μ obtained from equation (5.59) is denoted by

$$t = \tan \mu_1 = \frac{(1 + k)G - k(h \sin \mu)_1}{r}, \tag{5.65}$$

and if $(h \sin \mu)_2$ denotes a second approximation of $h \sin \mu$ based on the value of μ obtained from equation (5.65) and a value of h obtained from equation (5.63), then a second approximation for $h \sin \mu$ may be obtained as follows. A new value of h (denoted by h_1) is obtained from reapplication of the equation for the distance between a point and a line,

$$h_1 = r \cos \mu_1 + G \sin \mu_1 - a^2/N_1, \tag{5.66}$$

where

$$N_1 = a/(1 - \varepsilon^2 \sin^2\mu_1)^{1/2}. \tag{5.67}$$

Then, using the new value of h, $(h \sin \mu)_2$ can be expressed as

$$(h \sin \mu)_2 = h_1 \sin \mu_1. \tag{5.68}$$

The final value for μ may now be obtained by substituting the value of $h \sin \mu$ given by equation (5.68) into equation (5.65) and solving for μ as

$$\mu_2 = \arctan \frac{(1 + k)G - k(h \sin \mu)_2}{r}. \qquad (5.69)$$

The final value for h is computed from either

$$h = \frac{G}{\sin \mu_2} - (1 - e^2)N, \qquad (5.70)$$

or

$$h = \frac{r}{\cos \mu_2} - N. \qquad (5.71)$$

Greater accuracy is obtained from the equation having the larger denominator in the first term.

In the event that additional accuracy is required, a second pass through the approximating equations can be made using the value of μ obtained from equation (5.69) and the value of h obtained from either equation (5.70) or (5.71).

Summarizing, the steps used in calculating geodetic latitude and altitude are:

1. Obtain rough approximations of geodetic latitude and altitude using equations (5.60).
2. Use the values of h and μ obtained from step 1 in equation (5.65) to obtain an improved value for μ which is needed in equation (5.68). Also compute N from the new value of μ.
3. Solve equation (5.68) for an improved value of $h \sin \mu$.
4. Solve equation (5.69) for an improved value of μ.
5. Compute the final value of h from either equation (5.70) or (5.71).
6. If additional accuracy is required, use the values of μ and h from steps 4 and 5 to again compute μ using equation (5.65) and follow the same procedures as outlined in steps 3 to 5. Repeat as many times as necessary to reach the precision limit of the system.

Variable Names

Name	Description
Aa	Length of semimajor axis of reference spheroid in meters
Aa2	a^2 in meters squared
Alt	Height of off-spheroid point above reference spheroid in meters

Bb	Length of semiminor axis of reference spheroid in meters
Bb2	b^2 in meters squared
Cosu	Cosine of geodetic latitude
E	E coordinate in meters
E1	E coordinate in input units
E2	Eccentricity squared
F	F coordinate in meters
F1	F coordinate in input units
G	G coordinate in meters
G1	G coordinate in input units
Geoidsep	Separation of geoid in meters
Geosep	Separation of geoid in input units
H	Working height of target above reference spheroid in meters
Hsinu	Product of h and the sine of the target's geodetic latitude
K	$(a^2/b^2) - 1$
N	Length of great normal passing through the surface point in meters
R	Square root of R2 in meters
R2	$E^2 + F^2$ in meters squared
Sinu	Sine of geodetic latitude
T	Tangent of U
Times	Number of times the approximation equations are to be repeated
U	Geodetic latitude in degrees
Ucnv	Conversion factor

Computational Algorithms

The essential algorithms used for the Bowring approximation method are as follows:

A. Subroutine Bowring: Subroutine Bowring contains the primary computational algorithms needed for the Bowring solution. Upon initial entry into Bowring, the subroutine calls Osel to allow the operator to make the output device selection. After the output selection has been made, the program calls Efgentry which prompts the operator to enter the universal space rectangular coordinates of the target in the selected input/output units (E1-F1-G1). Subroutine Efgentry also requests the appropriate value of geoid separation in the selected units. If this value is unknown, zero should be entered. The program then returns to subroutine Bowring where the E1, F1, and G1 values are converted from input units into meters and stored in steps 9 to 11 as E, F, and G. Using the values of E and F, a value for R is computed (steps 12 and 13) for use in the approximating equations. The other input parameter required by the Bowring algorithm is G. In steps 15 and 17 the program implements equations (5.60) to obtain a first approximation of geodetic latitude and target altitude. A FOR-NEXT loop is established in step 18 to control the number of passes which will be made through the approximating equations. In the baseline program, the number of passes is operator selectable. For example, if two passes are desired, then the simple variable Times is set equal to 2 to provide two passes through the FOR-NEXT loop. Upon entry into the loop, steps 19 and 20 implement equation (5.65) and provide a better approximation of geodetic latitude, μ. The new value of μ is then used to improve on the estimate of target altitude by implementation of equations (5.67) and (5.66) (steps 23 to 25). Using the improved value of h from step 25, a new value of μ is obtained in step 27 through a direct implementation of equation (5.69). The value of N is again computed (step 29) using the latest value of μ, and a final value of h is determined from equation (5.70) or (5.71), depending on the magnitude of μ. At step 31 the the first pass is completed. If additional passes have been selected by the operator, step 32 will cause the program to return to step 18 where the same sequence of operations will be commenced using the values of μ and h returned from the pass just completed.

To determine whether the maximum accuracy has been achieved, test cases can be run using several iterations through the approximating equations. When the values stabilize, the approximations have reached the maximum accuracy attainable with the system. It has been found that one pass through the above algorithm reaches the precision limit of a 12-digit computational system for all but very high altitude points (10,000,000 meters or higher).

After the latitude and altitude computations have been completed, the longitude of the surface point is obtained from the usual trigonometric relations (step 34), and the results are displayed or printed in the universal Coordprint subroutine. If additional values are to be computed, the operator presses CONT and the program returns to the entry point.

```
Bowring:!
GOSUB Osel
Aa2=Aa*Aa
Bb2=Bb*Bb
K=Aa2/Bb2-1
Times=2
Bowring1:!
GOSUB Efgentry
E=E1/Ucnv
F=F1/Ucnv
G=G1-Ucnv
R2=E*E+F*F
R=SQR(R2)
DEFAULT ON
U=ATN(G/R)
DEFAULT OFF
H=SQR(R2+G*G)-(Aa+Bb)/2
FOR Count=1 TO Times
T=((1+K)*G-K*(H*SIN(U)))/R
DEFAULT ON
U=ATN(T)
DEFAULT OFF
GOSUB Ncalc
Cosu=COS(U)
H=R*Cosu+G*Sinu-Aa2/N
Hsinu=H*Sinu
U=ATN((1+K)*G-K*Hsinu)/R
Cosu=COS(U)
GOSUB Ncalc
IF Sinu>Cosu THEN H=G/Sinu-(1-E2)*N
IF Cosu>=Sinu THEN H=R/Cosu-N
NEXT Count
DEFAULT ON
Lon=180-ATN(F/E)
DEFAULT OFF
Lat=U
Alt=H
GOSUB Coordprint
PAUSE
GOTO Bowring1
```

B. Efgentry subroutine: The Efgentry subroutine is used to receive operator inputs of the universal space rectangular coordinates and geoid separation in the selected input/output units. These values are used by the Bowring subroutine.

C. Coordprint subroutine: The Coordprint subroutine is common to all the off-spheroid coordinate determination programs. It prints both the input values (E1-F1-G1) and the output values of target latitude, longitude, spheroid elevation, and geoid elevation.

Program Operation

The Bowring approximation solution is computed by the subroutines previously described. At the start of the main program, the operator is asked to select the desired input/output units and the spheroid/datum reference to be used in the calculations. Once these selections have been made, the main menu is displayed and the operator makes the EFG TO LAT, LON, and ALT selection. The CRT then displays a request for the operator to select the desired solution method as follows:

SELECT METHOD

0 = PURCELL AND COWAN
1 = LAGRANGE (CLOSED FORM)
2 = BOWRING
3 = GMD (CLOSED FORM)

To use the Bowring approximation, the operator enters 2.

The next CRT display requests that the operator select the desired output device.

SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER

The program then requests that the operator select the number of passes to be made through the approximation equations. This request is displayed as

ENTER NUMBER OF PASSES (DEFAULTS TO 2)

The last operator entries requested are the E-F-G coordinates and the geoid separation. This request is displayed sequentially as:

ENTER GEODETIC 'E' COORD IN (selected units)

ENTER GEODETIC 'F' COORD IN (selected units)

ENTER GEODETIC 'G' COORD IN (selected units)

ENTER GEOID SEP IN (selected units)

After each prompting message shown above is displayed on the CRT, the operator responds by entering the appropriate values in the units selected. The program converts all input values to meters and then calculates the target altitude and geodetic latitude. Unless another number of passes has been selected by the operator, the program makes two passes through the computational algorithms. When the altitude and geodetic latitude computations are completed, the program calculates target longitude and then calls Coordprint.

Subroutine Coordprint displays the computed results in the following form:

```
DATUM:  NORTH AMERICAN (NAD)    EARTH MODEL:  CLARKE 1866

GEOCENTRIC 'E' COORDINATE =  -2459439.14 METERS
GEOCENTRIC 'F' COORDINATE =  -4625532.27 METERS
GEOCENTRIC 'G' COORDINATE =   3643414.76 METERS

GEODETIC LATITUDE  =  34 59 59.9999   ( 34.99999998)   (0.610865238)
GEODETIC LONGITUDE = 118 00 00.0002   (118.00000005)   (2.059488518)
SPHEROID ELEVATION =          10000.00 METERS
GEOID ELEVATION    =          10014.28 METERS
```

Again, the unbracketed angle term is the angle value in degrees, minutes, and seconds. The first bracketed angle term is the angle value in degrees, and the second bracketed angle term is the angle value in radians. In the example shown, meters are the selected input/output units.

Program Validation

Validation of the Bowring approximation solutions is performed in the same manner as described for the other off-spheroid coordinate determination programs, and the same entry values are used to allow comparison of the computational accuracies of the four off-spheroid programs. It is interesting to note that the Bowring approximation method achieves better accuracy than the closed-form solutions when just one pass is used. This is due to a loss in accuracy when roots of large high-order terms in the quartic equations must be found.

Table 5.3 contains results from the Bowring method when one pass through the approximating equations is used. Table 5.4 contains results from the two-pass solution. Although the second pass provides some improvement in accuracy, it is not apparent in table 5.4 except for the 10,000,000 meter altitude point. The improvement at the lower levels occurs in digits beyond the precision shown in the tables and is far less than the least-significant-bit values of common tracking equipment.

TABLE 5.3. ONE-PASS BOWRING SOLUTION VALIDATIONS

Parameter	Actual value	Computed value
For E = -2455593.45 m, F = -4618299.59 m, and G = 3637679.00 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	118 00 00.0000
Altitude	0.00 m	0.00 m
For E = -2455978.02 m, F = -4619022.86 m, and G = 3638252.58 m		
Geodetic latitude	35 00 00.0000	35 00 00.0001
Longitude	118 00 00.0000	118 00 00.0000
Altitude	1000.00 m	1000.00 m
For E = -2459439.14 m, F = -4625532.27 m, and G = 3643414.76 m		
Geodetic latitude	35 00 00.0000	34 59 59.9999
Longitude	118 00 00.0000	118 00 00.0002
Altitude	10000.00 m	10000.00 m
For E = -2494050.31 m, F = -4690626.42 m, and G = 3695036.64 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	118 00 00.0001
Altitude	100000.00 m	100000.00 m
For E = -2840162.04 m, F = -5341567.92 m, and G = 4211255.44 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	117 59 59.9999
Altitude	1000000.00 m	1000000.00 m
For E = -6301279.35 m, F = -11850982.85 m, and G = 9373443.36 m		
Geodetic latitude	35 00 00.0000	34 59 59.9984
Longitude	118 00 00.0000	117 59 59.9999
Altitude	10000000.00 m	9999999.91 m

TABLE 5.4. TWO-PASS BOWRING SOLUTION VALIDATIONS

Parameter	Actual value	Computed value
For E = -2455593.45 m, F = -4618299.59 m, and G = 3637679.00 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	118 00 00.0000
Altitude	0.00 m	0.00 m
For E = -2455978.02 m, F = -4619022.86 m, and G = 3638252.58 m		
Geodetic latitude	35 00 00.0000	35 00 00.0001
Longitude	118 00 00.0000	118 00 00.0000
Altitude	1000.00 m	1000.00 m
For E = -2459439.14 m, F = -4625532.27 m, and G = 3643414.76 m		
Geodetic latitude	35 00 00.0000	34 59 59.9999
Longitude	118 00 00.0000	118 00 00.0002
Altitude	10000.00 m	10000.00 m
For E = -2494050.31 m, F = -4690626.42 m, and G = 3695036.64 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	118 00 00.0001
Altitude	100000.00 m	100000.00 m
For E = -2840162.04 m, F = -5341567.92 m, and G = 4211255.44 m		
Geodetic latitude	35 00 00.0000	35 00 00.0001
Longitude	118 00 00.0000	117 59 59.9999
Altitude	1000000.00 m	1000000.00 m
For E = -6301279.35 m, F = -11850982.85 m, and G = 9373443.36 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	117 59 59.9999
Altitude	10000000.00 m	10000000.00 m

The GMD Closed-Form Solution

The GMD closed-form solution computes the geodetic latitude and altitude of a target from universal space rectangular space coordinates by the use of two simultaneous equations. The first is the equation for the normal line to a meridional ellipse that passes through the target point. The second is the standard equation for the reference ellipse. Since the GMD method provides a direct closed-form solution, the accuracy of the result is dependent only on the precision limit of the system. With a 12-decimal-digit computational word length, latitude calculations are accurate to better than 0.0001 arc second and altitude calculations are accurate to better than 0.01 meter up to altitudes of 10,000,000 meters.

Figure 5.4 shows the first quadrant of a meridional ellipse. N is the normal line QP_0, the extension of which passes through any off-spheroid point P. G_0 is the G coordinate value of the surface point P_0, and G is the G coordinate value of the off-spheroid point P. The distance of the point P_0 from the rotational (G) axis is given by r_0, and the distance of the point P from the G axis is given by r. Obviously, $r_0^2 = E_0^2 + F_0^2$ and $r^2 = E^2 + F^2$. Note that the r and G coordinates and the geometrical relationships between the points P and P_0 are the same as those shown in figure 5.3 for the Bowring solution.

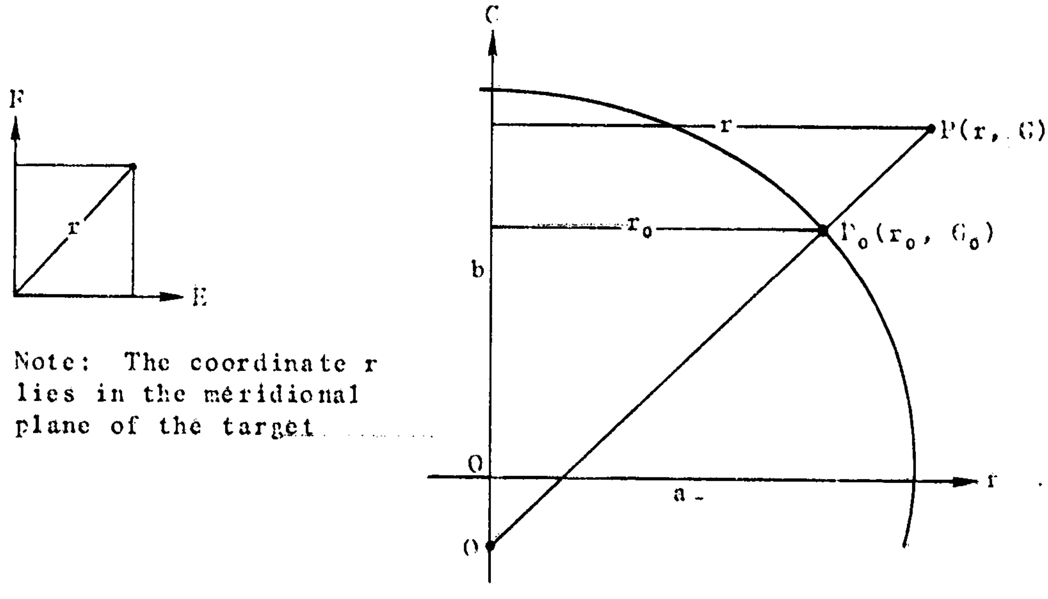

Figure 5.4.

The equation for the normal line to the meridional ellipse that passes through the points P and P_0 has been derived in the preceeding section as

$$G - G_0 = \frac{a^2}{b^2}\frac{G_0}{r_0}(r - r_0). \tag{5.72}$$

Equation (5.72) may be solved for G_0, yielding

$$G_0 = \frac{b^2 r_0 G}{a^2 r - r_0(a^2 - b^2)}. \tag{5.73}$$

The equation for the meridional ellipse shown in figure 5.4 is given by

$$\frac{r^2}{a^2} + \frac{G^2}{b^2} = 1. \tag{5.74}$$

The coordinates of the surface point $P_0(r_0, G_0)$ must therefore satisfy equation (5.74) as

$$\frac{r_0^2}{a^2} + \frac{G_0^2}{b^2} = 1. \tag{5.75}$$

Substituting the value for G_0 from equation (5.73) into equation (5.75), combining terms, and clearing fractions yields an expression for r_0 in the form of a quartic equation,

$$\begin{aligned}
&[(a^2 - b^2)^2 b^2]r_0^4 - [(2a^4b^2 - 2a^2b^4)r]r_0^3 \\
&+ [(a^4b^2)r^2 + (a^2b^4)G^2 - a^2b^2(a^2 - b^2)^2]r_0^2 \\
&+ [(2a^6b^2 - 2a^4b^4)r]r_0 - [(a^6b^2)r]^2 = 0.
\end{aligned} \tag{5.76}$$

Since a and b are known elliptical parameters, and both r and G are the known coordinates of the point P, the substitutions

$$A = (a^2 + b^2)^2 b^2 \tag{5.77}$$

$$B = -(2a^4b^2 - 2a^2b^4)r \tag{5.78}$$

$$C = (a^4b^2)r^2 + (a^2b^4)G^2 - a^2b^2(a^2 - b^2)^2 \tag{5.79}$$

$$D = (2a^6b^2 - 2a^4b^4)r \tag{5.80}$$

$$E = -(a^6b^2)r^2 \tag{5.81}$$

may be made to yield an equation in the form

$$Ar_0^4 + Br_0^3 + Cr_0^2 + Dr_0 + E = 0. \tag{5.82}$$

The solution of equation (5.82) will yield two real roots. One will be the value of r which lies on the normal line where it intersects the meridional ellipse directly below the target point. The other intersection will occur at the far side of the ellipse where the negative extension of the normal line again intersects the meridional ellipse. Two other roots will be obtained in the solution of the quartic equation. Both of these will be imaginary numbers with no significance to this solution. Knowing the nature of the roots, it can be seen from figure 12 that the real root whose sign is the same as the sign of r will be the proper root.

In the GMD algorithm, equation (5.82) is solved using subroutine Quartic, and the value of r_0 obtained from this subroutine is then substituted into equation (5.73) to obtain the corresponding value for G_0.

Knowing both r_0 and G_0, it is a simple matter to derive the geocentric latitude of the surface point P_0. Then, using equation (1.8), the geocentric latitude of the surface normal point can easily be converted to geodetic latitude. Having the exact geodetic latitude of the surface point, the altitude of the target can be obtained from either equation (5.70) or equation (5.71).

The longitude of the surface normal point is easily found from the expression

$$\lambda = 180 - \arctan(F/E). \tag{5.83}$$

This completes the derivation of the GMD closed-form solution to the problem of computing a target's geodetic coordinates and altitude from known E-F-G coordinates. It is basically a solution of two simultaneous equations with two unknowns. Unfortunately, the relationships are such that the solution involves finding the roots of a quartic equation, a task which is somewhat costly from the standpoint of computing time. However, the method is slightly faster than the Lagrange solution and is more direct in that the actual coordinate value is obtained rather than the secondary Lagrange multiplier term. It has been found that, for extremely high altitude targets, the GMD algorithms will provide somewhat better accuracy than the Lagrange solution. For example, the E-F-G coordinates of a target at 35 degrees latitude, 118 degrees longitude, and 1,000,000,000 meters altitude are:

$$E = -387024183.84,\ F = -727886625.27,\ \text{and}\ G = 577214115.35.$$

On a 12-digit computational system, the Lagrange solution will compute a target latitude of 35.00001604 degrees and a target altitude of 1,000,000,197.3 meters. Using the same E-F-G coordinates, the GMD algorithm will yield a latitude of 35.00000812 degrees and an altitude of 1,000,000,099.8 meters. Obviously, both solutions have reached the precision limit of the system because of the extremely large altitude computation. However, in this instance, it appears that the GMD solution yields accuracies that are slightly better than those obtained using the Lagrange solution. At a target height of 100,000,000 meters, the GMD routine returns a result that is accurate to within 4/10,000 of an arc second in angle and 0.2 meter in altitude. At and below 10,000,000 meters target altitude, the GMD routines return values accurate to within 1/10,000 arc second in angle and 0.01 meter in altitude.

Variable Names

Name	Description
A0	Constant portion of A term in equation (5.82)
Aa	Length of semimajor axis of reference spheroid in meters
Alt	Altitude of target above reference spheroid in meters
Amb	a - b
An	Normalized value of semimajor axis (An = 1)
Apb	a + b
B0	Constant portion of B term in equation (5.82)
B2	Second power of normalized b term
B4	Fourth power of normalized b term
Bb	Length of semiminor axis of reference spheroid in meters
Bn	Normalized value of semiminor axis
C0a	Constant portion of first part of C term
C0b	Constant portion of second part of C term
C0c	Constant portion of third part of C term
Cosu	Cosine of geodetic latitude
D0	Constant part of D term
E0	Constant part of E term
E1	Input E coordinate in selected units
E2	Eccentricity squared
En	Normalized E coordinate
F1	Input F coordinate in selected units
Fn	Normalized F coordinate
G1	Input G coordinate in selected units
Gn	Normalized G coordinate

Gn0	Normalized G coordinate of surface point
Lat	Geodetic latitude of surface point
Lon	Longitude of surface point
N	Great normal through surface point
Rn	Normalized r coordinate of target point
Rn0	Normalized r coordinate of surface point
Rn2	Square of normalized r coordinate of target point
Sinu	Sine of geodetic latitude of surface point
Tanup	Tangent of geocentric latitude of surface point
Terma	Term A in equation (5.82)
Termb	Term B in equation (5.82)
Termc	Term C in equation (5.82)
Termd	Term D in equation (5.82)
Terme	Term E in equation (5.82)
U	Geodetic latitude of surface point (also Lat)
Ucnv	Conversion factor
X1	First real root of quartic equation
X2	Second real root of quartic equation

Computational Algorithms

Algorithms essential to the GMD closed-form computation of geodetic latitude, longitude, and altitude from universal space rectangular coordinates are presented below.

A. Subroutine Gmd: Subroutine Gmd is the main computational subroutine for the GMD solution. Upon initial entry into Gmd, steps 1 to 16 compute the constant parts of the A, B, C, D, and E terms in equations (5.77) to (5.81). These are dependent only on the a and b values of the selected reference spheroid, and, once calculated, can be used throughout all subsequent computations. Both the a and b values are normalized (steps 2 and 3) to the length of the semimajor axis (Aa) of the reference spheroid. This causes a and all higher powers of a to equal 1 and reduces the b values to less than 1. All length parameters are normalized to prevent a precision overflow in the higher power terms.

At step 17 the program calls Osel to allow the selection of the desired output device. After the output device selection has been made, the program calls Efgentry (step 19) which prompts the operator to enter the universal space rectangular coordinates of the target (E1, F1, and G1) in the selected input/output units. In steps 20 to 22, these values are converted to meters and normalized. Steps 23 and 24 reduce the three dimensional problem to two dimensions by computing the value Rn, which is the normalized abscissa of the target lying in the plane of the meridional ellipse passing through the target point. In steps 25 through 29, the constant portions of terms A, B, C, D, and E (eqs. (5.77) to (5.81)) are combined with the normalized target coordinates Rn and Gn to obtain the complete coefficients of the quartic equation (5.82). Note that program variable Rn represents r in equations (5.77) to (5.81), and Gn represents G in equation (5.79). Subroutine Quartic is called at step 30 and four roots of equation (5.82) are computed. The two real roots (X1 and X2) are then tested in steps 31 and 32 to determine the root whose sign matches the sign of the target coordinate Rn. The simple variable Rn0 is set equal to the root with the matching sign. This root is the abscissa of the surface point whose normal passes through the target. At step 33, the surface coordinate Rn0 is substituted into equation (5.73) to yield a value of Gn0, the normalized ordinate of the surface point.

In the solution to equation (5.73), note that program variable Rn0 represents R_0, Gn0 represents G_0, Rn represents r, and Gn repesents G. Also note that the value of a in the same equation is equal to 1 because all variables have been normalized to the length of the semimajor axis. In the solution for Gn0 used in step 33, the $(a^2 - b^2)$ term in the denominator has been factored into (a + b) and (a - b), which are represented as program variables Apb and Amb.

Step 34 computes the tangent of the geocentric latitude (tanup) of the surface point, and step 36 applies equation (1.8) to convert geocentric latitude to geodetic latitude. The value of geodetic latitude is then used in step 40 to compute the E-W radius of curvature, N, that is needed for the derivation of the target altitude in equation (5.70) or (5.71). Again, note that the selection of equation (5.70) or (5.71) is made (steps 41 and 42) based on a test to determine which will yield the more accurate calculation of altitude for the specific value of μ.

The value of longitude is determined very simply in step 45 by resolving the target's E and F coordinates into angular form, and the universal subroutine Coordprint is called to display the results. After displaying the results, the program pauses (step 48). If another point is to be reduced, the program will next proceed to Step 18 where a new set of E-F-G coordinates may be entered.

It should also be mentioned that if a value for geoid separation were entered in subroutine Efgentry, then the final sea-level elevation value would be corrected by the same amount.

```
Gmd:!
An=1
Bn=Bb/An
B2=Bn*Bn
B4=B2*B2
Apb=An+Bn
Amb=An-Bn
Amb2=Amb*Amb
Apb2=Apb*Apb
A0=Amb2*Apb2*B2
B0=-2*B2*Apb*Amb
C0a=B2
C0b=B2*Amb2*Apb2
C0c=B4
D0=2*B2*Apb*Amb
E0=-B2
GOSUB Osel
Gmd1:!
GOSUB Efgentry
En=E1/Ucnv/Aa
Fn=F1/Ucnv/Aa
Gn=G1/Ucnv/Aa
Rn2=En*En+Fn*Fn
Rn=SQR(Rn2)
Terma=1
Termb=B0*Rn/A0
Termc=(C0a*Rn2+C0b+C0c*Gn*Gn)
Termd=D0*Rn/A0
Terme=E0*Rn2/A0
GOSUB Quartic
Rn0=X2
IF Sgn(X1)=SGN(Rn) THEN Rn0=X1
Gn0=Gn*B2*Rn0/(Rn-Rn0*Apb*Amb)
Tanup=Gn0/Rn0
DEFAULT ON
U=ATN(Tanup/(1-E2))
DEFAULT OFF
Sinu=SIN(U)
Cosu=COS(U)
GOSUB Ncalc
IF Sinu>Cosu THEN Alt=Gn*Aa/Sinu-(1-E2)*N
IF Cosu>=Sinu THEN Alt=Rn*Aa/Cosu-N
Lat=U
DEFAULT ON
Lon=180-ATN(Fn/En)
DEFAULT OFF
GOSUB Coordprint
PAUSE
GOTO Gmd1
```

B. Efgentry subroutine: The Efgentry subroutine has been described for the previous off-spheroid coordinate determination programs. The same subroutine is used by the GMD program.

C. Coordprint subroutine: The coordprint subroutine is common to all off-spheroid coordinate determination programs and has been described previously.

Program Operation.

The GMD closed-form solution uses the same input/output formats as described for the other off-spheroid coordinate determination programs. The initial display is:

SELECT METHOD

0 = PURCELL AND COWAN
1 = LAGRANGE (CLOSED FORM)
2 = BOWRING
3 = GMD (CLOSED FORM)

The operator enters 3 and presses CONT. The program next displays:

SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER

After the output device has been selected, the Efgentry subroutine sequentially displays:

ENTER GEODETIC 'E' COORD IN (selected units)

ENTER GEODETIC 'F' COORD IN (selected units)

ENTER GEODETIC 'G' COORD IN (selected units)

ENTER GEOID SEP IN (selected units)

If the value of geoid separation is unknown, 0 should be entered and geoid and spheroid elevation will be the same values in the result.

The program then enters the computational algorithm and computes the values of geodetic latitude, longitude, altitude above spheroid, and altitude above geoid. The results are displayed as:

```
DATUM: NORTH AMERICAN (NAD)   EARTH MODEL:  CLARKE 1866

GEOCENTRIC 'E' COORDINATE =   -2459439.14 METERS
GEOCENTRIC 'F' COORDINATE =   -4625532.27 METERS
GEOCENTRIC 'G' COORDINATE =    3643414.76 METERS

GEODETIC LATITUDE  =  35 00 00.0000    ( 35.00000001)    (0.610865238)
GEODETIC LONGITUDE = 118 00 00.0002    (118.00000005)    (2.059488518)
SPHEROID ELEVATION =        10000.00 METERS
GEOID ELEVATION    =        10014.28 METERS
```

Again, the unbracketed angle term is the angle value in degrees, minutes, and seconds. The first bracketed angle term is the angle value in degrees, and the second bracketed angle term is the angle value in radians. In the example shown, meters are the selected input/output units.

Program Validation

Validation of the GMD closed-form solution is performed in the same manner as described for the other off-spheroid coordinate determination programs, and the same entry values are used to allow comparison of the computational accuracies of the four programs. Table 5.5 contains results from the GMD solution.

TABLE 5.5. GMD CLOSED-FORM SOLUTION VALIDATIONS

Parameter	Actual value	Computed value
For E = -2455593.45 m, F = -4618299.59 m, and G = 3637679.00 m		
Geodetic latitude	35 00 00.0000	35 00 00.0001
Longitude	118 00 00.0000	118 00 00.0000
Altitude	0.00 m	0.00 m
For E = -2455978.02 m, F = -4619022.86 m, and G = 3638252.58 m		
Geodetic latitude	35 00 00.0000	35 00 00.0002
Longitude	118 00 00.0000	118 00 00.0000
Altitude	1000.00 m	1000.00 m
For E = -2459439.14 m, F = -4625532.27 m, and G = 3643414.76 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	118 00 00.0002
Altitude	10000.00 m	10000.00 m
For E = -2494050.31 m, F = -4690626.42 m, and G = 3695036.64 m		
Geodetic latitude	35 00 00.0000	35 00 00.0000
Longitude	118 00 00.0000	118 00 00.0001
Altitude	100000.00 m	100000.00 m
For E = -2840162.04 m, F = -5341567.92 m, and G = 4211255.44 m		
Geodetic latitude	35 00 00.0000	35 00 00.0002
Longitude	118 00 00.0000	117 59 59.9999
Altitude	1000000.00 m	1000000.01 m
For E = -6301279.35 m, F = -11850982.85 m, and G = 9373443.36 m		
Geodetic latitude	35 00 00.0000	35 00 00.0001
Longitude	118 00 00.0000	117 59 59.9999
Altitude	10000000.00 m	10000000.00 m

CHAPTER 6

ATMOSPHERIC REFRACTION

This chapter describes the refraction correction method used in the GMD baseline system. The software accepts raw range and elevation data from any local tracking system and performs the high-accuracy refraction corrections necessary for the analysis of test data. In addition, this chapter includes discussions of the methods used to determine atmospheric refractivity from locally observed psychrometric data or weather data.

Refractivity Calculations

General Theory

To accurately correct for the effects of atmospheric refraction, it is necessary to develop a refractivity model that closely approximates real-world conditions. Several methods are available for computing a refractivity profile from psychrometric or weather data. Generally, the projection of the refractivity profile is based on computed values of station refractivity, Ns, and an exponential decay factor or scale height, Hs. While Ns can be computed with a reasonable degree of accuracy, the computation of Hs is less accurate and can degrade the elevation correction significantly. Basically the problem with Hs is due to three factors. First, the makeup of the atmosphere is dependent on area weather and climatic conditions that greatly influence the refractivity profile, especially at the lower elevations. Second, most models are based on an exponential decay pattern, which may vary significantly from the true refractivity profile, especially when air masses move from water to desert regions, when intense local surface heating is present, or when optical or r-f energy must travel through nonuniform localized weather conditions. Last, it is difficult to make actual measurements of atmospheric conditions at elevations above the tracking site prior to each mission since this generally requires the use of specialized balloon-carried (radiosonde) equipment. Because of these problems, it has generally been the practice to use one or more methods of scale height computation based in part on the surface measurements and in part on seasonal averages of psychrometric profiles. This is somewhat in the nature of a Farmer's Almanac approach to the correction of otherwise high-precision tracking data, and, regardless of how closely the modeling equations match the long-term statistical data, there is really no way to be sure that the corrections for any given mission are reasonably near the correct values.

In the GMD baseline program, a refractivity profile can be obtained from any of four separate methods: (1) radiosonde data entered manually into the program, (2) extrapolation of local and distant (e.g. Sentinel Peak) measurements, (3) standard regression equations based on surface measurements and statistical decay patterns, or (4) combinations of the first three methods. Both the second and third approaches assume that refractivity varies exponentially with altitude and that the complete refractivity profile at any given time can be obtained by determining the refractivity that exists at the tracking station or at other nearby observation points of known elevation, or both.

Calculation of Ns From Psychrometric Data

In the absence of a direct-measurement refractometer, it is necessary to obtain accurate psychrometric or weather data from which the modulus of refraction can be computed. Psychrometric parameters are obtained as dry-bulb temperature, wet-bulb temperature, and barometric pressure. Weather data are obtained as temperature, pressure, and dew point or relative humidity. Three combinations of these parameters may be used in the reduction program that computes the partial pressure of water vapor and dry air pressure, which, along with temperature, are necessary for the calculation of station refractivity. The program also computes saturation vapor pressure, relative humidity (if not given as an input parameter), and absolute humidity. While the latter parameters are not required for the refractivity computations, they are important for comparing derived data with data obtained from radiosonde or other direct measurement sources.

If weather data are used, care must be exercised during below freezing conditions to insure that the constants used in the reduction equations are consistent with the method initially used to calculate the relative humidity or dew point. Most scientific and technical organizations performing meteorological observations now compute relative humidity with respect to water rather than ice at temperatures below 0 degrees Celsius. This procedure has been adopted by both the Smithsonian Institute and the U.S. Weather Service because it offers the following advantages:

1. Most hygrometers indicate relative humidity with respect to water at all temperatures.

2. The majority of clouds at temperatures below 0 degrees Celsius consist mainly of water.

3. Relative humidities in excess of 100 percent will not generally be obtained with respect to water.

4. The majority of existing tables are based on saturation with respect to water.

However, there are still many psychrometric tables and certain measurement devices that provide relative humidity data obtained with respect to ice for below freezing conditions. If the relative humidity measurement has been obtained from one of these sources, then the corresponding constants must be used in the equations for computing the partial pressure of water vapor.

The saturation vapor pressure of water, e_s, is obtained from the relation

$$e_s = (T + d)^a 10^{[c + b/(T + d)]} \qquad (6.1)$$

where T is the dry-bulb (normal) temperature, and a, b, c, and d are constants whose values are given in table 6.1. Since dry-bulb temperature is required by each of the input options, saturation vapor pressure is computed by the same method in all three cases.

If psychrometric data (dry-bulb temperature, wet-bulb temperature, and pressure) are input to the program, then the remaining parameters are computed as shown in equations (6.2) to (6.4). First, the partial pressure of water vapor is found from the relation

$$e_v = (W + d)^a 10^{[c + b/(W + d)]} - (f + gW)(P)(T - W) \qquad (6.2)$$

where T is the dry-bulb temperature, W is the wet-bulb temperature, and a, b, c, d, f, and g are constants given in table 6.1.

The relative humidity is given by

$$U = e_v/e_s. \qquad (6.3)$$

The absolute humidity, H, is given by

$$H = \frac{k}{T + d} e_v. \qquad (6.4)$$

T and W are the same as previously described, and a, b, c, d, f, g, and k are constants whose values are provided in table 6.1.

If the second input option (temperature, pressure, and dew point) is selected, e_v is computed from

$$e_v = (D + d)^a 10^{[c + b/(D + d)]} \qquad (6.5)$$

where D is the temperature of the dew point. Relative humidity and absolute humidity are again computed from equations (6.3) and (6.4).

If the third option is selected, then relative humidity, temperature, and pressure are the input parameters, and e_v is computed from

$$e_v = U\, e_s. \qquad (6.6)$$

Absolute humidity is again obtained from equation (6.4).

Note that in keeping with current conventions for the calculation of relative humidity, the constants a, b, and c remain the same for both above freezing and below freezing conditions. However, as previously indicated, if the value of U was obtained from a source that made the calculation using over-ice tables, then for consistency the over-ice values for a, b, and c must be used for below freezing conditions.

Once the partial pressure of water vapor has been obtained, the modulus of refraction at the measurement station can be obtained from the familiar r-f refractivity equation,

$$N \times 10^6 = 77.6 \frac{P_d}{T + d} + 72.0 \frac{e_v}{T + d} + 3.75 \times 10^5 \frac{e_v}{(T + d)^2}. \tag{6.7}$$

Another commonly used variation of equation (6.7) is

$$N \times 10^6 = 77.6 \frac{P}{T + d} + 3.73 \times 10^5 \frac{e_v}{(T + d)^2}. \tag{6.8}$$

Since the constants are given in SI units, the values of P_d and e_v must be expressed in millibars and T must be expressed in degrees Celsius. The value for d is taken from table 6.1 for SI units. The remaining constants used in equations (6.7) and (6.8) are those of Smith and Weintraub. They are considered accurate to 0.5 percent for frequencies up to 30,000 Mhz. The dry air term, P_d, is obtained from the close approximation

$$P_d \cong P - e_v \tag{6.9}$$

where P is the total barometric pressure, P_d is the dry air pressure, and e_v is the partial pressure of water vapor.

Optical refractivity for light of wavelength λ is given by

$$N \times 10^6 = \frac{77.5\, P_t}{T} (1 + 5.15 \times 10^{-3}/\lambda^2 + 1.07 \times 10^{-4}/\lambda^2). \tag{6.10}$$

When using equation (6.10), λ is expressed in microns. Typical values are 0.555 micron for normal white light and 0.75 micron for ruby laser light.

Equations (6.1) to (6.6) are based on a general expression of the Claperon-Clausius differential equation that relates saturation vapor pressure, absolute temperature, and latent heat of transformation. The expressions for vapor pressure, absolute humidity, and relative humidity are derived in reference 8.

Theory relating to equations (6.7) to (6.9) is presented in detail in reference 9.

Equation (6.10) was taken from reference 10, which also contains considerable information regarding the various techniques for the calculation of atmospheric scale height.

TABLE 6.1. CONSTANTS

Constant	Value of constants for—			
	U over water		U over ice	
	SI units	U.S. units	SI units	U.S. units
a	-4.92830	-4.92830	-0.32286	-0.32286
b	-2937.40	15287.32	-2705.21	-4869.38
c	23.5518	32.2801	11.4816	10.0343
d	273.15	459.67	273.15	459.67
f	6.600×10^{-4}	3.595×10^{-4}	6.660×10^{-4}	3.595×10^{-4}
g	7.570×10^{-7}	2.336×10^{-7}	7.750×10^{-7}	2.336×10^{-7}
k	0.21668	0.82455	0.21668	0.82455

Variable Names

Name	Description
A	Constant a in table 6.1
B	Constant b in table 6.1
C	Constant c in table 6.1
D	Constant d in table 6.1
Dew	Dew point in degrees Celsius
Dp	Dew point in kelvins
Es	Saturation vapor pressure of water
Ev	Partial pressure of water vapor
F	Constant f in table 6.1

G	Constant g in table 6.1
Ha	Absolute humidity
K	Constant k in table 6.1
K1	First Smith and Weintraub constant
K2	Second Smith and Weintraub constant
K3	Third Smith and Weintraub constant
Nop(1)	Refractivity for white (yellow-green) light
Nop(2)	Refractivity for ruby laser red light
Nrf	R-f refractivity
Press	Total barometric pressure
Tempd	Dry air temperature in degrees Celsius
Td	Dry air temperature in kelvins
Tempw	Wet-bulb temperature in degrees Celsius
Tw	Wet-bulb temperature in kelvins
U	Relative humidity in percent

Computational Algorithms

The algorithms used for computing the partial pressure of water vapor and for r-f and optical refractivity are provided below.

A. Executive Psy: Psy is a subprogram of GEOD. Upon entry into Psy, Osel is called to permit the operator to select the appropriate output device. Subroutine Psycon assigns values to the constants in table 6.1, and Psymode allows the operator to select one of the three possible input combinations, the selection being denoted by flag L. The program then goes to Psyinput to permit the operator to input the necessary psychrometric or weather parameters. When the parameters have been entered, the program branches to Psy1, Psy2, or Psy3 depending on the value of L (1, 2, or 3). In these subroutines, all the psychrometric parameters needed for the refractivity algorithms are computed. The executive routine then calls Refcomp, which computes optical and r-f refractivity, and Psyprint, which prints or displays the results. The program pauses after the printing is completed. To enter additional psychrometric or weather data, the operator presses CONT and the program returns to Psystart and subroutine Psyinput.

1. Psy:!
2. GOSUB Psycon
3. GOSUB Psymode
4. Psystart:!
5. GOSUB Psyinput
6. ON L GOSUB Psy1,Psy2,Psy3
7. GOSUB Refcomp
8. GOSUB Psyprint
9. GOTO Psystart

B. Subroutine Psycon: Subroutine Psycon sets up values for the various constant terms used in the psychrometric and refractivity computations. The values of these constants can be found in the SI column for over-water conditions in table 6.1.

1. Psycon:!
2. A=-4.9283
3. B=-2937.4
4. C=23.5518
5. D=273.15
6. F=6.6E-4
7. G=7.57E-7
8. K=.21668
9. K1=77.6
10. K2=72
11. K3=3.75E5

C. Subroutine Psymode: Subroutine Psymode prompts the operator to select the input mode. If mode 1 is selected, the input parameters are wet-bulb temperature, dry-bulb temperature, and pressure. If mode 2 is selected, the input parameters are temperature, dew point, and pressure. If mode 3 is selected, the input parameters are temperature, relative humidity, and dew point. Psymode also allows the operator to make the appropriate selection of over-water or over-ice constants for below freezing conditions.

D. Subroutine A: Subroutine A replaces constants used for over-water conditions with those used for over-ice conditions when relative humidity is one of the input parameters, and when its value has been computed using over-ice constants. The same over-ice constants are also provided in table 6.1.

1. A:!
2. A=-.32286
3. B=-2705.21
4. C=11.4816
5. RETURN

E. Subroutine Psy1: Subroutine Psy1 computes the saturation vapor pressure of water, the partial pressure of water vapor, the absolute humidity, and the relative humidity using dry-bulb temperature, wet-bulb temperature, and atmospheric pressure. Step 4 is a direct implementation of equation (6.1), step 5 implements equation (6.2), step 6 implements equation (6.3), and step 7 implements equation (6.4).

```
1.  Psy1:!
2.  Td=Tempd+D
3.  Tw=Tempw+D
4.  Es=Td**A1*10**(C+B/Td)
5.  Ev=Tw**A*10**(C+B/Tw)-(F+G*Tempw)*Press*(Td-Tw)
6.  U=Ev/Es
7.  Ha=K*1E3*Ev/Td
8.  RETURN
```

F. Subroutine Psy2: Subroutine Psy2 computes the partial pressure of water vapor from dew point. Steps 4, 5, 6, and 7 are implementations of equations (6.1), (6.5), (6.3), and (6.4), respectively.

```
1.  Psy2:!
2.  Td=Tempd+D
3.  Dp=Dew+D
4.  Es=Td**A*10**(C+B/Td)
5.  Ev=Dp**A*10**(C+B/Dp)
6.  U=Es/Ev
7.  Ha=K*1E3*Ev/Td
8.  RETURN
```

G. Subroutine Psy3: Subroutine Psy3 computes the partial pressure of water vapor from relative humidity and temperature. Steps 3, 4, and 5 implement equations (6.1), (6.6), and (6.4), respectively.

```
1.  Psy3:!
2.  Td=Tempd+D
3.  Es=Td**A*10**(C+B/Td)
4.  Ev=U*Es
5.  Ha=K*1E3*Ev/Td
6.  RETURN
```

H. Subroutine Refcomp: Subroutine Refr computes the modulus of refraction for normal white (yellow-green) light, ruby laser light, and C-band r-f energy. The FOR-NEXT loop initiated at step 3 uses a wavelength of 0.555 micron (the frequency of yellow-green light) for the first pass and 0.75 micron (the frequency of ruby laser light) for the second pass. Step 7 implements equation (6.10) for the two optical passes. Step 10 implements equation (6.7) to obtain a refractivity value for the r-f energy.

```
1.  Refcomp:!
2.  Td=Tempd+D
3.  FOR N=1 TO 2
4.  IF N=1 THEN Lam=.555
5.  IF N=2 THEN Lam=.75
6.  Lam2=Lam*Lam
7.  Lam4=Lam2*Lam2
8.  Nop(N)=77.5*Press/(Td*1E6)*(1+5.1E-3/Lam2+1.07E-4/Lam4)
```

9. NEXT N
10. Nrf=(K1*Press-Ev)/Td+K2*Ev/Td+K3*Ev/Td**2/1E6
11. RETURN

I. Subroutine B: Subroutine B is used when U.S. Customary units have been selected. It converts the pressure and density terms from SI to U.S. Customary units for printout. In steps 2, 3, and 4, pressures in millibars are converted to inches of mercury by the conversion equation 1 mb = 0.029529988 in. Hg. Step 5 converts density in grams per cubic meter to density in pounds per cubic foot using the relation 1 lb/ft³ = 6.24279606 x 10⁵ g/m³.

1. B:!
2. Ev=Ev*.029529988
3. Ep=Ep*.029529988
4. Press=Press*.029529988
5. Ha=Ha*6.24279606E-5
6. RETURN

J. Subroutine C: Subroutine C is used when the input temperatures are in U.S. Customary units. The subroutine converts input temperature parameters from Fahrenheit to Celsius.

1. C:!
2. Temp=5/9*(Temp-32)
3. RETURN

K. Subroutine D: Subroutine D is used when pressures are entered in inches of mercury. The conversion equation is 1 mb = 0.029529988 in. Hg.

1. D:!
2. Press=Press/.029529988
3. RETURN

Program Operation

The psychrometric and refractivity computations are selected from the same menu as described for other subprograms of GEOD. When the main menu is displayed, the operator selects RF AND OPTICAL REFRACTIVITY, and the program immediately requests the output device selection as described in previous programs.

Next, the program sequentially asks the operator to make the following selections.

SELECT UNITS

0 = SI
1 = US

SELECT INPUT

0 = DRY TEMP, WET TEMP, AND BARO PRESSURE
1 = DRY TEMP, DEW POINT, AND BARO PRESSURE
2 = DRY TEMP, REL HUMID, AND BARO PRESSURE

The program then sequentially requests operator inputs of the three parameters selected. Entries made by the operator are in the units selected (SI or U.S. Customary).

If 0 is selected, the results are printed in the selected units as:

TEMPERATURE (DRY)	=	75.0000 DEG FAHRENHEIT
TEMPERATURE (WET)	=	54.0000 DEG FAHRENHEIT
BAROMETRIC PRESSURE	=	29.9200 IN HG
SATURATION VAPOR PRESSURE OF WATER	=	29.9291 IN HG
PARTIAL PRESSURE OF WATER VAPOR	=	0.2502 IN HG
RELATIVE HUMIDITY	=	40.8601 PERCENT
ABSOLUTE HUMIDITY	=	0.0984910 LBM/CU FT
R-F REFRACTIVITY	=	0.0003254
OPTICAL REFRACT (WHITE LIGHT)	=	0.0002691
OPTICAL REFRACT (RUBY LASER)	=	0.0002669

If 1 is selected, the results are printed as:

TEMPERATURE (DRY)	=	75.0000 DEG FAHRENHEIT
DEW POINT TEMPERATURE	=	42.0000 DEG FAHRENHEIT
BAROMETRIC PRESSURE	=	29.9200 IN HG
SATURATION VAPOR PRESSURE OF WATER	=	29.9291 IN HG
PARTIAL PRESSURE OF WATER VAPOR	=	0.2705 IN HG
RELATIVE HUMIDITY	=	40.8601 PERCENT
ABSOLUTE HUMIDITY	=	0.0514954 LBM/CU FT
R-F REFRACTIVITY	=	0.0003254
OPTICAL REFRACT (WHITE LIGHT)	=	0.0002565
OPTICAL REFRACT (RUBY LASER)	=	0.0002532

If 2 is selected, the results are printed as:

TEMPERATURE (DRY)	=	75.0000 DEG FAHRENHEIT
BAROMETRIC PRESSURE	=	29.9200 IN HG
SATURATION VAPOR PRESSURE OF WATER	=	29.9291 IN HG
PARTIAL PRESSURE OF WATER VAPOR	=	0.3535 IN HG
RELATIVE HUMIDITY	=	40.0000 PERCENT
ABSOLUTE HUMIDITY	=	0.0005452 LBM/CU FT
R-F REFRACTIVITY	=	0.0003153
OPTICAL REFRACT (WHITE LIGHT)	=	0.0002691
OPTICAL REFRACT (RUBY LASER)	=	0.0002669

After the results have been printed on the output device, pressing CONT will return the program to the entry point for the next set of psychrometric or weather parameters. Note that if SI units were selected during the initiation, temperatures would have been in degrees Celsius, pressures in millibars, and density in grams per cubic meter.

Program Validation

Partial pressure calculations are validated against values from the Smithsonian tables. Refractivity values are validated against tables in reference 9. Sample validation values are provided in table 6.2.

TABLE 6.2. SAMPLE REFRACTIVITY CALCULATIONS

Rel hum, %	Barometric pressure, mbar (in. Hg)	Ambient temperature, deg C (deg F)	Computed e_s, mbar (in. Hg)	Computed e, mbar (in. Hg)	Computed $N \times 10^6$
68	760.0 (22.443)	34.0 (93.2)	53.255 (1.573)	36.213 (1.069)	335.3
80	850.0 (25.100)	42.0 (107.6)	82.091 (2.424)	65.673 (1.939)	456.1
39	800.0 (23.624)	29.0 (84.2)	40.095 (1.184)	15.637 (0.462)	269.4
67	1013.2 (29.920)	23.1 (75.0)	29.665 (0.876)	19.878 (0.587)	348.8
74	1017.6 (30.050)	8.9 (48.0)	11.412 (0.337)	8.432 (0.249)	319.6
50	1014.6 (29.960)	3.9 (39.0)	8.060 (0.238)	4.030 (0.119)	303.8
36	750.0 (22.147)	-1.0 (30.2)	5.679 (0.168)	2.044 (0.060)	224.2
47	700.0 (20.671)	-20.0 (-4.0)	1.254 (0.037)	0.589 (0.017)	218.0

The Baseline Refraction Correction Program

The baseline refraction algorithms were developed by GMD Systems to provide a refraction correction method that could be operated on a desktop computer of moderate word length (12 decimal digits) and still provide high accuracy results, especially at low or negative elevation angles. The program does not use the conventional Snell's Law approach since that technique either fails or becomes extremely inaccurate at the low elevation angles where the majority of the aerodynamic tracking operations are performed. Gradient refraction is an iterative technique that sequentially projects the wavefront along the propagation path for intervals whose exact lengths are determined by an optimizing algorithm within the program. At each iterative step, the velocity gradient perpendicular to the wave travel is calculated and used to compute the bending angle for that particular segment. The next sequential increment is then propagated normal to the adjusted plane of the wavefront.

Because of the large number of iterative steps required at low elevation angles, the length of the propagation path segments must be optimized so that both roundoff and truncation errors are minimized. If the segments are too long, the truncation errors will be excessive. If the segments are too short, the roundoff errors will be excessive. For a given range and angle condition, the optimizing algorithm should be designed to select an iteration interval that minimizes the total error.

The gradient refraction algorithms can operate at any elevation angle from -90 degrees to +90 degrees. The program has no singularity points and provides valid results at all angles. In addition, the mathematical algorithms have been designed to provide greater accuracy in the angle calculations through the measurement and accumulation of small angles rather than large angles. In the Snell's law approach, the angle of incidence is measured from the vertical at the point where the ray passes from one shell into another. This can lead to inaccuracies as demonstrated by the following example. At a typical tracking angle of 0.5 degree, the angle of incidence is large, 89.5 degrees. On a typical twelve-digit computer, the sine of 89.5 degrees is computed to be 0.999961923080. However, the arcsine of the same number is returned as 89.500000104300 degrees. This amounts to a combined error of 0.0000001043 degree in the single sine and arcsine calculation. On the other hand, if one were to take the sine of 1.23456789E-51 degree (50 zeros) on the same computer, the result would be 2.15472745200E-53. However, in this case, the arcsine is returned as 1.23456789E-51 degree, precisely the same as the starting value. In ray tracing solutions, where Snell's Law may be reapplied as many as 500 to 50,000 times for a single solution, the sine error can become excessive, even at moderately high elevation angles. In the gradient solution, the angle calculations have been designed (1) to use trigonometric functions in their most accurate regions, and (2) to prevent register saturation by the use of small, rather than large, angles. Because of this, gradient refraction is capable of delivering computational accuracies that are several orders of magnitude better than those possible when using the conventional Snell's Law approach. This also permits the use of fast small angle approximations when additional computational speed is required.

Gradient Refraction Solution

General Theory

Two principal angles are computed in the gradient refraction solution: the wave-front bending angle, and the earth interior angle subtended by the arc from the tracking antenna to the target. While the incremental values of these angles are extremely small (typically 0.0004 degree for the incremental bending angle and 0.0008 degree for the incremental interior angle), small angle approximations were found to cause a small but measurable difference in both range and angle values due to the long radius involved. Thus, for the highest accuracy solutions, the sine and tangent values are computed rather than approximated. For faster but less accurate solutions, small angle approximations are used. The geometry used in the gradient refraction solution is shown in figures 6.1(a) to 6.1(c).

Figure 6.1(a) shows the first segment of a beam transmitted from a tracker at P_0. The beam is defined by an upper ray, a central ray, and a lower ray. The refractivity at the midpoint of the central ray is given by N_1, the refractivity at the midpoint of the upper ray is given by Nu_1, and the refractivity at the midpoint of the lower ray is given by Nl_1. The rays are shown with a separation of 0.5 meter, and R_0 is the geocentric radius vector of the tracking site represented by the point P_0. The scale height (altitude at which the modulus of refractivity will have decayed to 1/e of its sea-level value) is given by hs. The height of the tracking site above sea level is given by h_0.

The distance traveled by the central ray in time Δt is approximately

$$D_1 = \frac{c \times \Delta t}{1 + N_1}. \tag{6.11}$$

The distance traveled by the upper ray in time Δt is approximately

$$Du_1 = \frac{c \times \Delta t}{1 + Nu_1}. \tag{6.12}$$

The distance traveled by the lower ray in time Δt is approximately

$$Dl_1 = \frac{c \times \Delta t}{1 + Nl_1}. \tag{6.13}$$

The altitude of the midpoint of the central ray above its starting point is

$$Dh_1 = 0.5\, D_1 \sin E_1. \tag{6.14}$$

Therefore, the refractivity at the midpoint of the central ray is given by

$$N_1 = N_0 e^{(-h_1/hs)}, \tag{6.15}$$

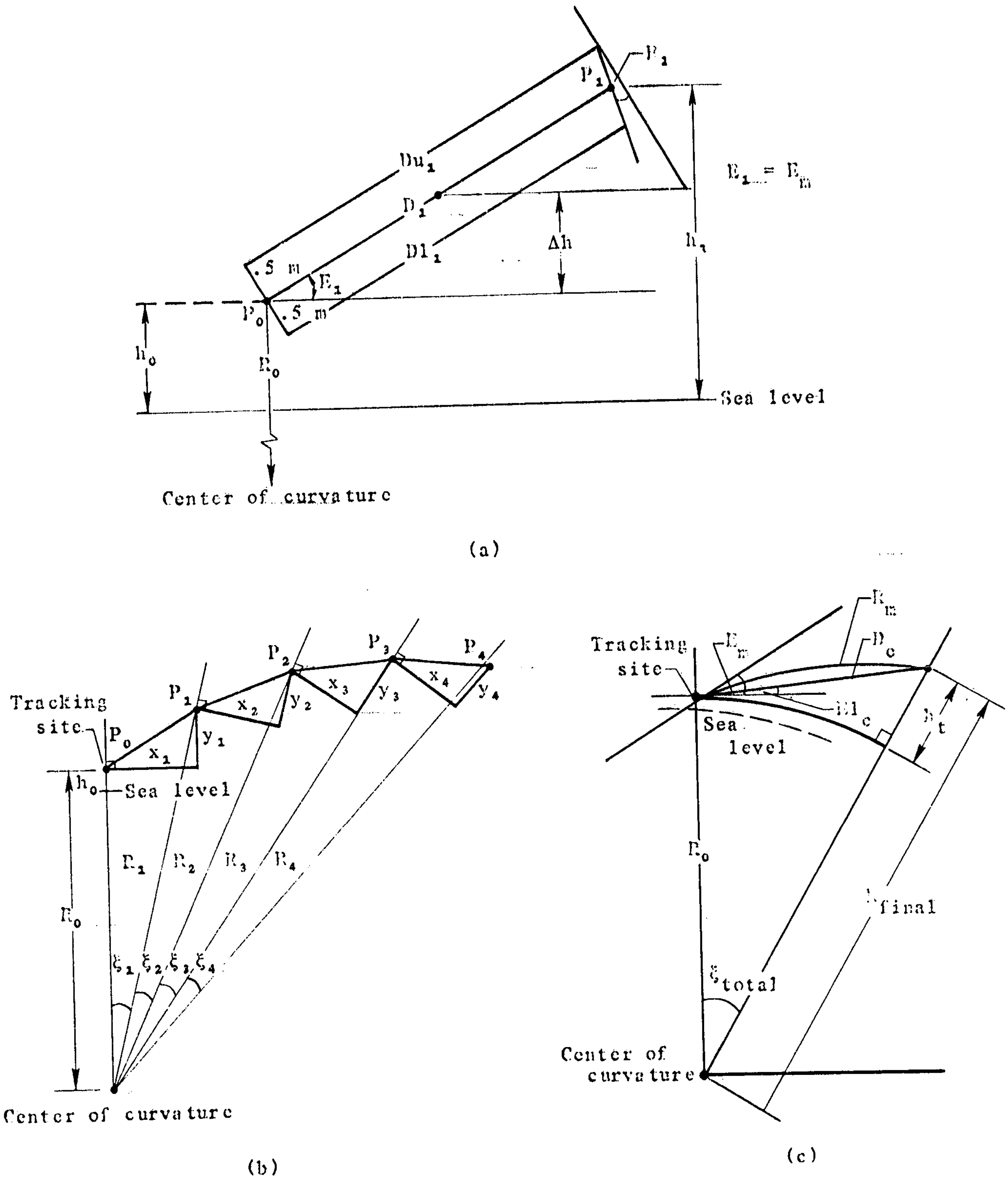

Figure 6.1.

and the refractivity gradient at the same point is given by

$$\frac{dn}{dh} = -N_0 \frac{h}{hs} e^{(-h_1/hs)}. \qquad (6.16)$$

Since the ray is traveling at an angle E_1 with respect to the local horizontal plane, the refraction gradient across the ray axis is given by

$$dN_1 = 0.5 \times dN \times \cos E_1. \qquad (6.17)$$

The wavefront bending angle, shown as B_1, is computed as

$$B_1 = \arcsin (Du_1 - Dl_1), \qquad (6.18)$$

and the X-Y elements of the line P_0P_1 are determined as

$$X_1 = D \sin E_1 \quad \text{and} \quad Y_1 = D \cos E_1. \qquad (6.19)$$

The earth interior angle subtended by the ray P_0P_1 is given by

$$\xi_1 = \arctan \frac{X}{R_0 + Y_1}, \qquad (6.20)$$

and the altitude of the point P_1 above the tracking site is given by

$$R_1 = \frac{X_1}{\sin \xi_1} \text{ (for E greater than or equal to 0.79 rad)} \qquad (6.21)$$

or

$$R_1 = \frac{Y_1 + R_1}{\cos \xi_1} \text{ (for E less than 0.79 rad)} \qquad (6.22)$$

Each iteration follows the same techniques as given above to find the bending angle B_i, the earth interior angle ξ_i, and the length of the new radius vector R_i. Each succeeding elevation angle E_{i+1} is found by subtracting each bending angle B_i and adding each earth interior angle ξ_i to the current elevation angle E_i. This causes the elevation angle E_i to be referenced to the local vertical at the appropriate segment starting point P_{i-1}. The values of earth interior angles ξ_i are accumulated as ξ_{total} until the full measured range Dm is depleted by subtracting the free-space value of D for the selected incremental time Δt ($D = c \times \Delta t$) from the measured distance remaining at the start of each succeeding iteration. When the measured distance reaches zero, the program interpolates to find the final values of ξ and R. The successive iteration steps are diagrammed in figure 6.1(b).

Values for corrected elevation (Elc), corrected distance (Dc), and target altitude (h_t) are then determined by finding the Cartesian elements of the target point with reference to the starting point by the simple geometry shown in figure 6.1(c). Since values for R_0, R_{final}, and ξ_{total} are now known, the Cartesian elements X and Y can be found from

$$X = R_{final} \times \sin \xi_{total} \tag{6.23}$$

and

$$Y = (R_{final} \times \cos \xi_{total}) - R_0 . \tag{6.24}$$

Noting the geometric relations shown in figure 6.1(c), it is now possible to compute Elc, Dc, and h_t from the following relations.

$$Elc = atn\ (Y/X) \tag{6.25}$$

$$Dc = (X^2 + Y^2)^{1/2} \tag{6.26}$$

$$h_t = R_{final} - R_0 + h_0 \tag{6.27}$$

Variable Names

Name	Description
Aa	Semimajor axis of reference spheroid
Bb	Semiminor axis of reference spheroid
Bi	Bending angle for ith segment
Bit	Total accumulated bending angle
C	Velocity of light (2.997924562E8) in meters per second
Cosei	Cos Ei
D	Distance traveled by ray in free space during time T
Da	Accumulated arc distance traveled by ray
Dc	Corrected target range
Dh	Approximate length of vertical component of ray segment
Dist	Refraction-corrected distance traveled by central leg of ray segment

D1	Distance traveled by lower leg of ray in time T
Dm	Measured range
Dn	Refractivity gradient at the midpoint of the ray referenced to the local vertical at the origin of the ray
Dni	Refractivity gradient at the midpoint of the ray computed perpendicular to the direction of the ray propagation
Dtot	Accumulated equivalent uncorrected (free-space) distance traveled by ray
Du	Distance traveled by upper leg of ray in time T
E2	Eccentricity squared
Ee	Eccentricity of reference spheroid
Ei	Instantaneous elevation angle referenced to local vertical at location of wavefront
Elc	Corrected target elevation angle
Em	Measured elevation
F	Factor to interpolate length of final ray segment when measured distance has been depleted
H	Spheroid elevation of tracking site
Ha1	Height of lower measurement station when using refractivity values from two stations to determine scale height
Ha2	Height of higher measurement station when using refractivity values from two stations to determine scale height
Hi	Average instantaneous height of the ray segment above the starting elevation
Hs	Atmospheric scale height
Ni	Average refractivity index over the center leg of the ray segment
N1	Average index of refraction over the upper leg of the ray segment
No	Station refractivity
No1	Refractivity at lower station when values from two stations are used to compute scale height

No2	Refractivity at higher station when values from two stations are used to compute scale height
Ns1	Sea-level refractivity
Nu	Average index of refraction over the upper leg of the ray segment
Num	Index showing number of iteration cycles completed
P2	2π
Pi	Earth interior angle subtended by the ith ray
Pit	Total accumulated earth interior angle
Ri	Instantaneous earth-centered ray height
Scaleh1	Manually entered value of scale height
Sflag	Flag indicating which Δt selection was made
Sinei	sin Ei
T	Incremental propagation time used in computations
To	Incremental propagation time optimized to reduce roundoff and truncation errors for various measured ranges
X	Horizontal component of ray segment referenced to the local vertical of the wavefront
Y	Vertical component of ray segment referenced to the local vertical of the wavefront

Computational Algorithms

The essential algorithms used in the gradient refraction program are presented below.

A. Gdref executive subroutine: Gdref controls the various calls needed for initialization, data entry, computation, and display.

1. GOSUB Gmode
2. GOSUB Ginit
3. Gdentry:!
4. GOSUB Entry
5. GOSUB Compute
6. GOSUB Elc
7. GOSUB Print
8. GOTO Gdentry

B. Subroutine Gmode: Subroutine Gmode requests operator selection of operating modes, tracking source, and iteration intervals. It produces the various menu statements that are fully described in the Program Operation section. These are straightforward INPUT statements and are omitted here to avoid unnecessary duplication. The subroutine also provides average earth radius values (Ro) for the various tracking site selections.

C. Subroutine Ginit: Subroutine Ginit initializes elevation, range, and earth radius parameters and sets up the selected Δt interval to be used in the iterations. At step 1 the radian mode is set. At step 2, the constant C is set equal to the free-space speed of light. In steps 3 through 6, starting values for the succeeding iterations are assigned the input values of H (tracking site elevation above sea level), Em (elevation measured in degrees), and Dm (distance measured in meters). Ro is the average earth radius at the tracking site. In step 8 the variables used to accumulate incremental values through the integration process are initialized to 0. In step 9, the time of travel (T) for each ray is initialized to TO, an optimum value selected to reduce the combined effects of roundoff and truncation error. If a faster mode has been selected, as indicated by the speed flag (Sflag), then T may be assigned values of 2 To, 5 To, 10 To, 25 To, or 100 To, which will decrease the computation times by factors of 2, 5, 10, 25, or 100, respectively. This increased speed increases the amount of truncation error (error arising from the fact that the curved ray path is being approximated by longer straight-line rays) and provides a less accurate solution.

```
 1.  RAD
 2.  C=2.997924562E8
 3.  Hi=H
 4.  Ei=Em*2*PI/360
 5.  Di=Dm
 6.  Ri=Ro
 7.  F=1
 8.  Ra=Ea=Da=Bit=Pit=Num=Dtot=Xtot=Ytot=Dh=0
 9.  T=To
10.  IF Sflag=2 THEN T=To*2
11.  IF Sflag=3 THEN T=To*5
12.  IF Sflag=4 THEN T=To*10
13.  IF Sflag=5 THEN T=To*25
14.  IF Sflag=6 THEN T=To*100
15.  ON Hsflg GOTO Hs1,Hs2,Hs3
16.  RETURN
```

D. Subroutine Hs1: Subroutine Hs1 computes the scale height (reciprocal of the exponential decay factor) for use in the refractivity equation. The algorithm uses a 10-step iterative process during which an assumed initial scale height of 6600 meters (based on the year-round average) is refined by use of the current station refractivity measurement. A value of sea-level refractivity is also computed. Steps 4 and 5 are an implementation of the HS1 equations given on page 4 of reference 10. This is the same method employed by Goddard Spaceflight Center to compute Hs values that are tabulated for use by Johnson Space Center.

```
1. Hs1:!
2. Hs=6600
3. FOR N=1 TO 10
4. Ns1=No*EXP(H/Hs)
5. Hs=1000/LOG(Ns1/Ns1-7.32E-6*EXP(5577*Ns1))
6. NEXT N
7. RETURN
```

E. Subroutine Hs2: Subroutine Hs2 assigns an operator-selectable value of scale height (Scaleh1) to Hs and computes sea-level refractivity based on that value. This solution assumes that scale height is known from radiosonde measurements or other means.

```
1. Hs2:!
2. Hs=Scaleh1
3. Ns1=No*EXP(H/Hs)
4. RETURN
```

F. Subroutine Hs3: Subroutine Hs3 computes the values of scale height and sea-level refractivity from refractivity measurements made at two known elevation points.

```
1. Hs3:!
2. Hs=(Ha2-Ha1)/LOG(No1/No2)
3. Ns1=No1*EXP(Ha1/Hs)
4. RETURN
```

G. Subroutine Cycle: Subroutine Cycle performs the iterative solution that computes the final values of bending angle, internal earth angle, and ray height. The variable Num counts the number of integration steps that have been performed. Ei is the instantaneous elevation angle of the ray as measured from the local horizontal at the start of the interval. At step 4, the free-space distance which would be traveled by the ray in time T is computed. C is the velocity of light. At step 5, the approximate increase in altitude to the midpoint of the ray is computed. In step 6, the altitude of the midpoint of the ray is added to the starting altitude value, and the sum is used to compute the average refractivity (eq. (6.15)) for the specific ray segment. In step 7, the value of dn/dh (equation (6.16)) is computed by combining equations (6.15) and (6.16). Step 8 implements equation (6.17), and steps 9 and 10 compute the values of the average modulus of refraction for the upper and lower ray segments (fig. 6.1). At step 11, the measured distance remaining is reduced by the equivalent free-space distance traveled by the ray segment. If some measured distance remains, the program branches to step 15. Otherwise, the program interpolates (step 13) to obtain a factor which must be applied to the last segment calculations to proportion the actual distance traveled by the last ray segment. This is accomplished in step 14 where the equivalent free-space distance is multiplied by the computed factor. In step 15, the actual distance traveled by the central ray is computed, and in steps 16 and 17 this value is adjusted to yield the distances traveled by the upper and lower segments of the ray (fig. 6.1). Da, the actual accumulated distance traveled by the central ray, is increased by the incremental travel of the central ray during this segment frame. In step 19, the bending angle for the current segment is computed and added to the total accumulated bending angle (step 20). Note that since the separation between the upper and lower segments is taken as being 1 meter, the actual bending angle relation (tan Bi = opposite/adjacent) has a denominator of 1, and the fact that Bi is a very small angle allows the substitution of Bi for tan Bi, which simply yields Bi = Du - Dl. The X and Y components of the ith ray segment are computed in steps 21 and 22, and these values are used in step 23 to calculate the central angle (Pi) subtended by the ith segment. In step 24 the incremental angular contribution of the ith segment is added to the total accumulated earth interior angle Pit. A new value for Ri is then computed from the simple trigonometry shown in figure 6.1, which is implemented in steps 26 and 27. Two methods of computing Ri are provided, and the method that provides the greatest accuracy for the specific Ei value is selected by the program logic. Hi, the altitude of the ray above the tracking site, is updated in step 29, and the total consumed free-space (or measured) distance is updated in step 30. If the measured distance remaining (Di) is greater than 0, the program recyles to step 1. The same process is repeated until the measured distance is consumed, at which point the program returns to the executive routine.

```
1.  Num=Num+1
2.  Cosei=COS(Ei)
3.  Sinei=SIN(Ei)
4.  D=C*T
5.  Dh=.5*D*Sinei
6.  Ni=Ns1*EXP(-(Hi+Dh)/Hs)
7.  Dn=-1/Hs*Ni
8.  Dni=.5*Dn*Cosei
9.  Nu=1+Ni+Dni
10. Nl=1+Ni-Dni
11. Di=Di-D
12. IF Di>0 THEN 15
13. F=(D+Di)/D
14. D=F*D
15. Dist=D/(1+Ni)
16. Du=D/Nu
17. Dl=D/Nl
18. Da=Dist+Da
19. Bi=Du-Dl
20. Bit=Bit+Bi
21. Y=Dist*Sinei
22. X=Dist*Cosei
23. Pi=ATN(X/(Ri+Y))
24. Pit=Pit+Pi
25. Ei=Ei-Bi+Pi
26. IF Ei>=.79 THEN Ri=X/SIN(Pi)
27. IF Ei<.79 THEN Ri=(Y+Ri)/COS(Pi)
28. Hi=Ri-Ro
```

```
29. Dtot=D+Dtot
30. IF Di>0 THEN 1
31. RETURN
```

H. Subroutine Elc: Subroutine Elc computes corrected elevation and corrected range of the target. Step 1 sets the degree mode. Step 2 computes the radian to degree conversion factor (Dcon). Steps 3 to 5 convert the total interior angle, total bending angle, and final elevation angle to degrees. Dx and Dy, the total X and Y components of the target position as shown in figure 6.1, are computed in steps 6 and 7. A special condition for a 90-degree corrected elevation angle is provided in steps 8 and 9. For all other cases, the corrected elevation angle and corrected distance are computed from the simple trigonometric relationships shown in steps 12 and 14. Default conditions are set for these calculations to prevent real precision overflows at elevation angles approaching 90 degrees. The program then returns to the main executive subroutine.

```
1. DEG
2. Dcon=180/PI
3. Pit=Pit*Dcon
4. Bit=Bit*Dcon
5. Ei=Ei*Dcon
6. Dx=Ri*SIN(Pit)
7. Dy=Ri*COS(Pit)-Ro
8. IF Dx<>0 THEN 11
9. Elc=90
10. GOTO 14
11. DEFAULT ON
12. Elc=ATN(Dy/Dx)
13. DEFAULT OFF
14. Dc=SQR(Dx**2+Dy**2)
15. RETURN
```

I. Subroutine Print: Subroutine Print causes the various input and output parameters to be displayed or printed. The format of the display is given in the Program Operation section, and is therefore omitted here to avoid duplication.

Program Operation

The gradient refraction program is a subprogram of GEOD. When GEOD is run, the operator is asked to select the units and datum/spheroid reference applicable to the computations to be performed. After these selections are made, the master menu selection is displayed. One of the menu items is GRADIENT REFRACTION. The operator makes the appropriate numerical entry and the main program calls subprogram Gdref.

Upon entry into Gdref, the operator is prompted to make several simple selections.

A. Output device selection

SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER

B. Tracking site selection

SELECT TRACKING SITE

1 = JSC MODEL
2 = FPS-16 (34)
3 = FPS-16 (38)
4 = FPS-16 (41)
5 = ELY HAIR SYSTEM
6 = MANUAL ENTRY

At this point the operator will select a prestored site or manual entry. If a manual entry is used, the program will sequentially ask for site latitude, site longitude, and site spheroid elevation. The latitude parameter is used to compute an average earth radius for the locale of the tracking site. This earth radius is then increased by the amount of the site's spheroid elevation, and the combined radius becomes the reference value (Ro) used in the refraction equations.

C. Entry of refractivity and scale height values

The operator is next asked to select between entry of a single station refractivity value or two values from sites at different elevations. This option was added to the program to permit refractivity taken at any one of the local radar sites to be used in conjunction with refractivity values obtained by telemetry from the weather station at Sentinel Peak (or at other similar future sites). The operator is prompted as follows:

MAKE SELECTION

0 = ENTRY OF SINGLE REFRACTIVITY VALUE
1 = ENTRY OF TWO REFRACTIVITY VALUES (COMPUTE SCALE HEIGHT)

If 0 is selected, the operator is asked to enter the value of station refractivity to be used. If 1 is selected, the operator is sequentially prompted as follows:

ENTER VALUE OF LOWER ELEVATION STATION REFRACTIVITY

ENTER LOWER STATION ELEVATION IN METERS

ENTER VALUE OF HIGHER ELEVATION STATION REFRACTIVITY

ENTER HIGHER STATION ELEVATION IN METERS

The program uses the two values of refractivity and the two station elevations to calculate the scale height to be used in the computations to follow. The program then proceeds to step D below.

Had 0 been selected, the operator would have been required to sequentially respond to the following prompt messages:

ENTER VALUE OF STATION REFRACTIVITY

IF MEASUREMENT STATION IS SAME AS TRACKING SITE PRESS CONT
IF MEASUREMENT STATION IS DIFFERENT ENTER ELEVATION IN (METERS)

The parentheses above indicate that the length units requested will be those selected by the operator at program initialization.

The following selection would then be required:

MAKE SELECTION

0 = COMPUTE SCALE HEIGHT FROM REGRESSION EQ
1 = MANUAL ENTRY OF SCALE HEIGHT TO BE USED
2 = PRECOMPUTED (STORED) VALUE OF SCALE HEIGHT TO BE USED

D. Incremental mode selection

At this point the operator is asked to select from several incremental modes that are available for computations.

MAKE SELECTION

0 = HIGH ACCURACY MODE (EXTREME ACCURACY BY LOW SPEED)
1 = VARIABLE INTERVAL (HI ACC AT LOW ANGLES BUT FASTER AT HI ANGLES)
2 = MODERATE SPEED SOLUTION (2 TIMES FASTER THAN MODE 0)
3 = HIGH SPEED SOLUTION (5 TIMES FASTER THAN MODE 0)
4 = VERY HIGH SPEED SOLUTION (10 TIMES FASTER THAN MODE 0)
5 = EXTREME SPEED SOLUTION (25 TIMES FASTER THAN MODE 0)
6 = ULTRA SPEED SOLUTION (100 TIMES FASTER THAN MODE 0)

Only mode 0 retains full accuracy. All other solutions sacrifice varying degrees of accuracy for speed. Generally, for baseline work, mode 0 would be selected. For faster response or real-time applications, one of the other modes would be used, with the specific selection depending on the speed of the processor and the time available within the real-time program.

E. Entry of measured values

At this point, the program is ready to accept keyboard-entered measurement values. The following prompt messages appear sequentially on the CRT. The operator responds to each by entering the appropriate values.

ENTER VALUE OF E MEASURED IN DEGREES

ENTER VALUE OF R MEASURED IN (METERS)

The parentheses indicate that the range data will be requested in the currently selected units. However, once entered, the input units are converted to radians and meters for use in the iteration sequence.

F. Output

When the iterations have been completed, the values of bending angle, central angle, and the magnitude of the final radius vector are used to compute corrected elevation, corrected range, and target altitude above sea level. These are displayed as:

SL REFRACTIVITY	.0002800
MEAS REFRACTIVITY	.0002513
SITE REFRACTIVITY	.0002513
SCALE HEIGHT	7515.0531 METERS
ELEV MEASURED	.3160 DEGREES
RANGE MEASURED	146400.93 METERS
CORRECTED EL	.1703 DEGREES
CORRECTED RNG	146364.04 METERS
TARGET ALTITUDE	2934 METERS
NO ITERATIONS	186
CENTRAL ANGLE	1.3211 DEGREES
BENDING ANGLE	0.2800 DEGREES
IMPINGING ANGLE	1.3570 DEGREES

After the results have been printed on the output device, pressing CONT will return the program to the entry point so that the next set of refraction data can be entered.

Program Validation

To validate the computational accuracy of the gradient refraction solution, results were compared with those from the JSC ray-tracing program described in reference 9. It should be noted that the JSC results were obtained on a CDC Cyber 74 computer using double precision (28 decimal digits). In those computations, 50,000 iterations were used for measured elevation angles of less than 0.5 degree, 5000 iterations were used for measured elevation angles from 0.5 to 2.5 degrees, and 500 iterations were used for higher elevation angles. The solution was based on a Snell's Law approach.

In general, the gradient refraction solutions agree with the JSC results to within approximatley one or two ten-thousandths of a degree in angle and one or two hundredths of a meter in range. Actual comparison values for elevation angles of 0, 20, 45, and 90 degrees, and ranges from 10 to 15,000,000 meters are presented in tables 6.3 to 6.6.

TABLE 6.3. COMPARISON OF JSC AND GMD RANGE AND ELEVATION VALUES FOR ELEVATION ANGLE OF 0 DEGREES

(No = 0.000386, Ro = 6378166 meters)

R measured, m	JSC Elc, deg	GMD Elc, deg	JSC Dc, m	GMD Dc, m
15,048.27	-0.0295	-0.0293	15,042.46	15,042.47
67,152.73	-0.1308	-0.1309	67,127.06	67,127.06
105,826.84	-0.2043	-0.2042	105,786.97	105,786.96
148,875.73	-0.2834	-0.2833	148,820.92	148,820.92
208,511.69	-0.3864	-0.3864	208,438.28	208,438.28
321,912.52	-0.5549	-0.5549	321,812.18	321,812.18
442,467.95	-0.6906	-0.6904	442,351.26	442,351.27
603,994.40	-0.8117	-0.8118	603,868.82	603,868.83
910,793.99	-0.9322	-0.9322	910,663.72	910,663.73
1,250,390.50	-0.9970	-0.9970	1,250,258.08	1,250,258.09
1,730,424.76	-1.0452	-1.0452	1,730,290.74	1,730,290.75
2,698,884.88	-1.0903	-1.0903	2,698,749.38	2,698,749.37
3,835,186.08	-1.1142	-1.1142	3,835,049.78	3,835,049.77
15,214,630.95*	-1.1566	-1.1566	15,214,493.25	15,214,493.06*

* Total JSC iterations - 50,000. Total GMD iterations - 3028

TABLE 6.4. COMPARISON OF JSC AND GMD RANGE AND ELEVATION VALUES FOR ELEVATION ANGLE OF 20 DEGREES

(No = 0.000386, Ro = 6378166 meters)

R measured, m	JSC Elc, deg	GMD Elc, deg	JSC Dc, m	GMD Dc, m
29.25	19.9999	20.0000	29.24	29.24
584.94	19.9989	19.9992	584.72	584.72
1,462.19	19.9974	19.9976	1,461.65	1,461.65
2,923.82	19.9949	19.9953	2,922.79	2,922.79
5,845.29	19.9904	19.9907	5,843.39	5,843.39
14,594.01	19.9795	19.9798	14,590.28	14,590.28
29,116.97	19.9678	19.9679	29,111.71	29,111.71
57,926.86	19.9561	19.9562	57,920.73	57,920.71
142,482.17	19.9465	19.9466	142,475.86	142,475.86
277,753.94	19.9432	19.9433	277,747.63	277,747.63
530,838.77	19.9415	19.9417	530,832.46	530,832.47
1,195,203.99	19.9405	19.9407	1,195,197.68	1,195,197.69
2,124,555.53	19.9401	19.9403	2,124,549.23	2,124,549.23
13,066,065.17	19.9397	19.9399	13,066,058.87	13,066,058.87

TABLE 6.5. COMPARISON OF JSC AND GMD RANGE AND ELEVATION VALUES FOR ELEVATION ANGLE OF 45 DEGREES

(No = 0.000386, Ro = 6378166 meters)

R measured, m	JSC Elc, deg	GMD Elc, deg	JSC Dc, m	GMD Dc, m
14.15	45.0000	44.9999	14.14	14.14
282.95	44.9996	44.9998	282.84	282.84
707.35	44.9990	44.9993	707.09	707.09
1,414.65	44.9981	44.9984	1,414.15	1,414.15
2,829.08	44.9965	44.9967	2,828.16	2,828.16
7,071.03	44.9925	44.9927	7,069.22	7,069.22
14,136.52	44.9882	44.9884	14,133.98	14,133.99
28,251.04	44.9840	44.9841	28,248.06	28,248.08
70,464.56	44.9804	44.9805	70,461.50	70,461.52
140,398.84	44.9792	44.9792	140,395.77	140,395.77
278,769.33	44.9786	44.9786	278,766.27	278,766.26
683,321.00	44.9782	44.9784	683,317.94	683,317.95
1,329,601.05	44.9781	44.9783	1,329,597.99	1,329,598.00
11,236,158.88	44.9780	44.9783	11,236,155.82	11,236,155.63

TABLE 6.6. COMPARISON OF JSC AND GMD RANGE AND ELEVATION VALUES FOR ELEVATION ANGLE OF 90 DEGREES

(No = 0.000386, Ro = 6378166 meters)

R measured, m	JSC Elc, deg	GMD Elc, deg	JSC Dc, m	GMD Dc, m
10.00	90.0000	90.0000	10.00	10.00
200.08	90.0000	90.0000	200.00	200.00
500.18	90.0000	90.0000	500.00	500.00
1,000.35	90.0000	90.0000	1,000.00	1,000.00
2,000.65	90.0000	90.0000	2,000.00	2,000.00
5,001.28	90.0000	90.0000	5,000.00	5,000.00
10,001.80	90.0000	90.0000	10,000.00	10,000.00
20,002.65	90.0000	90.0000	20,000.00	20,000.00
50,002.17*	90.0000	90.0000	50,000.00	50,000.00
100,002.17*	90.0000	90.0000	100,000.00	100,000.00
200,002.17*	90.0000	90.0000	200,000.00	200,000.01
500,002.17*	90.0000	90.0000	500,000.00	500,000.02
1,000,002.17*	90.0000	90.0000	1,000,000.00	1,000,000.04
10,000,002.17*	90.0000	90.0000	10,000,000.00	10,000,000.10

* The JSC atmosphere is terminated at 50 kilometers whereas the exponential atmosphere used in the GMD solution continues to exist in minor amounts above the 50-kilometer level.

CHAPTER 7

DATUM CONVERSIONS AND GENERATION OF DATA SHEETS

This chapter deals with the calculation of coordinate data with respect to any of the spheroid/datum models contained in the program. It also describes the program used to prepare horizontal control data sheets for survey points, calibration points, runway endpoints, and other locations in the Edwards horizontal control network.

Datum Conversion Theory

The orientation of a rigid body is defined by six quantities: three linear translational quantities and three angular or rotational quantities. If the rigid body is the earth spheroid, then two additional quantities are also required to establish the spheroid position of any surface point. However, by using suitable datum definitions, this number can be reduced. For example, if the axes of the spheroid are defined to be parallel to the earth's axis of rotation and to the meridian of Greenwich, then only five constants need be given.

In this document, all datums are referenced to WGS-72 and are considered to have their coordinate frames aligned with, although spatially offset from, the WGS-72 E-F-G triad. Recent satellite data indicate that some small discrepancies exist in the alignments of coordinate axes, especially with the older datums. However, for aerodynamic tracking applications, it is common practice to assume that alignment errors between the newer datums are small enough that only the translational corrections need be considered.

In the datum conversion program, the position of a starting point is established by knowing its latitude, longitude and spheroid elevation in a given spheroid/datum reference. Knowing the spheroid parameters, the E-F-G coordinates of the point can be easily computed with respect to the same datum reference. Next, three translational corrections (du2, dv2, dw2) between the new datum and the WGS-72 origin are subtracted from the three translational corrections (du1, du1, dw1) between the original datum and the WGS-72 origin to obtain the translational corrections (du, dv, dw) between the original and new datums. These corrections are applied to the E-F-G coordinates relative to the original datum to obtain the E-F-G coordinates of the point in the new datum (E2 = E1 + du, F2 = F1 + dv, G2 = G1 + dw). Finally, the latitude, longitude, and spheroid elevation of the point in the new datum are obtained using any one of the four off-spheroid coordinate determination programs described in chapter 5.

Again, it should be emphasized that the simple addition of translational elements neglects any minor tilt which may exist between the datums. However, since it is the common practice among range groups to use only the translational terms, this procedure has been followed in the baseline algorithms so that the results will agree with DMAC and USCGS data.

Datum Conversion Program

Variable Names

Name	Description
Aa	Length of semimajor axis of selected reference spheroid
Bb	Length of semiminor axis of selected reference spheroid
D1	Number used to identify the starting datum
D2	Number used to identify the ending datum
Du	Combined E translation from starting to ending datum
Du1	E translational component of starting datum
Du2	E translational component of ending datum
Dv	Combined F translation from starting to ending datum
Dv1	F translational component of starting datum
Dv2	F translational component of ending datum
Dw	Combined G translation from starting to ending datum
Dw1	G translational component of starting datum
Dw2	G translational component of ending datum
E01	E value of point in starting datum
E2	Square of the eccentricity of the reference spheroid
Ee	Eccentricity of the reference spheroid
En	Normalized E vector used in Lagrange solution
F01	F value of point in starting datum
G01	G value of point in starting datum

Computational Algorithms

The essential algorithms used in the datum conversion subroutine are presented below:

A. Datumconv subroutine: The Datumconv subroutine is the major executive subroutine containing the datum conversion algorithms and calls.

```
Datumconv:!
Call Spher(D1,Aa,Bb,E2,Ee,Du1,Dv1,Dw1)
GOSUB Uwhtoefg
Call Spher(D2,Aa,Bb,E2,Ee,Du2,Dv2,Dw2)
Du=Du1-Du2
Dv=Dv1-Dv2
Dw=Dw1-Dw2
E01=E
F01=F
G01=G
En=(E01+Du)/Aa
Fn=(F01+Dv)/Aa
Gn=(G01+Dw)/Aa
GOSUB Lagrangeterm
END
```

B. Subroutine Spher: Subroutine Spher is used to obtain the parameters for the selected spheroid datum. An index number D10(i) indicates the datum number. The other terms in the calling argument are length of the semimajor axis, length of the semiminor axis, eccentricity squared, eccentricity, and the three delta vectors.

C. Subroutine Lagrangeterm: Subroutine Lagrangeterm accepts the normalized E, F, and G coordinates of the target point in the selected spheroid/datum model and returns the latitude, longitude, and spheroid elevation of the point. This subroutine is fully covered in chapter 5.

Program Operation

The datum conversion routines are a subroutine of GEOD. When GEOD is run, the operator selects DATUM CONVERSION during program initialization. The following display sequence prompts the operator in making the necessary inputs.

A. Output device selection

```
SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER
```

B. Datum selection

SELECT STARTING DATUM

```
 1 = NAD 1927/CLARKE 1866             4 = WGS-72
 2 = MERCURY/FISHER 1960              5 = MERCURY/FISHER 1968
 3 = KAULA 1961                       6 = NWL-8E
----------------------------------------------------------------------
 7 = ADINDAN/CLARKE 1880              21 = SAO-66
 8 = ASCENSION (ASTRO-58)/INTNL       22 = SAO-67
 9 = AUSTRALIAN NATIONAL              23 = SAO-69
10 = CANTON IS (ASTRO-66)/INTNL       24 = SAO-73
11 = EUROPEAN (ed)/INTNL              25 = S AFRICAN 1950/CLARKE 1880
12 = GREAT BRITAIN 1936/AIRY          26 = SOUTH AMERICAN 1969
13 = GUAM 1963/CLARKE 1886            27 = SOUTH AFRICAN/FISHER 1960
14 = INDIAN (id)/EVEREST              28 = TOKYO (TD)/BESSEL 1841
15 = JOHNSON IS (ASTRO-61)/INTNL      29 = VANGUARD/HOUGH
16 = NANKING 1960/INTNL               30 = WAKE IS (ASTRO-52)/HOUGH
17 = GEM-6                            31 = WAKE-ENIWETOK 1960/HOUGH
18 = NWL-9C                           32 = WGS-60
19 = NWL-9D                           33 = WGS-66
20 = OLD HAWAIIAN (OHD)/CLARKE 1966
```

The operator makes the appropriate numerical entry and presses CONT. The program then displays

SELECT ENDING DATUM

along with the same menu of datum/spheroid models. Again the operator makes the appropriate numerical entry and presses CONT.

C. Entry of coordinates of point to be converted

```
ENTER (CLARKE 1866) COORDINATES

ENTER GEODETIC LAT AS D.MS (EG. 35 43 24.6789 = 35.43246789)

ENTER GEODETIC LON AS D.MS (EG. 117 54 38.1243 = 117.54381243)

ENTER ELEV ABOVE SPHEROID IN (METERS)
```

The items in parentheses will vary depending on the operator selections of starting datum and engineering units.

D. Output

```
DATUM:  (MERCURY)                        EARTH MODEL:  (FISHER 1960)

GEODETIC LATITUDE  =  34 57 39.4537  (34.96095937)  ( 0.610183851)
GEODETIC LONGITUDE = 117 54 40.0495  (117.91112486)  ( 2.057937354)
SPHEROID ELEVATION =          796.04 METERS
```

Program Validation

Validation of the datum conversion subroutines is accomplished by comparing computed results with those available from DMAC, USCGS, or NASA publications. Computed results for the various datum/spheroid models are provided in table 7.1 for a point at 35 N and 118 W in the Clarke/NAD model.

TABLE 7.1. COMPARISON OF CLARKE 1866/NAD AND OTHER DATUM COORDINATES (Clarke coordinates of point are 35 N, 118 W, and H = 500 meters)

Datum/Spheroid	Latitude, d-m-s	Longitude, d-m-s	Spheroid elev, m
Clarke 1866/NAD	35 00 00.0000	118 00 00.0000	500.00
Mercury/Fisher 60	35 00 00.4945	118 00 01.9504	508.95
Kaula/Kaula 61	35 00 00.0056	118 00 03.4295	480.89
WGS-72	34 59 59.8136	118 00 03.6724	489.09
Mercury/Fisher 68	34 59 59.7516	118 00 03.3110	484.32
NWL-8E	35 00 00.0997	118 00 03.7443	483.44
Adindan/Clarke 1880	35 00 06.5875	117 59 58.8616	290.91
Ascension/Intnl	35 00 01.7011	117 59 54.5370	222.14
Australian Natl	34 59 57.6970	118 00 00.1838	304.04
Canton Is/Intnl	35 00 14.8609	118 00 19.2199	392.61
European/Intnl	35 00 08.4047	118 00 02.6547	232.20
Great Britain/Airy	34 59 44.9009	118 00 18.7063	846.15
Guam/Clarke 1866	35 00 04.9542	118 00 04.9244	145.98
Indian/Everest	34 59 34.0183	117 59 56.5673	1741.11
Johnston Is/Intnl	35 00 07.5024	118 00 11.4494	418.35
Nanking 60/Intnl	35 00 09.7887	118 00 06.0328	-10.01
GEM-6	35 00 00.1463	118 00 03.7095	485.04
NWL-9C	34 59 59.8031	118 00 03.9324	484.29
NWL-9D	34 59 59.7834	118 00 03.6125	486.89
Old Hawaiian/Cl 1866	35 00 16.0369	118 00 10.6210	430.54
SAO-66	34 59 59.6864	118 00 03.8584	466.06
SAO-67	34 59 59.9965	118 00 03.7747	490.36
SAO-69	34 59 59.8593	118 00 03.7725	492.02
SAO-73	34 59 59.6949	118 00 03.9302	489.83
S African/Clarke 1880	35 00 16.9365	117 59 56.7560	696.68
S American 69	35 00 26.1387	118 00 00.9360	726.92
S Asia/Fisher 60	35 00 00.9467	118 00 05.5329	429.74
Tokyo/Bessel 1841	34 59 32.6510	117 59 49.2464	1138.55
Vanguard/Hough	35 00 00.4042	117 59 59.9423	499.43
Wake Is/Hough	34 59 57.1048	118 00 14.3399	262.30
Wake-Eniwetok 60	35 00 01.6636	118 00 06.3129	501.51
WGS-60	34 59 59.9382	118 00 04.1104	510.00
WGS-66	35 00 00.0887	118 00 03.6865	489.16

Data Sheet Preparation

The data sheet subroutine is simply an executive program that sequentially calls the various other subroutines and subprograms previously described. For this reason, the coding of the individual subroutines is not repeated here. However, the general sequence of operations and the operating instructions are provided.

The data sheet subroutine is intended to prepare CRT or hardcopy listings of horizontal control data referenced to any of the many control points generally used by the ATR. The program is designed to accept a manually entered control point or to sequentially read each control point from a master survey file. In either case, the geodetic latitude and longitude of the point must be entered or read in Clarke 1866/NAD coordinates.

The program then provides an output listing which consists of the name of the station (if named), the NAD coordinates of the point in dms, deg, and radian formats, the California zone 5 and zone 7 Lambert coordinates of the point, and the NAD E-F-G coordinates of the point.

Next, depending on the operator selection, the program will display or print the geodetic and E-F-G coordinates of the point in other selected datums.

Also, if selected by the operator, the program will display or print the range, azimuth, and elevation to all nearby survey points. This can be in either of two operator-selectable formats which are described in the program operation section below.

In making the necessary computations, the program makes use of the Lambert subroutines, the E-F-G subroutines, the datum conversion subroutines, and the various angle and distance routines described in earlier chapters.

Program Operation

Upon entry into GEOD, the operator is asked to select the operating mode. To prepare a data sheet the operator enters the number corresponding to DATA SHEET. The remaining operator entries are shown below.

A. Output device selection

SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER

B. Station selection

0 = MANUAL ENTRY OF STATION
1 = AUTOMATIC ENTRY (ALL STATIONS)
2 = ENTRY OF FILE STATION

If manual entry is selected, the following operator inputs are requested:

ENTER STATION NAME

ENTER STATION NAD LATITUDE IN D.MS FORM

ENTER STATION NAD LONGITUDE IN D.MS FORM

ENTER ELEVATION ABOVE SEA LEVEL IN (METERS)

ENTER GEOID SEPARATION IN (METERS)

If automatic entry is selected, the program sequentially reads one station after another from the survey file and prepares a separate horizontal data sheet for each station.

If the entry is to be a single file station, the program requests the following from the operator:

ENTER RECORD NUMBER
ENTER FILE NUMBER

This request refers to the record and file numbers shown when SURVEY DATA is selected and the survey data are printed during the initial program mode selection.

C. Datum conversions

The program next prompts the operator to enter the number of additional datums in which the coordinates are to be shown on the data sheet. If only Clarke/NAD coordinates are desired, the operator simply presses CONT. If the coordinates are to be shown in other datums, the operator enters the number of additional datums for which coordinates are to be shown. The program then asks the operator to

SELECT ADDITIONAL DATUM

and displays the menu on which all the datums are listed. This display is repeated the number of times corresponding to the number of additional datums previously specified by the operator.

D. Range, azimuth, and elevation to nearby points

The program operator is now asked to choose one of the two options in the following CRT message:

0 = NO ANGLE AND RANGE DATA FOR NEARBY SURVEY POINTS
1 = INCLUDE ANGLE AND RANGE DATA FOR NEARBY SURVEY POINTS

When 0 is selected (or CONT is depressed without a numerical entry), the data sheets are prepared without angle and range data for nearby survey points. When 1 is selected, the program displays the following request.

SELECT RANGE AND ANGLE FORMAT

0 = ANGLES IN D.MS FORMAT - RANGE IN METERS
1 = ANGLES IN DEG - RANGE IN METERS AND YARDS

E. Program Output

A typical horizontal control data form for FPS-16 number 34 is shown below:

HORIZONTAL CONTROL DATA

NAME OF STATION: FPS-16 NO. 34

CLARKE 1866 (NORTH AMERICAN (NAD-27)) COORDINATES ARE:

```
GEODETIC LATITUDE  =  34 57 38.9531    ( 34.96082031)    (0.610181424)
GEODETIC LONGITUDE = 117 54 38.1062    (117.91058506)    (2.057927932)

ELEVATION ABOVE SEA LEVEL:  811.566 METERS  ( 2633.613 FEET)
GEOD SEPARATION:            -24.400 METERS  (  -80.052 FEET)
SPHEROID ELEVATION:         787.166 METERS  ( 2582.560 FEET)

CAL ZONE 5 LAMBERT CRD: X =   617,767.24 METERS  Y =   162,041.17 METERS
                          = 2,026,791.36 FEET      =   531,630.07 FEET
CAL ZONE 7 LAMBERT CRD: X = 1,314,721.03 METERS  Y = 1,360,127.46 METERS
                          = 4,313,380.58 FEET      = 4,462,351.52 FEET
CLARKE 1866 (NORTH AMERICAN (NAD-27)) EFG COORD:
    E =  -2449851.59 M    F =  -4624898.18 M   G =    3634568.77 M
--------------------------------------------------------------------------
FISHER 1960 (MERCURY) EFG COORD:
    E =  -2449848.59 M    F =  -4624787.18 M   G =    3634793.77 M
    FISHER 1960 (MERCURY) GEODETIC COORD:
GEODETIC LATITUDE  =  34 57 39.4537    ( 34.96095937)    (0.610183851)
GEODETIC LONGITUDE = 117 54 40.0495    (117.91112486)    (2.057937354)
SPHEROID ELEVATION:         796.04 METERS
--------------------------------------------------------------------------
KAULA 1961 (KAULA 1961) EFG COORD:
    E =  -2449874.59 M    F =  -4624756.18 M   G =    3634764.77 M
    KAULA 1961 (KAULA 1961) GEODETIC COORD:
GEODETIC LATITUDE  =  34 57 38.9654    ( 34.96082371)    (0.610181483)
GEODETIC LONGITUDE = 117 54 41.5269    (117.91153525)    (2.057944516)
SPHEROID ELEVATION:         767.95 METERS
--------------------------------------------------------------------------
```

FPS-16 NO 34

RANGE, AZIMUTH, AND ELEVATION OF NEARBY SURVEY POINTS

DESCRIPTION OF POINT	AZIMUTH	ELEVATION	RANGE
BST 34 BRASS DISC	202 33 28.1	-2 50 22.2	675.11 METERS
BST 34 FEED HORN	202 41 17.6	0 50 39.8	674.36 METERS
MASTER NORTH BASE	175 53 51.0	-1 21 12.1	1631.24 METERS
MASTER SOUTH BASE	144 54 31.0	-1 15 41.4	5284.46 METERS
NASA 1	120 03 02.0	-2 43 35.4	2311.31 METERS

and so forth until all points have been shown

The above sample shows the output format when 0 is entered for range and angle format in step D above. Had 1 been entered, the format of the range and angle data would have been:

FPS-16 NO 34

RANGE, AZIMUTH, AND ELEVATION OF NEARBY SURVEY POINTS

DESCRIPTION OF POINT	AZIMUTH	ELEVATION	RANGE (M)	RANGE (YD)
BST 34 BRASS DISC	202.5578	-2.8395	675.11	738.31
BST 34 FEED HORN	202.6882	0.8444	674.36	737.48
MASTER NORTH BASE	175.8975	-1.3534	1631.24	1783.95
MASTER SOUTH BASE	144.9086	-1.2615	5284.46	5779.14
NASA 1	120.0506	-2.7265	2311.31	2527.67

and so forth until all points have been shown

Program Validation

Since the data sheet subprogram uses subroutines from other parts of the main program, it suffices to validate individual results obtained on the data sheets using the validation procedures for the individual subroutines. These validation procedures are contained in the sections describing the individual subroutines.

CHAPTER 8

RADAR CROSS SECTIONS AND ANTENNA GAIN PATTERNS

The radar cross section and antenna pattern coverage calculations are designed to use both ground-derived and air-derived data to compute target radar cross sections or antenna coverage patterns. Generally, all necessary data can be obtained from a single flight if the aircraft maneuvers are designed so that a wide range of aspect angles are obtained.

Radar Cross Section Calculations

To compute radar cross section as a function of the impinging angle of the r-f energy on the body axis triad, it is necessary to time correlate body axis data, radar receiver i-f signal levels, and tracking measurements (RAE). It is helpful if the radar system is capable of generating a time-tagged analog (duo-binary) or digital recording of the five radar parameters (time, range, azimuth, elevation, and received signal-to-noise ratio) needed for the analysis. Body axis data, obtained from a stable platform on board the test aircraft, is either transmitted to the ground in real-time or recorded on board for post mission analysis.

This program is not concerned with the manner in which the data was obtained (direct input from digital recordings or manual input from listings or stripchart recordings). It assumes that data from the necessary sources has been entered onto a digital magnetic tape in a format which can be used by the data reduction program.

The following sequence of operations are carried out by the program on data points recorded at 1 second intervals:

1. The aircraft's spatial position is converted from spherical R-A-E coordinates into radar-centered, E-N-V Cartesian form. This is accomplished by calling subroutine Raexyz (fig. 8.1).

2. The E-N-V cartesian coordinates are rotated into alignment with the earth-centered E-F-G Cartesian frame. This is accomplished by calling subroutine Xyzefg (fig. 8.2).

3. The aircraft position is translated to the E-F-G frame. This is accomplished by simple vector addition (fig. 8.3).

4. The ΔE-ΔF-ΔG vector from the aircraft to the radar is computed from the known E-F-G coordinates of the radar site and the computed E-F-G coordinates of the aircraft (fig. 8.4).

5. The geodetic coordinates of the aircraft are computed from its E-F-G coordinates. This is accomplished by the method of Purcell and Cowan which is fully covered in Chapter 5.

6. The ΔE-ΔF-ΔG coordinates of the radar position vector referenced to the aircraft centered, E-F-G-aligned triad are rotated to the aircraft-centered E-N-V frame (fig. 8.4).

7. The E-N-V coordinates of the radar position vector are rotated through the body-axis Euler angles to obtain the radar position with respect to the body axes of the aircraft. This is accomplished by means of a standard three-coordinate matrix rotation in which the direction cosines are computed from the three Euler angles describing the pitch, roll, and heading of the vehicle at that same moment in time.

8. Finally, the radar's Cartesian coordinates, referenced to the aircraft body axes, are converted into spherical form (range-azimuth-elevation) referenced to the same aircraft frame.

This provides the impinging angle of the r-f energy with respect to the aircraft body axes at one second intervals throughout the test period. To obtain a value for radar cross section that corresponds to this impinging angle, radar range values for the same time-correlated, one-second intervals are combined with known radar parameters in the logarithmic form of the classical radar equation,

$$[\sigma] = 4[R] + [B] + [NF] + [L] - [Pt] - 2[\lambda] - 2[G] + [S/N]. \qquad (8.1)$$

Here σ is the radar cross section in decibels referenced to 1 square meter, R is the range in decibels referenced to 1 nautical mile, B is the noise bandwidth of the intermediate-frequency amplifier of the radar in decibels referenced to 1 hertz, L is the transmitting and receiving line losses in decibels, NF is the operating noise figure of the radar in decibels, Pt is the peak transmitted power in decibels referenced to 1 watt, λ is the transmitted wave length referenced to 1 cm, G is the antenna gain over isotropic gain in decibels, and S/N is the ratio of receiver signal power to noise power in decibels. In equation (8.1), decibel values are used for all terms in the square brackets.

The various radar-dependent parameters used in equation (8.1) are determined from calibrated sphere tracks. Typical values recently computed for the DFRC AN/FPS-16 radar are:

1. B - 62 dBHz

2. NF - 11.2 dB

3. L - 5.5 dB

4. λ - 7.243 dBcm

5. G - 42.5 dB

Transmitted power is generally measured from a calibrated power meter on the radar console. One hundred percent power for AN/FPS-16 number 34 is 60 dBW.

In this program, the value of σ computed from equation (8.1) is stored along with the three body axis angles for the impinging signal at each data point. A plotting subroutine allows the value of σ to be plotted as a function of time or as a function of body azimuth angle for a selected elevation window.

Antenna Gain Pattern

To compute antenna gain as a function of impinging angle of the r-f energy on the body axis triad, it is again necessary to time correlate signal levels at the receiving system detector output with both body axis angles and precision tracking data. The ground station may either be a transmitter (if the antenna pattern of the on-board receiver is to be measured) or a receiver (if the antenna pattern of the on-board transmitter is to be measured). This type of calculation provides general antenna pattern information for line-of-sight type high-frequency signals (VHF and higher) only.

During a receiving antenna test, a continuous signal is transmitted from the ground station by means of a directional antenna slaved to a precision tracking source. The signal level at the detector output of the on-board receiver is recorded along with body axis information and timing data. During a transmitting antenna test, a continuous signal is transmitted from the flight vehicle and received by a directional antenna slaved to a precision tracking source. The received signal level is recorded at the receiving site for subsequent time base correlation with recorded tracking data and body axis information.

To perform the analysis, the same sequence of operations is used as described above for determining the angle of impingement (for ground-to-air transmission), or departure (for air-to-ground) transmission. In the case of ground-to-air checks, the gain of the on-board receiving antenna is computed for each 1-second data point by using a rearranged version of the standard transmission formula,

$$[Gra] = [S/N] - [Ptmw] - [Gg] + [Pl] + [Lt] + [Lr] - [Rs] \qquad (8.2)$$

where [Gra] is the gain of the airborne antenna (dB) for the instantaneous aspect angle of the received r-f energy, [Ptmw] is the transmitted power (dBmW), [Gg] is the gain of the ground antenna (dB), [Pl] is the path loss (dBm), [Lt] and [Lr] are the transmitting and receiving line losses (dB), and [Rs] is the receiver sensitivity (dBm). In equation (8.2), decibel values must be used for all terms in the square brackets.

To determine the gain pattern of an on-board transmitter antenna, the same procedure is followed except that the on-board transmitter is continuously keyed while the aircraft performs planned maneuvers which will provide data for all aspect angles of interest. In this case, the signal-to-noise level of the received signal is measured at the ground station and recorded along with the timing data. The equation for determining the gain pattern of the on-board transmitting antenna is

$$[Gta] = [S/N] - [Ptmw] - [Gg] + [Pl] + [Lt] + [Lr] - [Rs]. \qquad (8.3)$$

Constant values used in equations (8.2) and (8.3) are system dependent and entered into the program by the operator prior to commencing the data reduction operations.

In both equations (8.2) and (8.3) the path loss in dB is computed from

$$[P1] = 10 \log_{10} \frac{4\pi R^2}{\lambda^2} \tag{8.4}$$

where R is the vehicle range in meters and λ is the wave length of the transmitted signal in meters.

Variable Names

Name	Description
Adb	Additional losses in radar equation, decibels
Adldb	Additional losses in antenna gain equation, decibels
Alt	Target vehicle altitude above earth spheroid, meters
A(N,0)	N array elements holding target range values
A(N,1)	N array elements holding target azimuth values
A(N,2)	N array elements holding target elevation values
Bwhz	Noise bandwidth, hertz
B(N,0)	N array elements holding signal-to-noise values
B(N,1)	N array elements holding body azimuth values
B(N,2)	N array elements holding body elevation values
B(N,3)	N array elements reserved for parameter storage
B(N,4)	N array elements holding sigma or antenna gain values
B(0,0)	Array location holding tracker latitude
B(0,1)	Array location holding tracker longitude
B(0,2)	Array location holding tracker altitude
B(0,3)	Array location holding magnetic variation
B(0,4)	Array location holding geoid separation at tracker
C	Geodetic course of target

Cosc	Cosine of target course angle
Cosp	Cosine of target pitch angle
Cosr	Cosine of target roll angle
C(N,0)	N array elements holding target magnetic heading
C(N,1)	N array elements holding target geodetic course
C(N,2)	N array elements holding target pitch angle
C(N,3)	N array elements holding target roll angle
C(0,0)	Array element holding starting time of run
C(0,1)	Array element holding ending time of run
Dualdb	S/N loss due to dual mode, decibels
D(N,0)	N array elements holding target latitude, degrees
D(N,1)	N array elements holding target longitude, degrees
D(N,2)	N array elements holding target sea-level altitude, meters
E	E coordinate in E-F-G frame of reference
F	F coordinate in E-F-G frame of reference
Fmh	Radar frequency, megahertz
G	G coordinate in E-F-G frame of reference
Gadb	Gain of airborne antenna, decibels
Ggdb	Gain of ground antenna, decibels
Gsep	Geoid separation, meters
Gtdb	Gain of transmitting antenna, decibels
H	Altitude of target above sea level, meters
Hr	Hours after start of run
Lamdbcm	Wave length referenced to 1 centimeter, decibels
Lamdbm	Wave length referenced to 1 meter, decibels
Lamdbmi	Wave length reference to 1 nautical mile, decibels

Lam	Wave length in current units
Lam2	Square of wave length in current units
Lat	Target latitude, degrees
Lon	Target longitude, degrees
Lrdb	Receiver line losses, decibels
Ltdb	Transmitter line losses, decibels
Ltrdb	Combined transmitter and receiver line losses, decibels
MAT A2	Matrix used to store body axis direction cosines
MAT E	Generalized matrix used to store E-F-G coordinates
MAT Er	Matrix used to store radar E-F-G coordinates
MAT Ev	Matrix used to store vehicle E-F-G coordinates
MAT X	Generalized matrix used to store X-Y-Z coordinates
MAT X1	Matrix used to store vehicle-to-radar coordinates
Magvar	Input parameter for magnetic variation, degrees
Matflg	Flag indicating forward or reverse matrix rotation
Mhdg	Target magnetic heading
Min	Minutes after start of run
Nbwdbhz	Noise bandwidth referenced to 1 hertz, decibels
Nfdb	Radar noise figure, decibels
P	Working value of target pitch angle, degrees
Pit	Input target pitch parameter, degrees
Pldbm	Path loss computed on basis of 1 meter, decibels
Ptdbmwa	Airborne transmitter power referenced to 1 milliwatt, decibels
Ptdbmwg	Ground transmitter power referenced to 1 milliwatt, decibels
Pldbnm	Path loss referenced to 1 nautical mile, decibels
Ptdbw	Transmitter power referenced to 1 watt, decibels

R	Working value of target roll angle, degrees
Rol	Input value of target roll angle, degrees
Rsdb	Receiver sensitivity, decibels
Rsdba	Airborne receiver sensitivity, decibels
Rsdbg	Ground receiver sensitivity, decibels
Sec	Seconds after start of run
Sigma	Radar cross section referenced to 1 square meter, decibels
Sinc	Sine of target geodetic course
Sinp	Sine of target pitch angle
Sinr	Sine of target roll angle
Snr	Signal to noise ratio, decibels
U	Latitude transfer variable, degrees
Var	Magnetic variation, degrees
W	Longitude transfer variable, degrees
X	Transfer variable for radar-centered east coordinate
Y	Transfer variable for radar-centered north coordinate
Z	Transfer variable for radar-centered vertical coordinate

Computational Algorithms

Since all of the geodetic computations for this routine have been covered in earlier chapters, they will not be repeated here. The algorithms used to compute radar cross section and antenna gain are given below:

A. Radar cross section computation

Radar cross section is computed from the following algorithms:

```
Pldbnm=10*LGT(A(N,0)/1852)

Sigma=Ltrdb+Nfdb+Nbwdbhz+4*Pldbnm-2*Lamdbcm-2*Gtdb-Ptdbw+Snr+Adb
```

B. Antenna gain computation

Antenna gain is computed from the following algorithms:

```
Pldbm=10*LGT((4*PI*A(N,0))**2/Lam2)

Gadb=Snr-Ptdbmw-Ggdb+Pldbm+Ltdb+Lrdb-Rsdb
```

C. Angle calculation executive routine

The executive routine used to compute the impinging angles of the r-f energy on the vehicle body is given in steps 1 to 22. In entering this subroutine, values for pitch, roll, and course are provided as variables P, R, and C, respectively. Values for range, azimuth, and elevation are provided as Rng, Az, and El respectively. In step 2, the range, azimuth, and elevation coordinates of the target are converted to Cartesian E-N-V coordinates. In step 4, the E-N-V target coordinates are rotated into E-F-G alignment. Matflg is a flag which controls the direction of the rotation. When it is set equal to 1, the rotation in subroutine Xyzefg is from E-N-V to E-F-G. When Matflg is set equal to 2, subroutine Xyzefg rotates E-F-G coordinates into E-N-V alignment. In step 5, the target's radar-centered E-F-G coordinates are added to the radar's earth-centered E-F-G coordinates to yield the target position in the earth-centered E-F-G reference frame. In step 9, subroutine Purcell is called to compute the target's latitude, longitude, and elevation (Lat, Lon, and Alt) from its E-F-G coordinates. In step 12 the spheroid altitude of the target is converted to altitude above sea level by subtracting the geoid separation (Gsep) for the tracking site. In step 14, the radar-to-target E-F-G coordinates are reversed to yield the target-to-radar E-F-G coordinates, and these are then rotated into alignment with the E-N-V frame of reference at the position of the target aircraft. Subroutine Bodyangles is then called to compute the direction cosines needed to rotate the vehicle-to-radar E-F-G coordinates into alignment with the vehicle body axes. This is accomplished by using the three body-axis Euler angles P, R, and C to compute the six-dimensional rotation matrix A2. The actual rotation is accomplished in step 17 which yields the Cartesian coordinates of the radar site with respect to the vehicle body-axis triad. In step 21, the radar's Cartesian coordinates are converted to spherical form (body-axis azimuth, body-axis elevation, and range).

```
1.  Compentry:!
2.  GOSUB Raexyz
3.  Matflg=1
4.  GOSUB Xyzefg
5.  MAT Ev=Er+E
6.  E=Ev(0)
7.  F=Ev(1)
8.  G=Ev(2)
9.  GOSUB Purcell
10. U=Lat
11. W=Lon
12. H=Alt-Gsep
13. Matflg=2
```

14. MAT E=E*(-1)
15. GOSUB Xyzefg
16. GOSUB Bodyangles
17. MAT X1=A2*X
18. X=X1(0)
19. Y=X1(1)
20. Z=X1(2)
21. GOSUB Xyzrae
22. RETURN

D. Radar cross section calculation

The Compute subroutine initializes the radar coordinates, magnetic variation, and geoid separation which will be used for all of the subsequent calculations. In steps 2 through 6 the radar-dependent values are picked up from the holding elements in array B. In step 7, the sea-level altitude of the radar is converted to spheroid altitude by addition of the geoid separation factor (Gsep). In step 8, the program branches to subroutine Model where the earth model parameters are assigned values. In step 9, the value of the E-W radius of curvature for the radar site is computed, and, in steps 10 and 11, the earth-centered E-F-G coordinates of the radar are computed and stored in MAT Er. If Opmode has a value of 4, indicating COMPUTE A/C ANTENNA GAIN PATTERN (the fourth menu selection) had been chosen by the operator, the program branches to the antenna gain calcuation subroutine, Computegain. If Opmode has any other value, the program continues with step 13 where a FOR-NEXT loop is initiated. N, the loop index, is set to range from 1 to the maximum number of data points in the run. The total number of data points that must be processed is equal to the ending time of the run in seconds (stored in C(0,1)) minus the starting time of the run in seconds (stored in C(0,0)) plus one. At step 14 the program branches to subroutine Init where the transfer variables for radar-to-target range, azimuth, and elevation and the aircraft pitch, roll, and course are set equal to the proper stored array values for that second of the test. The program then branches to the angle computation routine and returns values for the body-referenced target-to-radar azimuth and elevation values and the target's geodetic coordinates for the same second. These values are stored in the specified holding array elements. At step 15, the signal-to-noise value for the same data point is assigned to Sndb. If the value happens to be 0, indicating that no data was taken for the point, the program sets Sigma equal to zero and bypasses the cross-section calculations. Otherwise, path loss is computed in step 19, and radar cross section is computed in step 20. The value of radar cross section for the data point is stored in array element B(N,4).

The subroutine then returns to the start of the loop and initializes for the same computations on the next data point.

1. Compute:!
2. U=Rlat=B(0,0)
3. W=Rlon=B(0,1)
4. Ralt=B(0,2)
5. Var=B(0,3)

```
6.  Gsep=B(0,4)
7.  H=Ralt+Gsep
8.  GOSUB Model
9.  GOSUB Ncalc
10. GOSUB Efgcalc
11. MAT Er=E
12. IF Opmode=4 THEN Computegain
13. FOR N=1 to C(0,1)-C(0,0)+1
14. GOSUB Init
15. Sndb=B(N,0)
16. IF Sndb<>0 THEN 19
17. Sigma=0
18. GOTO 21
19. Pldbnm=10*LGT(A(N,0)/1852)
20. Sigma=Ltrdb+Nfdb+Nbwdbhz+4*Pldbnm-2*Lamdbcm-2*Gtdb-Ptdbw+Sndb+Adb
21. B(N,4)=Sigma
22. NEXT N
```

E. Antenna gain pattern calculation

Antenna gain calculations use the same radar and earth model initializations as the radar cross section calculations (up through step 12 in the Compute subroutine). At step 12 in that subroutine, the program branches to the Computegain subroutine where a similar FOR-NEXT loop is established. The same data point initialization is performed in step 3, and the same angle values are returned. In step 4, the stored signal-to-noise measurement for the data point is assigned to Sndb. If the value is 0, indicating that no data was taken for that point, the antenna gain is set equal to zero and the program branches to step 10. If the signal-to-noise ratio (decibels) was other than 0, the program branches to steps 8 and 9 where the path loss and antenna gain computations are made. The computed antenna gain for the data point is stored in array location B(N,4), and the program recycles to the beginning of the loop where it initializes for the next data point and continues the same computations.

Note that the zero decibel value for the signal-to-noise ratio is used to detect a no-data condition, even though a valid zero value could be present. This causes no difficulty since the probability that the value is precisely zero (the necessary condition for the bypass branch) is exceedingly small. Should a true zero value be present, then one invalid data point would be obtained.

```
1. Computegain:!
2. FOR N=1 TO C(0,1)-C(0,0)+1
3. GOSUB Init
4. Sndb=B(N,0)
5. IF Sndb<>0 THEN 8
6. Gadb=0
7. GOTO 10
8. Pldbm=10*LGT((4*PI*A(N,0))**2/Lam2)
```

9. Gadb=Sndb-Ptdbmw-Ggdb+Pldbm-Rsdb+Ltdb+Lrdb
10. B(N,4)=Gadb
11. NEXT N

F. Subroutine Init

Subroutine Init initializes the transfer variables with radar range, azimuth, and elevation and vehicle pitch, roll, and yaw parameters for the data point for which the radar cross section or antenna gain value is to be computed. At step 9 the subroutine branches to Compentry where the impinging angles of the r-f energy on the body axes of the vehicle and the vehicle's geodetic coordinates are computed. These values are stored in the appropriate array locations in steps 10 through 14, and, in step 15, the subroutine returns to the calling program.

1. Rng=A(N,0)
2. Az=A(N,1)
3. El=A(N,2)
4. P=C(N,2)
5. R=C(N,3)
6. C=C(N,1)
7. Mhdg=C(N,0)
8. H=Ralt+Gsep
9. GOSUB Compentry
10. B(N,1)=Az
11. B(N,2)=El
12. D(N,0)=U
13. D(N,1)=W
14. D(N,2)=H
15. RETURN

Program Operation

Both XSECT (radar cross section and antenna gain program) and RFPL are part of a main program GEOD2. When Geod2 is entered, the following display appears on the CRT.

MAKE PROGRAM SELECTION

0 = CROSS SECTION/ANTENNA GAIN CALCULATIONS
1 = SKYSCREEN

If the operator selects 0 and presses CONT, the program enters XSECT. As soon as XSECT is entered, the following display appears:

```
MAKE MODE SELECTION

0 = MANUAL CALCULATION OF BODY AXIS ANGLES
1 = ENTRY OF S/N, RAE, OR ATTITUDE DATA
2 = CALCULATION OF SIGMA FROM STORED DATA
3 = PRINT DATA OR CALCULATED PARAMETERS
4 = COMPUTE A/C ANTENNA GAIN PATTERN
5 = READ DATA FROM TAPE
6 = STORE DATA ON TAPE
7 = PLOT SIGMA VALUES VS. BODY ANGLES
8 = PLOT ANTENNA GAIN VS. BODY ANGLES
9 = PLOT SIGNAL TO NOISE VS. TIME
```

The following sections describe the prompt displays and operator responses for each of the operating modes.

A. Manual calculation of body axis angles: If this mode is selected, the program prompts the following operator entries:

```
MAKE SELECTION

0 = FPS-16 NUMBER 34
1 = MANUAL ENTRY
```

If FPS-16 NUMBER 34 is selected, the program picks up stored values for latitude, longitude, sea-level elevation, geoid separation, and magnetic variation. If MANUAL ENTRY is selected, the operator is prompted to enter all of the same parameters as:

```
ENTER RADAR LATITUDE IN DEGREES
ENTER RADAR LONGITUDE IN DEGREES
ENTER SEA LEVEL ELEV OF RADAR IN METERS
ENTER GEOID SEPARATION AT RADAR IN METERS
ENTER MAGNETIC VARIATION AT RADAR IN DEGREES
```

This is followed by the following sequential prompts for target position and attitude data:

```
ENTER TARGET RANGE IN YARDS
ENTER TARGET AZIMUTH IN DEGREES
ENTER TARGET ELEVATION IN DEGREES
ENTER TARGET PITCH ANGLE (UP IS POS)
ENTER TARGET ROLL ANGLE (RT WING UP IS POS)
ENTER TARGET MAGNETIC HEADING
```

After making the last entry, the program computes the target-to-radar body angles and the target geodetic coordinates. The results are displayed as follows:

```
TARGET TO RADAR RANGE (YDS) = 20031.2836
TARGET TO RADAR AZIMUTH     =   268.4356
TARGET TO RADAR ELEVATION   =   -20.1274

TARGET LATITUDE             =  35.1235734235
TARGET LONGITUDE            = 117.9845374294
TARGET ALTITUDE (FT)        = 24015.2435
```

The program then returns to the operator entry point to await additional entries.

B. Entry of SN, RAE, or attitude data: If the entry selection is made by the operator, the following prompt message appears:

```
MAKE SELECTION

0 = ENTER S/N RATIO VS TIME
1 = ENTER RAE VS TIME
2 = ENTER PITCH VS TIME
3 = ENTER ROLL VS TIME
4 = ENTER MAG HDG VS TIME
```

If S/N data is to be entered, the program requests the starting and ending IRIG times for which the data is to be entered. This is prompted by:

```
ENTER STARTING IRIG TIME TO NEAREST SECOND AS HH:MM:SS
ENTER ENDING IRIG TIME TO NEAREST SECOND AS HH:MM:SS
```

The program also prompts the entry of the same radar parameters as described above (AN/FPS-16 34 or manual entry). After the radar entries are made, the program sequentially requests S/N data. The time is automatically indexed by 1 second after each entry. If an entry is to be made for a time other than that automatically displayed, the operator must enter the time before the S/N entry. This is accomplished by the following prompt messages:

```
AUTOTIME IS 13:23:12  IF OTHER TIME IS DESIRED ENTER AS HH:MM:SS
```

If the time for the entry is correct the operator presses CONT. Otherwise he enters the desired time and then presses CONT. The next prompt appears as:

```
ENTER S/N VALUE FOR 13:23:12
```

This sequence continues until all S/N points have been entered.

C. Entry of RAE data: If the RAE entry mode was selected, the program follows the same procedures as described for step B above except that three entries are made for each incremented or manually entered time. The prompt messages are:

```
ENTER RANGE (YDS) FOR 13:23:12
ENTER AZIMUTH (DEG) FOR 13:23:12
ENTER ELEVATION (DEG) FOR 13:23:12
```

D. Entry of pitch, roll, or yaw data: If pitch, roll, or yaw data is to be entered, the formats of the entry prompts are:

```
ENTER PITCH VALUE (NOISE UP IS POS) FOR 13:23:12
```

or

```
ENTER ROLL VALUE (RT WING UP IS POS) FOR 13:23:12
```

or

```
ENTER MAGNETIC HEADING VALUE FOR 13:23:12
```

If CALCULATION OF SIGMA FROM STORED DATA was selected at the main menu point, then the program uses the stored S/N and range values to compute radar cross section (Sigma). For the calculations to be performed, the RAE, body axis angles, and S/N data must be in the computer.

Prior to commencing the calculations, the program displays the default parameters which will be used in the calculations. If any of these must be changed, the operator may enter the number corresponding to the parameter which must be changed, and the program will prompt the operator to make the revised entry. During these entries, the parameter display portion of the CRT is locked so that the parameter values simply toggle as the operator selections are made. The operator prompt messages needed to make the revised entries all appear below the locked parameter display:

```
(1) SYSTEM PEAK POWER OUTPUT IN DBW       =  60.00
(2) ANTENNA GAIN IN DB                    =  42.50
(3) WAVE LENGTH IN DBCM                   =   7.24
(4) LINE LOSSES IN DB                     =   5.50
(5) NOISE FIGURE IN DB                    =  11.20
(6) NOISE BANDWIDTH IN DBHZ               =  62.04
(7) ADDED LOSSES (EX. DUAL BAND) IN DB    =   3.00
**************************************************
```

When all necessary revisions to the default parameters have been made, pressing CONT will take the program into the calculation mode.

As each calculation is made, the results are displayed on the CRT. As soon as the display stops scrolling, the calculations are complete and may be printed out or plotted by making the appropriate menu selections.

If PRINT DATA OR CALCULATED PARAMETERS was selected at the main menu point, a printout of any of the data files may be obtained. The following prompt message is displayed to the operator.

MAKE SELECTION

0 = TIME, SNR, AND MEASURED RAE VALUES
1 = TIME, TLAT, TLON, AND TALT
2 = TIME, RNG, BODY AZ, BODY EL, AND SIGMA
3 = TIME, RNG, BODY AZ, BODY EL, AND ANT GAIN
4 = TIME, PITCH, ROLL, AND COURSE

Had COMPUTE A/C ANTENNA GAIN PATTERN been selected at the main menu point, the program follows the same procedure as described for the calculation of sigma values, except that the default parameters needed for the antenna gain equation are first displayed to the operator. The form of this display is as follows.

(1) TRANSMITTED POWER IN DBMW	=	39.99
(2) GROUND ANTENNA GAIN IN DB	=	30.00
(3) ADDL LOSSES OR ADJUSTMENTS IN DB	=	0.00
(4) TRANSMIT LINE LOSS IN DB	=	0.00
(5) RECEIVE LINE LOSS IN DB	=	0.00
(6) RECEIVER SENSITIVITY IN DB	=	103.00
(7) WAVE LENGTH IN DBM	=	.87

****

Again the operator may modify any of the displayed parameters by entering the number of the parameter and pressing CONT. This will cause the program to branch to an entry subroutine that will request the revised values. Once the parameter table shown above has the desired values, the operator presses CONT and the program enters the computational mode.

As each value is computed, it is displayed on the operator CRT. When the CRT stops scrolling, the calculations are complete and the operator may select the desired print or plot mode by pressing CONT, which returns the program to the main menu point where the selections may be made.

The fifth and sixth main menu selections (READ DATA FROM TAPE and STORE DATA ON TAPE) are self-explanatory.

Three plotting selections are provided: (1) PLOT SIGMA VALUES VS. BODY ANGLES, (2) PLOT ANTENNA GAIN VS BODY ANGLES, and (3) PLOT SIGNAL-TO-NOISE VS. TIME. All of these modes are automatic. In each case, the program establishes maximum and minimum values for the plot axes based on the maximum and minimum values contained in the files. The proper scales, labels, and headings are placed on the plot without additional operator entries being made.

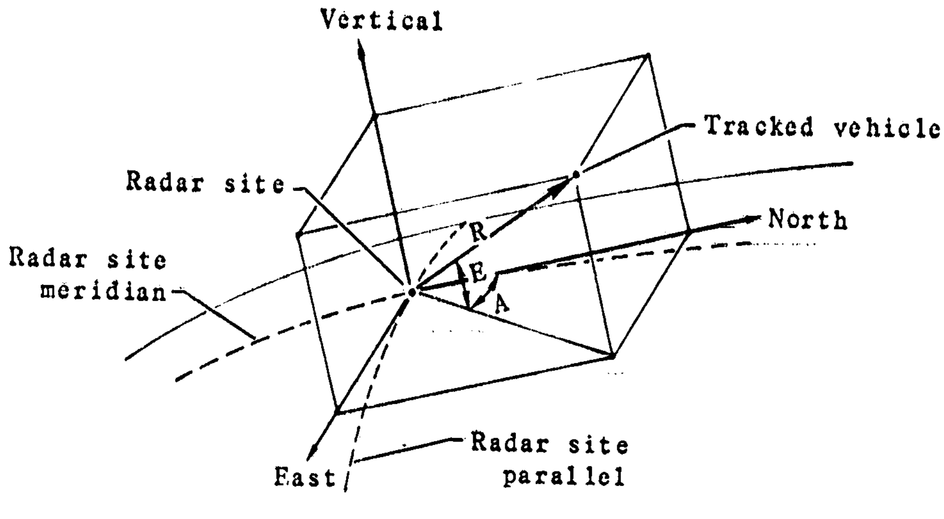

Figure 8.1.

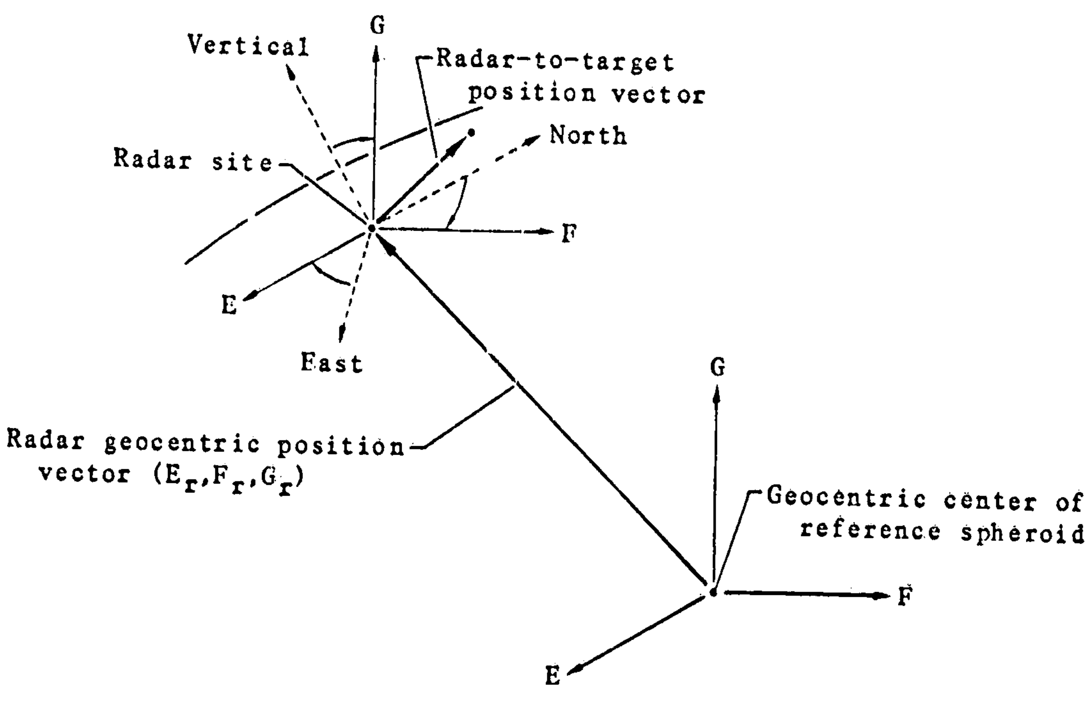

Figure 8.2.

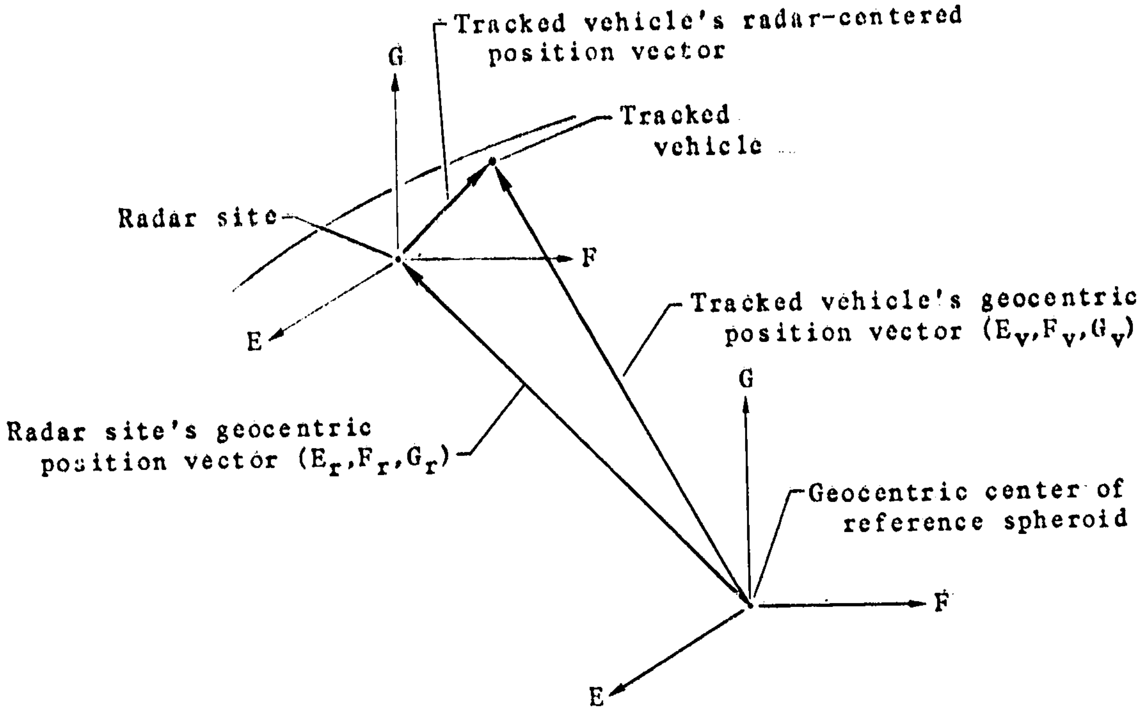

Figure 8.3.

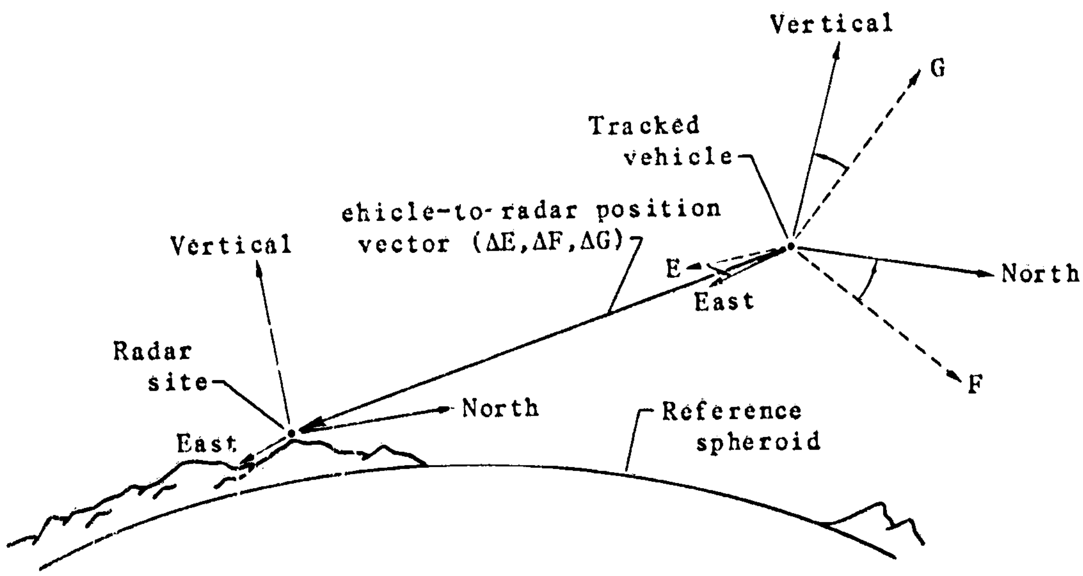

Figure 8.4.

CHAPTER 9

CIRCUIT MARGIN CALCULATIONS

This chapter provides theory and operating instructions for the circuit margin calculations made in subprogram RFPL. RFPL is used to compute circuit margin values for various standard transmitting and receiving systems used on the NASA Aerodynamic Test Range.

General Theory

In order to insure that the various r-f links used for tracking, telemetry, and communications purposes have sufficient signal-to-noise ratios to provide continuous communications throughout any specific mission, it is often desirable to compute the circuit margin values beforehand.

In RFPL, the circuit margin for the r-f link is determined from the relation

$$[Cm] = [Sn] - [Snr] \tag{9.1}$$

where [Cm] is the circuit margin in decibels, [Sn] is the actual or predicted signal-to-noise ratio at the detector of the receiver in decibels, and, [Snr] is the required signal-to-noise ratio in decibels. Note that throughout the derivations which follow, square brackets around parameters indicated that they must be expressed as decibel quantities.

The default values of required received power for various r-f links used on the ATR are given in table 9.1.

Table 9.1. REQUIRED S/N VALUES FOR ATR R-F LINKS

Modulation type	Reqd. receive power
FM/FM, PAM/FM, PDM/FM	9 - 12 dB
PCM/FM	15 dB
UHF voice	10 dB
On-board C-band beacon	12 dB
C-band radar	12 dB

[Cm] may be calculated from the well known r-f link equation

$$[Cm] = [Pt] - [Lt] + [Gt] - [Pl] + [Gr] - [Lr] + [Rs] - [Snr]. \tag{9.2}$$

where [Pt] is the transmitted power output in decibels referenced to one milliwatt, [Lt] is the transmitting line losses in decibels, [Gt] is the gain of the transmitting antenna in decibels, [Pl] is the path loss in decibels,

[Gr] is the receiver gain in decibels, [Lr] is the receiving line losses in decibels, and [Rs] is the receiver sensitivity in decibels referenced to 1 milliwatt.

Transmitted power output is computed from the relation

$$[Pt] \doteq 10 \log \frac{Pt \text{ (watts)}}{1 \text{ (mW)}}. \qquad (9.3)$$

Path loss is computed from the relation

$$[Pl] \doteq 10 \log \frac{(4\pi R^2)}{\lambda^2} \qquad (9.4)$$

where R is the length of the r-f path, and λ is the wavelength of the transmitted signal. The units used for R and λ in equation (9.5) must be consistent.

Receiver sensitivity in watts is found from the relation

$$Rs = k \; Te \; Bw \qquad (9.5)$$

where k is the Boltzmann constant ($1.38/10^{-23}$ joules per kelvin), Te is the equivalent noise temperature (kelvin), and Bw is the receiver noise bandwidth (hertz). Receiver sensitivity in dBm is obtained from

$$Rs = -10 \log \frac{k \; Te \; Bw}{1000}. \qquad (9.6)$$

Equivalent noise temperature is found from

$$Te = (Nfr - 1)T_0 \qquad (9.7)$$

where Te is the equivalent noise temperature (kelvin), Nfr is the receiver noise figure expressed as a ratio, and T_0 is the reference temperature (290 K).

[Lt] and [Lr] are measured parameters.

For ATR telemetry systems, the system r-f link characteristics are as shown in table 9.2

TABLE 9.2. ATR TELEMETRY SYSTEM CHARACTERISTICS

Parameter	Dual Band System	Triplexed System
Noise figure (preamp on)	5 dB	3 dB
Noise figure (preamp off)	9 dB	10 dB
Noise bandwidth	500 kHz	500 kHz
Receiving antenna gain	33 dB	35 dB
Receiving line losses	8 dB	8 dB
Typical frequency	1480.5 mHz	1480.5 mHz
Typical wavelength	0.2026 m	0.2026 m

A sample test calculation for the dual-band antenna is provided below. In this case, Pt has been taken as 5 watts, unity gain has been used for the transmitting (airborne) antenna, 100 statute miles have been taken for the distance, and 33 dB has been taken as the the receiving antenna gain. The preamplifier is assumed to be used for the sample solution.

$$[Pt] = 10 \log \text{Xmit Pwr (mW)}$$

$$= 10 \log 5 \times 10^3$$

$$= 36.9897 \text{ dBm}$$

$$[Pl] = 10 \log \frac{(4\pi R^2)}{\lambda^2}$$

$$= 10 \log \frac{(4\pi \times 1.609 \times 10^5 \text{ m})^2}{(0.2026 \text{ m})^2}$$

$$= 139.9825 \text{ dB}$$

$$Te = (Nfr - 1)T_0$$

$$= (3.16 - 1)\ 290 \text{ K}$$

$$= 627.06 \text{ K}$$

$$[Rs] = -10 \log (k\ Te\ Bw)$$

$$= -10 \log \frac{(1.38 \times 6.27.06 \times 5)}{10^{25}}$$

$$= 113.6384 \text{ dBm}$$

Therefore, for a PAM signal,

$$[CM] = [Pt] + [Gt] - [Lt] - [Pl] + [Gr] + [Rs] - [Lr] - [Snr]$$

$$= 37 + 0 - 0 - 140 + 33 + 114 - 8 - 12$$

$$= 24 \text{ dB}$$

For the triplexed antenna with preamplifier out and the same airborne conditions, different values for Gr and Rs would be used. The value for Gr, found from table 9.2, is 35 dB, and Te is found from

$$Te = (Nfr - 1)T_0$$

$$= (10 - 1)\ 290\ K$$

$$= 2620\ K$$

and [Rs] is found from

$$[Rs] = -10 \log \frac{(1.38 \times 2610 \times 5)}{10^{15}}$$

$$= 107.4451 \text{ dBm.}$$

Therefore, for a PCM signal,

$$[Cm] = [Pt] + [Gt] - [Lt] - [Pl] + [Gr] - [Lr] + [Rs] - [Snr]$$

$$= 37 + 0 - 0 - 140 + 35 + 107 - 8 - 15$$

$$= 16 \text{ dB}$$

The system specifications for the AN/FPS-16 radar are shown in table 9.3, and specifications for several common airborne beacon systems are shown in table 9.4.

TABLE 9.3. C-BAND LINK CHARACTERISTICS

Parameter	Value
Power output	1 MW
Receiver NF (paramps on)	4 dB
Receiver NF (paramps off)	11 dB
Receiver bandwidth.	2 MHz
Transmitting antenna gain	43 dB
Radar line losses (Xmit)	4 dB
Radar line losses (Rec)	2 dB

TABLE 9.4. TYPICAL AIRBORNE BEACON CHARACTERISTICS

Parameter	228	302	207	DPN-66
Transmitting power	5 W	300 W	40 W	500 W
Receiver sens.	40 dBm	70 dBm	70 dBm	70 dBM
Receiving ant. gain	0 dB	0 dB	0 dB	0 dB

For the radar-to-target link

$$[Pt] = 10 \log \frac{\text{Xmit power}}{1 \text{ mW}}$$

$$= 10 \log 1 \times 10^6$$

$$= 90 \text{ dBm}$$

$$[Pl] = 10 \log \frac{(4\pi R^2)}{\lambda^2}$$

$$= 10 \log \frac{(4\pi \times 1.609 \times 10^5)^2}{(0.0530 \text{ m})^2}$$

$$= 151.63$$

Therefore, with a 12 dB signal-to-noise requirement,

$$[Cm] = [Pt] - [Lt] + [Gt] - [Pl] + [Gr] - [Lr] + [Rs] - [Snr]$$

$$= 90 - 4 + 43 - 151 + 0 - 0 + 70 - 12$$

$$= 36 \text{ dB}$$

For the beacon return link the gain of the transmitting and receiving antennas is reversed, the receiving line losses of the radar must be considered, the ground receiver sensitivity must be computed, and the airborne transmitter power is used.

$$[Pt] = 10 \log 400 = 56 \text{ dBm}$$

$$Te = (Nfr - 1)T_0 = (2.51 - 1)\ 290 \text{ K} = 438.45 \text{ K}$$

$$[Rs] = -10 \log \frac{(1.38 \times 438.45 \times 2)}{10^{17}}$$

$$= 109.17 \text{ dBm}$$

Therefore, having a 12 dB requirement for lockon,

$$[Cm] = [Pt] + [Gt] - [Lt] - [Pl] + [Gr] - [Lr] + [Rs] - [Snr]$$

$$= 56 + 0 - 0 + 151 + 43 - 2 + 109 - 12$$

$$= 43 \text{ dB}$$

For uhf communications, typical values for the ground and airborne parameters are given in table 9.5

TABLE 9.5. UHF R-F LINK CHARACTERISTICS

Parameter	Value
Power output (air)	5, 10, or 20 W
Airborne receiver sens.	100 dB
Airborne antenna gain	0 dB
Frequency	225 to 400 MHz
Ground transmitter power	50, 75, or 100 W
Ground antenna gain (AT-1097 GR)	5 dB
Ground receiver sens.	103 dBm
Transmit line loss	1 dB
Receive line loss	1 dB

For the first uhf example, consider the case where a 20 watt airborne transmitter is installed.

$$[Pt] = 10 \log (20 \times 10^3) = 43.01 \text{ dBm}$$

Since the manufacturer's receiver sensitivity is already given, the downlink circuit margin can be calculated as

$$[Cm] = [Pt] + [Gt] - [Lt] - [Pl] + [Gr] - [Lr] + [Rs] - [Snr]$$

$$= 43 + 0 - 0 - 127 + 5 - 1 + 103 - 10$$

$$= 11 \text{ dB}$$

Voice uplink margin for the same 100 statute mile distance with 75 W of transmitted power would be computed from

$$[Pt] = 10 \log (100 \times 10^3) = 48.75 \text{ dBm}$$

and

$$[Cm] = [Pt] - [Lt] + [Gt] - [Pl] + [Gr] - [Lr] + [Rs] - [Snr]$$

$$= 48 - 1 + 5 - 127 + 0 - 0 + 100 - 10$$

$$= 15 \text{ dB}$$

It should be noted that a different equation must be used for the C-band radar link when in skin track mode. In this case the target's radar cross section must be known (from theoretical estimates or from actual measurements).

The classical radar equation is

$$S/N = \frac{Pt\ G^2\ \lambda^2\ \sigma}{R^4\ Bw\ Nf\ L} \times 1.07 \qquad (9.8)$$

where Pt is the transmitted power given in watts, σ is the radar cross section in m^2, R is the target range in nautical miles, B is the equivalent noise bandwidth in hertz, and Gain (G), noise factor (Nf), and combined line losses (L) are given as power ratios. Since the signal-to-noise calculations are only approximate, it is generally the practice to neglect the 1.07 factor since it is nearly unity.

Converting equation (9.9) to decibel form yields

$$[Cm] = [Pt] + 2[G] + 2[\lambda] + [\sigma] - 4[R] - [Bw] - [Nf] - [L] - [Snr] \qquad (9.9)$$

Considering an AN/FPS-16 skin tracking situation with a 1 m^2 target at 100 nautical miles and parametric amplifiers off, the values for the various parameters in equation (9.9) are

$[P_t] = 60$ dBw

$[G] = 43$ dB

$[\lambda] = 7.5$ dBcm

$[\sigma] = 0$

$[R] = 20$ dBnm

$[B_w] = 60$ dBhz

Substituting these values into equation (9.9) yields

$$[C_m] = 60 + 2(43) + 2(7.5) + 0 - 4(20) - 60 - 11 - 4 - 12 = -6 \text{ dB},$$

which indicates that the system would be 6 dB below the required 12 dB lockon signal strength.

Variable Names

Name	Description
Beacon	Flag indicating type of beacon
Bn	Noise bandwidth, dBHz
Bn16	Noise bandwidth of AN/FPS-16 radar, dBHz
Bw	Bandwidth, KHz
Bw16	Bandwidth of AN/FPS-16 radar, dBHz
C	Speed of light (3×10^8 m/s)
Change	Flag indicating certain default parameters must be revised
Cm	Circuit margin, dB
Freq	Frequency, MHz
Freqb	Beacon response frequency, MHz
Freq16	AN/FPS-16 transmitting frequency, MHz
Gr	Gain of receiving antenna, dB
Gr16	Receiving gain of AN/FPS-16 antenna, dB
Gt16	Transmitting gain of AN/FPS-16 antenna, dB
L	Wave length, cm

L16	Skin track wave length, cm
Lam	Wave length, dBm
Lr	Receiving line losses, dB
Lr16	Receiving line losses of AN/FPS-16 radar, dB
Lt	Transmitting line losses, dB
Mode	Flag indicating program mode selection
Mode2	Flag indicating dual band mode selection
Mode3	Flag indicating triplex antenna mode selection
Mode5	Flag indicating uhf mode selection
Mode6	Flag indicating radar mode selection
Nf	Noise figure, dB
Percent	Percent power
Pl	Path loss, dBnm
Pt	Transmitted power, dBW or dBmW
Pta	Actual transmitting power, W or KW
Pta16	AN/FPS-16 transmitting power, KW
R	Range, dBnm
Rng	Range, nm
Rs	Receiver sensitivity, dBm
Sigma	Radar cross section, dBm^2
Snr	Required signal-to-noise ratio, dB
Snr16	Required skin track signal-to-noise ratio, dB
Te	Equivalent noise temperature, K
Tlm	Flag indicating type of telemetry modulation

Computational Algorithms

The essential subroutines used in the RFPL subprogram are as follows:

A. Select1: Select1 allows the operator to select circuit margin calculations for r-f links associated with the dual-band telemetry system, the triplexed telemetry system, the Babcock uplink system, the voice communications (uhf) system, and the C-band radar system. Two additional selections are provided for calculating noise figure from equivalent noise temperature, and for calculating receiver sensitivity from the system noise factor. The operator makes his selection by entering the number which corresponds to the desired menu item. The number is stored as Mode.

B. Sysselect: Subroutine Sysselect allows the operator to select a particular system or configuration associated with the major link selection. These selections are self-explanatory and are fully covered in the Program Operation section of this Chapter.

C. Dualspec: Subroutine Dualspec provides the default specifications for the dual-band telemetry system.

D. Trispec: Subroutine Trispec provides the default specifications for the triplexed telemetry system.

E. Cspec: Subroutine Cspec provides the default specifications for the C-band video downlink system.

F. Babspec: Subroutine Babspec provides the default specifications for the Babcock uplink system.

G. Comspec: Subroutine Comspec provides the default specifications for the uhf communications system.

   1. Dnspec: Subroutine Dnspec is called by Comspec if the uhf downlink margin is to be computed. Dnspec provides default values for parameters unique to the uplink configuration.

   2. Upspec: Subroutine Upspec is called to Comspec if the uhf uplink margin is to be computed. Upspec provides default values for parameters unique to the uplink configuration.

H. Beaconspec: Subroutine Beaconspec provides the default specifications for the selected airbore beacon (228, 302, 207, or DPN-66).

I. Radspec: Subroutine Radspec provides default values for the AN/FPS-16 radar system.

J. Manspec: Subroutine Manspec permits the operator to make manual entries of the r-f link parameter values. Self-explanatory CRT messages prompt the operator entries.

K. Compute: Subroutine Compute prompts the operator to enter the transmission range in nautical miles for the current pass. It then computes path loss and circuit margin using equations (9.4) and (9.2).

L. Skincomp: Subroutine Skincomp computes the r-f circuit margin for C-band radar skin track conditions. This requires the use of equation (9.9).

Program Operation

Upon entry in to the RFPL subroutine, the following CRT message is displayed on the CRT.

```
PLEASE SELECT THE DESIRED OPERATING MODE

1 = CIRCUIT MARGIN CALCULATIONS FOR MANUAL ENTRIES
2 = CIRCUIT MARGIN CALCULATIONS FOR DUAL BAND SYSTEM
3 = CIRCUIT MARGIN CALCULATIONS FOR TRIPLEXED SYSTEM
4 = CIRCUIT MARGIN CALCULATIONS FOR BABCOCK (COMM ANT) UPLINK
5 = CIRCUIT MARGIN CALCULATIONS FOR UHF VOICE LINKS
6 = CIRCUIT MARGIN CALCULATIONS FOR AN/FPS-16 RADAR
7 = NOISE FIGURE FROM EQUIVALENT NOISE TEMPERATURE
8 = RECEIVER SENSITIVITY FROM SYSTEM NOISE FIGURE
```

The operator responds by entering the proper number and pressing CONT. Immediately after pressing CONT, the output selection is displayed.

```
SELECT OUTPUT DEVICE

0 = CRT
1 = THERMAL PRINTER
2 = LINE PRINTER
```

Again, the operator responds by entering the number of the correct selection and CONT. If calculations are to be made for the dual band antenna, the next display will be:

```
MAKE SELECTION

1 = DUAL BAND RECEPTION WITH PREAMPS ON
2 = DUAL BAND RECEPTION WITH PREAMPS OFF
```

Followed by:

```
PLEASE INDICATE TYPE OF PCM SIGNAL

1 = FM/FM, PAM/FM, PDM/FM
2 = PCM/FM
```

If, instead, calculations are to be made for the triplexed system, the following message will be displayed:

MAKE SELECTION

1 = TRIPLEX UPLINK WITH 25 WATTS OUTPUT
2 = TRIPLEX UPLINK WITH 50 WATTS OUTPUT
3 = TRIPLEX UPLINK WITH 100 WATTS OUTPUT
4 = TRIPLEX L-BAND DOWNLINK WITH PREAMPS ON
5 = TRIPLEX L-BAND DOWNLINK WITH PREAMPS OFF
6 = TRIPLEX C-BAND TV DOWNLINK WITH PREAMPS ON

Followed by:

PLEASE INDICATE TYPE OF PCM SIGNAL

1 = FM/FM, PAM/FM, PDM/FM
2 = PCM/FM

If the calculations are to be made for the uhf system, then the message will appear as:

MAKE SELECTION

1 = UHF COMM UPLINK (12-FOOT PARABOLIC DISH)
2 = UHF COMM DOWNLINK (12-FOOT PARABOLIC DISH)
3 = UHF COMM UPLINK (AT-1097-GR)
4 = UHF COMM DOWNLINK (AT-1097-GR)

If the calculations are to be made for the AN/FPS-16 radar, then the message will appear as:

MAKE SELECTION

1 = C-BAND RADAR BEACON DOWNLINK (PARAMPS ON)
2 = C-BAND RADAR BEACON DOWNLINK (PARAMPS OFF)
3 = C-BAND RADAR UPLINK (BEACON)
4 = C-BAND RADAR SKIN TRACK (PARAMPS ON)
5 = C-BAND RADAR SKIN TRACK (PARAMPS OFF)

After the appropriate operator selection has been made, if the calculations involve airborne beacon specifications, the next message displayed is:

ENTER TYPE OF BEACON IN USE

1 = 228
2 = 302
3 = 207
4 = DPN-66

The program now branches to the appropriate subroutines where default values for the parameters associated with the selected system configuration are assigned to the appropriate working variables. Immediately thereafter, the default values are displayed on the CRT. If the operator accepts the default values then he simply presses CONT. If he wishes to change any one or more of the default values, he presses 1 and CONT.

A typical format of the default parameter display is

THE DEFAULT VALUES USED FOR THE DUAL BAND TELEMETRY SYSTEM ARE

AIRBORNE TRANSMITTER POWER	5.0 W
TRANSMITTING FREQUENCY	1480.5 MHZ
TRANSMITTING ANTENNA GAIN	0.0 DB
TRANSMITTING LINE LOSSES	0.0 DB
NOISE FIGURE (PREAMPS OFF)	9.0 DB
RECEIVING LINE LOSSES	7.9 DBM
RECEIVING ANTENNA GAIN	33.0 DB
MINIMUM ACCEPTABLE SIGNAL LEVEL	15.0 DB

COMPUTED SPECIFICATIONS ARE

TRANSMITTER POWER	37.0 DBMW
NOISE FACTOR AS RATIO	7.9
EQUIVALENT NOISE TEMPERATURE	2013.6 K
RECEIVER SENSITIVITY	108.6 DBM

IF SPECS OK PRESS CONTINUE, IF CHANGES ARE NEEDED PRESS 1 AND CONT

Similar types of displays will appear for any system and configuration selection made by the operator. After any changes have been made to the default parameters, the operator presses CONT and the program provides the following prompt message:

ENTER TRANSMISSION RANGE IN NAUTICAL MILES

The operator responds by keying in the range from the transmitting aircraft to the ground. Upon pressing CONT the program enters the compute mode and displays the results as:

	TRANSMISSION RANGE	125.0 N.MI.
+Pt	TRANSMITTED POWER	37.0 DBMW
+Gt	TRANSMITTING ANTENNA GAIN	0.0 DB
-Lt	TRANSMITTING LINE LOSS	0.0 DB
-Pl	PATH LOSS	-143.1 DBM
+Gr	RECEIVING ANTENNA GAIN	33.0 DB
-Lr	RECEIVER LINE LOSSES	-7.9 DB
+Rs	RECEIVER SENSITIVITY	108.6 DBMW
-Snr	MINIMUM ALLOWABLE SIGNAL LEVEL	-15.0 DB
Cm	CIRCUIT MARGIN	12.5 DBMW

If CONT is depressed at this point, the program will return to the range entry point to permit a second distance to be checked.

Display formats for all of the various modes and configurations are similar to that shown above for the dual-band system. In the event that a skin-track radar link is being checked, then the display formats are slightly different. A typical default parameter display for the skin track mode is:

THESE ARE THE DEFAULT SPECIFICATIONS FOR C-BAND SKIN TRACK

PERCENT POWER SELECTED	100.0 PERCENT
TRANSMITTING POWER FOR ABOVE PERCENT	1000.0 KW
TRANSMITTING LINE LOSS	3.5 DB.
RECEIVING LINE LOSS	2.0 DB
ANTENNA GAIN	42.5 DB
TRANSMITTED WAVELENGTH	7.2 CM
RADAR SYSTEM NOISE FIGURE	4.0 DB
RADAR RCVR NOISE BANDWIDTH	63.0 DBHZ
MINIMUM ACCEPTABLE SIGNAL LEVEL	20.0 DBW

IF SPECS OK PRESS CONTINUE, IF CHANGES ARE NEEDED PRESS 1 AND CONT

If changes must be made to any of the system parameters, the operator presses 1 and CONT, and the program sequentially steps through all of the input selections. If no entry is made at any entry point, the program retains the last value (or the default value if no changes have been entered for that parameter).

When the displayed system parameters are correct, pressing CONT will cause the next message to appear on the CRT.

ENTER THE TARGET RADAR CROSS SECTION

The operator must make the appropriate entry (for example, 15 dBm^2) and press CONT. The program then requests the target range.

ENTER TARGET RANGE IN NAUTICAL MILES

After these entries have been made, the program enters the compute mode and displays the results to the operator. A typical display is:

TRANSMISSION RANGE		125.0 N.MI.
+Pt	TRANSMITTED POWER	60.0 DBW
+2G	2 X ANTENNA GAIN	85.0 DB
-4R	4 X TARGET RANGE	-83.9 DBNM
-Ltr	COMBINED LINE LOSSES	-5.5 DB
+2Lam	2 X TRANSMITTER WAVELENGTH	14.5 DBCM
-Nf	OPERATING NOISE FIGURE	-4.0 DB
-Bn	RCVR NOISE BANDWIDTH	-63.0 DBHZ
+Sig	RADAR CROSS SECTION	15.0 DBM2
-Snr	MINIMUM ALLOWABLE SIGNAL LEVEL	-20.0 DBMW
Cm	CIRCUIT MARGIN	-1.9 DB

CHAPTER 10

SKYSCREEN PROGRAM

The skyscreen program is used to compute line-of-sight coverage patterns for specific antennas for which horizon-blockage data has been entered. Data is acquired by means of theodolite measurements taken at the antenna site, or, in the case of tracking antennas with remote video installed, from elevation readings taken at the operator's console. In taking data, the horizon profile is measured at each 1-degree increment of azimuth angle. When the necessary mislevel corrections have been made to the measured data, they are entered into the skyscreen program and stored on tape for future use in generating coverage pattern plots for targets at specified operating altitudes.

General Theory

Surface-To-Air Calculations

The skyscreen program uses the elevation blockage data for each of 360 azimuth angles to compute the geometric line-of-sight range at which optical or r-f energy will intersect a given altitude shell when the antenna is depressed to the terrain clearance point for that angle. The calculations are carried out using the simple geometry shown in figure 10.1.

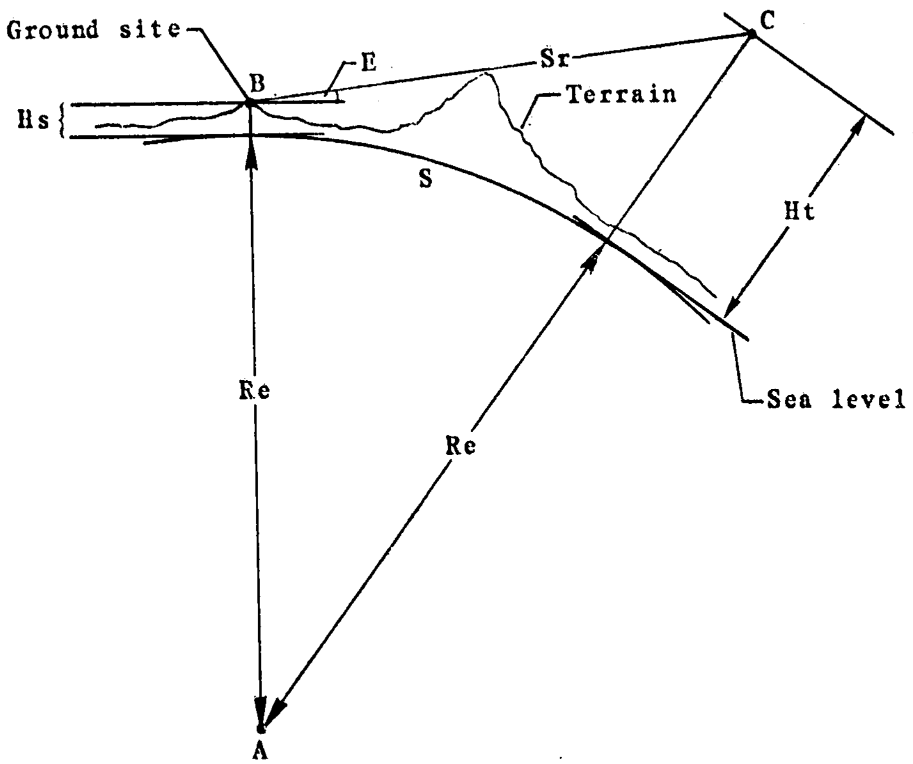

Figure 10.1

In figure 10.1, Ht is the selected target height, Hs is the elevation of the site above sea level, Re is the average radius of curvature, and E is the elevation clearance angle. If a ray were projected from the antenna site at an elevation angle E, it would intersect with the Ht shell at the point C. The slant range to the point of intersection is denoted by Sr, and the map distance (surface arc) between the tracking site and the maximum range point is denoted by S. The three opposite angles formed by the triangle legs Sr, Re + Ht, and Re + Hs are A, B, and C, respectively.

If the semimajor axis of the earth spheroid is denoted by a, the eccentricity by ε, and the latitude of the tracking site by μ, then the north-south (meridional) radius of curvature at the tracking site is given by

$$R = \frac{a(1 - \varepsilon^2)}{(1 - \varepsilon^2 \sin^2\mu)^{3/2}}, \tag{10.1}$$

and the east-west radius of curvature at the tracking site is given by

$$N = \frac{a}{(1 - \varepsilon^2 \sin^2\mu)^{1/2}}. \tag{10.2}$$

The average radius of curvature at the tracking site is found from

$$Re = (R \times N)^{1/2}. \tag{10.3}$$

Again consider the triangle ABC shown in figure 10.1. It is apparent that the angle B and E are related by

$$B = E + \pi/2. \tag{10.4}$$

The angle C is found from the law of sines as

$$C = \arcsin \frac{(Re + Hs)\ \sin B}{Re + Ht}, \tag{10.5}$$

and the angle A is found from

$$A = \pi - B - C. \tag{10.6}$$

The only remaining unknown is the slant range from the tracking site to the point of intersection of the elevation ray with the selected altitude shell. This is easily obtained by applying the law of sines,

$$Sr = \frac{(Re + Ht)\ \sin A}{\sin B}. \tag{10.7}$$

The surface arc distance (map distance) is found from the relation

$$S = Re \times A \tag{10.8}$$

In the baseline skyscreen program, the elevation clearance angles for a specific tracker are entered for each of 360-degrees of azimuth. The program then computes the map distance (S) and the slant-range distance (Sr) for each azimuth angle. A plot is then prepared for the selected tracker showing the line-of-sight coverage pattern for any designated target altitude.

If specified by the operator, the program will also call the gradient refraction subprogram to provide refraction adjustments to the slant range and map distance. This is accomplished by determining the true elevation and range values prior to computing the slant range and mapping distance. The theory of gradient refraction was presented in chapter 6.

Figure 10.2 shows a typical circular plot with both geometric and refraction-corrected coverage patterns.

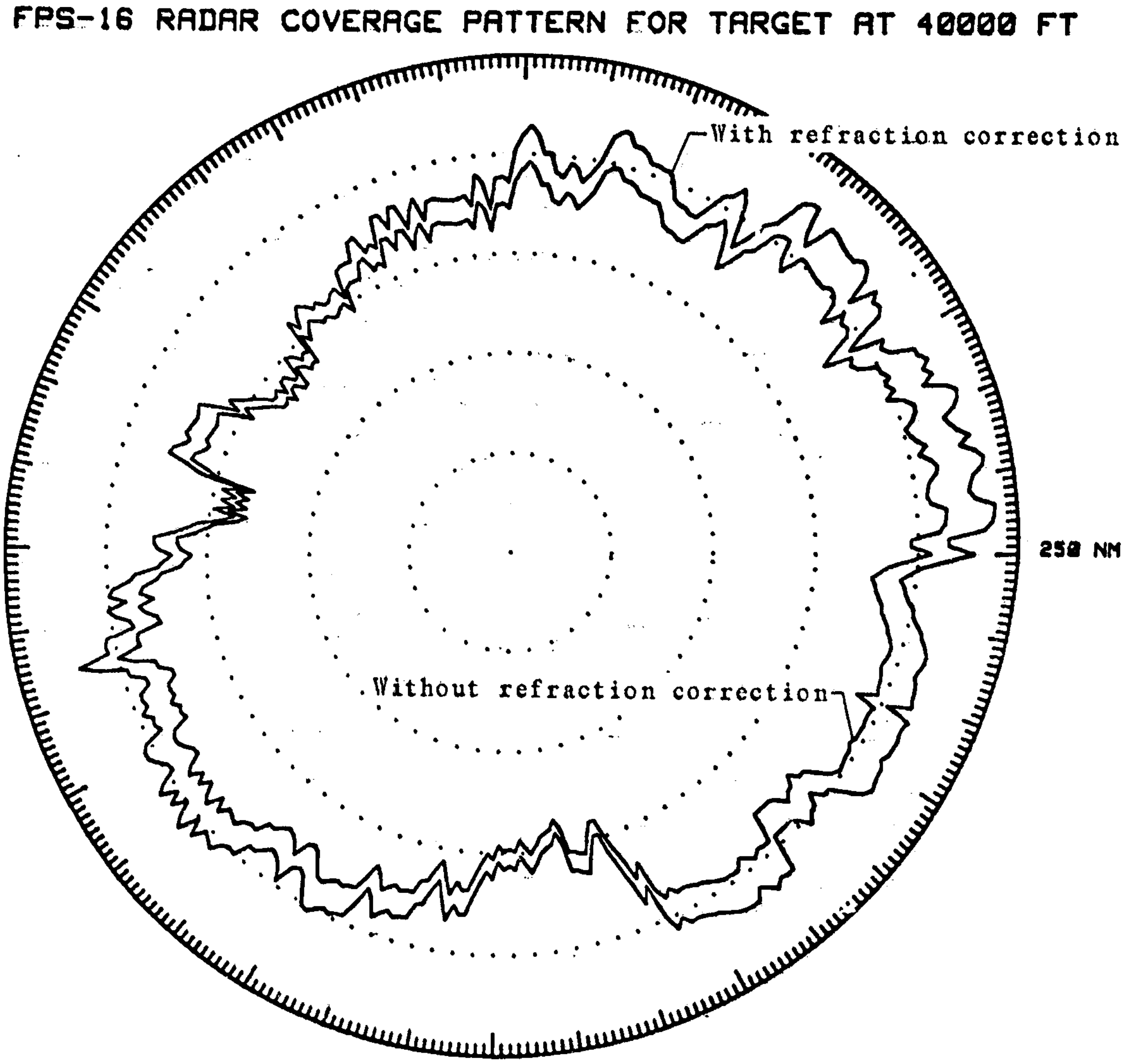

Figure 10.2

Figure 10.3 shows a plot of the elevation clearance angles versus azimuth angles for the AN/FPS-16 (34) radar at DFRC.

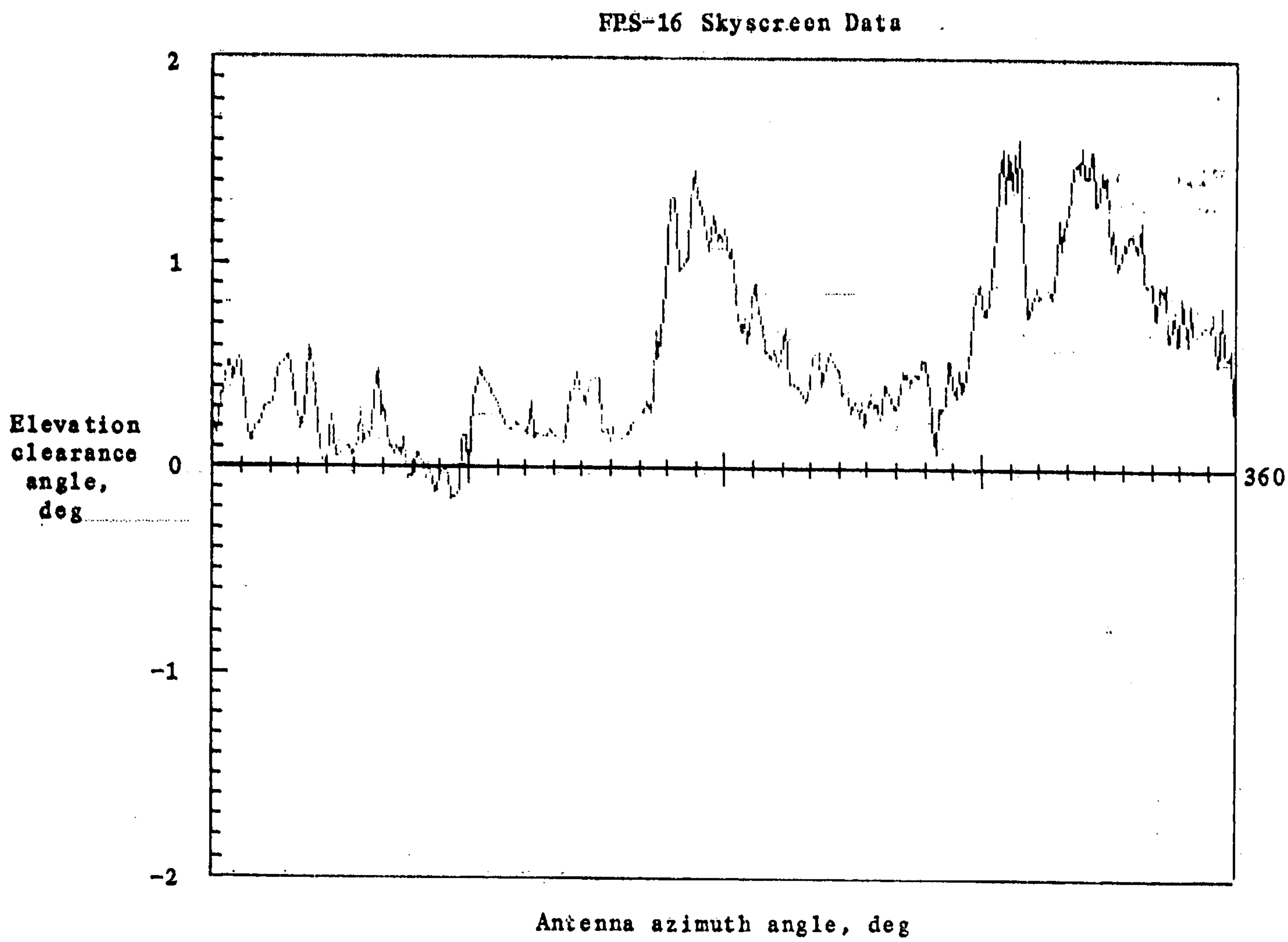

Figure 10.3

Air To Air Calculations

When an operation is conducted beyond line-of-sight communication range, it may be desirable to use an airborne repeater system to serve as a telemetry and communications link between the test vehicle and the ground control facilities. This requires certain additional calculations that will allow the test planners to compute a suitable station point for the airborne relay aircraft. Two possible mission scenarios exist in which this type of relay situation can be useful. The first is for missions conducted at normal flight altitudes but at extended ranges. The second involves tracking a low flying test aircraft at extended range, in which case terrain obstacles are a problem on both ends of the communications net.

High-Flying Test Aircraft Calculations

The geometry of the first scenario is provided in figure 10.4. To optimize the signal-to-noise ratios between the various transmitting and receiving units (tracking site, airborne relay, and target vehicle), it was decided that the relay aircraft should operate midway between the tracking station and the tracked vehicle so that S1 is approximately equal to S2. Obviously, since both the relay aircraft and the tracked vehicle are at considerably higher altitudes than that of the tracking site, it is simply necessary to compute minimum altitude at which the relay aircraft will have direct line-of-sight communications with the tracking site. This is accomplished as follows. First, the operator specifies the maximum anticipated distance between the test vehicle and the ground site. This distance (S) is entered by the operator. The geometry used to compute the position and minimum altitude for the tracker is the same as shown in figure 10.1. However, now the distance S is known and the altitude, Hr (altitude of the relay aircraft), must be determined.

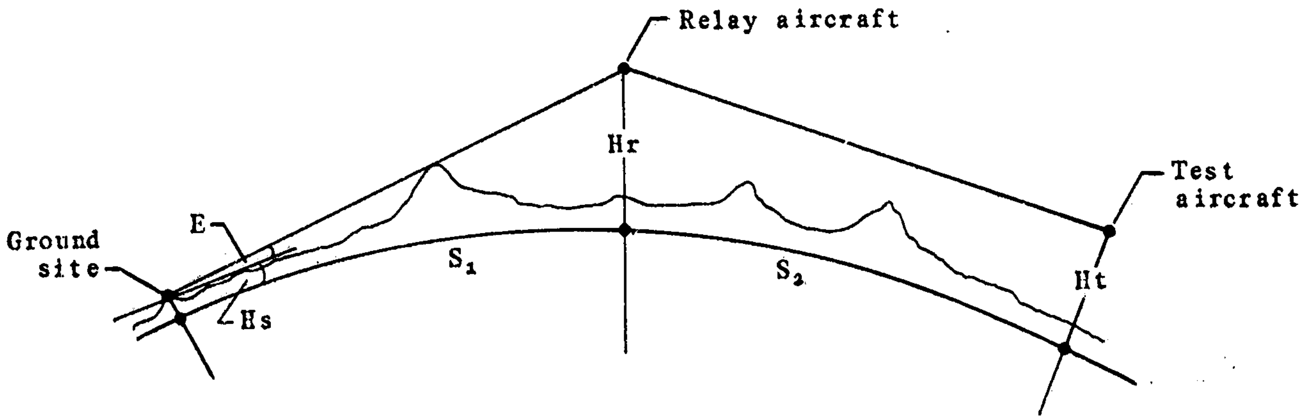

Figure 10.4

In this case the arc length S1 is one-half of the maximum anticipated operating distance S of the test aircraft, and the angle A (in radians) is simply

$$A = Re/S. \qquad (10.9)$$

For this calculation, the maximum obstruction angle (E) for the operating sector is found from the stored obstruction data. The program then computes B from equation (10.4) and C from equation (10.6). Knowing angles A, B, and C as well as Re and Hs, the minimum line-of-sight altitude (Hr) of the relay aircraft can be obtained from

$$Hr = \frac{\sin B\ (Re + Hs)}{\sin C} - Re. \qquad (10.10)$$

The slant range from the ground site to the relay aircraft is again found from equation (10.7)

Low-Flying Test Aircraft Calculations

When the test aircraft is operating at very low altitude, such as on terrain-following radar (TFR) missions, then it is necessary to consider both ends of the communications link. Here, one must first determine the worst case situation, such as a pass down a valley where a mountain range may lie between the test and relay aircraft. In this case the position of the relay aircraft is determined by noting the distance from the test aircraft's flight path to the highest obstruction measured in the sector from the test aircraft in which the transmissions to the relay aircraft will be made. Figure 10.5 shows the geometry of the two transmission links. The lowest altitude of the test aircraft is then entered, and the minimum line-of-sight elevation angle for the test aircraft to relay aircraft segment is determined by the procedure set forth for determining the blockage angle on the test aircraft link. Next, the operating sector from the ground station is entered and the highest stored operating sector from the ground station is entered and the highest stored obstruction angle for that sector is used as the obstruction angle for the site to relay link. The operator then enters the maximum surface distance (S) between the test aircraft's ground path and the ground site.

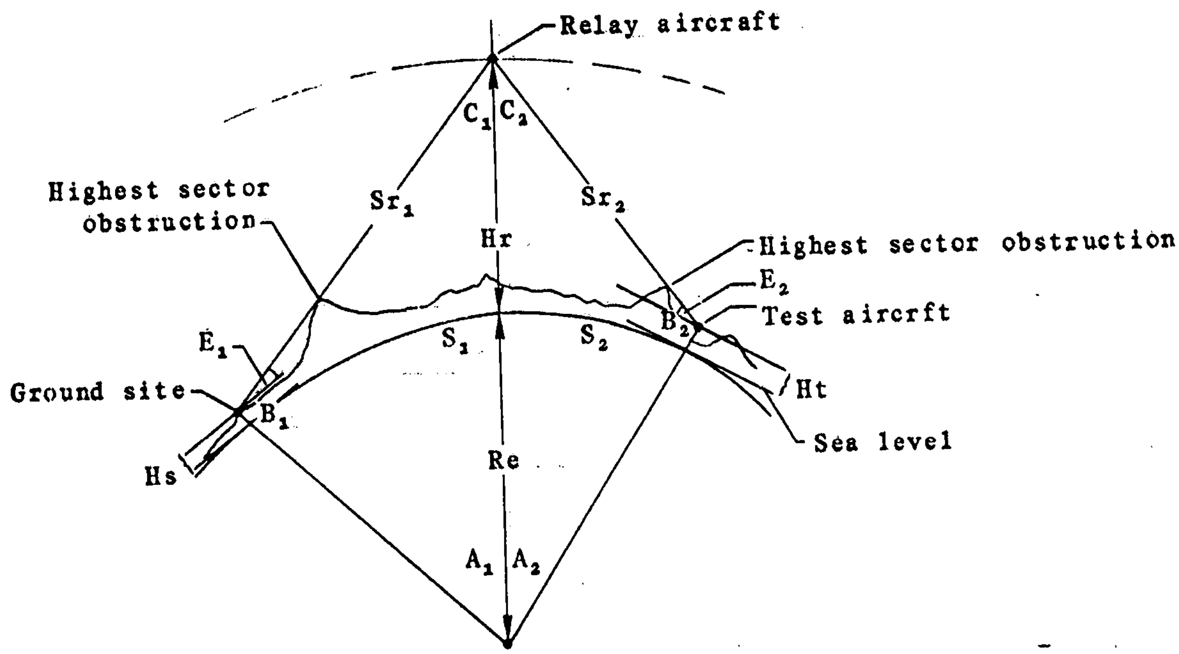

Figure 10.5

The blockage angle on the test aircraft link is determined by noting the geometry of figure 10.6. In this figure, D is the surface distance from the flight path of the aircraft to the highest obstruction in the sector in which the transmissions to the relay aircraft are to be made. The height of the obstruction is Hm and the height of the low flying target is Ht. It is first necessary to compute the angle B2. In figure 10.6, the known parameters are Re, Hm, Ht, and D. The leg Sr0 can be found from the law of cosines as

$$(Sr0)^2 = (Re + Hm)^2 + (Re + Ht)^2 - 2(Re + Hm)(Re + Ht)\cos(D/Re). \quad (10.11)$$

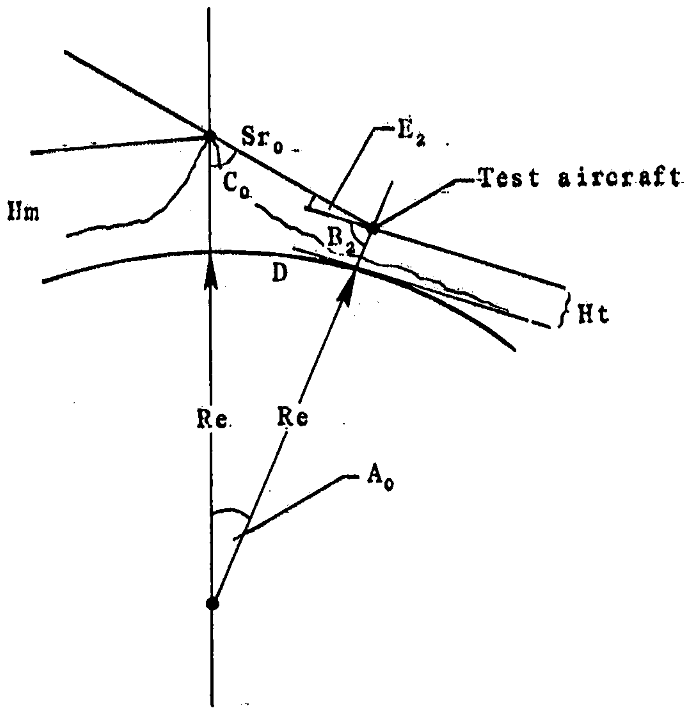

Figure 10.6

Knowing Sr, it is possible to find the angle B2 by application of the law of sines,

$$B2 = \arcsin\left[\frac{(Re + Hm)\sin a}{Sr}\right]. \qquad (10.12)$$

Returning to figure 10.5, it is apparent that the law of sines can be applied to equate functions of the unknown terms C1 and C2.

$$Re + hr = \frac{(Re + Hs)\sin B1}{\sin C1} = \frac{(Re + Ht)\sin B2}{\sin C2}. \qquad (10.13)$$

Grouping the known parameters Hs, Ht, B1, and B2 into terms K1 and K2 where

$$K1 = (Re + Hs)\sin B1, \qquad (10.14)$$

and

$$K2 = (Re + Hs)\sin B2 \qquad (10.15)$$

allows equation (10.13) to be simplified to

$$K1 \sin C2 = K2 \sin C1. \qquad (10.16)$$

It is also obvious from figure 10.5 that $A1 + B1 + C1 = \pi$, and $A2 + B2 + C2 = \pi$. Combining these two relations yields

$$A1 + A2 + B1 + B2 + C1 + C2 = 2\pi. \tag{10.17}$$

Recalling that

$$A1 + A2 = A = S/Re, \tag{10.18}$$

and knowing B1 and B2, it is now possible to find an expression for C2 as a function of C1.

$$C2 = [2\pi - S/Re - (B1 + B2)] - C1. \tag{10.19}$$

If K3 is substituted for the known quantity $[2\pi - S/Re - (B1 + B2)]$ in equation (10.19), and equations (10.19) and (10.16) are combined, a simplified expression may be written,

$$K1 \sin (K3 - C1) = K2 \sin C1. \tag{10.20}$$

Applying a standard double-angle formula to the left-hand side of equation (10.20) and rearranging terms yields

$$(K2/K1) \sin C1 = \sin K3 \cos C1 - \cos K3 \sin C1. \tag{10.21}$$

If the known terms in equation (10.21) are replaced with

$$P1 = K2/K1, \quad P2 = \sin K3, \quad \text{and} \quad P3 = \cos K3, \tag{10.22}$$

equation (10.21) may be rearranged to yield the following expression in which C1 is the only unknown:

$$P2 \cos C1 = (P1 + P3) \sin C1, \tag{10.23}$$

or

$$\frac{\sin C1}{\cos C1} = \frac{P2}{P1 + P3} = \tan C1. \tag{10.24}$$

Knowing C1 and B1, A1 can be simply found from the relation

$$A1 = \pi - B1 - C1, \tag{10.25}$$

and S1 can be found from

$$S1 = Re/A1. \tag{10.26}$$

A final calculation of hr is made using a rearranged form of equation (10.13),

$$hr = \left[(Re + Hs) \frac{\sin B1}{\sin C1}\right] - Re. \tag{10.27}$$

Thus, we have now determined the distance from the ground station to the point at which the relay plane should be stationed (S1) and the minimum altitude (hr) at which reception will be maintained with both the ground station and the test aircraft. Obviously, the fact that the relay aircraft must maintain some type of orbit requires that additional altitude be allowed for the off-point conditions. In the skyscreen program, the suggested minimum reception altitude for the relay aircraft is placed 5000 feet above the computed altitude.

Skyscreen Programs

The following descriptions cover the algorithms essential to carry out mathematical routines presented in this chapter.

Variable Names

Name	Description
A	Interior angle of figure 10.1
A1	Interior angle subtended by ground station to relay arc
Aa	Semimajor axis of reference spheroid
Az1	Azimuth at starting point of communications sector
Az2	Azimuth at ending point of communications sector
B	Elevation angle plus 90° as shown in figure 10.1
B1	Angle in A1-B1-C1 triangle in figure 10.5
C	Angle in A-B-C triangle of figure 10.1
C1	Angle in A1-B1-C1 triangle in figure 10.5
Conv	Conversion factor (radians to degrees)
D	Map distance to obstruction on test aircraft link
Den	Radical term in denominator of equations (10.1) and (10.2)
E1	Elevation angle of relay aircraft from ground station
E2	Elevation angle of relay aircraft from test aircraft
Elm	Maximum elevation angle for sector
H(N)	Elevation Array
Hm	Height of obstruction
Hr	Mainimum line-of-sight altitude of relay aircraft

Hs	Elevation of ground station
Ht	Altitude of test aircraft
I1	Integer value of sector starting azimuth
I2	Integer value of sector ending azimuth
J(M,N)	Array used to store elevation angles, ground distances and slant ranges
Mapdist	Spheroid distance from ground station to relay aircraft
Maxdist	Maximum coverage distance for all sector angles
Num	Numerator of equation (10.1)
Nz	N value at ground station as given by equation (10.2)
PI	π
Re	Average earth radius at ground station
Rz	R value at ground station as given by equation (10.1)
S	Distance to maximum coverage point for any azimuth angle
S1	Distance from ground station to relay aircraft in two link situation
Sinlat	Sine of ground station latitude
Sin2lat	Square of the sine of the ground station latitude
Sina	Sine of angle A in figure 10.1
Sinb	Sine of angle B in figure 10.1

Computational Algorithm

The essential algorithms used to compute the coverage ranges and relay aircraft positions are as follows:

A. Subroutine Skyscrl: Subroutine Skyscrl computes the overall coverage pattern around a specified tracker using stored terrain blockage data. Upon entry into the subroutine, step 4 allows the operator to select the desired ground station, and step 5 allows the output selection (CRT or plotter) to be made. At step 6, the program branches to Entry1 where the operator is prompted to enter the operating altitude (ft) of the target for which the skyscreen plot is desired. The format of the Entry1 display is provided in the Program Operation section.

In steps 8 and 9, the sin^2 value of the ground station latitude is computed and stored for use by the program. Steps 10 to 14 compute the two radii of curvature and the average radius of curvature at the ground station. In step 15 the radian mode is set and the radian to degree conversion factor is computed in step 16. In step 18 a FOR-NEXT loop is established which sequentially picks up the terrain blockage angle for each 360-degree of azimuth and performs calculations to determine the surface distance and slant range to the point where the elevation ray intersects with the Ht shell. The elevation angle, slant range, and surface distance are stored in a 360 by 2 dimensional array J. In step 30, the surface distance is compared with the maximum surface distance for this loop, and the maximum value is stored as Maxdist.

For correlation with the text, steps 8 to 14, implement equations (10.1) to (10.3), step 20 implements equation (10.4), step 22 implements equation (10.5), step 23 implements equation (10.6), step 25 implements equation (10.7), and step 26 implements equation (10.8).

```
 1.  Skyscr1:!
 2.  Htf=0
 3.  PRINT PAGE
 4.  GOSUB Radarsel
 5.  GOSUB Plotsel
 6.  GOSUB Entry1
 7.  DEG
 8.  Sinlat=SIN(Lat)
 9.  Sin2lat=Sinlat*Sinlat
10.  Num=Aa*(1-E2)
11.  Den=SQR(1-E2*Sin2lat)
12.  Rz=Num/Den**3
13.  Nz=Aa/Den
14.  Re=SQR(Rz*Nz)
15.  RAD
16.  Conv=360/(2*PI)
17.  Maxdist=0
18.  FOR N=1 TO 360
19.  E1=H(N)/Conv
20.  B=E1+PI/2
21.  Sinb=SIN(B)
22.  C=ASN((Re+Hs)*Sinb/(Re+Ht))
23.  A=PI-B-C
24.  Sina=SIN(A)
25.  Sr=(Re+Ht)*Sina/Sinb
26.  Mapdist=Re*A
27.  J(N,0)=E1*Conv
28.  J(N,1)=Sr
29.  J(N,2)=Mapdist
30.  IF Mapdist>Maxdist THEN Maxdist=Mapdist
31.  NEXT N
32.  DEG
33.  J(0,0)=Maxdist
34.  RETURN
```

B. Subroutine Skyscr2: Subroutine Skyscr2 computes the position and altitude of a relay aircraft used for communication with a high-flying, extended range target. After entry into the subroutine, the program branches to Subroutine Radarsel in which the operator is prompted to select one of the stored ground stations. This subroutine returns the altitude of the site (Hs) and its latitude (Lat). The latitude value is used in steps 4 to 10 to compute the two radii of curvature and the average radius of curvature at the ground station. At step 13, the program branches to Entry 2 where the operator is prompted to enter the maximum operating range of the test aircraft (n mi), and the starting and ending values of the operating sector as measured from the ground station (for example, 10 degrees to 40 degrees). In step 15, the maximum operating range is converted to meters. The two azimuth values are stored as Az1 and Az2, and in steps 16 to 18, they are set to the nearest integer values and adjusted for measurement across the 360/0 degree discontinuity. Note that the operating sector is always measured clockwise from Az1 to Az2. In step 19, a FOR-NEXT loop is established to find the maximum blockage angle in the selected operating sector. In step 24, the maximum value of elevation is stored in radian measure. The same procedure as used in Skyscr1 is implemented to compute the angle B1, and angle A1 is computed by implementing equation (10.9) in step 28. The angle C1 is found from equation (10.6) and, in step 30, the minimum reception altitude for the relay aircraft (Hr) is computed by implementing equation (10.10). Slant range is determined from equation (10.7) as before. The computed minimum reception altitude (Hr) is used to obtain a recommended relay altitude (Rr) by adding a constant 1500 meters. Steps 31 and 33 compute the same values in feet, and steps 34 and 35 round the values to the nearest hundred.

The formats of the Entry2 and Print2 displays are given in the Program Operation section.

```
 1.  Skyscr2:!
 2.  DEG
 3.  GOSUB Radarsel
 4.  Sinlat=SIN(Lat)
 5.  Sin2lat=Sinlat*Sinlat
 6.  Num=Aa*(1-E2)
 7.  Den=SQR(1-E2*Sin2lat)
 8.  Rz=Num/Den**3
 9.  Nz=Aa*Den
10.  Re=SQR(Rz*Nz)
11.  Conv=360/(2*PI)
12.  Skyscr2a:!
13.  GOSUB Entry2
14.  RAD
15.  Elm=-PI/2
16.  I1=INT(Az1)
17.  I2=INT(Az2)
18.  IF I2<I1 THEN I2=I2+360
19.  FOR N=I1 TO I2
20.  IF N>360 THEN N=N-360
21.  E1=H(N1)
```

```
22.  IF E1>E1m THEN E1m=E1
23.  NEXT N
24.  E1=E1m/Conv
25.  B1=E1+PI/2
26.  Sinb=SIN(B1)
27.  S1=S/2
28.  A1=S1/Re
29.  C1=PI-A1-B1
30.  Hr=(Sinb*(Re+Hs)/SIN(C1))-Re
31.  Hrf=Hr*3937/1200
32.  Rr=Hr+1500
33.  Rrf=Rr*3937/1200
34.  Rr=PROUND(Rr,2)
35.  Rrf=PROUND(Rrf,2)
36.  Sr1=(Re+Hr)*SIN(A1)/Sinb
37.  GOSUB Print2
38.  PAUSE
39.  GOTO Skyscr2a
```

C. Subroutine Skyscr3: Subroutine Skyscr3 is used to determine the optimum positioning for a relay aircraft used to maintain communications between a ground station and a low-flying test aircraft. Operator input parameters are the distance (S) to the test aircraft flight path point at which the blockage conditions are to be checked, the altitude of the test aircraft at that point (Ht), the altitude (Hm) of the highest obstruction in the test aircraft's transmitting sector, the distance (D) of the obstruction from the test aircraft flight path, and the azimuth angle Az from the ground station to the test aircraft point in question. The stored latitude (Lat) of the ground station is also used as a program input parameter.

Steps 2 to 11 are identical with those previously described for the Skyscr2 subroutine. At step 13 the program branches to subroutine Entry 3 to accept the operator inputs. At step 15, the angle A0 (shown on figure 10.6 is computed, and equation (10.11) is implemented in step 16. In steps 17 to 19 the sector azimuth angles are initialized to integer values and compensation is made if the sector crosses the 360/0 degree discontinuity. In step 20, a FOR-NEXT loop finds the highest stored elevation blockage angle for the ground station sector that lies 10 degrees on either side of the test aircraft's azimuth angle, and that value is stored as E1. In step 27, B1 is computed using equation (10.4), and, in step 28, B2 is computed using equation (10.12). The known terms K1, K2, K3, P1, P2, and P3 (eqs. (10.14) to (10.16) and (10.22)) are computed in steps 30 to 35. In steps 36 to 38, C1 is determined from equation (10.24), A1 is found from equation (10.25), and S1 is found from equation (10.26). In step 40, Hr is determined by implementation of equation (10.27), and the corresponding value in feet is computed in step 41. In steps 42 to 45, the recommended relay altitude in meters and feet is computed. Note that the recommended altitude is simply the minimum reception altitude plus 1500 meters. However, the recommended altitudes (in both meters and feet) are rounded to the nearest hundred. In step 46, Sr1 is determined by implementing equation (10.7)

At step 47, subroutine Print3 is called to printout the values for S1 (the distance to the relay point), Sr1 (the slant range distance to the relay aircraft), Hr (the minimum reception altitude), and Rr (the recommended relay aircraft altitude).

The formats of the Entry3 and Print3 displays are shown in the Program Operation section.

```
1.  Skyscr3:!
2.  DEG
3.  GOSUB Radarsel
4.  Sinlat=SIN(Lat)
5.  Sin2lat=Sinlat*Sinlat
6.  Num=Aa*(1-E2)
7.  Den=SQR(1-E2*Sin2lat)
8.  Rz=Num/Den]3
9.  Nz=Aa/Den
10. Re=SQR(Rz*Nz)
11. Conv=360/(2*PI)
12. Skyscr3a:!
13. GOSUB Entry3
14. Elm=-PI/2
15. A0=D/Re
16. Sr0=SQR((Re+Hm)**2+(Re+Ht)**2-2*(Re+Hm)*(Re+Ht)*COS(D/Re))
17. Az1=INT(Az-10)
18. Az2=INT(Az+10)
19. IF Az2<Az1 THEN Az2=Az2+360
20. FOR N=Az1 TO Az2
21. N1=N
22. IF N1>360 THEN N1=N1-360
23. E1=H(N1)
24. IF E1>Elm THEN Elm=E1
25. NEXT N
26. E1=Elm/Conv
27. B1=E1+PI/2
28. B2=ASN((Re+Hm)*SIN(A0)/Sr0)
29. IF Hm>Ht THEN B2=PI-B2
30. K1=(Re+Hs)*SIN(B1)
31. K2=(Re+Hs)*SIN(B2)
32. K3=2*PI-S/Re-(B1+B2)
33. P1=K2/K1
34. P2=SIN(K3)
35. P3=COS(K3)
36. C1=ATN(P2/(P1+P3))
37. A1=PI-B1-C1
38. S1=Re/A1
39. Sinb1=SIN(B1)
40. Hr=(Re+Hs)*Sinb1/Sin(C1)-Re
41. Hrf=Hr*3937/1200
42. Rr=Hr+1500
43. Rrf=Rr*3937/1200
44. Rr=PROUND(Rr,2)
```

```
45.  Rrf=PROUND(Rrf,2)
46.  Srl=(Ro+Ht)*Sin(A1)/Sinbl
47.  GOSUB Print3
48.  PAUSE
49.  GOTO Skyscr3a
```

Program Operation

The skyscreen program performs several separate functions. These are:

1. Entry and storage of terrain blockage data for specific locations.
2. Generation of skyscreen profiles for specified target latitudes
   a. Without refraction corrections
   b. With refraction corrections
3. Computation of relay aircraft location for long-range, high-altitude flights
4. Computation of relay aircraft location for long-range, low-altitude flights

Upon entry into the program, the following menu displays are provided for the operator.

A. Operating mode selection

SELECT OPERATING MODE

0 = KEYBOARD ENTRY OF BLOCKAGE ANGLES FOR SPECIFIC SITE
1 = GET BLOCKAGE DATA FROM A TAPE FILE
2 = STORE BLOCKAGE DATA ON A TAPE FILE
3 = PLOT ELEVATION BLOCKAGE ANGLES VS AZIMUTH ANGLES
4 = PRINTOUT OF ELEVATION BLOCKAGE ANGLES VS AZIMUTH ANGLES
5 = PLOT SKYSCREEN PATTERN WITHOUT REFRACTION CORRECTION
6 = PLOT SKYSCREEN PATTERN WITH REFRACTION CORRECTION
7 = PLOT SKYSCREEN PATTERN WITH AND WITHOUT REFRACTION CORRECTION
8 = COMPUTE POSITION OF RELAY AIRCRAFT FOR HI-ALITUTDE MISSION
9 = COMPUTE POSITION OF RELAY AIRCRAFT FOR LO-ALTITUDE MISSION

B. Entry of blockage angles from keyboard: In this mode, the program sequentially prompts the operator to enter the elevation blockage angle for each azimuth angle from 1 to 360 degrees. In the even the operator wishes to enter or correct a blockage angle for any specific azimuth, a negative number is entered, and the program branches to a loop which allows the operator to enter a specific azimuth angle. From that point the program will sequentially request entries for consecutive 1-degree azimuth increments.

ENTER EL ANGLE FOR AZ OF 1 DEGREE (ENTRY OF 888 ALLOWS NEW AZ ANGLE SEL)

ENTER EL ANGLE FOR AZ OF 2 DEGREES (ENTRY OF 888 ALLOWS NEW AZ ANGLE SEL)

⋮

C. Entry of data stored in tape file. This mode of program operation allows the operator to enter the name of the data file which is read into the elevation array, H(N). When the name has been entered, the operator presses CONT and the program enters the data from the selected tape file.

ENTER THE NAME OF THE TAPE FILE TO BE READ IN TO MEMORY

D. Storage of data on tape file: This mode of program operation allows the operator to store data entered from the keyboard into a specified tape file. The program requests the name of the file, and when CONT is depressed, stores the data from the H(N) array onto the specified tape file.

ENTER THE NAME OF THE TAPE FILE ON WHICH THE DATA IS TO BE STORED

E. Plot of elevation obstruction angles as a function of azimuth angle (fig. 10.3). The coordinates for the AN/FPS-16 radar, the communications building, and the main building are contained in the program. Others may be added by minor additions to the program subroutine Station.

SELECT GROUND STATION

0 = AN/FPS-16 (34)
1 = COMM BUILDING
2 = MAIN BUILDING

F. Skyscreen plots: When any of the skyscreen plots have been selected, the program requests the following information.

SELECT GROUND STATION

0 = AN/FPS-16 (34)
1 = COMM BUILDING
2 = MAIN BUILDING

The operator makes the appropriate selection and the program continues with the following prompt message:

ENTER THE OPERATING ALTITUDE FOR WHICH THE PLOT IS DESIRED (FT)

The last operator selection is requested with the following prompt message:

MAKE PLOTTER SELECTION

0 = CRT
1 = 9872A PLOTTER

When the appropriate operator selection has been made, the program starts the data plot on the selected display or output device.

G. Compute position of relay aircraft for high-altitude support mission: The following operator inputs are prompted by visual messages.

ENTER THE MAXIMUM GROUND STATION TO TEST AIRCRAFT RANGE (N MI)

ENTER THE OPERATING SECTOR AS AZ1, AZ2 (CLOCKWISE FM AZ1 TO AZ2)

The operator makes the two entries sequentially as requested. The first entry is made in nautical miles, and the second entries are made in degrees (for example 30,60 to represent the sector from 30 to 60 degrees azimuth as measured from the ground station). The program computes the minimum line-of-sight reception point for the relay aircraft based on the highest blockage angle in the operating sector. A 1500 meter margin is added to the computed minimum reception altitude, and the results are displayed as:

DISTANCE TO RELAY AIRCRAFT:	125 N MI	
SL RANGE TO RELAY AIRCRAFT:	231,711 METERS	
MINIMUM RECEPTION ALTITUDE:	6,971 METERS	(22,871 FT)
RECOMMENDED RELAY ALTITUDE:	8,500 METERS	(27,800 FT)

H. Compute distance of relay aircraft for low-altitude support mission: In this case the operator is prompted to make the following entries. Note that this program returns values for specific points which may be in question. For example, if the route of the test aircraft were to pass down a valley in which communications might be blocked by one or more mountains along the test route, the following inputs would be made for each point in question. The operator would respond to the various prompt messages shown below by sequentially entering the distance (n mi) from the ground station to the test aircraft, the azimuth angle (deg) from the ground station to the test aircraft, the altitude of the test aircraft (ft), elevation of the mountain or other obstruction along the test routine (ft), and distance of the obstruction from the test route as measured along the line from the test aircraft to the ground station (n mi) would be entered sequentially as indicated by prompt messages shown below.

ENTER THE DIST FROM THE GROUND STA TO TEST AIRCRAFT POSITION (N MI)

ENTER THE AZ ANGLE FROM THE GROUND STA TO POINT IN QUESTION (DEG)

ENTER THE ALTITUDE OF THE TEST AIRCRAFT (FT)

ENTER THE ELEV OF THE OBSTRUCTION ALONG THE TEST ROUTE (FT)

ENTER THE MAP DISTANCE OF THE OBSTRUCTION FROM THE TEST ROUTE (N MI)

The program will compute the position and minimum reception altitude for the relay aircraft based on the two line-of-sight links shown in figure 10.5. It will display to the operator:

DISTANCE TO RELAY AIRCRAFT:	175 N MI	
SL RANGE TO RELAY AIRCRAFT:	323,526 METERS	
MINIMUM RECEPTION ALTITUDE:	12,147 METERS	(39,854 FT)
RECOMMENDED RELAY ALTITUDE:	13,600 METERS	(44,800 FT)

The same procedure should be repeated for any points along the test aircraft route where blockage is anticipated. The recommended relay altitude for the worst case condition should be used.

APPENDIX

A CLOSED-FORM QUARTIC SOLUTION

The following is a procedure known as the Descartes technique for solving a fourth-degree polynomial equation. The steps given below closely parallel those presented in appendix 3 to reference 6. This procedure is implemented in subroutine Quartic, which is called by both the Lagrange and GMD off-spheroid coordinate determination programs.

Given an equation in the form

$$Ax^4 + Bx^3 + Cx^2 + Dx + E = 0, \tag{A.1}$$

it is possible to divide by A and obtain a new equation,

$$x^4 + B'x^3 + C'x^2 + D'x + E' = 0. \tag{A.2}$$

Equation (A.2) is then transformed into a reduced quartic equation (an equation in which the cubic term is eliminated) by making the substitutions

$$P = 6h^2 + 3B'h + C', \tag{A.3}$$

$$Q = 4h^3 + 3B'h^2 + 2C'h + D', \tag{A.4}$$

$$R = h^4 + B'h^3 + C'h^2 + D'h + E', \tag{A.5}$$

$$x = y + h, \tag{A.6}$$

$$h = -\frac{B'}{4}. \tag{A.7}$$

Substituting equations (A.3) to (A.7) in equation (A.2) yields the reduced quartic

$$y^4 + Py^2 + Qy + R = 0. \tag{A.8}$$

Now, by making the additional substitutions

$$f = \frac{1}{3}\left[3(P^2 - 4R) - 4P^2\right], \tag{A.9}$$

$$g = \frac{1}{27}\left[16P^3 - 18P(P^2 - 4R) - 27Q^2\right], \tag{A.10}$$

$$Z = t - s, \tag{A.11}$$

$$s = -\frac{2}{3}P, \tag{A.12}$$

a cubic resolvent may be obtained as

$$t^3 + 2Pt^2 + (P^2 - 4R)t - Q^2 = 0 \tag{A.13}$$

or

$$Z^3 + fZ \pm g = 0. \tag{A.14}$$

In the Descartes technique, a branch indicator Δ is computed by the relation

$$\Delta = \frac{f^3}{27} + \frac{g^2}{4}. \tag{A.15}$$

When $\Delta > 0$, the roots of the cubic resolvent are found from

$$Z_1 = \left[-\frac{g}{2} + (\Delta)^{1/2}\right]^{2/3} + \left[-\frac{g}{2} - (\Delta)^{1/2}\right]^{2/3}, \tag{A.16}$$

Z_2 = complex

Z_3 = complex

When $\Delta = 0$ the roots of the cubic resolvent are found from

$$Z_1 = 2(-g/2)^{3/2} \tag{A.17}$$

$$Z_2 = 2(g/2)^{3/2} \tag{A.18}$$

$$Z_3 = 2(g/2)^{3/2}. \tag{A.19}$$

And, when $\Delta < 0$, the roots of the cubic resolvent are found from

$$Z_1 = E_0\cos(\gamma/3) \tag{A.20}$$

$$Z_2 = E_0\cos(\gamma/3 + 2\pi/3) \tag{A.21}$$

$$Z_3 = E_0\cos(\gamma/3 + 4\pi/3) \tag{A.22}$$

where

$$E_0 = 2(-f/3)^{1/2} \quad \text{and} \quad \gamma = \arccos\left[-\frac{g}{2(-f^3/27)^{1/2}}\right]. \tag{A.23}$$

It is now possible to find the critical root (R') of equation (A.14), which is the maximum real number of

$$R' = \text{max real root } [Z_1 + s,\ Z_2 + s,\ Z_3 + s]. \tag{A.24}$$

A positive real root will always be found.

Once R' is obtained, two new parameters ξ and β can be found as

$$\xi = \frac{1}{2}\left[P + R' - Q/(R')^{1/2}\right] \tag{A.25}$$

and

$$\beta = \frac{1}{2}\left[P + R' + Q/(R')^{1/2}\right]. \tag{A.26}$$

It is now possible to factor equation (A.8) into two roots,

$$(y \;\; + y(R')^{1/2} + \xi)(y \;\; - y(R')^{1/2} + \beta) = 0. \tag{A.27}$$

The solution of the two quadratics given in equation (A.27) yields four root values for y. Thus the roots of equation (A.1) are

$$\begin{aligned} x_1 &= y_1 + h \\ x_2 &= y_2 + h \\ x_3 &= y_3 + h \\ x_4 &= y_4 + h \end{aligned} \tag{A.28}$$

In both the Hedgley and the GMD solutions, only the real roots are meaningful, and in each case two roots will be real and two will be complex. In the Hedgley solution, where the quartic solution is carried out to obtain a value of the Lagrange multiplier α, the proper root is found to be the one for which the value of target altitude is minimized. In the GMD solution, the correct root is the one having the same sign as the value of r.

Variable Names

Name	Description
Alph	The root used in the Lagrange multiplier solution
Delta	Δ in equation (A.15)
E0	E_0 in equation (A.23)
F2	f^2
F3	f^3
Ff1	f in equation (A.9)
Gam	γ in equations (A.20) to (A.23)

Gg1	g in equation (A.10)
H1	h in equation (A.7)
H2	h^2
H3	h^3
I1flg	1 if real roots are to be computed from Radic1
I2flg	1 if real roots are to be computed from Radic2
M1	ξ
N1	β
P1	P in equation (A.3)
P2	P^2
P3	P^3
Q1	Q in equation (A.4)
Q2	Q^2
Radic1	Generalized radical term used to solve equation (A.16) and (A.27)
Radic2	Generalized radical term used to solve equation (A.16) and (A.27)
Rp	R' term in equation (A.24)
S1	s term in equation (A.12)
Sign1	Sign of the value of the first radical in equation (A.16)
Sign2	Sign of the value of the second radical in equation (A.16)
Sqrad1	Square root of Radic1
Sqrad2	Square root of Radic2
Sqrdel	Square root of Δ
Sqrp	Square root of R'
Termb	B' term in equation (A.2)
Termc	C' term in equation (A.2)
Termd	D' term in equation (A.2)

Terme	E' term in equation (A.2)
X1	First root of equation (A.1)
X2	Second root of equation (A.1)
X3	Third root of equation (A.1)
X4	Fourth root of equation (A.1)
Zz1	First root of equation (A.14)
Zz2	Second root of equation (A.14)
Zz3	Third root of equation (A.14)

Algorithm

Subroutine Quartic is called in both the Hedgley and the GMD computations of off-spheroid latitude and altitude. The subroutine receives precomputed values of B', C', D', and E' (Termb, Termc, Termd, and Terme) from the main program. Equation (A.7) is implemented in step 2. Steps 3 and 4 form the power terms for equations (A.3) to (A.5), and steps 5 to 7 directly implement equations (A.3) to (A.5). Next, the power terms needed by equations (A.9) and (A.10) are computed in steps 8 to 10, and the value of s in equation (A.12) is computed in step 11. Ff1 and Gg1 correspond to the f and g terms in equations (A.9) and (A.10), and these equations are implemented in steps 12 and 13.

In steps 14 to 16, a value is computed for Δ by implementing equation (A.15). Zz1, Zz2, and Zz3, the three roots of equation (A.14), are initialized to 0 in step 18. When the value of Δ is positive, the subroutine computes values for Z starting at step 21 (Deltaplus). Steps 21 through 34 implement equation (A.16), which applies when the value of Δ is positive. If the value of Δ is 0, the program branches to Delta0 (step 35) and computes the values for Zz1, Zz2, and Zz3 using equations (A.17) to (A.19). If the computed value for Δ is negative, the program branches to Deltaminus (step 47) and computes the three roots of equation (A.14) using the trigonometric solution shown in equations (A.20) to (A.22). Regardless of which of the three solutions for the three roots of equation (A.14) is used, at least one positive real root will always be found. The program branches from each of the three root-finding sections to step 53 (Zplus) and determines the critical root (the maximum real root).

Using the critical root (R'), the values of ξ and β are found by implementation of equations (A.25) and (A.26) in steps 59 and 60. These values are the right-hand members of the two factors of equation (A.8) shown in equation (A.27). Since R', ξ, and β are all known, it is possible to solve each of the two bracketed terms in equation (A.27) by means of the quadratic equation. This yields the four y roots which can then be used in equations (A.28) to determine the four roots of equation (A.1). Step 63 forms the radical term of the quadratic equation for the first bracketed term of equation (A.27). If Radic1 is positive, indicating that the two values of y will be found by applying equation (A.28) (steps 68 and 69). IF Radic1 is negative, indicating that the roots are imaginary, the program jumps to step 70 to find the roots of the second bracketed term in equation (A.27). The same procedure is again followed and, if the radical (Radic2) is positive, two real values of x are computed. In the off-spheroid programs, two roots will always be real and two will always be imaginary. (If Radic1 is positive, Radic2 will be negative, or the reverse will be true.) Since only the real roots are meaningful for this solution, the imaginary roots are not computed. However, if the imaginary roots should be needed for some reason, they could easily be found by evaluating the negative radical.

```
 1.  Quartic:!
 2.  H1=-Termb/4
 3.  H2=H1*H1
 4.  H3=H1*H2
 5.  P1=6*H2+3*Termb*H1+Termc
 6.  Q1=4*H3+3*Termb*H2+2*Termc*H1+Termd
 7.  R1=H2*H2+Termb*H3+Termc*H2+Termd*H1+Terme
 8.  P2=P1*P1
 9.  P3=P1*P2
10.  Q2=Q1*Q1
11.  S1=-(2*P1)/3
12.  Ff1=(3*(P2-4*R1)-4*P2)/3
13.  Gg1=(16*P3-18*P1*(P2-4*R1)-27*Q2)/27
14.  F2=Ff1*Ff1
15.  F3=F2*Ff1
16.  Delta=F3/27+Gg1*Gg1/4
17.  Sign1=Sign2=1
18.  Zz1=Zz2=Zz3=0
19.  IF Delta=0 THEN Delta0
20.  IF Delta<0 THEN Deltaminus
21.  Deltaplus:!
22.  Sqrdel=SQR(Delta)
23.  Radic1=-Gg1/2+Sqrdel
24.  Radic2=-Gg1/2-Sqrdel
25.  IF Radic1>=0 THEN 28
26.  Sign1=-1
27.  Radic1=-Radic1
28.  IF Radic2>=0 THEN 31
29.  Sign2=-1
30.  Radic2=-Radic2
31.  Zz1=Sign1*Radic1**(1/3)+Sign2*Radic2**(1/3)
32.  Zz2=-1E99
33.  Zz3=-1E99
34.  GOTO Zplus
35.  Delta0:!
36.  Radic1=-Gg1/2
37.  Radic2=Gg1/2
38.  IF Radic1>0 THEN 41
39.  Sign1=-1
40.  Radic1=-Radic1
41.  IF Radic2>0 THEN 44
42.  Sign2=-1
```

```
43. Radic2=-Radic2
44. Zz1=2*Sign1*Radic1**(1/3)
45. Zz2=Zz3=Sign2*Radic2**(1/3)
46. GOTO Zplus
47. Deltaminus:!
48. E0=2*SQR(-Ff1/3)
49. Gam=ACS(Gg1/(2*SQR(-F3/27)))
50. Zz1=E0*COS(Gam/3)
51. Zz2=E0*COS(Gam/3+120)
52. Zz3=E0*COS(Gam/3+240)
53. Zplus:!
54. Rp=MAX(Zz1+S1,Zz2+S1,Zz3+S1)
55. IF Rp<0 THEN Rp=1E99
56. Sqrp=SQR(Rp)
57. Qdr=Q1/Sqrp
58. I1flg=I2flg=0
59. M1=(P1+Rp-Qdr)/2
60. N1=(P1+Rp+Qdr)/2
61. X1=X2=X3=X4=0
62. I1flg=I2flg=0
63. Radic1=Rp-4*M1
64. IF Radic1>=0 THEN 67
65. I1flg=1
66. GOTO 70
67. Sqrad1=SQR(Radic1)
68. X1=(-Sqrp+Sqrad1)/2+H1
69. X2=(-Sqrp-Sqrad1)/2+H1
70. Radic2=Rp-4*N1
71. IF Radic2>=0 THEN 74
72. I2flg=1
73. GOTO 77
74. Sqrad2=SQR(Radic2)
75. X3=(Sqrp+Sqrad2)/2+H1
76. X4=(Sqrp-Sqrad2)/2+H1
77. IF I1flg=1 THEN 81
78. IF I2flg=1 THEN 83
79. Alph=MAX(X1,X2,X3,X4)
80. RETURN
81. Alph=MAX(X3,X4)
82. RETURN
83. Alph=MAX(X1,X2)
84. RETURN
```

There are no operating instructions for this subroutine since it never receives operator inputs. It is called from both the Lagrange and GMD subprograms which provide the necessary input parameters, and it returns the necessary roots to the calling subprograms.

REFERENCES

1. Thomas, Paul D.: Conformal Projections in Geodesy and Cartography. Special Publication No. 251, U.S. Dept. of Commerce, 1952.

2. Clarke, A. R.: Geodesy. Oxford University Press, 1880.

3. Bomford, G.: Geodesy. Third ed. Oxford University Press, c. 1971.

4. Plane Coordinate Intersection Tables: Nevada. Special Publication No. 337, U.S. Dept. of Commerce, Coast and Geodetic Survey, 1955.

5. Hedgley, David R., Jr.: An Exact Transformation From Geocentric to Geodetic Coordinates for Nonzero Altitudes. NASA TR R-458, 1976.

6. Escobal, Pedro Ramon: Methods of Orbit Determination. John Wiley & Sons, Inc., c. 1965. (Reprinted by Robert E. Krieger Publishing Co., 1976.)

7. Bowring, B. R.: Transformation from spatial to geographical coordinates. Survey Review (Tolworth, England), vol. 23, no. 181, pp. 323-327.

8. Parish, O.O., and Putnam, T. W.: Equations for the Determination of Humidity From Dewpoint and Psychrometric Data. NASA TN D-8401, 1977.

9. Bean, B. R., and Dutton, E. J.: Radio Meteorology. Dover Publications, Inc., 1963.

10. Lear, William M.: Computing Atmospheric Scale Height for Refraction Corrections. NASA JSC Internal Note No. 80-FM-16, 1980.

11. Lear, William M.: Refraction Corrections for an Exponential Atmosphere. NASA JSC Internal Note No. 75-FM-60, 1975.