

NOAA Technical Report NOS 114
Charting and Geodetic Services Series CGS 7



Coordinate Conversion for Hydrographic Surveying

Richard P. Floyd

National Charting Research
and Development Laboratory

Rockville, MD
December 1985

**U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Ocean Service**

NOAA Technical Report NOS 114
Charting and Geodetic Services Series CGS 7



Coordinate Conversion for Hydrographic Surveying

Richard P. Floyd

National Charting Research
and Development Laboratory

Rockville, MD

December 1985

Reprinted August 1986

Reprinted August 1987

Reprinted November 1994

U.S. DEPARTMENT OF COMMERCE

Malcolm Baldrige, Secretary

National Oceanic and Atmospheric Administration

Anthony J. Calio, Administrator

National Ocean Service

Paul M. Wolff, Assistant Administrator

Charting and Geodetic Services

R. Adm. John D. Bossler, Director

Mention of a commercial company or product does not constitute an endorsement by the U.S. Government. Use for publicity or advertizing purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

CONTENTS

Acknowledgement	iv
Abstract	1
Introduction	1
Background	1
Scope	2
Terminology	3
Planar coordinates and their reference systems	3
Coordinates and origins	3
Scale	4
State Plane Coordinate System	5
Accuracy	6
Exact coordinate conversion	6
Practical coordinate conversion accuracies	6
Coordinate conversion procedures	7
Coordinate transformation algorithms	8
Isometric latitude	8
Inputs and outputs	9
Symbology	9
Normal Mercator projection	10
Transverse Mercator projection	16
Oblique Mercator projection	24
Lambert conformal conic projection	33
Polyconic projection	41
Azimuthal equidistant projection	49
Datum transformation	56
Appendices	
A. Ellipsoid parameters	63
B. State Plane Coordinate System of 1927 projection parameters	64
C. XYZ coordinate translation parameters	72
D. FORTRAN subroutines	75
E. Test points	105
References	119

ACKNOWLEDGEMENT

I thank Ruby L. Becker, of the National Charting Research and Development Laboratory, for coding the subroutines, helping to debug them, and assisting me in proofreading the manuscript. Her work was invaluable in making this an accurate document.

COORDINATE CONVERSION FOR HYDROGRAPHIC SURVEYING

Richard P. Floyd
National Charting Research and Development Laboratory
National Ocean Service, NOAA
Rockville, Maryland

ABSTRACT: Hydrographic survey positional data are processed, reported, and plotted using a variety of coordinate systems. Geodetic coordinate systems (latitude and longitude) provide worldwide coverage on reference surfaces that closely approximate the physical Earth. They form a basis upon which planar coordinate systems are founded. Planar systems are used extensively for their conceptual simplicity and their cartographic applications.

Information from various sources must be referenced to a common coordinate system before it can be related. Coordinate conversions, involving one or more coordinate transformations, are used to reduce coordinates to a common system. This report contains a general procedure for performing a coordinate conversion and detailed algorithms for making coordinate transformations. Explanations of various planar projections and limitations on their use are given.

INTRODUCTION

Background

Plotting soundings and features to allow comparison of historical with contemporary survey data is an important aspect of hydrographic surveying. In order to accomplish these tasks, it is necessary to reduce the positions associated with the information to a common frame of reference. Positional reference systems can be categorized into three broad groups: geodetic (also called geographic) in which coordinates are given in terms of a curvilinear lattice of latitudes and longitudes; planar, in which coordinates are given in terms of a rectilinear lattice of x's and y's, northings and eastings, or latitudes and departures; and space, in which coordinates are given in terms of a Cartesian system of X's, Y's, and Z's.

The measurements obtained to determine positions are made relative to the physical Earth, as characterized by its topography and by an undulating, semiphysical surface called the geoid, while the computations required for determining position must be based on a mathematically definable reference system. For geodetic coordinates, the mathematical reference system is the surface of an ellipsoid of specified size and shape, oriented to the surface of the Earth in a manner defined by the geodetic datum. Latitude and longitude are determined by projecting the point in question from its physical location to the ellipsoid, along a line normal to the ellipsoid. Planar coordinates are determined similarly; i.e., points are projected from their physical location onto a mathematically defined reference plane or developable

surface. (A developable surface is one having curvature in only one direction, such as a cone or cylinder, which can be "rolled out" with no angular or linear distortion into a plane.) Usually, points are projected from their physical location onto a geodetic surface, then projected from the geodetic surface to the plane or developable surface. Such a double projection may not be immediately apparent. Space coordinates depend only on the location of the coordinate origin and the orientation of the coordinate axes. Points are not projected onto a surface, as they are in geodetic and planar reference systems. Usually the space coordinate origin coincides with the center of a conventional ellipsoid.

The ellipsoid closely approximates the geoidal surface of the Earth. Thus, there is little difference between angles and distances measured on the topographic surface of the Earth and their geodetic counterparts represented on the ellipsoid. Any distortions are caused principally by irregularities in the gravity field as reflected in the geoid. Similarly, distortions between measured quantities and their counterparts in a spatial coordinate system are caused primarily by geoidal undulations. On the other hand, planar coordinates are tied to a frame of reference that is further from physical reality. Angles and distances measured on the Earth acquire greater distortion when represented in a planar system than when represented in a geodetic or a spatial system.

This discussion focuses on the following important concepts: (1) Distortions between measured and projected angles and distances can be mathematically accounted for regardless of the reference system employed, but distortions in a planar system are of greater magnitude and require more lengthy computations. (2) The ellipsoid is often used as an intermediate surface when projecting points from the physical surface of the Earth to a plane surface.

Historical hydrographic positional data might be referenced to one of several geodetic datums or to a great variety of planar reference systems. Coordinate conversions must be employed as required to represent all positions on a common basis before the information associated with the positions can be compared.

Scope

An automated hydrographic data acquisition and processing system must be capable of a variety of tasks required by the surveyor. Coordinate conversions is one such task. In converting coordinates from one reference system to another, one or more operations called "coordinate transformations" are required. The algorithms provided in this report are for those transformations. Subroutines coded from the algorithms, having both cartographic and survey applications, would lie at the lowest level of programming in an automated system. Transformation algorithms are provided for differing geodetic datums and for the following mapping projections:

1. Normal Mercator
2. Transverse Mercator
3. Oblique Mercator
4. Lambert conformal conic
5. Polyconic
6. Azimuthal equidistant

Terminology

Coordinate conversion is used in a general sense in this report to denote the process of changing the coordinates of a position represented in one reference system to the coordinates of that position represented in any other reference system. Coordinate conversions involve one or more coordinate transformations.

Coordinate transformation applies specifically to the process of converting geodetic coordinates to planar coordinates based on the same ellipsoid (forward transformation), converting planar coordinates to geodetic coordinates based on the same ellipsoid (inverse transformation), and converting geodetic coordinates to geodetic coordinates based on a different ellipsoid (datum transformation). Transformations are basic operations in performing the more general coordinate conversions.

A mapping projection, or simply projection, is a system whereby geodetic coordinates and planar coordinates are related with a one-to-one correspondence. (Note that the term "projection" applies here to mappings between mathematical surfaces, not to the projection of a point on the physical surface of the Earth to a mathematical surface.) Mapping projections are defined by specifying certain conditions that must be met. When a projection is referred by name, those conditions are implied. For example, in the transverse Mercator projection, angles between infinitesimal line segments are preserved, and scale is held constant along a selected meridian. Mapping projections are further defined by specifying certain projection parameters, thereby orienting the plane or developable surface to the ellipsoid. For instance, a specific transverse Mercator projection is defined by specifying the central meridian at 90° west longitude, where scale factor equals 1 (exactly). Most precisely, a mapping projection is defined by its mapping equations, which are used to make the one-to-one correspondence between coordinates. Mapping equations could be left in closed, exact form, but approximations must be incorporated to enable the computation of numerical results. Different mathematicians have used different approximations, yielding slightly different results (even though a one-to-one correspondence is maintained using a particular set of equations).

PLANAR COORDINATES AND THEIR REFERENCE SYSTEMS

Coordinates and Origins

Two types of planar coordinates and three planar coordinate origins are of interest in this report. "True" coordinates are those reckoned from the true origin of the projection. They are at a scale inherent with the projection, a scale dictated by the projection parameters. "Grid" coordinates are at the same inherent scale, but are referenced to an origin situated more conveniently for a particular area of interest. The true origin is the fundamental origin of the projection, lying at a point that orients the projection surface with the globe. Normally the grid origin is located by design to the west and south of the region of interest (or "zone"), so that resulting grid coordinates are positive-valued and of a desired magnitude.

A third origin, termed "false origin" in this report, is sometimes used as an intermediate origin when going from a true origin to a grid origin. The use of a false origin is strictly to facilitate the understanding of the translation of coordinates from the true origin to the grid origin; it has no other significance. Under certain circumstances, the false origin could be collocated with either the true origin or the grid origin. False eastings and false northings are x and y values used in the translation of coordinates from one origin to another. They can be thought of as coordinates in the grid system that are assigned to the true origin or to the false origin.

Scale

Scale, in the context of this report, is the ratio of distance over the projection surface to distance over the reference ellipsoid (geodetic distance). It is a quantity not clearly understood by many. Except along certain specific lines on some projections, scale varies from point to point. In general, scale even varies with direction from a point. In conformal map projections scale is independent of direction, though it is still dependent on position. Projections used for surveying are usually conformal.

Lines along which scale might be held constant are specified by the projection parameters. For example, on the transverse Mercator projection, scale is constant along the central meridian. On the Lambert conformal conic projection, scale is constant along parallels, although it generally differs from any given parallel to another. When scale is specified for a projection, it is implied, if not explicitly stated, along those specific lines.

The magnitude of a specified scale factor is near unity if the projection is to be used for survey coordinates. For cartographic applications, the specified scale factor is a small number, for example 1/10,000 or 1/250,000. Bear in mind that although the magnitudes of survey and cartographic scale factors are widely separated, they represent the same thing in essence. A specified cartographic scale factor applies to a specific line or point on the geodetic reference ellipsoid just as a specified survey scale factor does. One must be careful when interpreting cartographic scale factors.

Consider the following scenario. Plane survey coordinates are computed on a given projection having a scale factor of 0.9996 along its central line. The survey coordinates are then reduced by 1/10,000 to be plotted on a map. Subsequently, an inverse transformation is performed on the map coordinates to obtain geodetic coordinates for the survey points. Using a scale factor of 1/10,000 along the central line of the projection will result in substantial error in the computed geodetic coordinates. In this scenario, the scale factor that should have been used is $0.9996(1/10,000)$, or about 1/10,004. If, on the other hand, map coordinates for the survey points had been originally computed directly from geodetic coordinates, the scale factor of 1/10,000 in the inverse transformation would have been correct.

State Plane Coordinate System

The State Plane Coordinate System of 1927 (so called because it is based on the North American Datum of 1927) was devised by the U.S. Coast and Geodetic Survey (C&GS) in the 1930's. Its purpose was to allow surveyors and engineers to compute accurate coordinates using plane trigonometry. Corrections to observed angles and distances are made to account for discrepancies between planar and ellipsoidal computations. Originally, tables of constants that were computed by the C&GS using common logarithms were provided to simplify calculation of positions. Later, Claire (1973) of the C&GS provided algorithms and constants for machine computation of positions. These algorithms were designed to duplicate results obtained using the tables. For that reason, they are purposely inaccurate to a slight degree.

The State Plane Coordinate System of 1983 (so called because it is based on the North American Datum of 1983) was necessitated by the 1983 adjustment of the North American Datum. Mapping equations used by the National Geodetic Survey (NGS) of the National Ocean Service, National Oceanic and Atmospheric Administration, for transformation between 1983 coordinates provide very accurate results. Several of the algorithms in this report make use of the NGS equations, and should produce identical results. They will not produce results identical to the C&GS algorithms for 1927 coordinates. However, since the prevalent use of 1927 coordinate transformations would be in the inverse mode, it is more appropriate to obtain as accurate a transformation as possible than to duplicate the C&GS algorithms.

Projection parameters for the State Plane Coordinate System of 1927 were publicized in Mitchell and Simmons (1945). Some ambiguity could arise from using those parameters listed for the Lambert conformal conic zones due to the fact that mutually exclusive parameters are given. State plane coordinate zones using the Lambert conformal conic projection are defined with north and south standard parallels. The central parallel and its associated scale found in the Mitchell and Simmons are derived quantities, making them approximate values that should not be used for computations. Appendix B of this report contains the correct parameters for the State Plane Coordinate System of 1927.

The State of Michigan requires special attention. It was originally given three zones based on the transverse Mercator projection. In 1964 the Michigan coordinate system was revised to consist of three zones based on the Lambert conformal conic projection raised to an elevation of (nominally) 800 U.S. survey feet. The coordinate system elevated to that height is equivalent to a system at "sea level" on an ellipsoid with equivalent flattening, but with a semimajor axis defined as exactly 1.0000382 times the conventional semimajor axis (Berry 1971). Carrying out the multiplication, the semimajor axis for the 1964 Michigan coordinate system is 6,378,450.04748448 meters, exactly.

ACCURACY

Exact Coordinate Conversion

Consider the process by which field observations are used to compute coordinates. The basic steps are the same, irrespective of the coordinate system employed. First, measurements such as distances and angles are corrected for systematic errors attributable to the measuring system. Instrument error corrections account for errors directly attributable to the measuring equipment. Environmental corrections account for fluctuations in a measured value due to the effect that environmental fluctuations have on the measuring instrument. Second, the corrected measurements are "reduced" (changed, not necessarily made smaller) to a common surface on which coordinate computations can be made. And third, mathematical models based on geometric relationships are used to determine coordinates of unknown positions from those of known positions using the reduced measurements. Converting the coordinates thus obtained to coordinates in a different reference system can be accomplished in a straightforward manner by applying equations relating the two coordinate systems. Precisely correct results can be obtained if the following assumptions are valid:

1. Accurate corrections were applied to field observations to obtain accurate measurements.
2. Measurements were correctly reduced to the common reference surface.
3. The mathematical model for determining coordinates of unknown points was correct.
4. The coordinates of the known starting points were correct.
5. Equations relating the two coordinate systems were accurate.

In short, exact coordinates in a new reference system can be obtained only by applying exact transformation equations to exact coordinates in the old reference system.

Practical Coordinate Conversion Accuracies

In practice, exact coordinates in a new reference system are not obtained. To do so would require starting with coordinates known to be correct in the old system, converting those coordinates to the new reference system, applying corrections and reductions to the original field observations (assuming they were correct), then computing new coordinates using a mathematical model applicable to the new reference system. Starting with coordinates known to be correct in the old system is the crux of the problem.

Verification of historical data and information is not a function of an automated data acquisition and processing system. Coordinates of historical data points must be taken at face value, with the realization that such coordinates could be significantly in error. A rough idea of the magnitudes possible for such error follows:

<u>Type of error</u>	<u>Possible magnitude</u>
Measurement system	decimeters
Measurement reduction	meters
Coordinate computation model	centimeters

In general, the actual magnitude of these errors will not be known. Certainly, we do not want to increase them appreciably. Therefore, coordinate transformation equations should have accuracies on the order of millimeters.

COORDINATE CONVERSION PROCEDURES

Coordinate conversions fall into eight different cases, each involving one or more transformations.

- Case 1 - Geodetic to plane coordinates referenced to the same ellipsoid.
Operation required: forward transformation
- Case 2 - Plane to geodetic coordinates referenced to the same ellipsoid.
Operation required: inverse transformation
- Case 3 - Plane coordinates on one projection to plane coordinates on another projection, both projections referenced to the same ellipsoid.
Operations required: inverse transformation
forward transformation
- Case 4 - Geodetic coordinates based on one ellipsoid to geodetic coordinates based on another ellipsoid.
Operation required: datum transformation
- Case 5 - Geodetic to plane coordinates referenced to a different ellipsoid.
Operations required: datum transformation
forward transformation
- Case 6 - Plane to geodetic coordinates referenced to a different ellipsoid.
Operations required: inverse transformation
datum transformation
- Case 7 - Plane coordinates on one projection to plane coordinates on the same type of projection referenced to a different ellipsoid.
Operations required: inverse transformation
datum transformation
forward transformation
- Case 8 - Plane coordinates on one projection to plane coordinates on a different projection referenced to a different ellipsoid.
Operations required: inverse transformation
datum transformation
forward transformation

Assuming that a numeric code is known identifying the reference ellipsoid for geodetic coordinates, and assuming that numeric codes are known identifying a projection type and the associated reference ellipsoid for planar coordinates, coordinate conversions can be accomplished automatically. The procedure involves determining the case number, then calling transformation subroutines as required by the case. The case number can be determined using the following logic:

Assign a projection code of 0 to geodetic coordinates.

If old projection code = 0 (starting with geodetic coordinates)

If new projection code = 0 (going to geodetic coordinates) Case = 4

If new projection code \neq 0 (going to plane coordinates)

If old ellipsoid code = new ellipsoid code Case = 1

If old ellipsoid code \neq new ellipsoid code Case = 5

If old projection code \neq 0 (starting with plane coordinates)

If new projection code = 0 (going to geodetic coordinates)

If old ellipsoid code = new ellipsoid code Case = 2

If old ellipsoid code \neq new ellipsoid code Case = 6

If new projection code \neq 0 (going to plane coordinates)

If old ellipsoid code = new ellipsoid code Case = 3

If old ellipsoid code \neq new ellipsoid code

If old projection code = new projection code Case = 7

If old projection code \neq new projection code Case = 8

COORDINATE TRANSFORMATION ALGORITHMS

These algorithms are designed to be sufficiently general to allow their being used for a variety of purposes. The most basic ellipsoidal parameters and projection parameters are input, so that transformations can be performed with projections of any orientation to any ellipsoid. Provisions are made to allow the input and use of certain mutually exclusive defining parameters. (Of three related parameters, if any two can be selected as independent variables, only two can be considered defining parameters.) Coordinates are input in arrays dimensioned by variables, permitting transformations to be made either in groups or singly (by specifying an array of one pair of coordinates). Alternate entry points are provided between the computation of certain constants and computations unique to the specifically oriented projection, so that after initialization the subroutines can be used repeatedly without having to recompute constants common to the job.

Isometric Latitude

Isometric latitude (τ) is an auxiliary latitude used in several conformal projections. In most publications it is computed as follows:

$$\tau = \ln \left[\tan \left(\frac{\pi}{4} + \frac{\phi}{2} \right) \left(\frac{1 - e \sin \phi}{1 + e \sin \phi} \right)^{e/2} \right]$$

In this report $\left(\frac{1 + \sin\phi}{1 - \sin\phi}\right)^{1/2}$

is substituted for $\tan\left(\frac{\pi}{4} + \frac{\phi}{2}\right)$.

Furthermore, because $\exp(\tau)$ is usually the quantity of interest, the natural logarithm of the expression in brackets above is usually not taken.

Inputs and Outputs

All inputs of linear measure must be in like units, and all inputs of arc measure must be in units of radians, north latitudes and east longitudes positive. The following generic inputs are required:

1. Ellipsoidal parameters.
2. Projection parameters.
3. Row dimension declared for coordinate arrays in calling program.
4. Array of coordinates requiring transformation.
5. Number of pairs of coordinates requiring transformation.

Ellipsoidal parameters that are not unitless are usually given in metric units in reference literature, while projection parameters are often given in English units. Appropriate conversions must be made prior to passing such inputs.

Outputs are in the same units as the inputs, and must be converted as required after returning from the subroutines. Longitudes are output in the range $-\pi$ to less than or equal to π . The x values of points that are farther than 180° from the central meridian are computed in the opposite direction from the central meridian.

At least 12 significant figures are required for the desired accuracy in projection zone widths that may be encountered. Therefore, double precision variables will be required on most computers.

Symbology

Pseudocode based on the standard FORTRAN 77 programming language is used throughout the algorithms. Variable names and FORTRAN statements are capitalized. They are mixed with regular mathematical symbols and with symbols conventionally used to denote geodetic quantities. Preference is given to common symbology and English-like phrases, but these are supplemented with FORTRAN conventions to promote clarity and conciseness, and to facilitate translation into code. The following loose conventions are used in naming FORTRAN variables:

- o Variables that have no particular meaning other than a numerical quantity are generally given names with one or two characters.
- o Variables that represent recognizable quantities are generally given names of three to six characters.
- o GPRAD(I,1) denotes a geodetic latitude to be operated on, while a special latitude, such as one selected as a parametric value, is represented by PHI (ϕ).
- o GPRAD(I,2) and LAM (λ) are used similarly in denoting longitude.

Normal Mercator Projection

The normal Mercator (or simply Mercator) projection is good for surveying purposes in a band near the Equator. As the distance from the Equator increases, the change in scale factor increases more and more rapidly, making the projection less convenient for surveying purposes. In cartographic applications the projection is useful much further from the Equator, but as the poles are approached, convergence of the meridians becomes a problem. The north and south poles are undefined on the normal Mercator projection.

The true origin of the Mercator projection is at the intersection of the Equator with the meridian of zero longitude. A "central meridian" is arbitrarily chosen such that x coordinates in the area of interest remain positive-valued. Using the convention of east longitude positive, the central meridian is chosen west of the area of interest. To reduce the magnitude of the y coordinates, a grid origin may be established on the central meridian, just south of the area of interest. The establishment of the grid origin is accomplished by specifying a value other than 0 for the y coordinate of the true origin. In the northern hemisphere, a negative-valued y coordinate, or false northing, is assigned to the true origin, moving the grid origin north. In the southern hemisphere a false northing would be positive-valued, moving the grid origin south.

Normal Mercator Forward Transformation (MERFWD)

Input:

AMAJ	Semimajor axis of ellipsoid (a).
FINV	Reciprocal of flattening (1/f).
LAMCEN	Central meridian of projection (λ_0).
PHISF	Absolute value of the geodetic latitude where scale factor is known (ϕ_s).
SFPHI	Scale factor at PHISF.
FALSEN	False northing (y_0).
ROWS	Number of rows declared for arrays GPRAD and XYGRID in the calling program.
N	Number of positions to be transformed.

GPRAD(I,1) Geodetic latitude, in radians, of the i^{th} position to be transformed (ϕ).

GPRAD(I,2) Longitude of the position (λ).

Output:

XYGRID(I,1) x coordinate, relative to the grid origin, of the transformed i^{th} position.

XYGRID(I,2) y coordinate relative to the grid origin.

SF(I) Point scale factor, by which infinitesimal geodetic length at the i^{th} position is multiplied to obtain grid length.

Other meaningful variables:

FLAT Ellipsoidal flattening (f).

ECC First eccentricity of ellipsoid (e).

ESQ First eccentricity squared (e^2).

SFEQ Scale factor at the Equator.

TAU Isometric latitude (τ).

XTRUE x coordinate relative to the true origin of the projection.

YTRUE y coordinate relative to the true origin.

Algorithm:

1. Input ellipsoid parameters **AMAJ** and **FINV**.
2. Input projection parameters **LAMCEN**, **PHISF**, **SFPHI**, and **FALSEN**.
3. Input **ROWS**, **N**, and array **GPRAD**.

4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e = (e^2)^{\frac{1}{2}}$$

5. Compute the scale factor along the Equator,

$$SFEQ = SFPHI(\cos \phi_s)/(1 - e^2 \sin^2 \phi_s)^{\frac{1}{2}}$$

6. SAVE the constants ESQ, ECC, and SFEQ.

7. Provide an alternate ENTRY point named MERCF2 passing AMAJ, LAMCEN, FALSEN, ROWS, N, and GPRAD through the argument list.

8. If no more forward transformations to perform, RETURN. Output grid x's and y's, and their respective point scale factors.

9. For the next pair of ϕ and λ in the input array, compute the plane coordinates relative to the true origin,

$$XTRUE = a \lambda SFEQ$$

$$\tau = 1n \left\{ \left[\left(\frac{1 + \sin \phi}{1 - \sin \phi} \right) \left(\frac{1 - e \sin \phi}{1 + e \sin \phi} \right)^e \right]^{\frac{1}{2}} \right\}$$

$$YTRUE = a \tau SFEQ.$$

10. Compute the grid coordinates,

$$\text{IF } (-\pi \leq \lambda - \lambda_0 \leq \pi) \text{ XYGRID}(I,1) = XTRUE - a \lambda_0 SFEQ$$

$$\text{IF } (\lambda - \lambda_0 > \pi) \text{ XYGRID}(I,1) = XTRUE - a(\lambda_0 + 2\pi)SFEQ$$

$$\text{IF } (-\pi > \lambda - \lambda_0) \text{ XYGRID}(I,1) = XTRUE - a(\lambda_0 - 2\pi)SFEQ$$

$$\text{XYGRID}(I,2) = YTRUE + \text{FALSEN}.$$

11. Compute the point scale factor,

$$SF(I) = SFEQ(1 - e^2 \sin^2 \phi)^{\frac{1}{2}} / \cos \phi.$$

12. Repeat from step 8 for the next GP to be transformed.

References:

1. Thomas, 1952, pp.85-86.
2. Deetz and Adams, 1944, pp. 112-115.
3. Snyder, 1983, pp. 50-51.

Normal Mercator Inverse Transformation (MERINV)

Input:

- AMAJ Semimajor axis of ellipsoid (a).
- FINV Reciprocal of flattening (1/f).
- LAMCEN Central meridian of projection (λ_0).
- PHISF Absolute value of the geodetic latitude where scale factor is known (ϕ_s).
- SFPHI Scale factor at PHISF.
- FALSEN False northing (y_0).
- ROWS Number of rows declared for the arrays GPRAD and XYGRID in the calling program.
- N Number of positions to be transformed.
- XYGRID(I,1) x coordinate, relative to the grid origin, of the i^{th} position to be transformed.
- XYGRID(I,2) y coordinate relative to the grid origin.

Output:

GPRAD(I,1) Geodetic latitude, in radians, of the transformed i^{th} position (ϕ).

GPRAD(I,2) Longitude of the position (λ).

SF(I) Point scale factor, by which infinitesimal geodetic length at the i^{th} position is multiplied to obtain grid length.

Other meaningful variables:

FLAT Ellipsoidal flattening (f).

ECC First eccentricity of ellipsoid (e).

ESQ First eccentricity squared (e^2).

SFEQ Scale factor at the Equator.

YTRUE y coordinate relative to the true origin.

Algorithm:

1. Input ellipsoid parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, PHISF, SFPHI, and FALSEN.
3. Input ROWS, N, and array XYGRID.
4. Compute ellipsoidal constants,
 $f = 1/(1/f)$
 $e^2 = f(2 - f)$
 $e = (e^2)^{\frac{1}{2}}$.
5. Compute the scale factor along the Equator,
 $SFEQ = SFPHI(\cos\phi_s)/(1 - e^2\sin^2\phi_s)^{\frac{1}{2}}$.
6. SAVE the constants ESQ, ECC, and SFEQ.
7. Provide an alternate ENTRY point named MERCI2, passing AMAJ, LAMCEN, FALSEN, ROWS, N, and XYGRID through the argument list.

8. If no more inverse transformations to perform, RETURN. Output geodetic latitudes (ϕ) and longitudes (λ), and their respective point scale factors.
9. For the next pair of grid coordinates in the input array, compute the y coordinate relative to the true origin,

$$Y_{TRUE} = y - y_0.$$
10. Compute an initial approximation of geodetic latitude (ϕ_1),

$$t = 1 / \{ \exp[Y_{TRUE} / (a \text{SFEQ})] \}$$

$$\phi_1 = \pi / 2 - 2 \tan^{-1}(t).$$
11. Compute the next approximation of latitude (ϕ_{j+1} , $j = 1, 2, 3, \dots$),

$$\phi_{j+1} = \pi / 2 - 2 \tan^{-1} \{ t [(1 - e \sin \phi_j) / (1 + e \sin \phi_j)]^{e/2} \}.$$
12. Iterate step 11 until $|\phi_{j+1} - \phi_j| < 5 \times 10^{-9}$ rad.
13. Compute the longitude,

$$\lambda = \lambda_0 + x / [a(\text{SFEQ})]$$

$$\text{IF } (\lambda \leq -\pi) \text{ GPRAD}(I, 2) = \lambda + 2\pi$$

$$\text{IF } (\lambda > \pi) \text{ GPRAD}(I, 2) = \lambda - 2\pi.$$
14. Compute the point scale factor,

$$\text{SF}(I) = \text{SFEQ} (1 - e^2 \sin^2 \phi)^{\frac{1}{2}} / \cos \phi.$$
15. Repeat from step 8 for the next xy position to be transformed.

Reference:

Snyder, 1983, pp. 50-51.

Transverse Mercator Projection

The transverse Mercator projection is often used where the area of interest lies with its longer dimension in a north-south direction. The true origin of this projection is at the intersection of the central meridian with the Equator. To maintain coordinates that are convenient for the area of interest, a grid origin may be established to the south and west of that area. A two-step procedure allows definition of the north-south location of the grid origin in either of two ways: by specifying a latitude on which the grid origin lies or by designating a value other than 0 for the y coordinate of the true origin. The two-step procedure involves translation of coordinates from the true origin to a false origin, then to the grid origin.

The false origin lies on the central meridian of the projection and on the latitude of false origin. Y coordinates may be reduced in magnitude by specifying a latitude of false origin just to the south of the area of interest. If a false northing is specified rather than a latitude of false origin, the false origin is located at the intersection of the central meridian with the Equator. The false northing, or grid y-coordinate of the false origin, would be negative-valued in the northern hemisphere, moving the grid origin northward. In the southern hemisphere the false northing would be positive-valued. The false easting is the x coordinate assigned to the false origin. It would be positive in either hemisphere.

Mapping equations for the transverse Mercator projection become unstable near the poles and 90° off the central meridian. From a practical standpoint, use of the transverse Mercator projection should be limited to a region bounded by a maximum latitude (\pm) and a longitudinal distance from the central meridian. The limiting latitude and longitudinal distance will depend on the purpose of the projection.

Transverse Mercator Forward Transformation (TMFWD)

Input:

AMAJ	Semimajor axis of ellipsoid (a).
FINV	Reciprocal of flattening (1/f).
LAMCEN	Central meridian of projection (λ_0).
FALSEE	False easting (x_0).
FALSEN	False northing (y_0).
PHIFAL	Geodetic latitude of the false origin (ϕ_f).
SFCEN	Scale factor along central meridian (k_0).

ROWS Number of rows declared for the arrays GPRAD and XYGRID in the calling program.
N Number of positions to be transformed.
GPRAD(I,1) Geodetic latitude, in radians, of the i^{th} position to be transformed (ϕ).
GPRAD(I,2) Longitude of the position (λ).

Output:

XYGRID(I,1) x coordinate, relative to the grid origin, of the transformed i^{th} position.
XYGRID(I,2) y coordinate relative to the grid origin.
SF(I) Point scale factor, by which infinitesimal geodetic length at the i^{th} position is multiplied to obtain grid length.

Other meaningful variables:

FLAT Ellipsoidal flattening (f).
ESQ First eccentricity of ellipsoid, squared (e^2).
E2SQ Second eccentricity squared (e'^2).
ECC3 Third eccentricity (n).
RN Radius of curvature in prime vertical.
RREC Radius of rectifying sphere (r).
YVALUE y value of true origin relative to false origin.
OMEGA Rectifying latitude of the point in question (ω).
OMEGAF Rectifying latitude of false origin (ω_f).
S Meridional distance.
ETASQ Geodetic variable (n^2) representing the quantity $e'^2 \cos^2 \phi$.

XTRUE x coordinate relative to the true origin of the projection.
YTRUE y coordinate relative to the true origin.
XFALSE x coordinate relative to the false origin.
YFALSE y coordinate relative to the false origin.

Algorithm:

1. Input ellipsoid parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, FALSEE, FALSEN, PHIFAL, and SFCEN.
3. Input ROWS, N, and array GPRAD.
4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e'^2 = e^2/(1 - e^2)$$

$$n = f/(2 - f).$$

5. Compute constants for meridional distances,

$$r = a(1 - n)(1 - n^2)(1 + 9n^2/4 + 225n^4/64)$$

$$A_2 = -3n/2 + 9n^3/16$$

$$A_4 = 15n^2/16 - 15n^4/32$$

$$A_6 = -35n^3/48$$

$$A_8 = 315n^4/512$$

$$B_0 = 2(A_2 - 2A_4 + 3A_6 - 4A_8)$$

$$B_2 = 8(A_4 - 4A_6 + 10A_8)$$

$$B_4 = 32(A_6 - 6A_8)$$

$$B_6 = 128A_8.$$

6. Determine the y value of the true origin relative to the false origin,

$$\omega_f = \phi_f + \sin\phi_f \cos\phi_f (B_0 + B_2 \cos^2\phi_f + B_4 \cos^4\phi_f + B_6 \cos^6\phi_f)$$

$$YVALUE = -k_0 \omega_f r.$$

7. SAVE the constants ESQ, E2SQ, RREC, YVALUE, B₀, B₂, B₄, and B₆.
8. Provide an alternate ENTRY point named TMFWD2, passing AMAJ, LAMCEN, FALSEE, FALSEN, SFCEN, ROWS, N, and GPRAD through the argument list.
9. If no more forward transformations to perform, RETURN. Output grid x's and y's, and their respective point scale factors.
10. For the next pair of ϕ and λ in the input array, compute intermediate values needed for the transformation. $\cos\phi$ and $\tan\phi$ should be assigned to variable names to avoid repeated application of the intrinsic cosine and tangent functions. If $\phi = \pm\pi/2$, disable the tangent function; the tangent of ϕ has no effect on the outcome if $\phi = \pm\pi/2$.

$$\eta^2 = e^2 \cos^2 \phi$$

$$\text{IF } (-\pi \leq \lambda - \lambda_0 \leq \pi) \text{ L} = (\lambda - \lambda_0) \cos \phi$$

$$\text{IF } (\lambda - \lambda_0 > \pi) \text{ L} = (\lambda - \lambda_0 - 2\pi) \cos \phi$$

$$\text{IF } (-\pi > \lambda - \lambda_0) \text{ L} = (\lambda - \lambda_0 + 2\pi) \cos \phi$$

$$\omega = \phi + \sin \phi \cos \phi (B_0 + B_2 \cos^2 \phi + B_4 \cos^4 \phi + B_6 \cos^6 \phi)$$

$$S = \omega r$$

$$RN = a / (1 - e^2 \sin^2 \phi)^{1/2}$$

$$E_3 = (1 - \tan^2 \phi + \eta^2) / 6$$

$$E_4 = [5 - \tan^2 \phi + \eta^2 (9 + 4\eta^2)] / 12$$

$$E_5 = [5 - 18 \tan^2 \phi + \tan^4 \phi + \eta^2 (14 - 58 \tan^2 \phi)] / 120$$

$$E_6 = [61 - 58 \tan^2 \phi + \tan^4 \phi + \eta^2 (270 - 330 \tan^2 \phi)] / 360$$

$$E_7 = (61 - 479 \tan^2 \phi + 179 \tan^4 \phi - \tan^6 \phi) / 5040$$

$$F_2 = (1 + \eta^2) / 2$$

$$F_4 = [5 - 4 \tan^2 \phi + \eta^2 (9 - 24 \tan^2 \phi)] / 12.$$

11. Compute the plane coordinates relative to the true origin,

$$X_{TRUE} = k_0 (RN) L \{ 1 + L^2 [E_3 + L^2 (E_5 + E_7 L^2)] \}$$

$$Y_{TRUE} = k_0 \{ S + RN (\tan \phi) L^2 [1 + L^2 (E_4 + E_6 L^2)] / 2 \}.$$

12. Compute the plane coordinates relative to the false origin,

$$XFALSE = XTRUE$$

$$YFALSE = YTRUE + YVALUE.$$

13. Compute the grid coordinates,

$$XYGRID(I,1) = XFALSE + FALSEE$$

$$XYGRID(I,2) = YFALSE + FALSEN.$$

14. Compute the point scale factor,

$$SF(I) = k_0 [1 + F_2 L^2 (1 + F_4 L^2)].$$

15. Repeat from step 9 for the next GP to be transformed.

Reference:

Vincenty, 1984a.

Transverse Mercator Inverse Transformation (TMINV)

Input:

AMAJ Semimajor axis of ellipsoid (a).

FINV Reciprocal of flattening (1/f).

LAMCEN Central meridian of projection (λ_0).

FALSEE False easting (x_0).

FALSEN False northing (y_0).

PHIFAL Geodetic latitude of the false origin (ϕ_f).

SFCEN Scale factor along central meridian (k_0).

ROWS Number of rows declared for arrays GPRAD and XYGRID in the calling program.

N Number of positions to be transformed.

XYGRID(I,1) x coordinate, relative to the grid origin, of the i^{th} position to be transformed.

XYGRID(I,2) y coordinate relative to the grid origin.

Output:

GPRAD(I,1) Geodetic latitude, in radians, of the transformed i^{th} position (ϕ).

GPRAD(I,2) Longitude of the position (λ).

SF(I) Point scale factor, by which infinitesimal geodetic length at the i^{th} position is multiplied to obtain grid length.

Other meaningful variables:

FLAT Ellipsoidal flattening (f).

ESQ First eccentricity of ellipsoid, squared (e^2).

E2SQ Second eccentricity squared (e'^2).

ECC3 Third eccentricity (n).

RNFP Radius of curvature in prime vertical at the footpoint latitude.

RREC Radius of rectifying sphere (r).

YVALUE y value of true origin relative to false origin.

OMEGA Rectifying latitude of the point in question (ω).

OMEGAF Rectifying latitude of false origin (ω_f).

LATFP Footpoint latitude (ϕ_{fp}).

S Meridional distance.

ETASQ Geodetic variable (η^2) representing $e'^2 \cos^2 \phi$.

Algorithm:

1. Input ellipsoid parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, FALSEE, FALSEN, PHIFAL, and SFCEN.
3. Input ROWS, N, and array XYGRID.
4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e'^2 = e^2/(1 - e^2)$$

$$n = f/(2 - f)$$

5. Compute constants for meridional distances,

$$r = a(1 - n)(1 - n^2)(1 + 9n^2/4 + 225n^4/64)$$

$$A_2 = -3n/2 + 9n^3/16$$

$$A_4 = 15n^2/16 - 15n^4/32$$

$$A_6 = -35n^3/48$$

$$A_8 = 315n^4/512$$

$$B_0 = 2(A_2 - 2A_4 + 3A_6 - 4A_8)$$

$$B_2 = 8(A_4 - 4A_6 + 10A_8)$$

$$B_4 = 32(A_6 - 6A_8)$$

$$B_6 = 128A_8$$

$$C_2 = 3n/2 - 27n^3/32$$

$$C_4 = 21n^2/16 - 55n^4/32$$

$$C_6 = 151n^3/96$$

$$C_8 = 1097n^4/512$$

$$D_0 = 2(C_2 - 2C_4 + 3C_6 - 4C_8)$$

$$D_2 = 8(C_4 - 4C_6 + 10C_8)$$

$$D_4 = 32(C_6 - 6C_8)$$

$$D_6 = 128C_8$$

6. Determine the y value of the true origin relative to the false origin,

$$\omega_f = \phi_f + \sin\phi_f \cos\phi_f (B_0 + B_2 \cos^2 \phi_f + B_4 \cos^4 \phi_f + B_6 \cos^6 \phi_f)$$

$$YVALUE = -k_0 \omega_f r.$$

7. SAVE the constants ESQ, E2SQ, RREC, YVALUE, D_0 , D_2 , D_4 , and D_6 .

8. Provide an alternate ENTRY point named TMINV2, passing AMAJ, LAMCEN, FALSEE, FALSEN, SFCEN, ROWS, N, and XYGRID through the argument list.

9. If no more inverse transformations to perform, RETURN. Output geodetic latitudes (ϕ) and longitudes (λ), and their respective point scale factors.

10. For the next pair of grid coordinates in the input array, compute intermediate values needed for the transformation. $\cos\omega$ and $\tan\phi_{fp}$ should be assigned to variable names to avoid repeated application of the intrinsic cosine and tangent functions. If $\omega = \pm\pi/2$, computation of the remaining intermediate values in this step are skipped; in step 11, $\phi = \omega$, λ is indeterminate and may be set equal to λ_0 for convenience; and in step 12, the point scale factor equals k_0 .

$$\omega = (y - y_0 - YVALUE)/(k_0 r)$$

$$\phi_{fp} = \omega + \sin\omega \cos\omega (D_0 + D_2 \cos^2 \omega + D_4 \cos^4 \omega + D_6 \cos^6 \omega)$$

$$RNFP = a/(1 - e^2 \sin^2 \phi_{fp})^{1/2}$$

$$\eta_{fp}^2 = e^2 \cos^2 \phi_{fp}$$

$$G_2 = -\tan\phi_{fp} (1 + \eta_{fp}^2)/2$$

$$G_3 = -(1 + 2\tan^2 \phi_{fp} + \eta_{fp}^2)/6$$

$$G_4 = -[5 + 3\tan^2 \phi_{fp} + \eta_{fp}^2 (1 - 9\tan^2 \phi_{fp}) - 4\eta_{fp}^4]/12$$

$$G_5 = [5 + 28\tan^2 \phi_{fp} + 24\tan^4 \phi_{fp} + \eta_{fp}^2 (6 + 8\tan^2 \phi_{fp})]/120$$

$$G_6 = [61 + 90\tan^2 \phi_{fp} + 45\tan^4 \phi_{fp} + \eta_{fp}^2 (46 - 252\tan^2 \phi_{fp} - 90\tan^4 \phi_{fp})]/360$$

$$G_7 = -(61 + 662\tan^2\phi_{fp} + 1320\tan^4\phi_{fp} + 720\tan^6\phi_{fp})/5040$$

$$H_2 = (1 + \eta_{fp}^2)/2$$

$$H_4 = (1 + 5\eta_{fp}^2)/12$$

$$Q = (x - x_0)/(k_0 R_{NFP})$$

$$L = Q(1 + Q^2[G_3 + Q^2(G_5 + G_7Q^2)]).$$

11. Compute the geodetic latitude (ϕ) and longitude (λ),

$$\phi = \phi_{fp} + G_2Q^2[1 + Q^2(G_4 + G_6Q^2)]$$

$$\lambda = \lambda_0 + L/\cos\phi_{fp}$$

$$\text{IF } (\lambda < -\pi) \text{ GPRAD}(I,2) = \lambda + 2\pi$$

$$\text{IF } (\lambda > \pi) \text{ GPRAD}(I,2) = \lambda - 2\pi.$$

12. Compute the point scale factor,

$$\text{SF}(I) = k_0[1 + H_2Q^2(1 + H_4Q^2)].$$

13. Repeat from step 9 for the next xy position to be transformed.

Reference:

Vincenty, 1984a.

Oblique Mercator Projection

The oblique Mercator projection is used where the area of interest is oblong or rectangular and is skewed with respect to the meridians. Projection parameters for the oblique Mercator projection can be defined in one of two ways. (1) Two widely spaced points can be selected to define the central line of the projection. (The central line in this projection is a geodesic line running in the direction of the longer dimension of the area of interest.) The latitudes and longitudes of the two points and the scale factor on the central line at a selected latitude provide the projection parameters. (2) The latitude, longitude, and scale factor of a selected center point, and the azimuth at the center point of the skewed central line provide another set of projection parameters. This alternative is the one on which the algorithms herein are based.

The oblique Mercator projection and its transformation equations should not be used under certain circumstances.

1. If the center point of the area of interest lies near either pole, use the stereographic projection instead (not included in this report).
2. If one of the two widely spaced points defining the projection lies at either pole, use equations for the transverse Mercator projection instead.
3. If the two widely spaced points both lie on the Equator, use equations for the normal Mercator projection instead.
4. In general, if the two widely spaced points lie on the same parallel of latitude other than the Equator, use equations for the Lambert conformal conic projection instead. Note, however, that if the parallel of latitude on which the two defining points lie is close to the Equator, a normal Mercator projection may be used.

The true origin of the projection lies at the intersection of the central line with the Equator of the so-called aposphere (Hotine 1946-47), such point being near the true Equator of the Earth. For Alaska zone 1 (SE Alaska) the true origin is in the vicinity of 0° latitude, -101° longitude. The grid origin lies at a specified point near the area of interest, south and west of the center point. For Alaska zone 1, that point is 5,000,000 meters to the north and 5,000,000 meters to the west of the true origin.

The algorithms for transformation of coordinates are based on Hotine's "rectified skew orthomorphic" projection. Equations were obtained from T. Vincenty (1984b) of NGS and were further manipulated to eliminate application of the natural logarithm when the exponential function would subsequently be applied.

Oblique Mercator Forward Transformation (OMFWD)

Input:

AMAJ	Semimajor axis of ellipsoid (a).
FINV	Reciprocal of flattening (1/f).
PHICEN	Geodetic latitude of the center point of the projection (ϕ_c).
LAMCEN	Geodetic longitude of the center point (λ_c).
SFCEN	Scale factor at the center point (k_c).
AZICEN	Geodetic azimuth at the center point of the skewed center line (α_c).
FALSEN	False northing (y_0).
FALSEE	False easting (x_0).

ROWS Number of rows declared for array GPRAD and XYGRID in the calling program.
N Number of positions to be transformed.
GPRAD(I,1) Geodetic latitude, in radians, of the i^{th} position to be transformed (ϕ).
GPRAD(I,2) Longitude of the position (λ).

Output:

XYGRID(I,1) x coordinate, relative to the grid origin, of the transformed i^{th} position.
XYGRID(I,2) y coordinate relative to the grid origin.
SF(I) Point scale factor, by which infinitesimal geodetic length at the i^{th} position is multiplied to obtain grid length.

Other meaningful variables:

FLAT Ellipsoidal flattening (f).
ECC First eccentricity of ellipsoid (e).
ESQ First eccentricity squared (e^2).
E2SQ Second eccentricity squared (e'^2).
WSQ Geodetic variable (W^2) representing the quantity $1 - e^2 \sin^2 \phi$.
EXPT Natural base of logarithms raised to the T power, where T (τ) is the isometric latitude.
LAMO Geodetic longitude of the true origin (λ_0).
USKEW u coordinate in the rectilinear coordinate system with the origin at the center point and u axis along the projected skewed center line.
VSKEW v coordinate in the skewed rectilinear coordinate system.

Algorithm:

1. Input ellipsoid parameters AMAJ and FINV.
2. Input projection parameters PHICEN, LAMCEN, SFCEN, AZICEN, FALSEN, and FALSEE.
3. Input ROWS, N, and array GPRAD.
4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e = (e^2)^{\frac{1}{2}}$$

$$e'^2 = e^2/(1 - e^2).$$

5. Compute zone constants,

$$W_c^2 = 1 - e^2 \sin^2 \phi_c$$

$$B = (1 + e'^2 \cos^4 \phi_c)^{\frac{1}{2}}$$

$$A = B(1 - e^2)^{\frac{1}{2}}/W_c^2$$

$$EXPT_c = \left[\left(\frac{1 + \sin \phi_c}{1 - \sin \phi_c} \right) \left(\frac{1 - e \sin \phi_c}{1 + e \sin \phi_c} \right) e \right]^{\frac{1}{2}}$$

$$S_c = W_c A / \cos \phi_c$$

$$C_c = S_c + (S_c^2 - 1)^{\frac{1}{2}}$$

$$J_c = (C_c - 1/C_c)/2$$

$$D = k_c(A/B)a$$

$$F = \sin \alpha_0 = \sin \alpha_c \cos \phi_c / (W_c A)$$

$$G = \cos \alpha_0 = \cos(\sin^{-1} F)$$

$$\lambda_0 = \lambda_c - \sin^{-1}(J_c F/G)/B$$

$$H = k_c A.$$

6. SAVE the constants ESQ, ECC, B, EXPT_c, C_c, D, F, G, LAMO, and H.

7. Provide an alternate ENTRY point named OMFWD2, passing AZICEN, FALSEN, FALSEE, ROWS, N, and GPRAD through the argument list.
8. If no more forward transformations to perform, RETURN. Output grid x's and y's, and their respective point scale factors.
9. For the next pair of ϕ and λ in input array, compute intermediate values needed for the transformation,

$$L = (\lambda - \lambda_0)B$$

$$\text{EXPT}_\phi = \left[\left(\frac{1 + \sin\phi}{1 - \sin\phi} \right) \left(\frac{1 - e \sin\phi}{1 + e \sin\phi} \right)^e \right]^{\frac{1}{2}}$$

$$P = C_c (\text{EXPT}_\phi / \text{EXPT}_c)^B$$

$$J = (P - 1/P)/2$$

$$K = (P + 1/P)/2$$

$$u = D \tan^{-1} \{ [J(G) + F(\sin L)] / \cos L \}$$

$$v = (D/2) \ln \{ [K - F(J) + G(\sin L)] / [K + F(J) - G(\sin L)] \}.$$

10. Compute the grid coordinates,

$$\text{XYGRID}(I,1) = u \sin\alpha_c + v \cos\alpha_c + x_0$$

$$\text{XYGRID}(I,2) = u \cos\alpha_c - v \sin\alpha_c + y_0.$$

11. Compute the point scale factor,

$$\text{SF}(I) = H(1 - e^2 \sin^2\phi)^{\frac{1}{2}} \cos(u/D) / (\cos\phi \cos L).$$

12. Repeat from step 8 for the next GP to be transformed.

References:

1. Snyder, 1983, pp. 78-83.
2. Vincenty, 1984b.

Oblique Mercator Inverse Transformation (OMINV)

Input:

AMAJ	Semimajor axis of ellipsoid (a).
FINV	Reciprocal of flattening (1/f).
PHICEN	Geodetic latitude of the center point of the projection (ϕ_c).
LAMCEN	Geodetic longitude of the center point (λ_c).
SFCEN	Scale factor at the center point (k_c).
AZICEN	Geodetic azimuth at the center point of the skewed center line (α_c).
FALSEN	False northing (y_0).
FALSEE	False easting (x_0).
ROWS	Number of rows declared for arrays GPRAD and XYGRID in the calling program.
N	Number of positions to be transformed.
XYGRID(I,1)	x coordinate, relative to the grid origin, of the i^{th} position to be transformed.
XYGRID(I,2)	y coordinate relative to the grid origin.

Output:

GPRAD(I,1)	Geodetic latitude, in radians, of the transformed i^{th} position (ϕ).
GPRAD(I,2)	Longitude of the position (λ).
SF(I)	Point scale factor, by which infinitesimal geodetic length at the i^{th} position is multiplied to obtain grid length.

Other meaningful variables:

FLAT	Ellipsoidal flattening (f).
ECC	First eccentricity of ellipsoid (e).
ESQ	First eccentricity squared (e^2).
E2SQ	Second eccentricity squared (e'^2).
WSQ	Geodetic variable (W^2) representing the quantity $1 - e^2 \sin^2 \phi$.
EXPT	Natural base of logarithms raised to the T power, where T (τ) is the isometric latitude.
LAMO	Geodetic longitude of the true origin (λ_0).
CHI	Conformal latitude of the point in question (χ).
USKEW	u coordinate in the rectilinear coordinate system with the origin at the center point and u axis along the projected, skewed center line.
VSKEW	v coordinate in the skewed rectilinear coordinate system.

Algorithm:

1. Input ellipsoid parameters AMAJ and FINV.
2. Input projection parameters PHICEN, LAMCEN, SFCEN, AZICEN, FALSEN, and FALSEE.
3. Input ROWS, N, and array XYGRID.
4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e = (e^2)^{\frac{1}{2}}$$

$$e'^2 = e^2/(1 - e^2).$$

5. Compute zone constants,

$$W_c^2 = 1 - e^2 \sin^2 \phi_c$$

$$B = (1 + e^2 \cos^4 \phi_c)^{\frac{1}{2}}$$

$$A = B(1 - e^2)^{\frac{1}{2}} / W_c^2$$

$$\text{EXPT}_c = \left[\left(\frac{1 + \sin \phi_c}{1 - \sin \phi_c} \right) \left(\frac{1 - e \sin \phi_c}{1 + e \sin \phi_c} \right)^e \right]^{\frac{1}{2}}$$

$$S_c = W_c A / \cos \phi_c$$

$$C_c = S_c + (S_c^2 - 1)^{\frac{1}{2}}$$

$$J_c = (C_c - 1/C_c) / 2$$

$$D = k_c (A/B) a$$

$$F = \sin \alpha_0 = \sin \alpha_c \cos \phi_c / (W_c A)$$

$$G = \cos \alpha_0 = \cos(\sin^{-1} F)$$

$$\lambda_0 = \lambda_c - \sin^{-1}(J_c F/G) / B$$

$$H = k_c A$$

$$C_2 = e^2/2 + 5e^4/24 + e^6/12 + 13e^8/360$$

$$C_4 = 7e^4/48 + 29e^6/240 + 811e^8/11520$$

$$C_6 = 7e^6/120 + 81e^8/1120$$

$$C_8 = 4279e^8/161280$$

$$F_0 = 2(C_2 - 2C_4 + 3C_6 - 4C_8)$$

$$F_2 = 8(C_4 - 4C_6 + 10C_8)$$

$$F_4 = 32(C_6 - 6C_8)$$

$$F_6 = 128C_8.$$

6. SAVE the constants ESQ, B, EXPT_c, C_c, D, F, G, LAMO, H, F₀, F₂, F₄ and F₆

7. Provide an alternate ENTRY point named OMINV2, passing AZICEN, FALSEN,

FALSEE, ROWS, N, and XYGRID through the argument list.

8. If no more inverse transformations to perform, RETURN. Output geodetic latitudes (ϕ) and longitudes (λ), and their respective point scale factors..

9. For the next pair of grid coordinates in the input array, compute intermediate values needed for the transformation,

$$u = (x - x_0)\sin\alpha_c + (y - y_0)\cos\alpha_c$$

$$v = (x - x_0)\cos\alpha_c - (y - y_0)\sin\alpha_c$$

$$R_1 = \sinh(v/D)$$

$$R_2 = \cosh(v/D)$$

$$R_3 = \sin(u/D)$$

$$R_4 = \cos(u/D)$$

$$T = R_1F - R_3G$$

$$\text{EXPT}_\phi = \left[\left(\frac{R_2 - T}{R_2 + T} \right)^{\frac{1}{2}} / C_c \right]^{1/B} \text{EXPT}_c$$

$$\chi = 2 \tan^{-1} \left(\frac{\text{EXPT}_\phi - 1}{\text{EXPT}_\phi + 1} \right).$$

10. Compute the geodetic latitude (ϕ) and longitude (λ),

$$\phi = \chi + \sin\chi \cos\chi (F_0 + F_2 \cos^2\chi + F_4 \cos^4\chi + F_6 \cos^6\chi)$$

$$\lambda = \lambda_0 + \left\{ \tan^{-1} [(R_1G + R_3F)/R_4] \right\} / B$$

$$\text{IF } (\lambda < -\pi) \text{ GPRAD}(I,2) = \lambda + 2\pi$$

$$\text{IF } (\lambda > \pi) \text{ GPRAD}(I,2) = \lambda - 2\pi.$$

11. Compute the point scale factor,

$$\text{SF}(I) = H(1 - e^2 \sin^2\phi)^{\frac{1}{2}} R_4 / \{ \cos\phi \cos[(\lambda - \lambda_0)B] \}.$$

12. Repeat from step 8 for the next xy position to be transformed.

References:

1. Snyder, 1983, pp. 83-84.
2. Vincenty, 1984b.

Lambert Conformal Conic Projection

The Lambert conformal conic projection is often used where the area of interest lies with its longer dimension in an east-west direction. The fundamental projection parameters are central meridian, central parallel, and mapping radius of the Equator. Usually, north and south standard parallels are specified rather than a central parallel, which does not lie exactly halfway between the standard parallels. The next algorithm requires input of north and south parallels (not necessarily standard), and scale factor at the central parallel (not necessarily known). It handles three cases for defining the projection.

Case 1 - Secant projection with north and south standard parallels known: input a central scale factor of 0, and the north and south standard parallels.

Case 2 - Secant projection with central parallel and its scale factor known: input the known central scale factor, and north and south parallels equal to the central parallel.

Case 3 - Tangent projection: input the central scale factor of 1 (exactly), and north and south parallels equal to the central parallel.

The true origin of the Lambert conformal conic projection is at the intersection of the central meridian with the apex of the cone on which the projection is made. Thus, true y coordinates are all negative for projections in the northern hemisphere, and are all positive for projections in the southern hemisphere. To maintain positive-valued y coordinates in an area that was of interest when a particular zone was first defined, a latitude of false origin may have been specified to the south of the central parallel. The intersection of the central meridian with the latitude of false origin gives the false origin. Subsequent to the original definition of zone, a false northing may have been assigned to the false origin allowing the zone to be extended farther south without encountering negative coordinates. Application of the false northing and the false easting to the false origin provides the grid origin. The following algorithms are valid in either hemisphere when the conventional meanings of north and south are maintained in the above discussion.

Lambert Conformal Conic Forward Transformation (LCCFWD)

Input:

AMAJ Semimajor axis of ellipsoid (a).

FINV Reciprocal of flattening ($1/f$).

LAMCEN Central meridian of the projection (λ_0)

PHIN North parallel (ϕ_n).

PHIS South parallel (ϕ_s).

SFN Scale factor along the north (or south) parallel (k_n).

PHIFAL Geodetic latitude of false origin (ϕ_f).

FALSEN Northing value, usually 0, specified for the false origin (y_f).

FALSEE False easting (x_0).

ROWS Number of rows declared for arrays GPRAD and XYGRID in the calling program.

N Number of positions to be transformed.

BPRAD(I,1) Geodetic latitude, in radians, of the i^{th} position to be transformed (ϕ).

GPRAD(I,2) Longitude of the position (λ).

Output:

XYGRID(I,1) x coordinate, relative to the grid origin, of the transformed i^{th} position.

XYGRID(I,2) y coordinate relative to the grid origin.

SF(I) Point scale factor, by which infinitesimal geodetic length at the i^{th} position is multiplied to obtain grid length.

Other meaningful variables:

FLAT	Ellipsoidal flattening (f).
ECC	First eccentricity of ellipsoid (e).
ESQ	First eccentricity squared (e ²).
PHICEN	Central parallel (ϕ_0).
EXPT	Natural base of logarithms raised to the T power, where T (τ) is isometric latitude.
RAD	Mapping radius.
RADEQ	Mapping radius at Equator.
RADFAL	Mapping radius at latitude of false origin.
THETA	Mapping angle (θ).

Subscripts n, s, o, and f denote quantities associated with ϕ_n , ϕ_s , ϕ_o , and ϕ_f , respectively.

Algorithm:

1. Input ellipsoid parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, PHIN, PHIS, SFN, PHIFAL, FALSEN, and FALSEE.
3. Input ROWS, N, and array GPRAD.
4. For computation of zone constants, repeated computation of EXPT and of W merit function subprograms,

$$\text{EXPTAU}(\phi) = \left[\left(\frac{1 + \sin\phi}{1 - \sin\phi} \right) \left(\frac{1 - e \sin\phi}{1 + e \sin\phi} \right)^e \right]^{\frac{1}{2}}$$

$$W(\phi) = (1 - e^2 \sin^2\phi)^{\frac{1}{2}}.$$

5. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e = (e^2)^{\frac{1}{2}}$$

6. Compute zone constants,

IF ($\phi_n = \phi_s$) THEN

$$\phi_0 = \phi_n$$

$$\text{RADEQ} = k_n a \text{EXPT}_0^{\sin\phi_0} / (W_0 \tan\phi_0)$$

IF ($\phi_f \neq \phi_s$) THEN

$$\sin\phi_0 = \frac{\ln[W_n \cos\phi_s / (W_s \cos\phi_n)]}{\ln(\text{EXPT}_n / \text{EXPT}_s)}$$

$$\text{RADEQ} = k_n a \cos\phi_n (\text{EXPT}_n^{\sin\phi_0}) / (W_n \sin\phi_0)$$

$$\text{RADFAL} = \text{RADEQ} / (\text{EXPT}_f^{\sin\phi_0})$$

7. SAVE the constants ECC, ESQ, RADEQ, $\sin\phi_0$, and RADFAL.

8. Provide an alternate ENTRY point named LCCFD2, passing AMAJ, LAMCEN, FALSEN, FALSEE, ROWS, N, and GPRAD through the argument list.

9. If no more forward transformations to perform, RETURN. Output grid x's and y's and their respective point scale factors.

10. For the next pair of ϕ and λ in the input array, compute intermediate values needed for the transformation,

$$\text{RAD} = \text{RADEQ} / (\text{EXPT}^{\sin\phi_0})$$

$$\theta = (\lambda - \lambda_0) \sin\phi_0$$

11. Compute the grid coordinates,

$$\text{XYGRID}(I,1) = x_0 + \text{RAD} \sin\theta$$

$$\text{XYGRID}(I,2) = \text{RADFAL} + y_f - \text{RAD} \cos\theta$$

12. Compute the point scale factor,

$$SF(I) = W_{\phi}(\text{RADsin}\phi_0)/(a \cos\phi).$$

13. Repeat from step 9 for the next GP to be transformed.

Reference:

Vincenty, 1985.

Lambert Conformal Conic Inverse Transformation (LCCINV)

Input:

AMAJ Semimajor axis of ellipsoid (a).
FINV Reciprocal of flattening (1/f).
LAMCEN Central meridian of projection (λ_0).
PHIN North parallel (ϕ_n).
PHIS South parallel (ϕ_s).
SFN Scale factor along the north (or south) parallel (k_n).
PHIFAL Geodetic latitude of false origin (ϕ_f).
FALSEN Northing value, usually 0, specified for the false origin (y_f).
FALSEE False easting (x_0).
ROWS Number of rows declared for arrays GPRAD and XYGRID in the calling program.
N Number of positions to be transformed.
XYGRID(I,1) x coordinate, relative to the grid origin, of the i^{th} position to be transformed.
XYGRID(I,2) y coordinate relative to the grid origin.

Output:

GPRAD(I,1) Geodetic latitude, in radians, of the transformed i^{th} position (ϕ).
GPRAD(I,2) Longitude of the position (λ).
SF(I) Point scale factor, by which infinitesimal geodetic length at the i^{th} position is multiplied to obtain grid length.

Other meaningful variables:

FLAT Ellipsoidal flattening (f).
ECC First eccentricity of ellipsoid (e).
ESQ First eccentricity squared (e^2).
PHICEN Central parallel (ϕ_0).
TAU Isometric latitude (τ).
EXPT Natural base of logarithms raised to the T power, where T (τ) is the isometric latitude.
W Geodetic variable representing the quantity $(1 - e^2 \sin^2 \phi)^{\frac{1}{2}}$.
RAD Mapping radius.
RADEQ Mapping radius at Equator.
RADFAL Mapping radius at latitude of false origin.
XTRUE x coordinate relative to the true origin.
YPRIME Distance along central meridian from apex of projection to y coordinate of the point in question.
THETA Mapping angle (θ).

Subscripts n, s, o, and f denote quantities associated with ϕ_n , ϕ_s , ϕ_o , and ϕ_f , respectively.

Algorithm:

1. Input ellipsoid parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, PHIN, PHIS, FALSEE, PHIFAL, and FALSEN.
3. Input ROWS, N, and array XYGRID.
4. For computation of zone constants, repeated computation of EXPT and of W merit function subprograms,

$$\text{EXPTAU}(\phi) = \left[\left(\frac{1 + \sin\phi}{1 - \sin\phi} \right) \left(\frac{1 - e \sin\phi}{1 + e \sin\phi} \right)^e \right]^{\frac{1}{2}}$$

$$W(\phi) = (1 - e^2 \sin^2\phi)^{\frac{1}{2}}.$$

5. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e = (e^2)^{\frac{1}{2}}.$$

6. Compute zone constants,

IF ($\phi_n = \phi_s$) THEN

$$\phi_o = \phi_n$$

$$\text{RADEQ} = k_n a \text{EXPT}_o^{\sin\phi_o} / (W_o \tan\phi_o)$$

IF ($\phi_n \neq \phi_s$) THEN

$$\sin\phi_o = \frac{\ln[W_n \cos\phi_s / (W_s \cos\phi_n)]}{\ln(\text{EXPT}_n / \text{EXPT}_s)}$$

$$\text{RADEQ} = k_n a \cos\phi_n (\text{EXPT}_n^{\sin\phi_o}) / (W_n \sin\phi_o)$$

$$\text{RADFAL} = \text{RADEQ} / (\text{EXPT}_f^{\sin\phi_o}).$$

7. SAVE the constants ECC, ESQ, RADEQ, $\sin\phi_o$, and RADFAL.
8. Provide an alternate ENTRY point named LCCIN2, passing AMAJ, LAMCEN, FALSEN, FALSEE, ROWS, N, and XYGRID through the argument list.

9. If no more forward transformations to perform, RETURN. Output geodetic latitudes (ϕ) and longitudes (λ), and their respective point scale factors.

10. For the next pair of grid coordinates in the input array, compute intermediate values needed for the transformation,

$$XTRUE = x - x_0$$

$$YPRIME = RADFAL + y_f - y$$

$$\theta = \tan^{-1}(XTRUE/YPRIME)$$

$$RAD = (XTRUE^2 + YPRIME^2)^{\frac{1}{2}}$$

$$EXPT_{\phi} = |RADEQ/RAD| (1/\sin\phi_0).$$

11. Compute an initial approximation of the sine of the geodetic latitude, giving it the algebraic sign of ϕ_0 ,

$$\sin\phi_1 = (EXPT_{\phi}^2 - 1)/(EXPT_{\phi}^2 + 1).$$

12. Iterate for $\sin\phi_{j+1}$ three times ($j = 1,2,3$) as follows:

$$F_1 = \ln(EXPT_{\phi_j}/EXPT_{\phi})$$

$$F_2 = 1/(1 - \sin^2\phi_j) - e^2/(1 - e^2\sin^2\phi_j)$$

$$\sin\phi_{j+1} = \sin\phi_j - F_1/F_2.$$

13. Compute the geodetic latitude (ϕ) and longitude (λ),

$$\phi = \sin^{-1}(\sin\phi_4)$$

$$\lambda = \lambda_0 + \theta/\sin\phi_0$$

$$\text{IF } (\lambda \leq -\pi) \text{ GPRAD}(I,2) = \lambda + 2\pi$$

$$\text{IF } (\lambda > \pi) \text{ GPRAD}(I,2) = \lambda - 2\pi$$

14. Compute the point scale factor,

$$SF(I) = (1 - e^2\sin^2\phi_4)^{\frac{1}{2}}RAD\sin\phi_0/(a \cos\phi_4).$$

15. Repeat from step 9 for the next xy position to be transformed.

Reference:

Vincenty, 1985.

Polyconic Projection

There are several different polyconic projections. Each has a central meridian represented by a straight line and parallels of latitude represented by nonconcentric arcs of circles, the centers of which fall on the extension of the central meridian. In this report, the term "polyconic" refers to the ordinary (or American) polyconic projection. This projection has application in cartography, but not in surveying.

Scale along the central meridian is constant and equal to scale along all parallels of latitude. Elsewhere, scale varies with direction. The ordinary polyconic projection is not conformal, so point scale factors are not determined in the following algorithms.

The true origin of the polyconic projection lies at the intersection of the central meridian with the Equator. True planar coordinates are reckoned from the true origin. Grid coordinates are reckoned from a grid origin, which is defined by a specified latitude and a false easting. The algorithm for the inverse transformation (and the projection itself for that matter) must not be used at longitudes greater than 90° from the central meridian.

Polyconic Forward Transformation (PCFWD)

Input:

AMAJ	Semimajor axis of ellipsoid (a).
FINV	Reciprocal of flattening ($1/f$).
LAMCEN	Central meridian of projection (λ_0).
FALSEE	False easting (x_0).
PHIG	Geodetic latitude of the grid origin (ϕ_g).
SFCEN	Scale factor along the central meridian (for the case at hand, a cartographic scale factor) (k_0).
ROWS	Number of rows declared for arrays GPRAD and XYGRID in the calling program.

N Number of positions to be transformed.

GPRAD(I,1) Geodetic latitude, in radians, of the i^{th} position to be transformed (ϕ).

GPRAD(I,2) Longitude of the position (λ).

Output:

XYGRID(I,1) x coordinate, relative to the grid origin, of the transformed i^{th} position.

XYGRID(I,2) y coordinate relative to the grid origin.

Other meaningful variables:

FLAT Ellipsoidal flattening (f).

ESQ First eccentricity of ellipsoid, squared (e^2).

ECC3 Third eccentricity (n).

RN Radius of curvature in the prime vertical.

RREC Radius of rectifying sphere (r).

OMEGA Rectifying latitude of the point in question (ω).

OMEGAG Rectifying latitude of grid origin (ω_g).

YVALUE y value of true origin relative to grid origin (y_0).

S Meridional distance.

THETA Mapping angle (θ).

● XTRUE x coordinate relative to the true origin of the projection.

YTRUE y coordinate relative to the true origin.

Algorithm:

1. Input ellipsoidal parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, FALSEE, PHIG, and SFCEN.
3. Input ROWS, N, and array GPRAD.

4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$n = f/(2 - f).$$

5. Compute constants for meridional distances,

$$r = a(1 - n)(1 - n^2)(1 + 9n^2/4 + 225n^4/64)$$

$$A_2 = -3n/2 + 9n^3/16$$

$$A_4 = 15n^2/16 - 15n^4/32$$

$$A_6 = -35n^3/48$$

$$A_8 = 315n^4/512$$

$$B_0 = 2(A_2 - 2A_4 + 3A_6 - 4A_8)$$

$$B_2 = 8(A_4 - 4A_6 + 10A_8)$$

$$B_4 = 32(A_6 - 6A_8)$$

$$B_6 = 128A_8.$$

6. Determine the y value of the true origin relative to the grid origin
(which is the meridional distance to the grid origin),

$$\text{IF } (\phi_g = 0) \text{ YVALUE} = 0$$

$$\text{IF } (\phi_g \neq 0) \text{ THEN}$$

$$\omega_g = \phi_g + \sin\phi_g \cos\phi_g (B_0 + B_2 \cos^2\phi_g + B_4 \cos^4\phi_g + B_6 \cos^6\phi_g)$$

$$\text{YVALUE} = -k_0 \omega_g r.$$

7. SAVE the constants ESQ, RREC, YVALUE, B₀, B₂, B₄, and B₆.

8. Provide an alternate ENTRY point named PCFWD2, passing AMAJ, LAMCEN, FALSEE, SFCEN, ROWS, N, and GPRAD through the argument list.
9. If no more forward transformations to perform, RETURN. Output grid x's and y's.
10. For the next pair of ϕ and λ in the input array, compute intermediate values needed for the transformation,

$$\theta = (\lambda - \lambda_0) \sin \phi$$

$$\omega = \phi + \sin \phi \cos \phi (B_0 + B_2 \cos^2 \phi + B_4 \cos^4 \phi + B_6 \cos^6 \phi)$$

$$S = \omega r$$

$$RN = a / (1 - e^2 \sin^2 \phi)^{\frac{1}{2}}$$

11. Compute the plane coordinates relative to the true origin,

IF ($\phi = 0$) THEN

$$XTRUE = k_0 a (\lambda - \lambda_0)$$

$$YTRUE = 0$$

IF ($\phi \neq 0$) THEN

IF ($\theta = 0$) THEN

$$XTRUE = 0$$

$$YTRUE = k_0 S$$

IF ($\theta \neq 0$) THEN

$$XTRUE = k_0 (RN \sin \theta / \tan \phi)$$

$$YTRUE = k_0 [S + RN(1 - \cos \theta) / \tan \phi].$$

12. Compute the grid coordinates,

$$XYGRID(I,1) = XTRUE + FALSEE$$

$$XYGRID(I,2) = YTRUE + YVALUE.$$

13. Repeat from step 9 for the next GP to be transformed.

References:

1. Snyder, 1983, pp. 130.
2. Vincenty, 1984a (for computation of meridional distance).

Polyconic Inverse Transformation (PCINV)

Input:

- AMAJ** Semimajor axis of ellipsoid (a).
- FINV** Reciprocal of flattening ($1/f$).
- LAMCEN** Central meridian of projection (λ_0).
- FALSEE** False easting (x_0).
- PHIG** Geodetic latitude of the grid origin (ϕ_g).
- SFCEN** Scale factor along the central meridian (for the case at hand, a cartographic scale factor) (k_0).
- ROWS** Number of rows declared for arrays GPRAD and XYGRID in the calling program.
- N** Number of positions to be transformed.
- XYGRID(I,1)** x coordinate, relative to the grid origin, of the i^{th} position to be transformed.
- XYGRID(I,2)** y coordinate relative to the grid origin.

Output:

- GPRAD(I,1)** Geodetic latitude, in radians, of the transformed i^{th} position (ϕ).
- GPRAD(I,2)** Longitude of the position (λ).

Other meaningful variables:

FLAT	Ellipsoidal flattening (f).
ESQ	First eccentricity of ellipsoid, squared (e^2).
ECC3	Third eccentricity (n).
RREC	Radius of rectifying sphere (r).
OMEGA	Rectifying latitude of the point in question (ω).
OMEGAG	Rectifying latitude of grid origin (ω_g).
YVALUE	y value of true origin relative to grid origin (y_0).
XTRUE	x coordinate relative to the true origin of the projection.
YTRUE	y coordinate relative to the true origin.

Algorithm:

1. Input ellipsoidal parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, FALSEE, PHIG, and SFCEN.
3. Input ROWS, N, and array XYGRID.
4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$n = f/(2 - f).$$

5. Compute constants for meridional distances,

$$\text{FACTOR} = (1 - n)(1 - n^2)(1 + 9n^2/4 + 225n^4/64)$$

$$r = a(\text{FACTOR})$$

$$A_2 = -3n/2 + 9n^3/16$$

$$A_4 = 15n^2/16 - 15n^4/32$$

$$A_6 = -35n^3/48$$

$$A_8 = 315n^4/512$$

$$B_0 = 2(A_2 - 2A_4 + 3A_6 - 4A_8)$$

$$B_2 = 8(A_4 - 4A_6 + 10A_8)$$

$$B_4 = 32(A_6 - 6A_8)$$

$$B_6 = 128A_8$$

$$C_0 = 1 - 2A_2 + 4A_4 - 6A_6 + 8A_8$$

$$C_2 = 4(A_2 - 8A_4 + 27A_6 - 64A_8)$$

$$C_4 = 32(A_4 - 9A_6 + 40A_8)$$

$$C_6 = 64(3A_6 - 32A_8)$$

$$C_8 = 1024A_8$$

6. Determine the y value of the true origin relative to the grid origin (which is the meridional distance to the grid origin),

$$\text{IF } (\phi_g = 0) \text{ YVALUE} = 0$$

$$\text{IF } (\phi_g \neq 0) \text{ THEN}$$

$$\omega_g = \phi_g + \sin\phi_g \cos\phi_g (B_0 + B_2 \cos^2\phi_g + B_4 \cos^4\phi_g + B_6 \cos^6\phi_g)$$

$$\text{YVALUE} = -k_0 \omega_g r$$

7. SAVE the constants ESQ, FACTOR, RREC, YVALUE, B₀, B₂, B₄, B₆, C₀, C₂, C₄, C₆, and C₈.
8. Provide an alterante ENTRY point named PCINV2, passing AMAJ, LAMCEN, FALSEE, SFCEN, ROWS, N, and XYGRID through the argument list.
9. If no more inverse transformations to perform, RETURN. Output geodetic latitudes (ϕ) and longitudes (λ).
10. For the next pair of grid coordinates in the input array, compute plane coordinates relative to the true origin,

$$\text{XTRUE} = x - x_0$$

$$\text{YTRUE} = y - y_0$$

11. IF (YTRUE = 0) THEN

$$\phi = 0$$

$$\lambda = \lambda_0 + XTRUE/(k_0 a).$$

Skip to step 17.

12. Compute intermediate values needed for the transformation,

$$A = YTRUE/(k_0 a)$$

$$B = [XTRUE/(k_0 a)]^2 + A^2.$$

13. Let the first approximation of ϕ equal A,

$$\phi_1 = A.$$

14. Compute the next approximation of latitude (ϕ_{j+1} , $j = 1, 2, 3, \dots$),

$$C_j = (1 - e^2 \sin^2 \phi_j)^{\frac{1}{2}} \tan \phi_j$$

$$\omega_j = \phi_j + \sin \phi_j \cos \phi_j (B_0 + B_2 \cos^2 \phi_j + B_4 \cos^4 \phi_j + B_6 \cos^6 \phi_j)$$

$$R_j = \omega_j \text{FACTOR}$$

$$T_j = \text{FACTOR}(C_0 + C_2 \cos^2 \phi_j + C_4 \cos^4 \phi_j + C_6 \cos^6 \phi_j + C_8 \cos^8 \phi_j)$$

$$\phi_{j+1} = \phi_j - \{A(C_j R_j + 1) - R_j - C_j(R_j^2 + B)/2\} / \{2e^2 \sin \phi_j \cos \phi_j (R_j^2 + B - 2R_j A)/(4C_j) + (A - R_j)[C_j T_j - 1/(\sin \phi_j \cos \phi_j)] - T_j\}.$$

15. Iterate step 13 until $|\phi_{j+1} - \phi_j| < 5 \times 10^{-9}$ rad.

16. Compute the longitude,

$$\lambda = \lambda_0 + \sin^{-1}[C_j XTRUE/(k_0 a)]/\sin \phi.$$

17. Convert output to conventional range,

$$\text{IF } (\lambda \leq -\pi) \text{ GPRAD}(I, 2) = \lambda + 2\pi$$

$$\text{IF } (\lambda > \pi) \text{ GPRAD}(I, 2) = \lambda - 2\pi.$$

18. Repeat from step 9 for the next xy position to be transformed.

References:

1. Snyder, 1983, pp. 130-131.
2. Vincenty, 1984a (for computation of meridional distances).

Azimuthal Equidistant Projection

By definition, plane distances and azimuths from the center point of the azimuthal equidistant projection to all other points are equal to geodetic distances and azimuths. Computations involving geodetic distances and azimuths therefore lie at the heart of the transformation equations. Such computational procedures have been derived to varying degrees of accuracy and accompanying complexity. The following algorithms provide sufficient accuracy for geodetic survey purposes out to several hundred kilometers from the center point.

The azimuthal equidistant projection has been used for mapping but not charting in Guam and on individual islands of Micronesia. In Guam, both "semi-rigorous" (because certain approximations are unavoidable) and "approximate" transformation formulas have been used. The commonly used approximate method was derived by Claire (1973) to meet the needs of public and private organizations using electronic data processing equipment. Approximate inverse transformation formulas are not the exact reverse of approximate forward transformation formulas. Therefore, for our purposes, a semi-rigorous approach is appropriate.

A true origin for the azimuthal equidistant projection is specified at the center of the area of interest. A grid origin is defined by specifying coordinates for the true origin. When used for surveying, scale is held true at the true origin. Reducing scale at the true origin would contradict the main purpose of the projection, which is to provide correct geodetic distances from the center point. Of course, a cartographic scale factor may be applied to all coordinates for mapping purposes. Point scale factor is not computed for the azimuthal equidistant projection because the projection is not conformal.

Azimuthal Equidistant Forward Transformation (AEFWD)

Input:

AMAJ	Semimajor axis of ellipsoid (a).
FINV	Reciprocal of flattening (1/f).
LAMCEN	Central meridian (λ_0).

PHICEN Geodetic latitude of the center point (ϕ_0).
SFCEN Scale factor at the center point (for the case at hand, a
cartographic scale factor) (k_0).
FALSEN False northing (y_0).
FALSEE False easting (x_0).
ROWS Number of rows declared for the arrays GPRAD and XYGRID in the
calling program.
N Number of positions to be transformed.
GPRAD(I,1) Geodetic latitude, in radians, of the i^{th} position to be
transformed (ϕ).
GPRAD(I,2) Longitude of the position (λ).

Output:

XYGRID(I,1) x coordinate, relative to the grid origin, of the transformed
 i^{th} position.
XYGRID(I,2) y coordinate relative to the grid origin.

Other meaningful variables:

FLAT Ellipsoidal flattening (f).
ESQ First eccentricity of ellipsoid, squared (e^2).
ECC2 Second eccentricity (e').
E2SQ Second eccentricity squared (e'^2).
RN Radius of curvature in the prime vertical.
RNCEN RN at the center point (RN_0).
TAU Isometric latitude (τ).

AZI	Geodetic azimuth, measured clockwise from north, of the normal section from the central point to the point in question (α).
SIGMA	The angle at the intersection of the Earth's rotational axis with the normal to the ellipsoid through the central point of the projection, such angle between the normal and line segment running to the point in question (σ).
ETA	Geodetic variable (η) representing the quantity $e' \cos \phi$.
DIST	Geodetic distance between center point and point in question.
XTRUE	x coordinate relative to the true origin of the projection.
YTRUE	y coordinate relative to the true origin.

Algorithm:

1. Input ellipsoidal parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, PHICEN, SFCEN, FALSEN, and FALSEE.

3. Input ROWS, N, and array GPRAD.

4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e'^2 = e^2/(1 - e^2)$$

$$e' = (e'^2)^{\frac{1}{2}}$$

5. Compute constants needed for the transformation,

$$RN_0 = a/(1 - e^2 \sin^2 \phi_0)^{\frac{1}{2}}$$

$$G_0 = e' \sin \phi_0$$

$$GSQ_0 = e'^2 \sin^2 \phi_0$$

$$\eta_0 = e' \cos \phi_0$$

6. SAVE the constants ESQ , RN_0 , G_0 , GSQ_0 , and ETA_0 .
7. Provide an alternate ENTRY point named AEFWD2, passing AMAJ, LAMCEN, SFCEN, FALSEN, FALSEE, ROWS, N, and GPRAD through the argument list.
8. If no more forward transformations to perform, RETURN. Output geodetic latitudes (ϕ) and longitudes (λ).
9. For the next pair of ϕ and λ in the input array, compute intermediate values needed for the transformation. In finding the arctangent of τ use the intrinsic FORTRAN function DATAN because $-\pi/2 < \tau < \pi/2$. For α , however, use DATAN2.

$$\tau = \tan^{-1}[(1 - e^2)\tan\phi + e^2RN_0\sin\phi_0/(RN\cos\phi)]$$

$$\alpha = \tan^{-1}\{\sin(\lambda - \lambda_0)/[\cos\phi_0\tan\tau - \sin\phi_0\cos(\lambda - \lambda_0)]\}$$

$$\text{IF } (\alpha = 0) \quad \sigma = \tau - \phi_0$$

$$\text{IF } (\alpha = \pi) \quad \sigma = \phi_0 - \tau$$

$$\text{IF } (\alpha \neq 0 \text{ and } \alpha \neq \pi) \quad \sigma = \sin^{-1}[\sin(\lambda - \lambda_0)\cos\tau/\sin\alpha]$$

$$H = \eta_0\cos\alpha$$

$$HSQ = H^2$$

$$\text{DIST} = RN_0\sigma \left\{ 1 - \sigma^2HSQ(1 - HSQ)/6 + (\sigma^3/8)G_0H(1 - 2HSQ) + (\sigma^4/120) \right. \\ \left. [HSQ(4 - 7HSQ) - 3GSQ_0(1 - 7HSQ)] - (\sigma^5/48)G_0H \right\}$$

10. Compute the true coordinates,

$$XTRUE = k_0\text{DIST}\sin\alpha$$

$$YTRUE = k_0\text{DIST}\cos\alpha.$$

11. Compute the grid coordinates,

$$XYGRID(I,1) = XTRUE + x_0$$

$$XYGRID(I,2) = YTRUE + y_0.$$

12. Repeat from step 8 for the next GP to be transformed.

Reference:

Snyder, 1983, pp. 188-189.

Azimuthal Equidistant Inverse Transformation (AEINV)

Input:

AMAJ Semimajor axis of ellipsoid (a).

FINV Reciprocal of flattening ($1/f$).

LAMCEN Central meridian (λ_0).

PHICEN Geodetic latitude of the center point (ϕ_0).

SFCEN Scale factor at the center point (for the case at hand, a cartographic scale factor) (k_0).

FALSEN False northing (y_0).

FALSEE False easting (x_0).

ROWS Number of rows declared for the arrays GPRAD and XYGRID in the calling program.

N Number of positions to be transformed.

XYGRID(I,1) x coordinate, relative to the grid origin, of the i^{th} position to be transformed.

XYGRID(I,2) y coordinate relative to the grid origin.

Output:

GPRAD(I,1) Geodetic latitude, in radians, of the transformed i^{th} position (ϕ).

GPRAD(I,2) Longitude of the position (λ).

Other meaningful variables:

FLAT	Ellipsoidal flattening (f).
ESQ	First eccentricity of ellipsoid, squared (e^2).
E2SQ	Second eccentricity squared (e'^2).
ETASQ	Geodetic variable (η^2) representing the quantity $e'^2 \cos^2 \phi$.
RN	Radius of curvature in the prime vertical.
RNCEN	RN at the center point (RN_0).
DIST	Geodetic distance between center point and point in question.
AZI	Geodetic azimuth, measured clockwise from north, of the normal section from the central point to the point in question (α).
TAU	Isometric latitude (τ).
XTRUE	x coordinate relative to the true origin of the projection.
YTRUE	y coordinate relative to the true origin.

Algorithm:

1. Input ellipsoid parameters AMAJ and FINV.
2. Input projection parameters LAMCEN, PHICEN, SFCEN, FALSEN, and FALSEE.

3. Input ROWS, N, and array XYGRID.

4. Compute ellipsoidal constants,

$$f = 1/(1/f)$$

$$e^2 = f(2 - f)$$

$$e'^2 = e^2/(1 - e^2).$$

5. Compute constants needed for the transformation,

$$RN_0 = a/(1 - e^2 \sin^2 \phi_0)^{1/2}$$

$$\eta_0^2 = e'^2 \cos^2 \phi_0.$$

6. SAVE the constants ESQ, E2SQ, RN₀ and ETASQ₀.
7. Provide an alternate ENTRY point named AEINV2, passing AMAJ, LAMCEN, SFCEN, FALSEN, FALSEE, ROWS, N, and XYGRID through the argument list.
8. If no more inverse transformations to perform, RETURN. Output geodetic latitudes (ϕ) and longitudes (λ).
9. For the next pair of grid coordinates in the input array, compute intermediate values needed for the transformation. In finding the arctangent of α , use the intrinsic FORTRAN function DATAN2. For ϕ use DATAN because $-\pi/2 \leq \phi \leq \pi/2$.

$$XTRUE = x - x_0$$

$$YTRUE = y - y_0$$

$$DIST = (XTRUE^2 + YTRUE^2)^{1/2} / k_0$$

$$\alpha = \tan^{-1}(XTRUE/YTRUE)$$

$$A = \eta_0^2 \cos^2 \alpha$$

$$B = 3e'^2(1 + A)\sin\phi_0 \cos\phi_0 \cos\alpha$$

$$D = DIST/RN_0$$

$$E = D + A(1 - A)D^3/6 - B(1 - 3A)D^4/24$$

$$F = 1 + A(E^2/2) - B(E^3/6)$$

$$\tau = \sin^{-1}(\sin\phi_0 \cos E + \cos\phi_0 \sin E \cos\alpha).$$

10. Compute the geodetic latitude (ϕ) and longitude (λ),

$$\phi = \tan^{-1}[(\tan\tau - e^2 F \sin\phi_0 / \cos\tau) / (1 - e^2)]$$

$$\lambda = \lambda_0 + \sin^{-1}(\sin\alpha \sin E / \cos\tau)$$

$$\text{IF } (\lambda \leq -\pi) \text{ GPRAD}(I,2) = \lambda + 2\pi$$

$$\text{IF } (\lambda > \pi) \text{ GPRAD}(I,2) = \lambda - 2\pi.$$

11. Repeat from step 8 for the next pair of xy coordinates to be transformed.

Reference:

Snyder, 1983, pp. 191-192.

Datum Transformation

Geodetic Datums

A classical horizontal geodetic datum is simple in concept. It is a mathematical surface, an ellipsoid, oriented by definition to the topographic surface of the Earth. On an ideal planet, survey observations could be entered directly into an ellipsoidal model of the Earth, and accurate positions would be obtained.

In reality, survey observations are affected by the distribution of mass within the Earth. Instruments used in classical geodetic surveying are oriented to the Earth by gravity, which is irregular due to uneven distribution of mass. Observations are therefore "tilted" at varying degrees with respect to the ellipsoid. The tilt is given in geodetic terms as "deflection of the vertical". Additionally, position computations must account for the irregular topography over which the observations were obtained. Thus, the mathematical surface upon which position computations are made is only a good approximation of the physical Earth.

A minimum of five parameters are needed to define a geodetic datum. Two of the parameters define the size and shape of the ellipsoid. Length of the semimajor axis and ellipsoidal flattening are commonly used. The other three parameters are latitude and longitude of a survey monument used as a starting point (or geodetic origin), and azimuth from the origin to another nearby monument. The latter three parameters could be assigned arbitrarily, but in practice they are usually determined by astronomic observations. Since they are determined by observation, they are influenced by gravity and topography by an amount that can be determined only with further observations.

If the origin happened to be located where elevation was known and where the gravity field behaved favorably, the datum would be a fairly good one. Deflection of the vertical at other survey points throughout the region in which the datum was employed would tend to average out. More typically, gravity anomalies exist in the vicinity of the origin. With the results of a preliminary survey, a deflection of the vertical can be assigned to the origin, which will minimize deflection of the vertical averaged for survey points throughout the region.

Deflection of the vertical is quantified in two ways, each requiring two parameters. Either a total deflection and an azimuth in the direction of maximum deflection, or components of the deflection in a north-south (prime meridian) and east-west (prime vertical) direction can be used. The parameters defining the deflection of the vertical at the origin, along with the other five parameters discussed earlier, amount to seven parameters classically used to define a geodetic datum.

Definition of a geodetic datum can be seen as an iterative procedure. The seven defining parameters provide the basis for computation of all other positions in the network. As geodetic and astronomic positions of more and more survey points are determined, they can be used to redefine the deflection of the vertical at the origin, thereby making the datum more closely fit the physical Earth. A geodetic datum, therefore, might be defined as the locus of all monumented points in the geodetic network rather than a list of seven parameters.

Three conclusions drawn from this discussion will help in the understanding of datum transformations:

1. Classical geodetic datums are designed to fit a particular region. A datum that is suitable for one region might not be suitable elsewhere.
2. A geodetic datum is not necessarily geocentric, but in a well-designed datum the equatorial plane of the ellipsoid is nearly coincident with the equatorial plane of the Earth, and the minor axis of the ellipsoid is nearly parallel with the average spin axis of the Earth. This is because astronomic observations interspersed throughout the network help define the orientation of the reference ellipsoid.
3. Because the geodetic datum can be thought of as the locus of all monumented points in the regional network, each subset of points, i.e., an area within the region, can be thought of as defining a slightly different datum.

Space Coordinates

A space coordinate system is simply a three-dimensional system of Cartesian coordinates. When used for satellite surveying, the origin of the XYZ space coordinate system is located at the Earth's center of mass, providing truly geocentric coordinates. For our purposes the origin of the XYZ coordinate system is located at the center of the reference ellipsoid upon which a geodetic datum is based. It may or may not be truly Earth-centered. The XY plane of the system is coincident with the equatorial plane of the ellipsoid, the Z axis is coincident with the minor axis of the ellipsoid, and the XZ plane contains the meridian of zero longitude.

Transformation Overview

There are many approaches to performing datum transformations. The following scheme is used here:

1. Geodetic coordinates in the "old" datum are converted to old XYZ coordinates.
2. The old XYZ coordinates are transformed to "new" XYZ coordinates. The XYZ coordinate transformation involves a translation of the origin, rotation of the axes, and a scale change.
3. The new XYZ coordinates are converted to geodetic coordinates in the new datum.

The conversion of geodetic coordinates to XYZ coordinates in step 1 requires knowledge of the height of the point in question above the ellipsoid. Height

above the ellipsoid is the sum of orthometric height (elevation above sea level) and geoidal separation. For cartographic applications, geoidal height, if unknown, can be assumed to be 0.

The transformation of original XYZ coordinates to new XYZ coordinates is the crux of the problem. There are a number of ways to accomplish this. The method discussed here is the Bursa-Wolf model.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{new}} = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}_0 + \begin{bmatrix} 1 & \omega & -\psi \\ -\omega & 1 & \epsilon \\ \psi & -\epsilon & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{old}} (1 + \Delta k)$$

where

$(\Delta X, \Delta Y, \Delta Z)_0$ are the components of translation of the origin (new minus old)
 $\omega, \epsilon,$ and ψ are rotations about the Z, X and Y axes, respectively
 Δk is the change in scale.

The rotation matrix in the Bursa-Wolf model is an approximation, making use of the fact that $\omega, \epsilon,$ and ψ are small; therefore the sines of these angles are approximately equal to the angles themselves, the product of the sines approximately equal 0, and the cosines approximately equal 1. Carrying out the multiplication and rearranging, the following equations are obtained:

$$X_n = \Delta X + (X_0 + \omega Y_0 - \psi Z_0)(1 + \Delta k)$$

$$Y_n = \Delta Y + (Y_0 - \omega X_0 + \epsilon Z_0)(1 + \Delta k)$$

$$Z_n = \Delta Z + (Z_0 + \psi X_0 - \epsilon Y_0)(1 + \Delta k)$$

where subscripts n and o denote new and old.

$\Delta X, \Delta Y, \Delta Z, \omega, \epsilon, \psi,$ and Δk are seven transformation parameters needed to compute new X's, Y's, and Z's. Their values are determined by a least squares method in which the model is supplied with known coordinates in the old and new coordinate systems. In practice it is difficult to determine the correct combination of translation, rotation, and scale change. If the geodetic datum was developed with care, ω , and especially ϵ and ψ are indeed very small, indicating the reference ellipsoid is properly oriented to the physical Earth. Small rotation angles complicated by distortions in the geodetic network make it practical to consider datum shifts in subregions of a geodetic network to be caused entirely by translation of XYZ origins. This argument is supported by conclusion 3, above.

Estimated values of $\Delta X, \Delta Y,$ and ΔZ to preliminary NAD 83 have been computed by the National Geodetic Survey for stations in the United States. Mean shifts calculated from these values are listed in appendix C. Mean shifts relating other local datums to the World Geodetic System of 1972 (WGS 72) have been tabulated by the Defense Mapping Agency Hydrographic/Topographic Center,

Geodesy and Surveys Department (DMATC) (Ayres 1983). To translate coordinates from these local datums to NAD 83, add $\Delta X = 0$, $\Delta Y = 0$, and $\Delta Z = 4.5$ meters to the DMATC parameters used to obtain WGS 72 coordinates. Shifts of hydrographic positions at sea are assumed to be the same as the shifts along the coast where the shore control was located.

Note well that the datum transformation thus obtained is generally not suitable for precise positioning, but can be safely used in cartographic applications. If ω , ϵ , ψ , and Δk are supplied, and if the elevation and geoidal separation are known, the transformation can be used for geodetic applications.

Input:

AMAJ Semimajor axis of ellipsoid (a).
FINV Reciprocal of flattening (1/f).
DELTAX Translation in the X direction in the sense new minus old (ΔX).
DELTAY Translation in the Y direction in the sense new minus old (ΔY).
DELTAZ Translation in the Z direction in the sense new minus old (ΔZ).
OMEGA Angle of rotation of the initial XYZ system about the Z axis, positive in the counter-clockwise direction as perceived from positive Z toward the origin (ω).
EPSIL Angle of rotation in same sense as ω , about the X axis after ω has been applied (ϵ).
PSI Angle of rotation, in same sense as ω , about the Y axis after ω and ϵ have been applied (ψ).
DELTAK Change in scale factor in the sense new minus old (Δk).
ROWS Number of rows declared for the arrays GPORAD and GPNRAD in the calling program.
N Number of positions to be transformed.
GPORAD(I,1) Old geodetic latitude, in radians, of the i^{th} position to be transformed (ϕ_0).
GPORAD(I,2) Old longitude of the position (λ_0).

ELEV(I) Elevation (orthometric height) of the i^{th} position (H).
NSEPO(I) Geoidal separation of the old datum at the i^{th} position (NSEP_0).

Output:

GPNRAD(I,1) New geodetic latitude, in radians, of the transformed i^{th} position (ϕ_n).
GPNRAD(I,2) New longitude of position (λ_n).
HTO(I) Height of the i^{th} position above the old ellipsoid (h_0).
HTN(I) Height of the i^{th} position above the new ellipsoid (h_n).
NSEPN(I) Geoidal separation of the old datum at the i^{th} position (NSEP_n).

Other Meaningful Variables:

FLAT Ellipsoidal flattening (f).
BMIN Semiminor axis of ellipsoid (b).
ESQ First eccentricity of ellipsoid, squared (e^2).
E2SQ Second eccentricity squared (e'^2).
RN Radius of curvature in prime vertical.

Suffixes O and N denote "old" and "new".

Algorithm:

1. Input ellipsoid parameters AMAJO, FINVO, AMAJN, and FINVN.
2. Input transformation parameters DELTAX, DELTAY, DELTAZ, OMEGA, EPSIL, PSI, and DELTAK.
3. Input ROWS, N, and arrays GPORAD ELEV, and NSEPO.

4. Compute ellipsoidal constants,

$$f_0 = 1/(1/f_0)$$

$$f_n = 1/(1/f_n)$$

$$b_n = a_n(1 - f_n)$$

$$e_0^2 = 2f_0 - f_0^2$$

$$e_n^2 = 2f_n - f_n^2$$

$$e_n'^2 = e_n^2/(1 - e_n^2).$$

5. SAVE the constants FLATO, FLATN, BMINN, ESQO, ESQN and E2SQN.

6. Provide an alternate ENTRY point named DATUM2, passing AMAJO, AMAJN, DELTAX, DELTAY, DELTAZ, OMEGA, EPSIL, PSI, DELTAK, ROWS, N, GPORAD, ELEV, and NSEPO through the argument list.

7. If no more transformations to perform, RETURN. Output new geodetic latitudes (ϕ_n), longitudes (λ_n), ellipsoidal heights (h_0 and h_n), and geoidal separations ($NSEP_n$).

8. For the next set of ϕ_0 , λ_0 , and h_0 in the input arrays, compute the radius of curvature and height above ellipsoid in the old system,

$$RN_0 = a/(1 - e^2 \sin^2 \phi_0)^{1/2}$$

$$h_0 = H + NSEP_0.$$

9. Compute Cartesian coordinates in the old system,

$$X_0 = (RN_0 + h_0) \cos \phi_0 \cos \lambda_0$$

$$Y_0 = (RN_0 + h_0) \cos \phi_0 \sin \lambda_0$$

$$Z_0 = [RN_0(1 - e_0^2) + h_0] \sin \phi_0.$$

10. Transform the Cartesian coordinates from the old to the new system,

$$X_n = \Delta X + (X_0 + \omega Y_0 - \psi Z_0) (1 + \Delta k)$$

$$Y_n = \Delta Y + (Y_0 - \omega X_0 + \epsilon Z_0) (1 + \Delta k)$$

$$Z_n = \Delta Z + (Z_0 + \psi X_0 - \epsilon Y_0) (1 + \Delta k).$$

11. Compute geodetic coordinates in the new datum, using the DATAN2 function for calculating λ ,

$$p = (X_n^2 + Y_n^2)^{\frac{1}{2}}$$

$$\theta = \tan^{-1}[a_n Z_n / (b_n p)]$$

$$\phi_n = \tan^{-1}[(Z_n + e_n^2 b_n \sin^3 \theta) / (p - e_n^2 a_n \cos^3 \theta)]$$

$$\lambda_n \equiv \tan^{-1}(Y_n / X_n)$$

$$\text{IF } (\lambda_n < -\pi) \text{ GPNRAD}(I,2) = \lambda_n + 2\pi$$

$$\text{IF } (\lambda_n > \pi) \text{ GPNRAD}(I,2) = \lambda_n - 2\pi$$

$$RN_n = a_n / (1 - e_n^2 \sin^2 \phi_n)^{\frac{1}{2}}$$

$$h_n \equiv p / \cos \phi_n - RN_n$$

$$\text{NSEP}_n = h_n - H.$$

12. Repeat from step 7 for the next GP to be transformed.

References:

1. Rapp, 1981, pp. 53-57, 66.
2. Bowring, 1976, pp. 323-327.

APPENDIX A. -- ELLIPSOID PARAMETERS

<u>Ellipsoid</u>	<u>Datums</u> ⁽¹⁾	<u>Semimajor Axis</u> <u>(m)</u>	<u>Flattening</u> <u>Reciprocal</u>
Clarke 1866	NAD 27 LUZON (Philippines) Old Hawaiian Guam 1963	6,378,206.4 ⁽²⁾	294.978698
GRS 80	NAD 83	6,378,137.	298.257222101
International	Old Hawaiian Provisional S. American S. American 1969	6,378,388.	297.

(1) See Ayres (1983) for a more complete listing of geodetic datums and their associated reference ellipsoids.

(2) For the 1964 Michigan state plane system use 6,378,450.04748448 (Berry 1971).

APPENDIX B. -- STATE PLANE COORDINATE SYSTEM OF 1927 PROJECTION PARAMETERS⁽¹⁾

State Zone ⁽²⁾	Code (3)	Central meridian LAMCEN ⁽⁴⁾ ° ' "	Scale (5)	N std parallel PHIN ° ' "	S std parallel PHIS ° ' "	False origin		
						Latitude PHIFAL ° ' "	Easting FALSEE ft	Northing FALSEN ft
Alabama								
TM East	0101	85 50	25,000			30 30	500,000	0
TM West	0102	87 30	15,000			30 00	500,000	0
Alaska								
OM Zone 1 ⁽⁶⁾	5001	133 40	10,000				5,000,000	5,000,000
TM Zone 2	5002	142 00	10,000			54 00	500,000	0
TM Zone 3	5003	146 00	10,000			54 00	500,000	0
TM Zone 4	5004	150 00	10,000			54 00	500,000	0
TM Zone 5	5005	154 00	10,000			54 00	500,000	0
TM Zone 6	5006	158 00	10,000			54 00	500,000	0
TM Zone 7	5007	162 00	10,000			54 00	700,000	0
TM Zone 8	5008	166 00	10,000			54 00	500,000	0
TM Zone 9	5009	170 00	10,000			54 00	600,000	0
LCC Zone 10	5010	176 00		53 50	51 50	51 00	3,000,000	0

(1) From Mitchell and Simmons (1974).

(2) TM denotes transverse Mercator, OM oblique Mercator, LCC Lambert conformal conic, and AE azimuthal equidistant.

(3) State plane coordinate zone code designated by the National Geodetic Survey (Pfeifer 1980).

(4) All central meridians except the one for Guam are negative valued (west).

(5) Scale is specified for the central line (or point) of TM, OM, and AE zones. $SFCEN = 1 - 1/Scale$. The scale factor for all LCC zones is unity at the north and south standard parallels.

(6) $PHICEN = 57^\circ 00'$. $AZICEN = \arctan(-3/4)$. False easting and false northing are in meters.

State Zone (2)	Code (3)	Central meridian LAMCEN (4)	Scale (5)	N std parallel PHIN ° '	S std parallel PHIS ° '	False origin		
						Latitude PHIFAL ° '	Easting FALSEE ft	Northing FALSEN ft
American Samoa LCC		170 00		-14 16	-14 16	-14 16	500,000	312,234.65
Arizona								
TM East	0201	110 10	10,000			31 00	500,000	0
TM Central	0202	111 55	10,000			31 00	500,000	0
TM West	0203	113 45	15,000			31 00	500,000	0
Arkansas								
LCC North	0301	92 00		36 14	34 56	34 20	2,000,000	0
LCC South	0302	92 00		34 46	33 18	32 40	2,000,000	0
California								
LCC Zone 1	0401	122 00		41 40	40 00	39 20	2,000,000	0
LCC Zone 2	0402	122 00		39 50	38 20	37 40	2,000,000	0
LCC Zone 3	0403	120 30		38 26	37 30	36 04	2,000,000	0
LCC Zone 4	0404	119 00		37 15	36 00	35 20	2,000,000	0
LCC Zone 5	0405	118 00		35 28	34 02	33 30	2,000,000	0
LCC Zone 6	0406	116 15		33 53	32 47	32 10	2,000,000	0
LCC Zone 7	0407	118 20		34 25	33 52	34 08	(7)	(7)
Colorado								
LCC North	0501	105 30		40 47	39 43	39 20	2,000,000	0
LCC Central	0502	105 30		39 45	38 27	37 50	2,000,000	0
LCC South	0503	105 30		38 26	37 14	36 40	2,000,000	0
Connecticut								
LCC	0600	72 45		41 52	41 12	40 50	600,000	0
Delaware								
TM	0700	75 25	200,000			38 00	500,000	0

(7) FALSEE = 4,186,692.58. FALSEN = 4,160,926.74.

State Zone ⁽²⁾	Code (3)	Central meridian LAMCEN ⁽⁴⁾ ° ' "	Scale (5)	N std parallel PHIN ° ' "	S std parallel PHIS ° ' "	False origin		
						Latitude	Easting	Northing
						PHIFAL ° ' "	FALSEE ft	FALSEN ft
Florida								
TM East	0901	81 00	17,000			24 20	500,000	0
TM West	0902	82 00	17,000			24 20	500,000	0
LCC North	0903	84 30		30 45	29 35	29 00	2,000,000	0
Georgia								
TM East	1001	82 10	10,000			30 00	500,000	0
TM West	1002	84 10	10,000			30 00	500,000	0
Guam⁽⁸⁾								
AE			Exact				50,000	50,000
Hawaii								
TM Zone 1	5101	155 30	30,000			18 50	500,000	0
TM Zone 2	5102	156 40	30,000			20 20	500,000	0
TM Zone 3	5103	158 00	100,000			21 10	500,000	0
TM Zone 4	5104	159 30	100,000			21 50	500,000	0
TM Zone 5	5105	160 10	Exact			21 40	500,000	0
Idaho								
TM East	1101	112 10	19,000			41 40	500,000	0
TM Central	1102	114 00	19,000			41 40	500,000	0
TM West	1103	115 45	15,000			41 40	500,000	0
Illinois								
TM East	1201	88 20	40,000			36 40	500,000	0
TM West	1202	90 10	17,000			36 40	500,000	0
Indiana								
TM East	1301	85 40	30,000			37 30	500,000	0
TM West	1302	87 05	30,000			37 30	500,000	0

(8) False easting and false northing are in meters. LAMCEN = east (+)144°44'55.50254". PHICEN = north 13°28'20.87887".

State Zone ⁽²⁾	Code (3)	Central meridian LAMCEN ⁽⁴⁾ ° ' "	Scale (5)	N std parallel PHIN ° ' "	S std parallel PHIS ° ' "	False origin		
						Latitude PHIFAL ° ' "	Easting FALSEE ft	Northing FALSEN ft
Iowa								
LCC North	1401	93 30		43 16	42 04	41 30	2,000,000	0
LCC South	1402	93 30		41 47	40 37	40 00	2,000,000	0
Kansas								
LCC North	1501	98 00		39 47	38 43	38 20	2,000,000	0
LCC South	1502	98 30		38 34	37 16	36 40	2,000,000	0
Kentucky								
LCC North	1601	84 15		38 58	37 58	37 30	2,000,000	0
LCC South	1602	85 45		37 56	36 44	36 20	2,000,000	0
Louisiana								
LCC North	1701	92 30		32 40	31 10	30 40	2,000,000	0
LCC South	1702	91 20		30 42	29 18	28 40	2,000,000	0
LCC Offshore	1703	91 20		27 50	26 10	25 40	2,000,000	0
Maine								
TM East	1801	68 30	10,000			43 50	500,000	0
TM West	1802	70 10	30,000			42 50	500,000	0
Maryland								
LCC	1900	77 00		39 27	38 18	37 50	800,000	0
Massachusetts								
LCC Mainland	2001	71 30		42 41	41 43	41 30	600,000	0
LCC Island	2002	70 30		41 29	41 17	41 00	200,000	0
Michigan (1934)								
TM East	2101	83 40	17,500			41 30	500,000	0
TM Central	2102	85 45	11,000			41 30	500,000	0
TM West	2103	88 45	11,000			41 30	500,000	0

State Zone ⁽²⁾	Code (3)	Central meridian LAMCEN ⁽⁴⁾	Scale (5)	N std parallel PHIN	S std parallel PHIS	False origin		
						Latitude PHIFAL	Easting FALSEE	Northing FALSEN
						° ' "	ft	ft
Michigan (1964)								
LCC North	2111	87 00		47 05	45 29	44 47	2,000,000	0
LCC Central	2112	84 20		45 42	44 11	43 19	2,000,000	0
LCC South	2113	84 20		43 40	42 06	41 30	2,000,000	0
Minnesota								
LCC North	2201	93 06		48 38	47 02	46 30	2,000,000	0
LCC Central	2202	94 15		47 03	45 37	45 00	2,000,000	0
LCC South	2203	94 00		45 13	43 47	43 00	2,000,000	0
Mississippi								
TM East	2301	88 50	25,000			29 40	500,000	0
TM West	2302	90 20	17,000			30 30	500,000	0
Missouri								
TM East	2401	90 30	15,000			35 50 ⁽⁹⁾	500,000	0
TM Central	2402	92 30	15,000			35 50	500,000	0
TM West	2403	94 30	17,000			36 10	500,000	0
Montana								
LCC North	2501	109 30		48 43	47 51	47 00	2,000,000	0
LCC Central	2502	109 30		47 53	46 27	45 50	2,000,000	0
LCC South	2503	109 30		46 24	44 52	44 00	2,000,000	0
Nebraska								
LCC North	2601	100 00		42 49	41 51	41 20	2,000,000	0
LCC South	2602	99 30		41 43	40 17	39 40	2,000,000	0
Nevada								
TM East	2701	115 35	10,000			34 45	500,000	0
TM Central	2702	116 40	10,000			34 45	500,000	0
TM West	2703	118 35	10,000			34 45	500,000	0

(9) Mitchell and Simmons (1974) erroneously listed 35°30'.

State Zone ⁽²⁾	Code (3)	Central meridian LAMCEN ⁽⁴⁾ ° ' "	Scale (5)	N std parallel PHIN ° ' "	S std parallel PHIS ° ' "	False origin		
						Latitude PHIFAL ° ' "	Easting FALSEE ft	Northing FALSEN ft
New Hampshire TM	2800	71 40	30,000			42 30	500,000	0
New Jersey TM	2900	74 40	40,000			38 50	2,000,000	0
New Mexico TM East	3001	104 20	11,000			31 00	500,000	0
TM Central	3002	106 15	10,000			31 00	500,000	0
TM West	3003	107 50	12,000			31 00	500,000	0
New York LCC Long Is.	3104	74 00		41 02	40 40	40 30	2,000,000	100,000
TM East	3101	74 20	30,000			40 00	500,000	0
TM Central	3102	76 35	16,000			40 00	500,000	0
TM West	3103	78 35	16,000			40 00	500,000	0
North Carolina LCC	3200	79 00		36 10	34 20	33 45	2,000,000	0
North Dakota LCC North	3301	100 30		48 44	47 26	47 00	2,000,000	0
LCC South	3302	100 30		47 29	46 11	45 40	2,000,000	0
Ohio LCC North	3401	82 30		41 42	40 26	39 40	2,000,000	0
LCC South	3402	82 30		40 02	38 44	38 00	2,000,000	0
Oklahoma LCC North	3501	98 00		36 46	35 34	35 00	2,000,000	0
LCC South	3502	98 00		35 14	33 56	33 20	2,000,000	0

State Zone (2)	Code (3)	Central meridian LAMCEN (4)	Scale (5)	N std parallel PHIN ° ' "	S std parallel PHIS ° ' "	False origin		
						Latitude PHIFAL ° ' "	Easting FALSEE ft	Northing FALSEN ft
Oregon								
LCC North	3601	120 30		46 00	44 20	43 40	2,000,000	0
LCC South	3602	120 30		44 00	42 20	41 40	2,000,000	0
Pennsylvania								
LCC North	3701	77 45		41 57	40 53	40 10	2,000,000	0
LCC South	3702	77 45		40 58	39 56	39 20	2,000,000	0
Pr. Rico & Virgin Is.								
LCC Zone 1		66 26		18 26	18 02	17 50	500,000	0
LCC Zone 2		66 26		18 26	18 02	17 50	500,000	100,000
Rhode Island								
TM	3800	71 30	160,000			41 05	500,000	0
South Carolina								
LCC North	3901	81 00		34 58	33 46	33 00	2,000,000	0
LCC South	3902	81 00		33 40	32 20	31 50	2,000,000	0
South Dakota								
LCC North	4001	100 00		45 41	44 25	43 50	2,000,000	0
LCC South	4002	100 20		44 24	42 50	42 20	2,000,000	0
Tennessee								
LCC	4100	86 00		36 25	35 15	34 40	2,000,000	100,000
Texas								
LCC North	4201	101 30		36 11	34 39	34 00	2,000,000	0
LCC N Cent.	4202	97 30		33 58	32 08	31 40	2,000,000	0
LCC Central	4203	100 20		31 53	30 07	29 40	2,000,000	0
LCC S Cent.	4204	99 00		30 17	28 23	27 50	2,000,000	0
LCC South	4205	98 30		27 50	26 10	25 40	2,000,000	0

State Zone (2)	Code (3)	Central meridian LAMCEN (4)	Scale (5)	N std parallel PHIN	S std parallel PHIS	False origin		
						Latitude PHIFAL	Easting FALSEE	Northing FALSEN
		° ' "		° ' "	° ' "	ft	ft	ft
Utah								
LCC North	4301	111 30		41 47	40 43	40 20	2,000,000	0
LCC Central	4302	111 30		40 39	39 01	38 20	2,000,000	0
LCC South	4303	111 30		38 21	37 13	36 40	2,000,000	0
Vermont								
TM	4400	72 30	28,000			42 30	500,000	0
Virginia								
LCC North	4501	78 30		39 12	38 02	37 40	2,000,000	0
LCC South	4502	78 30		37 58	36 46	36 20	2,000,000	0
Washington								
LCC North	4601	120 50		48 44	47 30	47 00	2,000,000	0
LCC South	4602	120 30		47 20	45 50	45 20	2,000,000	0
West Virginia								
LCC North	4701	79 30		40 15	39 00	38 30	2,000,000	0
LCC South	4702	81 00		38 53	37 29	37 00	2,000,000	0
Wisconsin								
LCC North	4801	90 00		46 46	45 34	45 10	2,000,000	0
LCC Central	4802	90 00		45 30	44 15	43 50	2,000,000	0
LCC South	4803	90 00		44 04	42 44	42 00	2,000,000	0
Wyoming								
TM Zone I	4901	105 10	17,000			40 40	500,000	0
TM Zone II	4902	107 20	17,000			40 40	500,000	0
TM Zone III	4903	108 45	17,000			40 40	500,000	0
TM Zone IV	4904	110 05	17,000			40 40	500,000	0

APPENDIX C. -- XYZ COORDINATE TRANSLATION PARAMETERS⁽¹⁾

NAD 27 TO PRELIMINARY NAD 83

<u>Area</u>	<u>Delta X</u> <u>(m)</u>	<u>Delta Y</u> <u>(m)</u>	<u>Delta Z</u> <u>(m)</u>
Contiguous United States			
Maine	23.1	-161.0	-187.5
Massachusetts	19.1	-161.5	-189.2
Cape May	17.5	-161.4	-186.8
Virginia	15.5	-158.4	-186.1
Pamlico Sound	12.8	-161.5	-187.2
Cape Fear	15.0	-157.2	-182.3
South Carolina	17.0	-154.0	-180.3
Georgia	11.8	-152.4	-181.9
Florida (peninsula)	11.6	-153.6	-178.5
Florida (panhandle)	12.2	-150.6	-175.7
Mississippi Delta	15.5	-150.8	-176.0
Texas	15.6	-150.8	-178.3
California			
San Diego area	21.4	-153.8	-180.9
Los Angeles area	22.5	-154.2	-182.3
Santa Maria	25.4	-155.6	-182.2
Monterey area	24.1	-158.5	-186.7
San Francisco area	22.8	-158.3	-187.3
Ukiah to Eureka	22.7	-156.8	-188.4
Klamath	19.3	-161.4	-182.9
Washington	18.7	-157.1	-187.4
Alaska			
Southeast Alaska			
Shoal Cove	17.6	-154.6	-182.9
Craig	13.4	-151.0	-182.6
Wrangell to Biorka Island	13.9	-146.4	-180.2
Juneau	10.3	-145.1	-181.3

(1) For coastal areas. Original data furnished by Astronomy and Space Geodesy Section, National Geodetic Survey, NOAA. Some values represent averages for the area; others are for single stations in the area.

<u>Area</u>	<u>Delta X</u> (m)	<u>Delta Y</u> (m)	<u>Delta Z</u> (m)
Southeast Alaska (continued)			
Haines	9.9	-143.6	-180.9
Yakutat	Irregular		
Southern Alaska			
Cape Yakataga	-0.8	-134.1	-184.7
Prince William Sound	11.7	-130.3	-177.3
Anchorage and Kenia Peninsula (exc. Nuka Island)	10.8	-136.9	-180.7
Nuka Island	17.4	-131.9	-176.1
Kamishak Bay	11.5	-137.0	-181.1
Kodiak Island	Irregular		
Alaska Peninsula			
King Salmon	Irregular		
Port Heiden	21.6	-149.6	-179.0
Cape Kekurnoi	18.4	-151.8	-181.4
Kumlik Island	17.2	-153.7	-182.8
Port Moller	17.4	-142.1	-180.1
Perryville	15.8	-143.6	-181.3
Unga Island	16.0	-143.2	-180.6
Tip of peninsula and Unimak Island	13.4	-136.1	-181.4
Aleutian Islands			
Adak	41.5	-173.3	-165.4
Attu Island	38.6	-217.9	-162.2
St. Paul Island	-20.7	-280.1	-220.9
Western Alaska			
Mosquito Point	11.6	-146.0	-183.9
Platinum	10.5	-138.8	-183.7
Cape Avinof to Dall Point	6.7	-135.0	-185.5
St. Lawrence Island	-56.9	26.2	-204.6
Unalakleet to Kotzebue	10.5	-130.7	-182.3
Point Hope	12.8	-127.0	-181.4

<u>Area</u>	<u>Delta X</u> <u>(m)</u>	<u>Delta Y</u> <u>(m)</u>	<u>Delta Z</u> <u>(m)</u>
Alaska (continued)			
North Slope			
Icy Cape	15.1	-132.6	-180.4
Smith Bay	11.6	-134.2	-180.9
Oliktok Point	8.3	-133.8	-181.2
Point Gordon	5.7	-133.4	-181.6
Barter Island	3.9	-133.4	-181.4

OLD HAWIIAN TO PRILIMINARY NAD 83

<u>Area</u>	<u>Delta X</u> <u>(m)</u>	<u>Delta Y</u> <u>(m)</u>	<u>Delta Z</u> <u>(m)</u>
Hawaiian Islands			
Kauai	-0.8	297.9	155.5
Oahu	-14.2	290.8	165.3
Molokai	-22.3	296.1	170.3
Maui	-29.1	296.0	177.6
Hawi, Hawaii	-38.1	292.5	168.6
Kileuea Crater, Hawaii	-38.7	296.1	199.9
Remainder of Hawaii	-45.7	289.3	168.9

PUERTO RICAN TO PRILIMINARY NAD 83

<u>Area</u>	<u>Delta X</u> <u>(m)</u>	<u>Delta Y</u> <u>(m)</u>	<u>Delta Z</u> <u>(m)</u>
Puerto Rico and Virgin Islands	-6.7	-64.7	102.2

APPENDIX D. -- FORTRAN SUBROUTINES

FORTRAN subroutines are listed in the order shown below. They conform to the ANSI X3.9-1978 (full-language) standard.

- MERFWD - Transforms geodetic latitude and longitude to x and y on the normal Mercator projection.
- MERINV - Transforms x and y on the normal Mercator projection to geodetic latitude and longitude.
- TMFWD - Transforms geodetic latitude and longitude to x and y on the transverse Mercator projection.
- TMINV - Transforms x and y on the transverse Mercator projection to geodetic latitude and longitude.
- OMFWD - Transforms geodetic latitude and longitude to x and y on the oblique Mercator projection.
- OMINV - Transforms x and y on the oblique Mercator projection to geodetic latitude and longitude.
- LCCFWD - Transforms geodetic latitude and longitude to x and y on the Lambert conformal conic projection.
- LCCINV - Transforms x and y on the Lambert conformal conic projection to geodetic latitude and longitude.
- PCFWD - Transforms geodetic latitude and longitude to x and y on the American polyconic projection.
- PCINV - Transforms x and y on the American polyconic projection to geodetic latitude and longitude.
- AEFWD - Transforms geodetic latitude and longitude to x and y on the azimuthal equidistant projection.
- AEINV - Transforms x and y on the azimuthal equidistant projection to geodetic latitude and longitude.
- DATUMT - Transforms geodetic latitude, longitude, and height on an "old" datum to geodetic latitude, longitude, and height on a "new" datum.


```

SUBROUTINE MERFWD (AMAJ, FINV, LANCEN, PHISF, SFPHI,
1                 FALSEN, N, ROWS, GPRAD, XYGRID, SF)
C
C     PURPOSE:  THIS ROUTINE TRANSFORMS GEODETIC COORDINATES
C               (LATITUDE, LONGITUDE) INTO NORMAL MERCATOR
C               COORDINATES (X,Y).
C
C     GLOBAL DECLARATIONS:
C     INTEGER N, ROWS
C     DOUBLE PRECISION AMAJ, FALSEN, FINV, GPRAD(ROWS,2), LANCEN,
1     PHISF, SF(ROWS), SFPHI, XYGRID(ROWS,2)
C
C     LOCAL DECLARATIONS:
C     DOUBLE PRECISION ECC, ESQ, FLAT, PI, PINUL2, SFEQ, TAU,
1     TEMP1, XTRUE, YTRUE
C
C     PARAMETER (PI      = 3.1415926535897932DO ,
1              PINUL2   = 2.0DO * PI)
C     SAVE ESQ, ECC, SFEQ
C
C     ***COMPUTE THE INITIALIZATION CONSTANTS***
C
C     FLAT = 1.0DO/FINV
C     ESQ  = (2.0DO - FLAT) * FLAT
C     ECC  = DSQRT(ESQ)
C     SFEQ = SFPHI * DCOS(PHISF) /
1         DSQRT( 1.0DO - ESQ*DSIN(PHISF)**2 )
C
C     ***ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
C     PREVIOUSLY COMPUTED***
C
C     ENTRY MERCF2 (AMAJ, LANCEN, FALSEN, N, ROWS, GPRAD, XYGRID, SF)
C
C     DO 1 I = 1, N
C       XTRUE = AMAJ * GPRAD(I,2) * SFEQ
C       TEMP1 = DSIN(GPRAD(I,1))
C       TAU   = DLOG( DSQRT
1             ( (1.0DO+TEMP1) / (1.0DO-TEMP1) *
2             ( (1.0DO-ECC*TEMP1) / (1.0DO+ECC*TEMP1) )
3             ** ECC ) )
C       YTRUE = AMAJ * TAU * SFEQ
C       IF ( (GPRAD(I,2) - LANCEN) .LE. PI
1         .AND.
2         (GPRAD(I,2) - LANCEN) .GE. -PI )
3         XYGRID(I,1) = XTRUE - AMAJ * LANCEN * SFEQ
C       IF ( (GPRAD(I,2) - LANCEN) .GT. PI )
1         XYGRID(I,1) = XTRUE - AMAJ * (LANCEN + PINUL2) * SFEQ
C       IF ( (GPRAD(I,2) - LANCEN) .LT. -PI )
1         XYGRID(I,1) = XTRUE - AMAJ * (LANCEN - PINUL2) * SFEQ
C       XYGRID(I,2) = YTRUE + FALSEN
C       SF(I)      = SFEQ *

```

```
1          DSQRT(1.000 - ESQ * TEMP1**2) /  
2          DCOS( GPRAD(I,1) )  
1  CONTINUE  
   RETURN  
   END
```

1 SUBROUTINE MERINV (AMAJ, FINV, LANCEN, PHISF, SFPHI,
 FALSEN, N, ROWS, XYGRID, GPRAD, SF)

C
 C PURPOSE: THIS ROUTINE TRANSFORMS NORMAL MERCATOR
 C COORDINATES (X,Y) INTO GEODETIC COORDINATES
 C (LATITUDE, LONGITUDE).
 C

C GLOBAL DECLARATIONS:

C INTEGER N, ROWS, SLP
 C DOUBLE PRECISION AMAJ, FALSEN, FINV, GPRAD(ROWS,2), LANCEN,
 1 PHISF, SF(ROWS), SFPHI, XYGRID(ROWS,2)

C LOCAL DECLARATIONS:

C DOUBLE PRECISION ECC, ESQ, FLAT, PHIEST, PI, PIDIV2, PINUL2,
 1 SFEQ, T, TEMP1, YTRUE

C
 C PARAMETER (PI = 3.14159265358979320 ,
 1 PIDIV2 = PI / 2.000 ,
 2 PINUL2 = PI * 2.000)

C COMMON /IO/ SLP
 C SAVE ECC, ESQ, SFEQ

C *** COMPUTE THE INITIALIZATION CONSTANTS ***

C FLAT = 1.000/FINV
 C ESQ = (2.000 - FLAT) * FLAT
 C ECC = DSQRT(ESQ)
 C SFEQ = SFPHI * DCOS(PHISF) /
 1 DSQRT(1.000 - ESQ*DSIN(PHISF)**2)

C *** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
 C PREVIOUSLY COMPUTED ***

C ENTRY MERCI2 (AMAJ, LANCEN, FALSEN, N, ROWS, XYGRID, GPRAD, SF)

C DO 2 I = 1, N
 C YTRUE = XYGRID(I,2) - FALSEN
 C COMPUTE AN INITIAL APPROXIMATION OF GEODETIC LATITUDE.
 T = 1.000 / DEXP(YTRUE/(AMAJ*SFEQ))
 GPRAD(I,1) = PIDIV2 - 2.000*DATAN(T)

C PERFORM SUCCESSIVE APPROXIMATIONS TO OBTAIN THE LATITUDE FOR
 C THIS POSITION.

C DO 1 K=1,5
 C TEMP1 = ECC*DSIN(GPRAD(I,1))
 C COMPUTE NEXT APPROXIMATION OF LATITUDE (PHIEST).
 PHIEST = PIDIV2 -
 1 2.000*DATAN(T*
 2 DSQRT(((1.000-TEMP1) / (1.000+TEMP1))
 3 **ECC))

```

      IF (DABS(PHIEST - GPRAD(I,1)) .LE. 5.D-9) THEN
C      CONVERGENCE HAS OCCURRED.
      GPRAD(I,1) = PHIEST
      GPRAD(I,2) = LANCEN + XYGRID(I,1)/(AMAJ*SFEQ)
      IF (GPRAD(I,2) .LE. -PI) GPRAD(I,2) = GPRAD(I,2) + PIMUL2
      IF (GPRAD(I,2) .GT. PI) GPRAD(I,2) = GPRAD(I,2) - PIMUL2
      SF(I) = SFEQ * DSQRT( 1.DO - ESQ * DSIN(GPRAD(I,1))**2 ) /
1      DCOS(GPRAD(I,1))
      GO TO 2
      ELSE
      GPRAD(I,1) = PHIEST
      ENDIF
1      CONTINUE
C
C      CONVERGENCE DID NOT OCCUR WITHIN 5 ITERATIONS.
C      SET LATITUDE TO BOGUS VALUE AND PRINT OUT ERROR MESSAGE.
C
      GPRAD(I,1) = PIMUL2
      WRITE (SLP,1000) I
1000  FORMAT ('1*** WARNING ***' / 'OERROR HAS OCCURRED IN SUBROUTINE M
1ERINV.' / 'OGEODETIC LATITUDE DID NOT CONVERGE IN 5 ITERATIONS FOR
2THE ', I5, 'TH POSITION.')
2      CONTINUE
      RETURN
      END

```

SUBROUTINE TMFWD (AMAJ, FINV, LANCEN, FALSEE, FALSEN, PHIFAL,
1 SFCEN, N, ROWS, GPRAD, XYGRID, SF)

C
C
C
C
C
C

PURPOSE: THIS ROUTINE TRANSFORMS GEODETIC COORDINATES
(LATITUDE, LONGITUDE) INTO TRANSVERSE MERCATOR
COORDINATES (X, Y).

GLOBAL DECLARATIONS:

INTEGER N, ROWS
DOUBLE PRECISION AMAJ, FALSEE, FALSEN, FINV, GPRAD(ROWS, 2),
1 LANCEN, PHIFAL, SF(ROWS), SFCEN, XYGRID(ROWS, 2)

C
C

LOCAL DECLARATIONS:

DOUBLE PRECISION A2, A4, A6, A8, B0, B2, B4, B6, C, E3, E4, E5,
1 E6, E7, ECC3, ESQ, E2SQ, E3SQ, ETASQ,
2 F2, F4, FLAT, L, L2, OMEGA, OMEGAF, PI,
3 PIDIV2, PINUL2, RN, RREC, S, T, T2, T4,
4 XFALSE, XTRUE, YFALSE, YTRUE, YVALUE

C
C
C
C

THE FOLLOWING ARE TEMPORARY STORAGE VARIABLES:
C, E3SQ, L2, T, T2, T4.

PARAMETER (PI = 3.1415926535897932DO,
1 PIDIV2 = 0.5DO * PI,
2 PINUL2 = 2.0DO * PI)
SAVE ESQ, E2SQ, RREC, YVALUE, B0, B2, B4, B6

C
C
C

*** COMPUTE THE INITIALIZATION CONSTANTS ***

FLAT = 1.DO / FINV
ESQ = (2.DO - FLAT) * FLAT
E2SQ = ESQ / (1.DO - ESQ)
ECC3 = FLAT / (2.DO - FLAT)
E3SQ = ECC3 ** 2
RREC = AMAJ * (1.DO - ECC3) * (1.DO - E3SQ) *
1 (1.DO + 2.25DO * E3SQ + 225.DO * E3SQ ** 2 / 64.DO)
A2 = ECC3 * (-1.5DO + 9.DO * E3SQ / 16.DO)
A4 = 15.DO * E3SQ / 32.DO * (2.DO - E3SQ)
A6 = -35.DO * E3SQ * ECC3 / 48.DO
A8 = 315.DO * E3SQ ** 2 / 512.DO
B0 = 2.DO * (A2 - 2.DO * A4 + 3.DO * A6 - 4.DO * A8)
B2 = 8.DO * (A4 - 4.DO * A6 + 10.DO * A8)
B4 = 32.DO * (A6 - 6.DO * A8)
B6 = 128.DO * A8
C = DCOS(PHIFAL)
OMEGAF = PHIFAL + C * DSIN(PHIFAL) *
1 (B0 + B2 * C ** 2 + B4 * C ** 4 + B6 * C ** 6)
YVALUE = -SFCEN * OMEGAF * RREC

C
C
C

*** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
PREVIOUSLY COMPUTED ***

```

C
ENTRY TNFWD2 (ANAJ, LANCEN, FALSEE, FALSEN, SFCEN, N, ROWS,
1          GPRAD, XYGRID, SF)
C
DO 1 I = 1,N
  C      = DCOS(GPRAD(I,1))
C
C      IF LATITUDE EQUALS PI/2, DISABLE THE TANGENT FUNCTION.
C      SINCE THE INTERNAL REPRESENTATION OF LATITUDE IN RADIANS
C      MAY NOT BE EXACTLY EQUAL TO THAT OF PIDIV2, DISABLE THE
C      TANGENT FUNCTION IF THE LATITUDE IS WITHIN 1.0D-15 OF PIDIV2.
C
  IF (DABS (GPRAD(I,1) - PIDIV2) .LT. 1.0D-15
1      .OR.
2      DABS (GPRAD(I,1) + PIDIV2) .LT. 1.0D-15) THEN
    T = 0.DO
  ELSE
    T = DTAN(GPRAD(I,1))
  ENDIF
  T2      = T**2
  T4      = T2**2
  ETASQ   = E2SQ * C**2
  IF ( (GPRAD(I,2) - LANCEN) .LE. PI
1      .AND.
2      (GPRAD(I,2) - LANCEN) .GE. -PI )
3      L = (GPRAD(I,2) - LANCEN) * C
  IF ( (GPRAD(I,2) - LANCEN) .GT. PI )
1      L = (GPRAD(I,2) - LANCEN - PIMUL2) * C
  IF ( (GPRAD(I,2) - LANCEN) .LT. -PI )
1      L = (GPRAD(I,2) - LANCEN + PIMUL2) * C
C
C      TO AVOID FLOATING POINT UNDERFLOW ERROR ON PERKIN-ELNER 3205
C      COMPUTER, REMOVE 4TH AND 6TH ORDER TERMS IN CALCULATION OF OMEGA
C      WHEN C IS NEAR ZERO (WITHIN A NEIGHBORHOOD OF 1.0D-10).
C
  IF (C .LT. 1.0D-10) THEN
    OMEGA = GPRAD(I,1) + C * DSIN(GPRAD(I,1)) * (B0 + B2*C**2)
  ELSE
    OMEGA = GPRAD(I,1) + C * DSIN(GPRAD(I,1)) *
1      (B0 + B2*C**2 + B4*C**4 + B6*C**6)
  ENDIF
  S      = OMEGA * RREC
  RN     = ANAJ / DSQRT( 1.DO - ESQ*DSIN(GPRAD(I,1))**2 )
  E3     = (1.DO - T2 + ETASQ) / 6.DO
  E4     = ( 5.DO - T2 + ETASQ * (9.DO + 4.DO*ETASQ) ) / 12.DO
  E5     = ( 5.DO - 18.DO*T2 + T4 + ETASQ*(14.DO - 58.DO*T2) ) /
1      120.DO
  E6     = ( 61.DO - 58.DO*T2 + T4 + ETASQ*(270.DO - 330.DO*T2) ) /
1      360.DO
  E7     = (61.DO - 479.DO*T2 + 179.DO*T4 - T2*T4) / 5040.DO
  F2     = (1.DO + ETASQ) / 2.DO

```

```

F4      = ( 5.DO - 4.DO*T2 + ETASQ * (9.DO - 24.DO*T2) ) / 12.DO
L2      = L**2
XTRUE   = SFCEN * RN * L *
1        ( 1.DO + L2 * ( E3 + L2*(E5 + E7*L2) ) )
YTRUE   = SFCEN * ( S + RN*T*L2 * ( 1.DO + L2*(E4 + E6*L2) ) /
1        2.DO )
XFALSE  = XTRUE
YFALSE  = YTRUE + YVALUE
XYGRID(I,1) = XFALSE + FALSEE
XYGRID(I,2) = YFALSE + FALSEN
SF(I)    = SFCEN * ( 1.DO + F2*L2 * (1.DO + F4*L2) )
1        CONTINUE
        RETURN
        END

```

SUBROUTINE THINV (ANAJ, FINV, LANCEN, FALSEE, FALSEN, PHIFAL,
 1 SFCEN, N, ROWS, XYGRID, GPRAD, SF)

C
C
C
C
C
C

PURPOSE: THIS ROUTINE TRANSFORMS TRANSVERSE MERCATOR
 COORDINATES (X,Y) INTO GEODETIC COORDINATES
 (LATITUDE, LONGITUDE).

GLOBAL DECLARATIONS:

INTEGER N, ROWS
 DOUBLE PRECISION ANAJ, FALSEE, FALSEN, FINV, GPRAD(ROWS,2),
 1 LANCEN, PHIFAL, SF(ROWS), SFCEN, XYGRID(ROWS,2)

C
C

LOCAL DECLARATIONS:

DOUBLE PRECISION A2, A4, A6, A8, B0, B2, B4, B6, C,
 1 C2, C4, C6, C8, D0, D2, D4, D6,
 2 ECC3, ESQ, E2SQ, E3SQ, ETASQ,
 3 FLAT, G2, G3, G4, G5, G6, G7, H2, H4,
 4 L, LATFP, OMEGA, OMEGAF, PI, PINUL2,
 5 Q, Q2, RNFP, RREC, S, T, T2, T4, YVALUE

C
C
C
C

THE FOLLOWING ARE TEMPORARY STORAGE VARIABLES:
 C, E3SQ, Q2, T, T2, T4.

PARAMETER (PI = 3.1415926535897932D0 ,
 1 PINUL2 = 2.0D0 * PI)
 SAVE ESQ, E2SQ, RREC, YVALUE, D0, D2, D4, D6

C
C
C

*** COMPUTE THE INITIALIZATION CONSTANTS ***

FLAT = 1.D0 / FINV
 ESQ = (2.D0 - FLAT) * FLAT
 E2SQ = ESQ / (1.D0 - ESQ)
 ECC3 = FLAT / (2.D0 - FLAT)
 E3SQ = ECC3 ** 2
 RREC = ANAJ * (1.D0 - ECC3) * (1.D0 - E3SQ) *
 1 (1.D0 + 2.25D0 * E3SQ + 225.D0 * E3SQ ** 2 / 64.D0)
 A2 = ECC3 * (-1.5D0 + 9.D0 * E3SQ / 16.D0)
 A4 = 15.D0 * E3SQ / 32.D0 * (2.D0 - E3SQ)
 A6 = -35.D0 * E3SQ * ECC3 / 48.D0
 A8 = 315.D0 * E3SQ ** 2 / 512.D0
 B0 = 2.D0 * (A2 - 2.D0 * A4 + 3.D0 * A6 - 4.D0 * A8)
 B2 = 8.D0 * (A4 - 4.D0 * A6 + 10.D0 * A8)
 B4 = 32.D0 * (A6 - 6.D0 * A8)
 B6 = 128.D0 * A8
 C2 = 1.5D0 * ECC3 * (1.D0 - 9.D0 * E3SQ / 16.D0)
 C4 = 21.D0 * E3SQ / 16.D0 - 55.D0 * E3SQ ** 2 / 32.D0
 C6 = 151.D0 * ECC3 * E3SQ / 96.D0
 C8 = 1097.D0 * E3SQ ** 2 / 512.D0
 D0 = 2.D0 * (C2 - 2.D0 * C4 + 3.D0 * C6 - 4.D0 * C8)
 D2 = 8.D0 * (C4 - 4.D0 * C6 + 10.D0 * C8)
 D4 = 32.D0 * (C6 - 6.D0 * C8)


```

D6      = 128.DO * C8
C       = DCOS(PHIFAL)
OMEGAF = PHIFAL + C * DSIN(PHIFAL) *
1      (B0 + B2*C**2 + B4*C**4 + B6*C**6)
YVALUE = -SFCEN * OMEGAF * RREC

C
C     *** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
C     PREVIOUSLY COMPUTED ***
C
ENTRY TNINV2 (ANAJ, LANCEN, FALSEE, FALSEN, SFCEN, N, ROWS,
1          XYGRID, GPRAD, SF)
C
DO 1 I = 1, N
  OMEGA = (XYGRID(I,2) - FALSEN - YVALUE) / (SFCEN * RREC)
  C     = DCOS(OMEGA)
C     TO TEST WHETHER OMEGA IS EQUAL TO PI/2 OR -PI/2, WE EQUIVALENTLY
C     TEST TO SEE IF COS(OMEGA) IS SUFFICIENTLY CLOSE TO ZERO.
  IF (DABS(C) .LE. 1.0D-10) THEN
    GPRAD(I,1) = OMEGA
    LONGITUDE IS INDETERMINATE IN THIS CASE.
    SET TO LANCEN FOR CONVENIENCE.
    GPRAD(I,2) = LANCEN
    SF(I)      = SFCEN
  ELSE
    LATFP = OMEGA + DSIN(OMEGA) * C *
1      (D0 + D2*C**2 + D4*C**4 + D6*C**6)
    RNFP = ANAJ / DSQRT( 1.DO - ESQ*DSIN(LATFP)**2 )
    ETASQ = E2SQ * DCOS(LATFP)**2
    T      = DTAN(LATFP)
    T2     = T**2
    T4     = T2**2
    G2     = -T * (1.DO + ETASQ) / 2.DO
    G3     = -(1.DO + 2.DO*T2 + ETASQ) / 6.DO
    G4     = -( 5.DO + 3.DO*T2 + ETASQ*(1.DO - 9.DO*T2) -
1      4.DO*ETASQ**2 ) / 12.DO
    G5     = ( 5.DO + 28.DO*T2 + 24.DO*T4 +
1      ETASQ*(6.DO + 8.DO*T2) ) / 120.DO
    G6     = ( 61.DO + 90.DO*T2 + 45.DO*T4 +
1      ETASQ*(46.DO - 252.DO*T2 - 90.DO*T4) ) / 360.DO
    G7     = -(61.DO + 662.DO*T2 + 1320.DO*T4 + 720.DO*T2*T4) /
1      5040.DO
    H2     = (1.DO + ETASQ) / 2.DO
    H4     = (1.DO + 5.DO*ETASQ) / 12.DO
    Q      = (XYGRID(I,1) - FALSEE) / (SFCEN * RNFP)
    Q2     = Q**2
    L      = Q * ( 1.DO + Q2*( G3 + Q2*(G5 + G7*Q2) ) )
    GPRAD(I,1) = LATFP + G2*Q2 * ( 1.DO + Q2*(G4+G6*Q2) )
    GPRAD(I,2) = LANCEN + L/DCOS(LATFP)
    IF (GPRAD(I,2) .LE. -PI) GPRAD(I,2) = GPRAD(I,2) + PINUL2
    IF (GPRAD(I,2) .GT. PI) GPRAD(I,2) = GPRAD(I,2) - PINUL2
    SF(I)   = SFCEN * ( 1.DO + H2*Q2 * (1.DO + H4*Q2) )

```

1 **ENDIF**
 CONTINUE
 RETURN
 END

SUBROUTINE ONFWD (ANAJ, FINV, PHICEN, LANCEN, SFCEN, AZICEN,
 1 FALSEN, FALSEE, N, ROWS, GPRAD, XYGRID, SF)

C
C
C
C
C
C

PURPOSE: THIS ROUTINE TRANSFORMS GEODETIC COORDINATES
 (LATITUDE, LONGITUDE) INTO OBLIQUE MERCATOR COORDINATES
 (X, Y).

GLOBAL DECLARATIONS:

INTEGER N, ROWS
 DOUBLE PRECISION ANAJ, AZICEN, FALSEE, FALSEN, FINV,
 1 GPRAD(ROWS, 2), LANCEN, PHICEN, SF(ROWS),
 2 SFCEN, XYGRID(ROWS, 2)

C
C

LOCAL DECLARATIONS:

DOUBLE PRECISION A, B, C, CC, D, ECC, ESQ, E2SQ, EXPT,
 1 EXPTC, F, FLAT, G, H, J, JC, K, L, LAMO,
 2 P, S, SC, TEMP, USKEW, VSKEW, WSQ

C
C

THE FOLLOWING ARE TEMPORARY STORAGE VARIABLES: C, S, TEMP
 SAVE ESQ, ECC, B, EXPTC, CC, D, F, G, LAMO, H

C
C
C
C

*** COMPUTE THE INITIALIZATION CONSTANTS. ***

FLAT = 1.DO / FINV
 ESQ = (2.DO - FLAT) * FLAT
 ECC = DSQRT(ESQ)
 E2SQ = ESQ / (1.DO - ESQ)
 S = DSIN(PHICEN)
 C = DCOS(PHICEN)
 WSQ = 1.DO - ESQ*S**2
 B = DSQRT(1.DO + E2SQ*C**4)
 A = B * DSQRT(1.DO - ESQ) / WSQ
 EXPTC = DSQRT((1.DO + S)/(1.DO - S) *
 1 . ((1.DO - ECC*S)/(1.DO + ECC*S)) ** ECC)
 TEMP = DSQRT(WSQ) * A
 SC = TEMP / C
 CC = SC + DSQRT(SC**2 - 1.DO)
 JC = (CC - 1.DO/CC) / 2.DO
 D = SFCEN * (A/B) * ANAJ
 F = DSIN(AZICEN) * C / TEMP
 G = DCOS(DSIN(F))
 LAMO = LANCEN - DASIN(JC*F/G) / B
 H = SFCEN * A

C
C
C
C
C

*** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
 PREVIOUSLY COMPUTED ***

ENTRY ONFWD2 (AZICEN, FALSEN, FALSEE, N, ROWS, GPRAD,
 1 XYGRID, SF)

C

DO 1 I = 1, N

```

L          = (GPRAD(I,2) - LAMO) * B
S          = DSIN(GPRAD(I,1))
EXPT      = DSQRT( (1.DO + S)/(1.DO - S) *
1          ( (1.DO - ECC*S)/(1.DO + ECC*S) ) ** ECC )
P          = CC * (EXPT/EXPTC)**B
J          = 0.5DO * (P - 1.DO/P)
K          = 0.5DO * (P + 1.DO/P)
USKEW     = D * DATAN2( (J*G + F*DSIN(L)) , DCOS(L) )
TEMP      = F*J - G*DSIN(L)
VSKEW     = D/2.DO * DLOG( (K - TEMP) / (K + TEMP) )
XYGRID(I,1) = USKEW*DSIN(AZICEN) + VSKEW*DCOS(AZICEN) + FALSEE
XYGRID(I,2) = USKEW*DCOS(AZICEN) - VSKEW*DSIN(AZICEN) + FALSEN
SF(I)     = H * DSQRT(1.DO - ESQ*S**2) * DCOS(USKEW/D) /
1          (DCOS(GPRAD(I,1)) * DCOS(L))
1 CONTINUE
RETURN
END

```

SUBROUTINE OMINV (ANAJ, FINV, PHICEN, LANCEN, SFCEN, AZICEN,
 1 FALSEN, FALSEE, N, ROWS, XYGRID, GPRAD, SF)

C
C
C
C
C
C

PURPOSE: THIS ROUTINE TRANSFORMS OBLIQUE MERCATOR
 COORDINATES (X,Y) INTO GEODETIC COORDINATES (LATITUDE,
 (LONGITUDE).

GLOBAL DECLARATIONS:

INTEGER N, ROWS
 DOUBLE PRECISION ANAJ, AZICEN, FALSEE, FALSEN, FINV,
 1 GPRAD(ROWS,2), LANCEN, PHICEN, SF(ROWS),
 2 SFCEN, XYGRID(ROWS,2)

C
C

LOCAL DECLARATIONS:

DOUBLE PRECISION A, B, C, CC, CHI, C2, C4, C6, C8,
 1 D, ECC, ESQ, E2SQ, EXPT, EXPTC,
 2 E4, E6, F, FLAT, FO, F2, F4, F6,
 3 G, H, JC, LAMO, PI, PINUL2, R1, R2, R3, R4,
 4 S, SC, TEMP, USKEW, VSKEW, WSQ

C
C
C
C

THE FOLLOWING ARE TEMPORARY STORAGE VARIABLES:

C, E4, E6, S, TEMP

PARAMETER (PI = 3.1415926535897932DO ,
 1 PINUL2 = 2.0DO * PI)
 SAVE ESQ, B, EXPTC, CC, D, F, G, LAMO, H, FO, F2, F4, F6

C
C
C

*** COMPUTE THE INITIALIZATION CONSTANTS. ***

FLAT = 1.DO / FINV
 ESQ = (2.DO - FLAT) * FLAT
 ECC = DSQRT(ESQ)
 E2SQ = ESQ / (1.DO - ESQ)
 S = DSIN(PHICEN)
 C = DCOS(PHICEN)
 WSQ = 1.DO - ESQ*S**2
 B = DSQRT(1.DO + E2SQ*C**4)
 A = B * DSQRT(1.DO - ESQ) / WSQ
 EXPTC = DSQRT((1.DO + S)/(1.DO - S) *
 1 ((1.DO - ECC*S)/(1.DO + ECC*S)) ** ECC)
 TEMP = DSQRT(WSQ) * A
 SC = TEMP / C
 CC = SC + DSQRT(SC**2 - 1.DO)
 JC = (CC - 1.DO/CC) / 2.DO
 D = SFCEN * (A/B) * ANAJ
 F = DSIN(AZICEN) * C / TEMP
 G = DCOS(DSIN(F))
 LAMO = LANCEN - DASIN(JC*F/G) / B
 H = SFCEN * A
 E4 = ESQ ** 2
 E6 = E4 * ESQ

```

C2      = ESQ/2.DO * (1.DO + 5.DO*ESQ/12.DO + E4/6.DO +
1      13.DO*E6/180.DO)
C4      = E4 * (7.DO/48.DO + 29.DO*ESQ/240.DO + 811.DO*E4/11520.DO)
C6      = E6 * (7.DO/120.DO + 81.DO*ESQ/1120.DO)
C8      = 4279.DO * E4**2 / 161280.DO
F0      = 2.DO * (C2 - 2.DO*C4 + 3.DO*C6 - 4.DO*C8)
F2      = 8.DO * (C4 - 4.DO*C6 + 10.DO*C8)
F4      = 32.DO * (C6 - 6.DO*C8)
F6      = 128.DO * C8

```

C
C
C
C

```

*** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
PREVIOUSLY COMPUTED ***

```

```

ENTRY OMINV2 (AZICEN, FALSEN, FALSEE, N, ROWS, XYGRID,
1      GPRAD, SF)

```

C

```

DO 1 I = 1,N
  USKEW      = (XYGRID(I,1) - FALSEE) * DSIN(AZICEN) +
1      (XYGRID(I,2) - FALSEN) * DCOS(AZICEN)
  VSKEW      = (XYGRID(I,1) - FALSEE) * DCOS(AZICEN) -
1      (XYGRID(I,2) - FALSEN) * DSIN(AZICEN)
  R1          = DSINH(VSKEW/D)
  R2          = DCOSH(VSKEW/D)
  R3          = DSIN(USKEW/D)
  R4          = DCOS(USKEW/D)
  TEMP        = R1*F - R3*G
  EXPT        = (DSQRT( (R2 - TEMP) / (R2 + TEMP) ) / CC)
1      ** (1.DO/B) * EXPTC
  CHI         = 2.DO * DATAN( (EXPT - 1.DO) / (EXPT + 1.DO) )
  S           = DSIN(CHI)
  C           = DCOS(CHI)
  GPRAD(I,1) = CHI + S*C * (F0 + F2*C**2 + F4*C**4 + F6*C**6)
  GPRAD(I,2) = LAMO + ( DATAN2( (R1*G + R3*F),R4 ) ) / B
  IF (GPRAD(I,2) .LE. -PI) GPRAD(I,2) = GPRAD(I,2) + PINUL2
  IF (GPRAD(I,2) .GT. PI) GPRAD(I,2) = GPRAD(I,2) - PINUL2
  SF(I)      = H * DSQRT(1.DO - ESQ*DSIN(GPRAD(I,1))**2) * R4 /
1      ( DCOS(GPRAD(I,1))*DCOS((GPRAD(I,2) - LAMO) * B) )
1 CONTINUE
RETURN
END

```

1

```

SUBROUTINE LCCFWD (AMAJ, FINV, LANCEN, PHIN, PHIS, SFN, PHIFAL,
1      FALSEN, FALSEE, N, ROWS, GPRAD, XYGRID, SF)
C
C      PURPOSE: THIS ROUTINE TRANSFORMS GEODETIC COORDINATES
C      (LATITUDE, LONGITUDE) INTO LAMBERT CONFORMAL CONIC
C      COORDINATES (X, Y).
C
C      GLOBAL DECLARATIONS:
C      INTEGER N, ROWS
C      DOUBLE PRECISION AMAJ, FALSEE, FALSEN, FINV, GPRAD(ROWS,2),
1      LANCEN, PHIFAL, PHIN, PHIS, SF(ROWS), SFN,
2      XYGRID(ROWS,2)
C
C      LOCAL DECLARATIONS:
C      DOUBLE PRECISION ECC, ESQ, EXPT, EXPTAU, FLAT, PHICEN,
1      RAD, RADEQ, RADFAL, SCEN, THETA, X, W
C
C      SAVE ECC, ESQ, RADEQ, SCEN, RADFAL
C
C      EXPTAU AND W ARE STATEMENT FUNCTIONS USED IN COMPUTATION OF
C      ZONE CONSTANTS BELOW. THE ARGUMENT OF BOTH FUNCTIONS IS THE
C      SINE OF THE APPROPRIATE LATITUDE.
C
C      EXPTAU(X) = DSQRT( (1.DO+X) / (1.DO-X) *
1      ( (1.DO-ECC*X) / (1.DO+ECC*X) ) ** ECC )
W(X) = DSQRT(1.DO - ESQ * X**2)
C
C      *** COMPUTE THE INITIALIZATION CONSTANTS. ***
C
C      FLAT = 1.DO / FINV
C      ESQ = (2.DO - FLAT) * FLAT
C      ECC = DSQRT(ESQ)
C
C      IF (PHIN .EQ. PHIS) THEN
C          PHICEN = PHIN
C          SCEN = DSIN(PHICEN)
C          RADEQ = SFN * AMAJ * EXPTAU(SCEN)**SCEN /
1      ( W(SCEN) * DTAN(PHICEN) )
C      ELSE
C          SCEN = DLOG( W(DSIN(PHIN)) * DCOS(PHIS) /
1      ( W(DSIN(PHIS)) * DCOS(PHIN)) ) /
2      DLOG( EXPTAU(DSIN(PHIN)) / EXPTAU(DSIN(PHIS)) )
C          RADEQ = SFN * AMAJ * DCOS(PHIN) * EXPTAU(DSIN(PHIN))**SCEN /
1      ( W(DSIN(PHIN)) * SCEN )
C      ENDIF
C      RADFAL = RADEQ / EXPTAU(DSIN(PHIFAL))**SCEN
C
C      ***ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
C      PREVIOUSLY COMPUTED.***
C
C      ENTRY LCCFD2 (AMAJ, LANCEN, FALSEN, FALSEE, N, ROWS,

```

```

1          GPRAD, XYGRID, SF)
C
  DO 1 I = 1,N
    RAD      = RADEQ / EXPTAU( DSIN(GPRAD(I,1)) )**SCEN
    THETA    = (GPRAD(I,2) - LANCEN) * SCEN
    XYGRID(I,1) = FALSEE + RAD * DSIN(THETA)
    XYGRID(I,2) = RADFAL + FALSEN - RAD * DCOS(THETA)
    SF(I)    = W( DSIN(GPRAD(I,1)) ) * RAD * SCEN /
              ( AMAJ * DCOS(GPRAD(I,1)) )
1
1  CONTINUE
  RETURN
  END

```



```

SUBROUTINE LCCINV (AMAJ, FINV, LANCEN, PHIN, PHIS, SFN, PHIFAL,
1      FALSEN, FALSEE, N, ROWS, XYGRID, GPRAD, SF)
C
C      PURPOSE: THIS ROUTINE TRANSFORMS LAMBERT CONFORMAL CONIC
C      COORDINATES (X,Y) INTO GEODETIC COORDINATES (LATITUDE,
C      LONGITUDE).
C
      INTEGER N, ROWS
      DOUBLE PRECISION AMAJ, FALSEE, FALSEN, FINV, GPRAD(ROWS,2),
1      LANCEN, PHIFAL, PHIN, PHIS, SF(ROWS), SFN,
2      XYGRID(ROWS,2)
C
      LOCAL DECLARATIONS:
      DOUBLE PRECISION ECC, ESQ, EXPT, EXPTAU, FLAT, F1, F2, PHICEN,
1      PI, PINUL2, RAD, RADEQ, RADFAL, SCEN, SEST,
2      THETA, X, XTRUE, YPRIME, W
C
      PARAMETER (PI      = 3.1415926535897932DO ,
1      PINUL2      = 2.0DO * PI)
      SAVE ECC, ESQ, RADEQ, SCEN, RADFAL
C
      EXPTAU AND W ARE STATEMENT FUNCTIONS USED IN COMPUTATION OF
      ZONE CONSTANTS BELOW. THE ARGUMENT OF BOTH FUNCTIONS IS THE
      SINE OF THE APPROPRIATE LATITUDE.
C
      EXPTAU(X) = DSQRT( (1.DO+X) / (1.DO-X) *
1      ( (1.DO-ECC*X) / (1.DO+ECC*X) ) ** ECC )
      W(X)      = DSQRT(1.DO - ESQ * X**2)
C
      *** COMPUTE THE INITIALIZATION CONSTANTS. ***
C
      FLAT      = 1.DO / FINV
      ESQ       = (2.DO - FLAT) * FLAT
      ECC       = DSQRT(ESQ)
C
      IF (PHIN .EQ. PHIS) THEN
          PHICEN = PHIN
          SCEN   = DSIN(PHICEN)
          RADEQ  = SFN * AMAJ * EXPTAU(SCEN)**SCEN /
1              ( W(SCEN) * DTAN(PHICEN) )
      ELSE
          SCEN   = DLOG( W(DSIN(PHIN)) * DCOS(PHIS) /
1              (W(DSIN(PHIS)) * DCOS(PHIN)) ) /
2              DLOG( EXPTAU(DSIN(PHIN)) / EXPTAU(DSIN(PHIS)) )
          RADEQ  = SFN * AMAJ * DCOS(PHIN) * EXPTAU(DSIN(PHIN))**SCEN /
1              ( W(DSIN(PHIN)) * SCEN )
      ENDIF
      RADFAL = RADEQ / EXPTAU(DSIN(PHIFAL))**SCEN
C
      *** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
      PREVIOUSLY COMPUTED. ***

```

```

C      ENTRY LCCIN2 (AMAJ, LANCEN, FALSEN, FALSEE, N, ROWS,
1      XYGRID, GPRAD, SF)
C
      DO 2 I = 1,N
          XTRUE      = XYGRID(I,1) - FALSEE
          YPRIME     = RADFAL + FALSEN - XYGRID(I,2)
          THETA      = DATAN(XTRUE / YPRIME)
          RAD        = DSQRT(XTRUE**2 + YPRIME**2)
          EXPT       = DABS(RADEQ/RAD)**(1.DO/SCEN)
C      COMPUTE AN INITIAL APPROXIMATION OF SINE OF GEODETIC LATITUDE.
          SEST       = DSIGN((EXPT**2 - 1.DO) / (EXPT**2 + 1.DO), SCEN)
C      ITERATE THREE TIMES FOR THE SINE OF THE GEODETIC LATITUDE.
          DO 1 J = 1,3
              F1     = DLOG(EXPTAU(SEST) / EXPT)
              F2     = 1.DO / (1.DO-SEST**2) - ESQ / (1.DO-ESQ*SEST**2)
              SEST   = SEST - F1/F2
1          CONTINUE
          GPRAD(I,1) = DASIN(SEST)
          GPRAD(I,2) = LANCEN + THETA / SCEN
          IF (GPRAD(I,2) .LE. -PI) GPRAD(I,2) = GPRAD(I,2) + PIMUL2
          IF (GPRAD(I,2) .GT.  PI) GPRAD(I,2) = GPRAD(I,2) - PIMUL2
          SF(I)      = W(SEST) * RAD * SCEN / ( AMAJ * DCOS(GPRAD(I,1)) )
2          CONTINUE
          RETURN
          END

```

1 SUBROUTINE PCFWD (AMAJ, FINV, LANCEN, FALSEE, PHIG,
SFCEN, N, ROWS, GPRAD, XYGRID)

C
C
C
C
C

PURPOSE: THIS ROUTINE TRANSFORMS GEODETIC COORDINATES
(LATITUDE, LONGITUDE) INTO POLYCONIC COORDINATES (X, Y).

GLOBAL DECLARATIONS:

1 INTEGER N, ROWS
DOUBLE PRECISION AMAJ, FALSEE, FINV, GPRAD(ROWS, 2), LANCEN,
PHIG, SFCEN, XYGRID(ROWS, 2)

C
C

LOCAL DECLARATIONS:

1 DOUBLE PRECISION A2, A4, A6, A8, B0, B2, B4, B6, C,
ECC3, ESQ, E3SQ, FLAT, OMEGA, OMEGAG,
2 RN, RREC, S, S1, THETA, XTRUE, YTRUE, YVALUE

C
C
C
C

THE FOLLOWING ARE TEMPORARY STORAGE VARIABLES:
C, E3SQ, S1

SAVE ESQ, RREC, YVALUE, B0, B2, B4, B6

C
C
C
C

*** COMPUTE THE INITIALIZATION CONSTANTS ***

FLAT = 1.DO / FINV
ESQ = (2.DO - FLAT) * FLAT
ECC3 = FLAT / (2.DO - FLAT)
E3SQ = ECC3 ** 2
RREC = AMAJ * (1.DO - ECC3) * (1.DO - E3SQ) *
1 (1.DO + 2.25DO * E3SQ + 225.DO * E3SQ ** 2 / 64.DO)
A2 = ECC3 * (-1.5DO + 9.DO * E3SQ / 16.DO)
A4 = 15.DO * E3SQ / 32.DO * (2.DO - E3SQ)
A6 = -35.DO * E3SQ * ECC3 / 48.DO
A8 = 315.DO * E3SQ ** 2 / 512.DO
B0 = 2.DO * (A2 - 2.DO * A4 + 3.DO * A6 - 4.DO * A8)
B2 = 8.DO * (A4 - 4.DO * A6 + 10.DO * A8)
B4 = 32.DO * (A6 - 6.DO * A8)
B6 = 128.DO * A8
IF (PHIG .EQ. 0.DO) YVALUE = 0.DO
IF (PHIG .NE. 0.DO) THEN
C = DCOS(PHIG)
OMEGAG = PHIG + DSIN(PHIG) * C * (B0 + B2 * C ** 2 + B4 * C ** 4 + B6 * C ** 6)
YVALUE = -SFCEN * OMEGAG * RREC
ENDIF

C
C
C
C

*** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
PREVIOUSLY COMPUTED ***

1 ENTRY PCFWD2 (AMAJ, LANCEN, FALSEE, SFCEN, N, ROWS,
GPRAD, XYGRID)

C

DO 1 I = 1, N

```

C      = DCOS(GPRAD(I,1))
S1     = DSIN(GPRAD(I,1))
THETA  = (GPRAD(I,2) - LANCEN) * S1
OMEGA  = GPRAD(I,1) + S1 * C * (B0 + B2*C**2 + B4*C**4 + B6*C**6)
S      = OMEGA * RREC
RN     = AMAJ / DSQRT( 1.DO - ESQ * S1**2)
IF (GPRAD(I,1) .EQ. 0.DO) THEN
  XTRUE = SFCEN * AMAJ * (GPRAD(I,2) - LANCEN)
  YTRUE = 0.DO
ELSE
  IF (THETA .EQ. 0.DO) THEN
    XTRUE = 0.DO
    YTRUE = SFCEN * S
  ELSE
    XTRUE = SFCEN * ( RN * DSIN(THETA) / DTAN(GPRAD(I,1)) )
    YTRUE = SFCEN *
1      ( S + RN * (1.DO - DCOS(THETA)) / DTAN(GPRAD(I,1)) )
  ENDIF
ENDIF
XYGRID(I,1) = XTRUE + FALSEE
XYGRID(I,2) = YTRUE + YVALUE
1 CONTINUE
RETURN
END

```

1 SUBROUTINE PCINV (AMAJ, FINV, LANCEN, FALSEE, PHIG,
SFCEN, N, ROWS, XYGRID, GPRAD)

C
C
C
C
C

PURPOSE: THIS ROUTINE TRANSFORMS POLYCONIC COORDINATES
(X,Y) INTO GEODETIC COORDINATES (LATITUDE, LONGITUDE).

GLOBAL DECLARATIONS:

INTEGER N, ROWS, SLP

1 DOUBLE PRECISION AMAJ, FALSEE, FINV, GPRAD(ROWS,2), LANCEN,
PHIG, SFCEN, XYGRID(ROWS,2)

C
C

LOCAL DECLARATIONS:

DOUBLE PRECISION A, A2, A4, A6, A8, B, B0, B2, B4, B6, C
1 CJ, CO, C2, C4, C6, C8, ECC3, ESQ, E3SQ,
2 FACTOR, FLAT, OMEGAG, OMEGAJ, PHIEST, PI,
3 PIMUL2, RJ, RREC, S1, THETA, TJ,
4 XTRUE, YTRUE, YVALUE

C
C
C

THE FOLLOWING ARE TEMPORARY STORAGE VARIABLES: C, E3SQ, S1

COMMON /IO/ SLP

PARAMETER (PI = 3.1415926535897932DO ,
1 PIMUL2 = 2.0DO * PI)
SAVE ESQ, FACTOR, RREC, YVALUE, B0, B2, B4, B6, C0, C2, C4, C6, C8

C
C
C

*** COMPUTE THE INITIALIZATION CONSTANTS ***

FLAT = 1.DO / FINV
ESQ = (2.DO - FLAT) * FLAT
ECC3 = FLAT / (2.DO - FLAT)
E3SQ = ECC3 ** 2
FACTOR = (1.DO - ECC3) * (1.DO - E3SQ) *
1 (1.DO + 2.25DO * E3SQ + 225.DO * E3SQ ** 2 / 64.DO)
RREC = AMAJ * FACTOR
A2 = ECC3 * (-1.5DO + 9.DO * E3SQ / 16.DO)
A4 = 15.DO * E3SQ / 32.DO * (2.DO - E3SQ)
A6 = -35.DO * E3SQ * ECC3 / 48.DO
A8 = 315.DO * E3SQ ** 2 / 512.DO
B0 = 2.DO * (A2 - 2.DO * A4 + 3.DO * A6 - 4.DO * A8)
B2 = 8.DO * (A4 - 4.DO * A6 + 10.DO * A8)
B4 = 32.DO * (A6 - 6.DO * A8)
B6 = 128.DO * A8
C0 = 1.DO - 2.DO * A2 + 4.DO * A4 - 6.DO * A6 + 8.DO * A8
C2 = 4.DO * (A2 - 8.DO * A4 + 27.DO * A6 - 64.DO * A8)
C4 = 32.DO * (A4 - 9.DO * A6 + 40.DO * A8)
C6 = 64.DO * (3.DO * A6 - 32.DO * A8)
C8 = 1024.DO * A8
IF (PHIG .EQ. 0.DO) YVALUE = 0.DO
IF (PHIG .NE. 0.DO) THEN
C = DCOS(PHIG)
OMEGAG = PHIG + DSIN(PHIG) * C * (B0 + B2 * C ** 2 + B4 * C ** 4 + B6 * C ** 6)

```

      YVALUE = -SFCEN * OMEGAG * RREC
    ENDIF
C
C   *** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
C   PREVIOUSLY COMPUTED ***
C
    ENTRY PCINV2 (AMAJ, LANCEN, FALSEE, SFCEN, N, ROWS,
1      XYGRID, GPRAD)
C
    DO 3 I = 1,N
      XTRUE = XYGRID(I,1) - FALSEE
      YTRUE = XYGRID(I,2) - YVALUE
      IF (YTRUE .EQ. 0.DO) THEN
        GPRAD(I,1) = 0.DO
        GPRAD(I,2) = LANCEN + XTRUE / (SFCEN * AMAJ)
      ELSE
        A = YTRUE / (SFCEN * AMAJ)
        B = ( XTRUE / (SFCEN*AMAJ) )**2 + A**2
C       SET FIRST APPROXIMATION OF GEODETIC LATITUDE.
        GPRAD(I,1) = A
C
C       PERFORM SUCCESSIVE APPROXIMATIONS TO OBTAIN THE LATITUDE
C       FOR THIS POSITION.
C
        DO 1 J= 1,5
          C = DCOS(GPRAD(I,1))
          S1 = DSIN(GPRAD(I,1))
          CJ = DSQRT(1.DO - ESQ*S1**2) * DTAN(GPRAD(I,1))
          OMEGAJ = GPRAD(I,1) +
1          S1*C * (B0 + B2*C**2 + B4*C**4 + B6*C**6)
          RJ = OMEGAJ * FACTOR
          TJ = FACTOR *
1          (C0 + C2*C**2 + C4*C**4 + C6*C**6 + C8*C**8)
C          COMPUTE NEXT APPROXIMATION OF LATITUDE (PHIEST).
          PHIEST = GPRAD(I,1) -
1          ( A*(CJ*RJ + 1.DO) - RJ - CJ*(RJ**2 + B)/2.DO ) /
2          ( 2.DO*ESQ*S1*C*(RJ**2 + B - 2.DO*RJ*A) / (4.DO*CJ)
3          + (A - RJ)*(CJ*TJ - 1.DO/(S1*C)) - TJ )
          IF (DABS(PHIEST - GPRAD(I,1)) .LT. 5.D-9) THEN
C            CONVERGENCE HAS OCCURRED.
            GPRAD(I,1) = PHIEST
            GPRAD(I,2) = LANCEN + DASIN( CJ*XTRUE / (SFCEN*AMAJ) ) /
1            DSIN(PHIEST)
            GO TO 2
          ELSE
            GPRAD(I,1) = PHIEST
          ENDIF
        CONTINUE
1
C
C   CONVERGENCE DID NOT OCCUR WITHIN 5 ITERATIONS.
C   SET LATITUDE TO BOGUS VALUE AND PRINT OUT ERROR MESSAGE.

```

C

```
GPRAD(I,1) = 8.DO * DATAN(1.DO)
WRITE (SLP,1000) I
1000  FORMAT('O*** WARNING ***' / 'OERROR HAS OCCURRED IN SUBROUTIN
1E PCINV.' / 'OGEODETC LATITUDE DID NOT CONVERGE IN 5 ITERATIONS F
2OR THE ', I5, 'TH POSITION.')
      ENDIF
2     IF (GPRAD(I,2) .LE. -PI) GPRAD(I,2) = GPRAD(I,2) + PINUL2
      IF (GPRAD(I,2) .GT. PI) GPRAD(I,2) = GPRAD(I,2) - PINUL2
3     CONTINUE
      RETURN
      END
```

```

SUBROUTINE AEFWD (AMAJ, FINV, LANCEN, PHICEN, SFCEN,
1          FALSEN, FALSEE, N, ROWS, GPRAD, XYGRID)

C
C
C          PURPOSE: THIS ROUTINE TRANSFORMS GEODETIC COORDINATES
C          (LATITUDE, LONGITUDE) INTO AZIMUTHAL EQUIDISTANT COORDINATES
C          (X,Y).
C
C          GLOBAL DECLARATIONS:
C          INTEGER N, ROWS
C          DOUBLE PRECISION AMAJ, FALSEE, FALSEN, FINV, GPRAD(ROWS,2),
1          LANCEN, PHICEN, SFCEN, XYGRID(ROWS,2)
C
C          LOCAL DECLARATIONS:
C          DOUBLE PRECISION AZI, C, DIST, ECC2, ESQ, ETAO, E2SQ,
1          FLAT, GO, GSQO, H, HSQ, PI, RN, RNCEN,
2          S, SIGMA, TAU, XTRUE, YTRUE
C
C          THE FOLLOWING ARE TEMPORARY STORAGE VARIABLES: C, S
C
C          PARAMETER (PI      = 3.1415926535897932DO)
C          SAVE ESQ, RNCEN, GO, GSQO, ETAO
C
C          *** COMPUTE THE INITIALIZATION CONSTANTS. ***
C
C          FLAT  = 1.DO / FINV
C          ESQ   = (2.DO - FLAT) * FLAT
C          E2SQ  = ESQ / (1.DO - ESQ)
C          ECC2  = DSQRT(E2SQ)
C          S     = DSIN(PHICEN)
C          C     = DCOS(PHICEN)
C          RNCEN = AMAJ / DSQRT(1.DO - ESQ * S**2)
C          GO    = ECC2 * S
C          GSQO  = GO**2
C          ETAO  = ECC2 * C
C
C          *** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
C          PREVIOUSLY COMPUTED ***
C
C          ENTRY AEFWD2 (AMAJ, LANCEN, SFCEN, FALSEN, FALSEE, N, ROWS,
1          GPRAD, XYGRID)
C
C          DO 1 I = 1, N
C          RN = AMAJ / DSQRT(1.DO - ESQ * DSIN(GPRAD(I,1))**2)
C          TAU = DATAN( (1.DO - ESQ) * DTAN(GPRAD(I,1)) +
1          ESQ * RNCEN * S / ( RN * DCOS(GPRAD(I,1)) ) )
C          AZI = DATAN2( DSIN(GPRAD(I,2)) - LANCEN,
1          C*DTAN(TAU) - S*DCOS(GPRAD(I,2)) - LANCEN )
C
C          IF (AZI .EQ. 0.DO) SIGMA = TAU - PHICEN
C          IF (AZI .EQ. PI)  SIGMA = PHICEN - TAU
C          IF (AZI .NE. 0.DO .AND. AZI .NE. PI)

```



```

1      SIGMA = DASIN(DSIN(GPRAD(I,2)-LANCEN) * DCOS(TAU)/DSIN(AZI))
C
      H      = ETAO * DCOS(AZI)
      HSQ    = H**2
      DIST   = RNCEN * SIGMA *
1          ( 1.DO - SIGMA**2 * HSQ * (1.DO - HSQ) / 6.DO +
2          (SIGMA**3 / 8.DO) * GO * H * (1.DO - 2.DO*HSQ) +
3          (SIGMA**4 / 120.DO) * (HSQ * (4.DO - 7.DO*HSQ)
4          - 3.DO * GSQO * (1.DO - 7.DO*HSQ)) -
5          (SIGMA**5 / 48.DO) * GO * H )
      XTRUE  = SFCEN * DIST * DSIN(AZI)
      YTRUE  = SFCEN * DIST * DCOS(AZI)
      XYGRID(I,1) = XTRUE + FALSEE
      XYGRID(I,2) = YTRUE + FALSEN
1      CONTINUE
      RETURN
      END

```

```

SUBROUTINE AEINV (AMAJ, FINV, LANCEN, PHICEN, SFCEN,
1          FALSEN, FALSEE, N, ROWS, XYGRID, GPRAD)
C
C          PURPOSE: THIS ROUTINE TRANSFORMS AZINUTHAL EQUIDISTANT
C          COORDINATES (X,Y) INTO GEODETIC COORDINATES (LATITUDE,
C          LONGITUDE).
C
C          GLOBAL DECLARATIONS:
C          INTEGER N, ROWS
C          DOUBLE PRECISION AMAJ, FALSEE, FALSEN, FINV, GPRAD(ROWS,2),
1          LANCEN, PHICEN, SFCEN, XYGRID(ROWS,2)
C
C          LOCAL DECLARATIONS:
C          DOUBLE PRECISION A, AZI, B, C, CA, D, DIST, E, ESQ, ETASQO, E2SQ,
1          F, FLAT, PI, PINUL2, RN, RNCEN, S, TAU, XTRUE,
2          YTRUE
C
C          THE FOLLOWING ARE TEMPORARY STORAGE VARIABLES: C, CA, S
C
C          PARAMETER (PI      = 3.1415926535897932DO,
1          PINUL2 = 2.DO * PI)
C          SAVE ESQ, E2SQ, RNCEN, ETASQO
C
C          *** COMPUTE THE INITIALIZATION CONSTANTS. ***
C
C          FLAT  = 1.DO / FINV
C          ESQ   = (2.DO - FLAT) * FLAT
C          E2SQ  = ESQ / (1.DO - ESQ)
C          S     = DSIN(PHICEN)
C          C     = DCOS(PHICEN)
C          RNCEN = AMAJ / DSQRT(1.DO - ESQ * S**2)
C          ETASQO = E2SQ * C**2
C
C          *** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
C          PREVIOUSLY COMPUTED ***
C
C          ENTRY AEINV2 (AMAJ, LANCEN, SFCEN, FALSEN, FALSEE, N, ROWS,
1          XYGRID, GPRAD)
C
C          DO 1 I = 1,N
C          XTRUE = XYGRID(I,1) - FALSEE
C          YTRUE = XYGRID(I,2) - FALSEN
C          DIST  = DSQRT(XTRUE**2 + YTRUE**2) / SFCEN
C          AZI   = DATAN2(XTRUE, YTRUE)
C          CA    = DCOS(AZI)
C          A     = ETASQO * CA**2
C          B     = 3.DO * E2SQ * (1.DO + A) * S * C * CA
C          D     = DIST / RNCEN
C          E     = D + A * (1.DO - A) * D**3 / 6.DO -
1          B * (1.DO - 3.DO*A) * D**4 / 24.DO
C          F     = 1.DO + A * (E**2 / 2.DO) - B * (E**3 / 6.DO)

```

```

    TAU = DASIN(S * DCOS(E) + C * DSIN(E) * CA)
    GPRAD(I,1) = DATAN( (DTAN(TAU) - ESQ * F * S / DCOS(TAU)) /
2      (1.DO - ESQ) )
    GPRAD(I,2) = LANCEN + DASIN( DSIN(AZI) * DSIN(E) / DCOS(TAU) )
    IF (GPRAD(I,2) .LE. -PI) GPRAD(I,2) = GPRAD(I,2) + PIMUL2
    IF (GPRAD(I,2) .GT. PI) GPRAD(I,2) = GPRAD(I,2) - PIMUL2
1  CONTINUE
    RETURN
    END

```

```

SUBROUTINE DATUMT (AMAJO, FINVO, AMAJN, FINVN, DELTAX, DELTAY,
1          DELTAZ, OMEGA, EPSIL, PSI, DELTAK, N, ROWS,
2          GPORAD, GPNRAD, ELEV, NSEPO, NSEPN, HTO, HTN)

C
C      PURPOSE:  THIS ROUTINE TRANSFORMS GEODETIC COORDINATES
C      (LATITUDE, LONGITUDE) FROM ONE GEODETIC DATUM TO ANOTHER.
C
C      GLOBAL DECLARATIONS:
C      INTEGER N, ROWS
C      DOUBLE PRECISION AMAJO, AMAJN, DELTAK, DELTAX, DELTAY, DELTAZ,
1          ELEV(ROWS), EPSIL, FINVO, FINVN, GPORAD(ROWS,2),
2          GPNRAD(ROWS,2), HTO(ROWS), HTN(ROWS),
3          NSEPO(ROWS), NSEPN(ROWS)

C
C      LOCAL DECLARATIONS:
C      DOUBLE PRECISION BHINN, ESQO, ESQN, E2SQN, FLATO, FLATN,
1          P, PI, PIMUL2, RNO, RNN, THETA, XO, YO, ZO,
2          XN, YN, ZN

C
C      PARAMETER (PI      = 3.1415926535897932D0,
1          PIMUL2 = 2.DO * PI)
C      SAVE FLATO, FLATN, BHINN, ESQO, ESQN, E2SQN

C
C      *** COMPUTE THE INITIALIZATION CONSTANTS. ***

C
C      FLATO = 1.DO / FINVO
C      FLATN = 1.DO / FINVN
C      BHINN = AMAJN * (1.DO - FLATN)
C      ESQO  = (2.DO - FLATO) * FLATO
C      ESQN  = (2.DO - FLATN) * FLATN
C      E2SQN = ESQN / (1.DO - ESQN)

C
C      *** ENTER HERE IF INITIALIZATION CONSTANTS HAVE BEEN
C      PREVIOUSLY COMPUTED ***

C
C      ENTRY DATUM2 (AMAJO, AMAJN, DELTAX, DELTAY, DELTAZ, OMEGA,
1          EPSIL, PSI, DELTAK, N, ROWS, GPORAD, GPNRAD,
2          ELEV, NSEPO, NSEPN, HTO, HTN)

C
C      DO 1 I = 1, N
C          RNO = AMAJO / DSQRT( 1.DO - ESQO * DSIN(GPORAD(I,1))**2 )
C          HTO(I) = ELEV(I) + NSEPO(I)
C          XO = (RNO + HTO(I)) * DCOS(GPORAD(I,1)) * DCOS(GPORAD(I,2))
C          YO = (RNO + HTO(I)) * DCOS(GPORAD(I,1)) * DSIN(GPORAD(I,2))
C          ZO = (RNO * (1.DO - ESQO) + HTO(I)) * DSIN(GPORAD(I,1))
C          XN = DELTAX + (XO + OMEGA*YO - PSI*ZO) * (1.DO * DELTAK)
C          YN = DELTAY + (YO - OMEGA*XO + EPSIL*ZO) * (1.DO * DELTAK)
C          ZN = DELTAZ + (ZO + PSI*XO - EPSIL*YO) * (1.DO * DELTAK)
C          P = DSQRT(XN**2 + YN**2)
C          THETA = DATAN(AMAJN * ZN / (BHINN * P))
C          GPNRAD(I,1) = DATAN( (ZN + E2SQN * BHINN * DSIN(THETA)**3) /

```

```

1          (P - ESON * AMAJN * DCOS(THETA)**3 )
GPNRAD(I,2) = DATAN2(YN, XN)
IF (GPNRAD(I,2) .LE. -PI) GPNRAD(I,2) = GPNRAD(I,2) + PINUL2
IF (GPNRAD(I,2) .GT. PI) GPNRAD(I,2) = GPNRAD(I,2) - PINUL2
RNN       = AMAJN / DSQRT(1.DO - ESON * DSIN(GPNRAD(I,1))**2)
HTN(I)    = P / DCOS(GPNRAD(I,1)) - RNN
NSEPN(I)  = HTN(I) - ELEV(I)
1 CONTINUE
RETURN
END

```

APPENDIX E. -- TEST POINTS

Test points are included in this appendix to provide a standard against which to compare results in the event the subroutines are someday revised. Parameters and coordinates have been chosen not as those that would be typically found in practice, but rather, those that thoroughly exercise the transformations. Whole-degree latitudes and longitudes are transformed to x's and y's, which are then transformed back to latitudes and longitudes. The "Round Trip Error" is the difference in degrees between the starting and ending coordinates. Both the planar coordinates and round trip error should be used to check the results of future revisions.

NORMAL MERCATOR TRANSFORMATION TEST POINTS ON GRS 80-ELLIPSOID.
 SEMIMAJOR AXIS = 6378137.
 RECIPROCAL OF FLATTENING = 298.257222101

PROJECTION PARAMETERS :
 CENTRAL MERIDIAN = 0. DEGREES
 REFERENCE LATITUDE FOR SCALE FACTOR = 0.
 SCALE FACTOR AT REFERENCE LATITUDE = 1/ 1
 FALSE NORTHING = 0.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
0.	0.	0.000000000000E+00	0.000000000000E+00	-.12722E-13	0.00000E+00
0.	-60.	-.667916944760E+07	0.000000000000E+00	-.12722E-13	0.35527E-14
0.	120.	0.133583388952E+08	0.000000000000E+00	-.12722E-13	0.35527E-14
0.	-180.	-.200375083428E+08	0.000000000000E+00	-.12722E-13	0.00000E+00
0.	240.	-.133583388952E+08	0.000000000000E+00	-.12722E-13	0.00000E+00

PROJECTION PARAMETERS :
 CENTRAL MERIDIAN = 90. DEGREES
 REFERENCE LATITUDE FOR SCALE FACTOR = 25.
 SCALE FACTOR AT REFERENCE LATITUDE = 1/ 100
 FALSE NORTHING = -25000.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
29.	0.	-.908550812557E+05	0.542423039994E+04	0.11278E-09	0.00000E+00
29.	-60.	-.151425135426E+06	0.542423039994E+04	0.11278E-09	0.35527E-14
29.	120.	0.302850270852E+05	0.542423039994E+04	0.11278E-09	0.35527E-14
29.	-180.	0.908550812557E+05	0.542423039994E+04	0.11278E-09	0.00000E+00
29.	240.	0.151425135426E+06	0.542423039994E+04	0.11278E-09	0.00000E+00

PROJECTION PARAMETERS :
 CENTRAL MERIDIAN = -180. DEGREES
 REFERENCE LATITUDE FOR SCALE FACTOR = 50.
 SCALE FACTOR AT REFERENCE LATITUDE = 1/ 10000
 FALSE NORTHING = 500.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
-58.	0.	0.129052356510E+04	-.108023511948E+02	-.22240E-11	0.00000E+00
-58.	-60.	0.860349043400E+03	-.108023511948E+02	-.22240E-11	0.35527E-14
-58.	120.	-.430174521700E+03	-.108023511948E+02	-.22240E-11	0.35527E-14
-58.	-180.	0.000000000000E+00	-.108023511948E+02	-.22240E-11	0.00000E+00
-58.	240.	0.430174521700E+03	-.108023511948E+02	-.22240E-11	0.00000E+00

PROJECTION PARAMETERS :
 CENTRAL MERIDIAN = 270. DEGREES
 REFERENCE LATITUDE FOR SCALE FACTOR = 75.
 SCALE FACTOR AT REFERENCE LATITUDE = 1/1000000
 FALSE NORTHING = -7.5

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
87.	0.	0.260118052292E+01	-.147919489501E+01	-.35527E-14	0.00000E+00
87.	-60.	0.867060174308E+00	-.147919489501E+01	-.35527E-14	0.35527E-14
87.	120.	-.433530087154E+01	-.147919489501E+01	-.35527E-14	0.17764E-13
87.	-180.	-.260118052292E+01	-.147919489501E+01	-.35527E-14	0.00000E+00
87.	240.	-.867060174308E+00	-.147919489501E+01	-.35527E-14	0.00000E+00

TRANSVERSE MERCATOR TRANSFORMATION TEST POINTS ON GRS 80 ELLIPSOID:
 SEMIMAJOR AXIS = 6378137.
 RECIPROCAL OF FLATTENING = 298.257222101

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 0. DEGREES
 LATITUDE OF FALSE ORIGIN = 0. DEGREES
 SCALE FACTOR ALONG CENTRAL MERIDAN = 1/ 1
 FALSE NORTHING = 100000.0
 FALSE EASTING = 0.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
0.	0.	0.000000000000E+00	0.100000000000E+06	0.00000E+00	0.00000E+00
0.	-2.	-.222684513479E+06	0.100000000000E+06	0.00000E+00	-.12366E-10
0.	4.	0.445642555682E+06	0.100000000000E+06	0.00000E+00	0.51664E-09
0.	-6.	-.669149347436E+06	0.100000000000E+06	0.00000E+00	-.57793E-08
0.	8.	0.893483518340E+06	0.100000000000E+06	0.00000E+00	0.38211E-07

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 90. DEGREES
 LATITUDE OF FALSE ORIGIN = -25. DEGREES
 SCALE FACTOR ALONG CENTRAL MERIDAN = 1/ 100
 FALSE NORTHING = 3000.0
 FALSE EASTING = 10000.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
-22.	90.	0.100000000000E+05	0.632253691352E+04	0.26539E-11	-.71054E-14
-22.	88.	0.793445318814E+04	0.630902834553E+04	0.13856E-12	0.77094E-12
-22.	94.	0.141329175271E+05	0.626843336086E+04	-.56462E-09	-.86409E-10
-22.	84.	0.379605916890E+04	0.620054333499E+04	-.14244E-07	0.90809E-09
-22.	98.	0.182804549437E+05	0.610500795230E+04	-.14214E-06	-.37515E-09

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = -180. DEGREES
 LATITUDE OF FALSE ORIGIN = 50. DEGREES
 SCALE FACTOR ALONG CENTRAL MERIDAN = 1/ 10000
 FALSE NORTHING = 90.0
 FALSE EASTING = 200.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
68.	-180.	0.200000000000E+03	0.290503560859E+03	-.26752E-11	0.00000E+00
68.	-182.	0.191636915907E+03	0.290638913337E+03	-.10445E-11	-.56843E-13
68.	-176.	0.216718834120E+03	0.291044944735E+03	0.41643E-09	0.20584E-09
68.	-186.	0.174940100580E+03	0.291721575926E+03	0.10845E-07	-.80304E-08
68.	-172.	0.233378896294E+03	0.292668671623E+03	0.10978E-06	0.10858E-06

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 270. DEGREES

LATITUDE OF FALSE ORIGIN = -75. DEGREES

SCALE FACTOR ALONG CENTRAL MERIDAN = 1/1000000

FALSE NORTHING = 2.7

FALSE EASTING = 3.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
6.	270.	0.30000000000E+01	0.116904075425E+02	0.50160E-12	0.00000E+00
6.	268.	0.277852826580E+01	0.116908117121E+02	0.16467E-11	-.10914E-10
6.	274.	0.344320954773E+01	0.116920266851E+02	0.24745E-09	0.44832E-09
6.	264.	0.233451890015E+01	0.116940598906E+02	0.61325E-08	-.49868E-08
6.	278.	0.388855723131E+01	0.116969238317E+02	0.60809E-07	0.33358E-07

OBLIQUE MERCATOR TRANSFORMATION TEST POINTS ON GRS 80 ELLIPSOID.

SEMINAJOR AXIS = 6378137.

RECIPROCAL OF FLATTENING = 298.257222101

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 0.00 DEGREES
LATITUDE OF CENTER POINT = 0. DEGREES
AZIMUTH AT THE CENTER POINT = 45.0 DEGREES
SCALE FACTOR AT THE CENTER POINT = 1.0000
FALSE NORTHING = 0.0
FALSE EASTING = 0.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
-10.	-10.	-.110194509751E+07	-.111143297110E+07	-.10378E-09	0.11102E-13
-10.	-20.	-.220018786094E+07	-.114745882272E+07	-.10379E-09	0.21316E-13
-10.	-40.	-.444086598323E+07	-.139398738032E+07	-.10379E-09	0.35527E-13
20.	10.	0.108799115133E+07	0.222080182547E+07	0.83642E-10	-.62172E-14
20.	20.	0.213683632920E+07	0.225442871465E+07	0.83649E-10	-.17764E-13
20.	40.	0.422344883446E+07	0.251949322560E+07	0.83659E-10	-.35527E-13
-40.	-10.	-.113217442646E+07	-.451317936140E+07	0.29416E-10	0.15543E-14
-40.	-20.	-.203119384651E+07	-.449804564754E+07	0.29420E-10	0.14211E-13
-40.	-40.	-.375633710710E+07	-.469685925067E+07	0.29413E-10	0.31974E-13

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = -133.67 DEGREES
LATITUDE OF CENTER POINT = 57. DEGREES
AZIMUTH AT THE CENTER POINT = -36.9 DEGREES
SCALE FACTOR AT THE CENTER POINT = 0.9999
FALSE NORTHING = -5000000.0
FALSE EASTING = 5000000.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
47.	-144.	0.540346197512E+05	-.491973627967E+06	-.13756E-10	0.35527E-14
47.	-154.	-.710944286498E+06	-.332025509173E+06	-.13774E-10	0.71054E-14
47.	-174.	-.215972046918E+07	0.356223176130E+06	-.13767E-10	-.17764E-13
77.	-124.	0.105214522995E+07	0.285000987399E+07	0.17799E-11	-.10658E-13
77.	-114.	0.130168082168E+07	0.292134430020E+07	0.18048E-11	0.14211E-13
77.	-94.	0.175553703149E+07	0.318750963565E+07	0.17799E-11	0.00000E+00
17.	-144.	-.186339090971E+06	-.407175427553E+07	0.10250E-09	-.71054E-14
17.	-154.	-.147815861961E+07	-.402517396969E+07	0.10250E-09	0.71054E-14
17.	-174.	-.434826072973E+07	-.322320266923E+07	0.10250E-09	-.71054E-14

LAMBERT CONFORMAL CONIC TRANSFORMATION TEST POINTS ON GRS 80 ELLIPSOID.
 SEMIMAJOR AXIS = 6378137.
 RECIPROCAL OF FLATTENING = 298.257222101

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 90. DEGREES
 NORTH PARALLEL = 45. DEGREES
 SOUTH PARALLEL = 45. DEGREES
 SCALE FACTOR ALONG NORTH (OR SOUTH) PARALLEL = 1/ 1
 LATITUDE OF FALSE ORIGIN = 45. DEGREES
 FALSE NORTHING = 0.0
 FALSE EASTING = 100000.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
45.	90.	0.100000000000E+06	0.000000000000E+00	0.49738E-13	-.71054E-14
51.	90.	0.100000000000E+06	0.668394931088E+06	0.42633E-13	-.71054E-14
33.	90.	0.100000000000E+06	-.134153384686E+07	0.92371E-13	-.71054E-14
63.	90.	0.100000000000E+06	0.204064546893E+07	0.28422E-13	-.71054E-14
21.	90.	0.100000000000E+06	-.273554239821E+07	-.35527E-13	-.71054E-14

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = -90. DEGREES
 NORTH PARALLEL = -45. DEGREES
 SOUTH PARALLEL = -47. DEGREES
 SCALE FACTOR ALONG NORTH (OR SOUTH) PARALLEL = 1/ 100
 LATITUDE OF FALSE ORIGIN = -48. DEGREES
 FALSE NORTHING = 1000.0
 FALSE EASTING = 3000.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
-46.	-87.	0.532299678162E+04	0.317977671096E+04	-.46185E-13	0.35527E-14
-40.	-87.	0.557441498108E+04	0.985147204252E+04	-.67502E-13	0.35527E-14
-58.	-87.	0.481632795939E+04	-.102653120822E+05	-.24869E-13	0.35527E-14
-28.	-87.	0.608687863566E+04	0.234503339357E+05	0.17764E-13	0.35527E-14
-70.	-87.	0.428076855600E+04	-.244770484606E+05	-.71054E-14	0.35527E-14

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 90. DEGREES
 NORTH PARALLEL = 47. DEGREES
 SOUTH PARALLEL = 41. DEGREES
 SCALE FACTOR ALONG NORTH (OR SOUTH) PARALLEL = 1/ 10000
 LATITUDE OF FALSE ORIGIN = 37. DEGREES
 FALSE NORTHING = 20.0
 FALSE EASTING = 90.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
44.	72.	-.530312026126E+02	0.113491331664E+03	0.42633E-13	0.35527E-14
50.	72.	-.385761804883E+02	0.178641775311E+03	0.35527E-13	0.35527E-14

32.	72.	-.820463175976E+02	-.172831214965E+02	0.74607E-13	0.35527E-14
62.	72.	-.890450684381E+01	0.312375413700E+03	0.39080E-13	0.35527E-14
20.	72.	-.112203315241E+03	-.153204170845E+03	-.21316E-13	0.35527E-14

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = -90. DEGREES
 NORTH PARALLEL = -41. DEGREES
 SOUTH PARALLEL = -63. DEGREES
 SCALE FACTOR ALONG NORTH (OR SOUTH) PARALLEL = 1/1000000
 LATITUDE OF FALSE ORIGIN = -72. DEGREES
 FALSE NORTHING = 0.3
 FALSE EASTING = 2.7

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
-52.	-9.	0.708627916328E+01	-.216266453769E+00	-.46185E-13	-.66613E-15
-46.	-9.	0.767742915085E+01	0.690324367425E-01	-.56843E-13	-.66613E-15
-64.	-9.	0.589648620445E+01	-.790480474290E+00	-.35527E-13	-.66613E-15
-34.	-9.	0.888326834518E+01	0.650990639417E+00	-.81712E-13	-.66613E-15
-76.	-9.	0.463806124612E+01	-.139781744790E+01	-.21316E-13	-.66613E-15

POLYCONIC TRANSFORMATION TEST POINTS ON GRS 80 ELLIPSOID
 SEMIMAJOR AXIS = 6378137.
 RECIPROCAL OF FLATTENING = 298.257222101

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 0. DEGREES
 LATITUDE OF GRID ORIGIN = 0. DEGREES
 SCALE FACTOR ALONG CENTRAL MERIDIAN = 1/ 1
 FALSE EASTING = 0.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
0.	0.	0.000000000000E+00	0.000000000000E+00	0.00000E+00	0.00000E+00
0.	-2.	-.222638981587E+06	0.000000000000E+00	0.00000E+00	0.00000E+00
0.	4.	0.445277963173E+06	0.000000000000E+00	0.00000E+00	0.22204E-15
0.	-6.	-.667916944760E+06	0.000000000000E+00	0.00000E+00	0.00000E+00
0.	8.	0.890555926346E+06	0.000000000000E+00	0.00000E+00	-.44409E-15

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 90. DEGREES
 LATITUDE OF GRID ORIGIN = -25. DEGREES
 SCALE FACTOR ALONG CENTRAL MERIDIAN = 1/ 100
 FALSE EASTING = 10000.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
-22.	90.	0.100000000000E+05	0.332253691352E+04	-.15016E-08	-.71054E-14
-22.	88.	0.793481586721E+04	0.330903430263E+04	-.41594E-09	-.17808E-06
-22.	94.	0.141300151490E+05	0.326852877871E+04	0.28557E-08	0.35678E-06
-22.	84.	0.380586000770E+04	0.320102726762E+04	0.83267E-08	-.53674E-06
-22.	98.	0.182572057273E+05	0.310654131114E+04	0.16025E-07	0.71859E-06

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = -180. DEGREES
 LATITUDE OF GRID ORIGIN = 50. DEGREES
 SCALE FACTOR ALONG CENTRAL MERIDIAN = 1/ 10000
 FALSE EASTING = 200.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
68.	-180.	0.200000000000E+03	0.200503560859E+03	0.44052E-08	0.00000E+00
68.	-182.	0.191637154283E+03	0.200638903680E+03	0.36550E-10	-.71821E-06
68.	-176.	0.216716932271E+03	0.201044790387E+03	-.13159E-07	0.14383E-05
68.	-186.	0.174946490328E+03	0.201720795857E+03	-.34925E-07	-.21622E-05
68.	-172.	0.233363846271E+03	0.202666212049E+03	-.65744E-07	0.28917E-05

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 270. DEGREES
 LATITUDE OF GRID ORIGIN = -75. DEGREES
 SCALE FACTOR ALONG CENTRAL MERIDIAN = 1/1000000
 FALSE EASTING = 3.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
6.	270.	0.300000000000E+01	0.899040754253E+01	0.15740E-08	0.00000E+00
6.	268.	0.277857305116E+01	0.899081150674E+01	-.42497E-09	-.10967E-05
6.	274.	0.344285094977E+01	0.899202339399E+01	-.64194E-08	0.21975E-05
6.	264.	0.233573094508E+01	0.899404318816E+01	-.16397E-07	-.33062E-05
6.	278.	0.388567831651E+01	0.899687086234E+01	-.30334E-07	0.44269E-05

AZIMUTHAL EQUIDISTANT TRANSFORMATION TEST POINTS ON GRS 80 ELLIPSOID.
SEMINOR AXIS = 6378137.
RECIPROCAL OF FLATTENING = 298.257222101

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 0. DEGREES
LATITUDE OF CENTER POINT = 0. DEGREES
SCALE FACTOR AT CENTER POINT = 1/ 1
FALSE NORTHING = 100000.0
FALSE EASTING = 0.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LOX	X	Y	LAT	LOX
0.	0.	0.000000000000E+00	0.100000000000E+06	0.00000E+00	0.00000E+00
0.	-4.	-.445277963173E+06	0.100000000000E+06	-.24638E-15	0.62172E-14
0.	8.	0.890555926346E+06	0.100000000000E+06	-.49156E-15	-.13101E-13
0.	-12.	-.133583388952E+07	0.100000000000E+06	-.73434E-15	0.20872E-13
0.	16.	0.178111185269E+07	0.100000000000E+06	-.97355E-15	-.31974E-13

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 90. DEGREES
LATITUDE OF CENTER POINT = -25. DEGREES
SCALE FACTOR AT CENTER POINT = 1/ 100
FALSE NORTHING = 3000.0
FALSE EASTING = 10000.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LOX	X	Y	LAT	LOX
-21.	90.	0.100000000000E+05	0.742977300029E+04	-.17150E-07	-.71054E-14
-21.	86.	0.583832194842E+04	0.737140323260E+04	-.26883E-07	0.21661E-07
-21.	98.	0.183202425567E+05	0.719610147008E+04	-.24700E-07	-.38441E-07
-21.	78.	-.247254370047E+04	0.690328843789E+04	0.93255E-07	-.21072E-06
-21.	106.	0.266153580596E+05	0.649199356040E+04	0.53314E-06	0.15008E-05

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = -180. DEGREES
LATITUDE OF CENTER POINT = 50. DEGREES
SCALE FACTOR AT CENTER POINT = 1/ 10000
FALSE NORTHING = 90.0
FALSE EASTING = 200.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LOX	X	Y	LAT	LOX
58.	-180.	0.200000000000E+03	0.179043563031E+03	0.38432E-07	0.00000E+00
58.	-184.	0.176282478599E+03	0.179702783757E+03	0.64368E-07	-.44858E-07
58.	-172.	0.247358939396E+03	0.181679446287E+03	0.16013E-06	0.22781E-06
58.	-192.	0.129151985260E+03	0.184970535438E+03	0.37251E-06	-.83417E-06
58.	-164.	0.294108239265E+03	0.189570971987E+03	0.75603E-06	0.24376E-05

PROJECTION PARAMETERS :

CENTRAL MERIDIAN = 270. DEGREES
LATITUDE OF CENTER POINT = -75. DEGREES
SCALE FACTOR AT CENTER POINT = 1/1000000
FALSE NORTHING = 2.7
FALSE EASTING = 3.0

ORIGINAL POSITION		TRANSFORMED POSITION		ROUND TRIP ERROR	
LAT	LON	X	Y	LAT	LON
-63.	270.	0.300000000000E+01	0.403855301619E+01	0.33071E-05	0.00000E+00
-63.	266.	0.279596594015E+01	0.403179976301E+01	0.34135E-05	-.60862E-06
-63.	278.	0.340719162497E+01	0.401156514413E+01	0.37440E-05	0.13219E-05
-63.	258.	0.239140154783E+01	0.397792458331E+01	0.43323E-05	-.22572E-05
-63.	286.	0.380738481912E+01	0.393100379177E+01	0.52362E-05	0.35562E-05

DATUM TRANSFORMATION TEST POINTS FROM NAD 27 (SEMINAJOR AXIS = 6378206.4, RECIPROCAL OF FLATTENING = 294.978698)
 TO NAD 83 (SEMINAJOR AXIS = 6378137., RECIPROCAL OF FLATTENING = 298.257222101).

PROJECTION PARAMETERS :

DELTA X = 0. METERS
 DELTA Y = 0. METERS
 DELTA Z = 0. METERS
 OMEGA = 0.0 SECONDS
 EPSILON = 0.00 SECONDS
 PSI = 0.0 SECONDS
 DELTA K = 0.00E+00

POSITION ON NAD 27				POSITION ON NAD 83			ROUND TRIP ERROR		
LAT	LON	ELEV	NSEP	LAT	LON	HT	LAT	LON	NSEP
0.0	0.	1000.	0.	0.000000000	0.000000000	1069.40000	0.00000E+00	0.00000E+00	0.00000E+00
22.4	45.	-2000.	-20.	22.398489256	45.000000000	-1985.04682	-0.23093E-12	0.00000E+00	-0.10245E-07
44.8	-90.	3000.	40.	44.797860212	-90.000000000	2991.47876	-0.14815E-11	0.71054E-14	-0.16391E-06
-67.2	-135.	-4000.	-60.	-67.198471215	-135.000000000	-4192.68196	0.10232E-11	0.35527E-14	-0.26962E-06
-89.6	180.	5000.	80.	-89.599970182	180.000000000	4911.49748	0.00000E+00	0.00000E+00	0.00000E+00

PROJECTION PARAMETERS :

DELTA X = 20. METERS
 DELTA Y = -25. METERS
 DELTA Z = 5. METERS
 OMEGA = 0.3 SECONDS
 EPSILON = 0.05 SECONDS
 PSI = 0.1 SECONDS
 DELTA K = -.10E-06

POSITION ON NAD 27				POSITION ON NAD 83			ROUND TRIP ERROR		
LAT	LON	ELEV	NSEP	LAT	LON	HT	LAT	LON	NSEP
0.0	0.	1000.	0.	0.000073176	-0.000307874	1088.76218	0.15830E-09	-0.22119E-09	-0.55780E-04
22.4	45.	-2000.	-20.	22.398553066	44.999619615	-1987.04499	0.35650E-09	0.11674E-09	-0.46066E-04
44.8	-90.	3000.	40.	44.797747595	-89.999858081	3012.10988	-0.11269E-10	-0.50204E-09	0.16177E-04
-67.2	-135.	-4000.	-60.	-67.198434372	-134.999277757	-4196.55394	0.43280E-09	-0.21319E-09	0.35572E-05
-89.6	180.	5000.	80.	-89.600176311	-179.966055878	4905.72204	0.38503E-09	0.38271E-07	-0.16804E-04

PROJECTION PARAMETERS :

DELTA X = -40. METERS
 DELTA Y = 50. METERS
 DELTA Z = -50. METERS
 OMEGA = -0.9 SECONDS
 EPSILON = -.15 SECONDS
 PSI = -.2 SECONDS
 DELTA K = 0.20E-05

POSITION ON NAD 27				POSITION ON NAD 83			ROUND TRIP ERROR		
LAT	Lon	ELEV	NSEP	LAT	Lon	HT	LAT	Lon	NSEP
0.0	0.	1000.	0.	-0.000508041	0.000699086	1042.15914	-0.41706E-10	-0.30244E-09	-0.47410E-03
22.4	45.	-2000.	-20.	22.398037183	45.000840080	-1984.81722	0.13008E-08	0.21483E-08	-0.34293E-03
44.8	-90.	3000.	40.	44.797815914	-90.000200507	2933.49403	0.11601E-09	-0.42401E-08	-0.85144E-04
-67.2	-135.	-4000.	-60.	-67.198693505	-135.001384358	-4136.62038	0.20128E-08	-0.49320E-08	0.12569E-03
-89.6	180.	5000.	80.	-89.599559985	179.930326288	4974.49897	0.32724E-08	0.83439E-07	0.18302E-04

PROJECTION PARAMETERS :

DELTA X = 60. METERS
 DELTA Y = -75. METERS
 DELTA Z = -375. METERS
 OMEGA = 2.1 SECONDS
 EPSILON = 0.35 SECONDS
 PSI = -.3 SECONDS
 DELTA K = -.30E-04

POSITION ON NAD 27				POSITION ON NAD 83			ROUND TRIP ERROR		
LAT	Lon	ELEV	NSEP	LAT	Lon	HT	LAT	Lon	NSEP
0.0	0.	1000.	0.	-0.003474778	-0.001256965	938.03700	0.10309E-06	0.88658E-08	-0.37838E-02
22.4	45.	-2000.	-20.	22.395269447	44.998493320	-2328.95865	0.10160E-06	0.27692E-07	0.23364E-02
44.8	-90.	3000.	40.	44.795094311	-89.999743168	2589.39877	0.81340E-07	-0.38158E-07	0.69795E-02
-67.2	-135.	-4000.	-60.	-67.199564450	-134.998352192	-4033.58247	0.49519E-07	-0.70343E-07	-0.99545E-02
-89.6	180.	5000.	80.	-89.600446926	-179.890522738	5095.21984	0.29460E-07	-0.14059E-05	-0.11299E-01

REFERENCES

- Adams, Oscar S., 1949: Latitude developments in connection with geodesy and cartography, Special Publication 67, U.S. Coast and Geodetic Survey, Washington, D.C., pp. 89, 125, National Geodetic Information Center, NOAA, Rockville, MD 20852.
- Ayres, James A., 1983: Circular Letter 44/1983, International Hydrographic Organization, Monaco.
- Berry, Ralph M., 1971: The Michigan coordinate system, paper presented to Michigan Association of Registered Land Surveyors, pp. 22-25, National Geodetic Information Center, NOAA, Rockville, MD 20852.
- Bowring, B.R., 1976: Transformation from spacial to geographical coordinates, Survey Review, no. 181, pp. 323-327.
- Claire, Charles N., 1968, rev. 1973: State plane coordinates by automatic data processing, Publication 62-4, U.S. Coast and Geodetic Survey, Washington, D.C., National Geodetic Information Center, NOAA, Rockville, MD 20852.
- Deetz, Charles H., and Adams, Oscar S., 1944: Elements of map projection with applications to map and chart construction, Special Publication 68, 5th ed., U.S. Coast and Geodetic Survey, Washington, D.C., pp. 112-114.
- Hotine, Martin, 1945-46: The orthomorphic projection of the spheroid, Empire Survey Review, vols. VIII and IX, no. 62-66.
- Mitchell, Hugh C., and Simmons, Lancing G., 1945: The state coordinate systems (a manual for surveyors), Special Publication 235, U.S. Coast and Geodetic Survey, Washington, D.C.
- Pfeifer, Ludvik, 1980: Input Formats and Specifications of the National Geodetic Survey Data Base, vol. I: Horizontal control data, Federal Geodetic Control Committee, pp. B-1 - B-2, National Geodetic Information Center, NOAA, Rockville, MD 20852.
- Rapp, Richard H., 1981: Transformation of geodetic data between reference datums, Geometric Geodesy, vol. II, Ohio State University, Columbus, pp. 53-57, 66.
- Snyder, James P., 1983: Map projections used by the U.S. Geological Survey, Geological Survey Bulletin 1532, 2nd ed., U.S. Geological Survey, Washington, D.C., 313 pp.
- Thomas, Paul D., 1952: Conformal projections in geodesy and cartography, Special Publication 251, U.S. Coast and Geodetic Survey, Washington, D.C. pp. 85-90.

Vincenty, T., (National Geodetic Survey, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, Rockville, MD) 1984a:
Transverse Mercator projection formulas for GRS80 ellipsoid, 7 pp.,
Aug. 31, 1984 (unpublished file report).

Vincenty, T., (National Geodetic Survey, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, Rockville, MD) 1984b:
Oblique Mercator projection formulas for GRS80 ellipsoid, 4 pp.,
Sept. 1, 1984 (unpublished file report).

Vincenty, T., (National Geodetic Survey, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, Rockville, MD) 1985:
Lambert conformal conic projection formulas for GRS80 ellipsoid, 5 pp.,
Mar. 1, 1985 (unpublished file report).