NASA Technical Memorandum 83233

Grid Systems for Earth Radiation Budget Experiment Applications

David R. Brooks

DECEMBER 1981

NASA

NASA Technical Memorandum 83233

Grid Systems for Earth Radiation Budget Experiment Applications

David R. Brooks
Langley Research Center
Hampton, Virginia

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1981

SUMMARY

The Earth Radiation Budget Experiment (ERBE) has given rise to numerous analyses involving various data sets. Spatial coordinate transformations are developed for several different grid systems used for the compressed storage of global satellite data. These include grids for the ERBE project, for analysis of existing satellite radiation data, and for the U.S. Air Force cloud data base. Generally the grids are defined in terms of a two-dimensional index system, with associated regional box numbers. The transformations associate longitude and latitude with a particular grid location, and vice versa. In addition, the relationship between longitude and right ascension is given. This transformation combines position information for the Sun with longitude-latitude coordinates as the Earth rotates in inertial space; from it, certain information about the Sun's position relative to an observer can be obtained. The transformations developed in this memorandum are implemented in FORTRAN IV in an appendix. Such implementation is suggested by the fact that the spatial grid systems are defined, rather than just described, by their transformation algorithms and that actual computer-coded versions of the transformations are the best way to avoid some potential ambiguities in each system.

INTRODUCTION

Development of the Earth Radiation Budget Experiment (ERBE, described in ref. 1) has resulted in a large number of directly related and supporting software tasks. The latter involve other existing or potential data sources, including archived U.S. Air Force cloud data (the so-called 3DNEPH data), radiation data from the eastern Geostationary Operational Environmental Satellite (GOES-east, hereinafter referred to simply as GOES), and the Nimbus-Earth Radiation Budget experiment. Examination of such data requires many spatial transformations back and forth between locations on the Earth's surface (longitude and latitude) and a grid system suitable for analysis and compressed storage of large quantities of information. In addition, analysis of variables which exhibit diurnal behavior often requires temporal transformations between a standardized global time and a local clock or solar time. Thus, this memorandum has two purposes: first, to provide spatial transformation equations relating position on the Earth's surface to locations in a grid system and, second, to provide transformations relating inertial space to coordinate systems on the Earth's rotating surface. The latter equations are equivalent to relating universal time (UT) to a local or solar time; they allow position information for the Sun and an observer to be compared in the same coordinate system.

Spatial transformations in algorithm form not only make data manipulation possible but also serve to actually define the coordinate systems involved. From a computational point of view, careful definitions are especially important at the boundaries of grid boxes. One way of looking at this problem is to ask whether points exactly on the Equator belong in Northern or Southern Hemisphere boxes. Similar care is needed at each pole and at a longitude of 0° and 360°, where indexing algorithms may produce indices outside the expected range. Pictures of the grid do not resolve these details, so that precise algorithms are necessary to avoid ambiguity and computational problems in a computer code.

SYMBOLS

a	relative azimuth between an observer and the Sun, as defined in figure 6, deg
A	array used in defining the Nimbus-ERB grid
B	box number in any of the grid systems
C	variable related to longitude, for defining the Nimbus-ERB grid, deg
COL	matrix column identifier within boxes of the 3DNEPH grid
d	integer part of $(B_{3D} - 1)/8$, for defining the 3DNEPH grid
D	array used in defining the GOES grid
f	angular variable used in defining the 3DNEPH grid
G	array used in defining the Nimbus-ERB grid
I	longitude increment in a latitude band of the Nimbus-ERB grid, deg
INT	FORTRAN function: $\mathrm{INT}(m) = (\mathrm{sign}\ m)\left(\mathrm{Largest\ integer} \le \lvert m \rvert\right)$
L	geocentric longitude, measured positively to the east from the Greenwich meridian, deg
LAT	integer latitude band identifier
LON	integer longitude box identifier
m	dummy variable for defining INT and MOD functions
MOD	FORTRAN function: $\mathrm{MOD}(m,n) = m - n\ \mathrm{INT}(m/n)$
n	number of boxes in a latitude band of an ERBE grid, or dummy variable for defining the MOD function
$\vec{r}$	position vector, any units
ROW	matrix row identifier within boxes of the 3DNEPH grid
s	size of sides on one of three ERBE grids, 2.5°, 5°, or 10° (also see subscript list)
T	time in Julian centuries (36 525 days) from Jan. 0.5, 1900, Julian Date 241 5020.0
UT	universal time, hours, minutes, and seconds, or fractions of a day
u,v	dimensionless variables used in defining the 3DNEPH grid
x,y,z	vector components, any units

Z	array used in defining the Langley GOES grid
α	right ascension, deg
δ	declination, deg
ε	angle between ecliptic plane and Earth's Equator, $\simeq 23.45^{\circ}$
θ	zenith angle, deg
λ	geocentric latitude, deg ($90^{\circ} \leq \lambda \leq -90^{\circ}$)
λ_{co}	geocentric colatitude, deg ($0^{\circ} \leq \lambda_{co} \leq 180^{\circ}$)
♈	first point of Aries
ω	Earth's rotation rate, 360.9856473 deg/day

Subscripts:

E	Earth
g	Greenwich meridian
g,o	Greenwich meridian at 0^{h} UT
grd	a place on the Earth's surface
G	Langley GOES grid
m	meridian
mid	midpoint of grid box
min	minimum
max	maximum
N	Nimbus-ERB grid
obs	orbiting observer of the Earth's surface
s	one of the three ERBE grid systems when used with LAT, LON, or B; or to the Sun in the discussion of the relationship between longitude and right ascension
3D	3DNEPH grid

Superscripts:

h,m,s	hours, minutes, seconds
$\rightarrow$	dimensioned vector
^	unit vector

THE ERBE GRID SYSTEM

Selection of a suitable grid system for the ERBE project is an important process, affecting literally millions of computer calculations during the life of the project. Representing spacecraft data on a grid system implies that it makes sense to portray information at the spatial resolution of the grid. A basic requirement of the ERBE project is that it produce "regional" monthly averages of the Earth's radiation budget. A "region" is loosely defined as an area a few hundred (250-500) kilometers on a side (ref. 1). Within the constraints of this requirement, several different grid systems have been proposed. "Constant area" systems start with a box of appropriate size at the Equator and seek to define longitude-latitude boundaries over the globe in a systematic way which preserves at least approximately the area of the equatorial reference box. This is done in one of two ways: either the latitude boundaries of the box are kept constant while the longitude boundaries are adjusted, or vice versa. An alternative system gives up constant area in favor of constant longitude-latitude boundaries. This leads to boxes which vary greatly in area from Equator to pole.

The ERBE Science and Data Management Teams have informally evaluated several proposed grid systems on the basis of meeting resolution requirements, ease of nesting from one resolution to another, ease of indexing, and possibilities for a reasonable global representation of the ERBE data products. The equal longitude-latitude system has been chosen over any of the equal area systems, despite the problems of interpreting regional data in grid boxes which vary in area by more than an order of magnitude. The layout of a 2.5° system is given in figure 1. To simplify indexing, latitude λ is replaced with colatitude, where $\lambda_{co} = 90^{\circ} - \lambda$, so that $0^{\circ} \leq \lambda_{co} \leq 180^{\circ}$. The boxes at the Equator have areas of 77 426 km^2, while those nearest the poles have areas of 1686 km^2. Because of its simple boundary definition (fixed increments in longitude and colatitude), the ERBE grid is easy to define and work with. Actually, three different grid systems are going to be employed in ERBE analysis, with angular resolutions s of 2.5°, 5°, and 10°. These contain 10 368, 2592, and 648 boxes, respectively, the number of boxes n in any latitude band being 144, 72, and 36, respectively. The following equations define each of the systems by giving transformations from spatial coordinates to a location in the grid, and vice versa. Transformations between the systems are also given; all the transformations in this memorandum are defined specifically in terms of geocentric, as opposed to geodetic, coordinates. Given a colatitude λ_{co} and longitude L $(0^{\circ} \leq L \leq 360^{\circ})$, the location in the grid (LAT_s, LON_s) is

$$LAT_s = \begin{cases} INT(\lambda_{co}/s + 1) & (\lambda_{co} \neq 180^{\circ}) \\ n/2 & (\lambda_{co} = 180^{\circ}) \end{cases}$$

$$LON_s = \begin{cases} INT(L/s + 1) & (L \neq 360^{\circ}) \\ n & (L = 360^{\circ}) \end{cases}$$

The INT function truncates (does not round off) its argument:

$$INT(m) = (\text{sign } m)\left(\text{Largest integer} \leq |m|\right)$$

The specific definitions at $\lambda_{co} = 180^{\circ}$ and $L = 360^{\circ}$ are given to prevent, for example, $LAT_{2.5} = 73$ and $LON_{2.5} = 145$. Each test for these limits in computer-coded form requires the equivalent of a FORTRAN IF statement and so should be avoided if possible by restraining the values of λ_{co} and L. This grid system numbers the latitude boxes from north to south and longitude boxes from west to east.

A consecutively numbered box system starting at the North Pole and Greenwich meridian, and winding eastward around the globe is given by

$$B_s = (LAT_s - 1)n + LON_s$$

Note that this system is defined so that in the 2.5° grid, for example, a point at $\lambda_{co} = 90^{\circ}$ and $L = 0^{\circ}$ is in the upper left-hand corner of the box identified by $LAT_s = 37$ and $LON_s = 1$, or $B_s = 5185$. That is, points on the Equator belong in Southern Hemisphere boxes. More generally, each box contains points on its northern and western sides, but not on its southern and eastern sides. Given a box B_s, LAT_s, and LON_s are given by

$$LAT_s = INT[(B_s - 1)/n] + 1$$

$$LON_s = B_s - (LAT_s - 1)n$$

Matrix references at one resolution can be obtained from those at another. Let $X_s = LAT_s$ or LON_s. Then

$$X_5 = INT[(X_{2.5} - 1)/2] + 1$$

$$X_{10} = INT[(X_{2.5} - 1)/4] + 1$$

$$= INT[(X_5 - 1)/2] + 1$$

Consecutively numbered boxes at different resolutions can be related as follows:

$$B_5 = 72\ INT\,[(B_{2.5} - 1)/288] + INT\{MOD\,[(B_{2.5} - 1),144]/2\} + 1$$

$$B_{10} = 36\ INT\,[(B_{2.5} - 1)/576] + INT\{MOD\,[(B_{2.5} - 1),144]/4\} + 1$$

$$= 36\ INT\,[(B_5 - 1)/144] + INT\{MOD\,[(B_5 - 1),72]/2\} + 1$$

where $MOD(m,n) = m - n\ INT(m/n)$. The reverse calculation finds all the smaller boxes included in a given larger box. First find the upper left-hand corner box:

$$B_{2.5}(\text{corner}) = 576\ INT[(B_{10} - 1)/36] + 4\ MOD[(B_{10} - 1),36] + 1$$

The remaining 15 boxes in the larger box are

$$B_{2.5}(\text{corner}) + 1,2,3,144,145,146,147,288,289,290,291,432,433,434,435$$

Similarly,

$$B_5(\text{corner}) = 144 \text{ INT}\left[(B_{10} - 1)/36\right] + 2 \text{ MOD}\left[(B_{10} - 1),36\right] + 1$$

$$B_5(\text{corner}) + 1,72,73$$

and

$$B_{2.5}(\text{corner}) = 288 \text{ INT}\left[(B_5 - 1)/72\right] + 2 \text{ MOD}\left[(B_5 - 1),72\right] + 1$$

$$B_{2.5}(\text{corner}) + 1,144,145$$

Given a particular box identified by LON_S and LAT_S, the longitude and colatitude coordinates of the midpoint are given by

$$\lambda_{co,mid} = (LAT_S - 1)s + s/2$$

$$L_{mid} = (LON_S - 1)s + s/2$$

OTHER GRID SYSTEMS OF INTEREST

The Langley GOES Grid

ERBE-related research at NASA Langley Research Center has included significant use of GOES data. This has been processed at a basic resolution of 8 km and finally on a constant area grid system which starts with $2.25^\circ \times 2.25^\circ$ boxes at the Equator (ref. 2). Much of the work has involved 1600 grid boxes in the Western Hemisphere, centered around the Equator and extending to $\pm 45^\circ$ in latitude. The Langley GOES grid is shown in figure 2. The conversion from latitude and longitude to a 2-digit reference in the GOES grid system is as follows:

$$LAT_G = \begin{cases} \text{INT}\left[(47.25 - \lambda)/2.25\right] & (-45^\circ < \lambda \leq 45^\circ) \\ 40 & (\lambda = -45^\circ) \end{cases}$$

$$LON_G = \text{INT}\left[(Z(i) + L)/D(i)\right]$$

where

$$i = \mathrm{INT}\left(|\lambda|/18 + 1\right)$$

and

i	Z(i)	D(i)	*L_{min}	†L_{max}
1	123.75	2.25	-121.5	-31.5
2	132.50	2.50	-130.0	-30.0
3	141.00	3.00	-138.0	-18.0

*$L \geq L_{min}$.
†$L < L_{max}$.

The GOES grid boxes are numbered consecutively from top (+45° latitude) to bottom (-45°), west to east, with 40 boxes in each row. Each box contains points on its northern and western sides, but not on its southern and eastern sides, so that points directly on the Equator belong in Southern Hemisphere boxes. The box number is

$$B_G = (LAT_G - 1)40 + LON_G$$

The reverse procedure is to find the longitude and latitude of the midpoint of each GOES box. Given a box number B_G,

$$LAT_G = \mathrm{INT}[(B_G - 1)/40] + 1$$

$$LON_G = B_G - (LAT_G - 1)40$$

The midpoints are

$$\lambda_{mid} = 46.125^\circ - 2.25^\circ LAT_G$$

$$L_{mid} = D(i)\ LON_G - [Z(i) - D(i)/2]$$

where

$$i = \mathrm{INT}[(28 - k)/8]$$

$$k = \begin{cases} LAT_G & (LAT_G \leq 20) \\ 41 - LAT_G & (LAT_G > 20) \end{cases}$$

The Nimbus-ERB Grid System

The Nimbus Earth Radiation Budget (ERB) Experiment has been conducted on the Nimbus 6 and 7 spacecraft (ref. 3). Radiation budget data have been analyzed globally in "target areas" defined by a constant area grid system of the type considered, but not chosen, for the ERBE. It consists of 2070 consecutively numbered boxes starting at the South Pole and spiralling westward up to the North Pole. Each box contains points on its southern and eastern sides, but not on its northern and western sides, so that points directly on the Equator belong in Northern Hemisphere boxes. (Note that this is opposite from the ERBE and Langley GOES grids.) The grid system is pictured in figure 3. Boxes at the Equator are $4.5^{\circ} \times 4.5^{\circ}$; they are defined as follows:[1]

$$LAT_N = \begin{cases} \mathrm{INT}[(\lambda + 90^{\circ})/4.5] + 1 & (-90^{\circ} \leq \lambda < 90^{\circ}) \\ 40 & (\lambda = 90^{\circ}) \end{cases}$$

$$LON_N = \mathrm{INT}(C/I) + 1$$

where for $0^{\circ} \leq L \leq 360^{\circ}$,

$$C = 360^{\circ} - L$$

and the longitude increment for each value of LAT_N is

$$I = 360^{\circ}/A(i)$$

$$i = \begin{cases} LAT_N & (LAT_N \leq 20) \\ 41 - LAT_N & (LAT_N > 20) \end{cases}$$

The array A gives the number of boxes in each latitude band from pole to Equator. (The same number in A applies to corresponding latitude bands in each hemisphere.) Then

$$B_N = \begin{cases} G(i) + LON_N & (L \neq 0) \\ G(i) + LON_N - A(i) & (L = 0) \end{cases}$$

The array G gives the sequential number of the last box in the previous latitude band, starting at the South Pole. The arrays A and G are

---

[1]These equations are based on FORTRAN subroutines supplied by Guy T. Germana, Systems and Applied Sciences Corporation, Riverdale, Maryland, March 1981.

A: 3,9,16,20,30,36,40,45,48,60,60,60,72,72,72,72,80,80,80,80

G:							
0	3	12	28	48	78	114	154
199	247	307	367	427	499	571	643
715	795	875	955	1035	1115	1195	1275
1355	1427	1499	1571	1643	1703	1763	1823
1871	1916	1956	1992	2022	2042	2058	2067

The reverse process gives midpoint longitude and latitude for each box. First find the smallest element j in array G such that $G(j) \geq B_N$. Then

$$LAT_N = \begin{cases} j - 1 & (1 \leq B_N \leq 2067) \\ 40 & (2068 \leq B_N \leq 2070) \end{cases}$$

The column identifier is

$$LON_N = [B_N - G(LAT_N)]$$

For I, as previously defined, the midpoint longitude and latitude are

$$L_{mid} = 360^\circ - [I(LON_N - 1) + I/2]$$

$$\lambda_{mid} = -90^\circ + (LAT_N - 1)4.5^\circ + 2.25^\circ$$

The 3DNEPH Grid System

For a number of years the U.S. Air Force has produced global cloud cover information which is archived at the Environmental Technical Applications Center, Asheville, North Carolina. This is available to civilian users through the National Climatic Center, Asheville, North Carolina. The grid system is an 8 × 8 set of boxes superimposed on a polar stereographic projection, one set of 64 boxes for each hemisphere. Points on the Equator belong in Northern Hemisphere boxes. This grid is described in reference 4 and pictured in figure 4. The four corner boxes are not used. Alignment of the grid along $L = 10^\circ$ is an apparently arbitrary choice. Each box is subdivided into a 64 × 64 array of points spaced a nominal 25 nautical miles apart. Point (1,1) occupies the upper left-hand corner of its box as shown in figure 4(c). Longitude ($-180^\circ \leq L \leq 180^\circ$) and latitude are related to a particular box and to a row and column in that box as follows:

$$u = \frac{\cos(350^\circ + L)}{|\cos(350^\circ + L)|} \sqrt{\frac{(62317.6272)\left(1 - \sin|\lambda|\right)}{\left(1 + \sin|\lambda|\right)[1 + \tan^2(350^\circ + L)]}} \qquad (L \neq 100^\circ \text{ or } -80^\circ)$$

Since u is undefined for $L = 100^\circ$ or -80°, it is necessary (and sufficient) for computation to let

$$L = 99.9999^\circ \qquad \left(|100^\circ - L| < 10^{-4}\right)$$

$$L = -79.9999^\circ \qquad \left(|-80^\circ - L| < 10^{-4}\right)$$

Then

$$v = u \tan(350^\circ + L)$$

Let

$$i = 257 + \mathrm{INT}(u + 0.5)$$

$$j = 257 - \mathrm{sign}(\lambda)[\mathrm{INT}(v + 0.5)]$$

Then let

$$\mathrm{ROW} = j - 64\ \mathrm{INT}[(k - 1)/8]$$

where k is the smallest nonnegative integer such that $1 \le \mathrm{ROW} \le 64$. Further, let

$$\mathrm{COL} = i - 64\ell$$

where ℓ is the smallest nonnegative integer such that $1 \le \mathrm{COL} \le 64$. Finally, ROW and COL are the row and column identifiers in the box numbered

$$B_{3D} = k + \ell$$

As an example of locations on the 3DNEPH grid, the grid point closest to 0° longitude and 0° latitude is in Northern Hemisphere box 40, row 43, column 55. The reverse operation is to convert B_{3D}, ROW, and COL into longitude and latitude:

$$L = \mathrm{MOD}\,[(\tan^{-1}(v/u) - 350^\circ), 360^\circ]$$

$$\lambda = \begin{cases} \sin^{-1} f & \text{(Northern Hemisphere)} \\ -\sin^{-1} f & \text{(Southern Hemisphere)} \end{cases}$$

where

$$\sin f = [62317.6272 - (u^2 + v^2)]/[62317.6272 + (u^2 + v^2)]$$

$$u = i - 257$$

$$v = \begin{cases} 257 - j & \text{(Northern Hemisphere)} \\ j - 257 & \text{(Southern Hemisphere)} \end{cases}$$

$$i = [(B_{3D} - 1) - 8d]64 + COL$$

$$j = 64d + ROW$$

$$d = INT\,[(B_{3D} - 1)/8]$$

Care must be taken to obtain the proper quadrant from the inverse tangent function. The longitude and latitude of the grid point for the above example (box 40, row 43, column 55) are $L = 0.311^\circ$ and $\lambda = 0.017^\circ$.

THE RELATIONSHIP BETWEEN LONGITUDE AND RIGHT ASCENSION

The astronomical equations relating the Sun's position to the Earth are given in a (nearly) inertial framework: an Earth-equatorial system with the x-axis pointing to a particular place in the sky (the first point of Aries, denoted by ♈) which remains fixed for most practical purposes. The basic time system for processing ERBE data is universal time (UT). This is a globally constant mean solar time in which $0^h0^m0^s$ UT starts the calendar day (refs. 5 and 6). UT has the advantage of being a centralized time system which does not depend on an observer's location on the globe. However, there are disadvantages when data must be presented which show diurnal variations related to solar activity. If 12^h UT is local noon at one place on the globe, it is midnight on the other side of the globe. Physically, then, analyses of diurnal behavior are better treated in a local time system with, perhaps, a UT time scale also maintained to retain a standardized link to other data. The equivalence between local time and UT is in one sense a spatial transformation between the inertial coordinate system and the Earth's own rotating longitude-latitude system. UT is defined from the Greenwich meridian (0° longitude), so that it is local mean solar time ("clock time") at Greenwich $\pm$ 7.5° of longitude. Every 15° of longitude corresponds to a 1-hour shift, neglecting local variations and "daylight savings" time. The adjustment is measured positively to the east, so that moving east by 15° requires setting a clock ahead by 1 hour. Given a particular UT and measuring longitude positively from Greenwich eastward,

$$\text{Local hour} = \text{MOD}\left(\left\{\text{UT hour} + \text{INT}\,[(L + 7.5^\circ)/15^\circ]\right\},\ 24\right)$$

Whereas longitude is the measure of angular distance in the Earth's rotating coordinate system, right ascension is the corresponding measure in the inertial system. Longitude and right ascension are related through the equation of time, which contains astronomical information about the Earth's movement relative to the Sun. The geometry is illustrated in figure 5. At Greenwich midnight, the start of a calendar day, the right ascension of the Greenwich meridian is

$$\alpha_{g,o} = 99.6909833^o + 36000.7689^o T + 0.00038708^o T^2$$

where T is the time in Julian centuries from January 0.5, 1900:

$$T = (\text{Julian date} - 241\ 5020.0)/36525$$

A table of Julian dates and a detailed explanation of the use of these equations can be found in reference 6. To determine the right ascension (called sidereal time) of any meridian on the Earth's surface,

$$\alpha_m = \alpha_{g,o} + \omega t + L_m$$

where t is fraction of a day measured from $0^h 0^m 0^s$ UT, ω is the Earth's rotation rate, 360.9856473 deg/day, and L_m is the longitude of the meridian.

THE RELATIONSHIP BETWEEN AN OBSERVER AND THE SUN

The Sun's location in geocentric inertial space can be determined from the equations of the Earth's orbital motion around the Sun. Let $\vec{r}_E$ be the Earth's position vector in heliocentric ecliptic space, where the ecliptic plane is defined as the plane in which the Earth travels around the Sun. (The necessary equations can be found in reference 5 or 6, or in the implementation of this transformation in the appendix.) Then, the unit vector defining the Sun's position relative to the Earth in this same coordinate system is

$$\hat{r}'_s = \left(-x_E/|\vec{r}_E|, -y_E/|\vec{r}_E|, -z_E/|\vec{r}_E|\right)$$

The Sun's unit position vector is rotated from the ecliptic to an Earth-equatorial system:

$$x_s = x'_s$$

$$y_s = y'_s \cos \varepsilon - z'_s \sin \varepsilon$$

$$z_s = y'_s \sin \varepsilon + z'_s \cos \varepsilon$$

where ε is the angle between the ecliptic plane and the plane of the Earth's equator, about 23.45^o. Now, observers on the Earth can relate themselves to the Sun by comparing their right ascension and declination to that of the Sun. Declination is equivalent to latitude and right ascension is found from the equations in the previous section. The geometry is shown in figure 6. In terms of solar right ascension α_s and declination δ_s, the unit vector of the Sun is

$$x_s = \cos \delta_s \cos \alpha_s$$

$$y_s = \cos \delta_s \sin \alpha_s$$

$$z_s = \sin \delta_s$$

Then

$$\delta_s = \sin^{-1} z_s$$

$$\alpha_s = \tan^{-1}(y_s/x_s)$$

where care must be taken to obtain the proper quadrant from the inverse tangent function. The angular distance between the observer's and the Sun's right ascensions is a measure of true local time. Noon is defined as the instant at which the declinations are the same, i.e., when the two meridians coincide. This time is not, in general, the same as clock noon because clock time is based on a fictitious mean Sun which moves with constant speed along the Equator's projection on the celestial sphere. True noon and clock noon always differ by less than 20 minutes, neglecting irregularities in local time zones.

Other useful relationships between an observer and the Sun follow when both position vectors are expressed in the same coordinate system. The solar zenith angle at a ground observation point, as defined in figure 6, is

$$\theta_s = \cos^{-1}\left(\hat{r}_{grd} \cdot \hat{r}_s\right)$$

If an orbiting observer at $\hat{r}_{obs}$ views a point at $\hat{r}_{grd}$, the relative azimuth between Sun and observer is defined in figure 7 as

$$a = \cos^{-1}\left[\left(\hat{r}_{grd} \times \hat{r}_s\right) \cdot \left(\hat{r}_{grd} \times \hat{r}_{obs}\right)\right]$$

CONCLUDING REMARKS

Spatial transformation algorithms have been given for each of four large data systems: the Earth Radiation Budget Experiment (ERBE), eastern Geostationary Operational Environmental Satellite (GOES), Nimbus-Earth Radiation Budget experiment, and U.S. Air Force cloud data (called 3DNEPH). In each case the algorithms deal only with data at a particular resolution level, from tens to hundreds of kilometers, which is much coarser than the basic resolution of the various systems. The information is almost entirely satellite derived from sensors having resolutions of a few kilometers (the 3DNEPH cloud data contain other types of observations as well) and has been extensively preprocessed before this stage of interest. Thus, the equations presented here do not deal directly with the satellite systems themselves, but rather with specific, well-documented data products of interest to ERBE scientific investigations.

Intercomparison of data between grid systems is not a straightforward matter, and this memorandum does not address the large-scale transfer of information from one grid to another. Single points may be transformed from one grid to longitude-latitude and then onto another grid, but this is not the same thing as expressing the distribution of a variable in one system in terms of another system. The transformation between longitude and right ascension uses standard astronomical equations; it is included for a certain completeness of use to the ERBE project. In particular, diurnal models of radiative behavior require specification of solar zenith angle, which always necessitates definition of Earth coordinates relative to the Sun.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
November 24, 1981

APPENDIX

FORTRAN IV LISTINGS

The transformations included in the preceding sections have been implemented in FORTRAN IV. A listing of three main programs with subroutines is given below; they operate in an interactive mode with NASA Langley Research Center central computers, through a remote terminal. The grid system program is LLGRID. There are two subroutines for each grid: one for converting longitude and latitude to grid coordinates and another for converting back again. LLGRID expects terminal keyboard input in free format (list directed READ) of longitude and latitude, from which it obtains grid coordinates for each system. The reverse transformation gives the midpoint longitude and latitude of the grid box or, in the case of 3DNEPH, the longitude and latitude of the closest grid point. Program LONGRA receives a UT calendar date input from the keyboard and converts it to Julian date. It then calculates the right ascension of the input longitude. The relationship between a ground-based observer and the Sun is illustrated by program ZENITH, which calculates the solar zenith angle for a given longitude, latitude, and UT calendar date. Sample input and output follow each program listing.

```
      PROGRAM LLGRID(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     CONVERTS LON AND LAT TO INDICES ON A 3DNEPH,GOES,NIMBUS,
C     OR ERBE GRID. THEN CONVERTS BACK TO CLOSEST GRID LONGITUDE
C     AND LATITUDE FOR 3DNEPH GRID AND TO MIDPOINT LONGITUDE AND
C     LATITUDE FOR THE OTHER GRIDS.
      DIMENSION IROW(3),ICOL(3),IBOX(3)
C     INPUT LONGITUDE AND LATITUDE
    1 PRINT1010
      READ*,XLON,XLAT
      IF(XLON.EQ.999.) GOTO9999
      PRINT1017,XLON,XLAT
C     3DNEPH SYSTEM
      CALL LLNEPH(XLON,XLAT,NEPHB,NROW,NCOL)
      PRINT1000
      IF(XLAT.GE.0.) PRINT1005
      IH=1
      IF(XLAT.LT.0.)IH=2
      IF(XLAT.LT.0.) PRINT1006
      PRINT1001,NEPHB,NROW,NCOL
C     ERBE GRID SYSTEM
      PRINT1015
      S=1.25
      DO 10 I=1,3
      S=S*2.
      CALL LLERB(XLON,XLAT,S,IBOX(I),IROW(I),ICOL(I))
      PRINT1003,S
      PRINT1001,IBOX(I),IROW(I),ICOL(I)
   10 CONTINUE
C     LONGITUDE, LATITUDE TO GOES GRID
      CALL LLGOES(XLON,XLAT,IGOES,LATG,LONG,IERR)
      IF(IERR.EQ.0) GOTO18
      PRINT1007
      GOTO19
   18 PRINT1008
      PRINT1001,IGOES,LATG,LONG
   19 CONTINUE
C     LONGITUDE, LATITUDE TO NIMBUS-ERB GRID
      CALL LLNIM(XLON,XLAT,NIMB,NIMLAT,NIMLON)
      PRINT1012
      PRINT1001,NIMB,NIMLAT,NIMLON
C     RETURN MIDPOINTS FROM GOES GRID
      IF(IERR.NE.0) GOTO20
      CALL GOESLL(LATG,LONG,GLON,GLAT)
      PRINT1014
      PRINT1002,GLON,GLAT
   20 CONTINUE
C     BACK TO LONGITUDE, LATITUDE FROM 3DNEPH GRID
      CALL NEPHLL(NEPHB,IH,NROW,NCOL,ZLON,ZLAT)
      PRINT1004
```

```
      PRINT1002,ZLON,ZLAT
C     RETURN MIDPOINTS FROM ERBE GRIDS
      PRINT1016
      S=1.25
      DO 200 I=1,3
      S=S*2.
      CALL ERBLL(IROW(I),ICOL(I),S,CLON,CLAT)
      PRINT1003,S
      PRINT1009,CLON,CLAT
  200 CONTINUE
C     RETURN MIDPOINTS FROM NIMBUS-ERB GRID
      CALL NIMLL(NIMB,CIMLON,CIMLAT)
      PRINT1013
      PRINT1002,CIMLON,CIMLAT
      GOTO1
 1000 FORMAT(* 3DNEPH LOCATION - BOX, ROW AND COL IDENTIFIERS*)
 1001 FORMAT(3I6)
 1002 FORMAT(* LONGITUDE AND LATITUDE *2F8.3)
 1003 FORMAT(1XF5.1,* DEGREE ERBE GRID*)
 1004 FORMAT(* RETURN LONGITUDE, LATITUDE FROM 3DNEPH BOX*)
 1005 FORMAT(* NORTHERN HEMISPHERE*)
 1006 FORMAT(* SOUTHERN HEMISPHERE*)
 1007 FORMAT(* LONG. OR LAT. OUTSIDE OF RANGE FOR GOES GRID*)
 1008 FORMAT(* GOES GRID BOX, ROW, COLUMN*)
 1009 FORMAT(* CENTER LONGITUDE AND LATITUDE *2F8.3)
 1010 FORMAT(//* INPUT LONGITUDE AND LATITUDE, FREE FORMAT*)
 1012 FORMAT(* NIMBUS-ERB TARGET AREA, LAT AND LON REFERENCE*)
 1013 FORMAT(* RETURN LONGITUDE, LATITUDE FROM NIMBUS-ERB BOX*)
 1014 FORMAT(* RETURN LONGITUDE, LATITUDE FROM GOES GRID*)
 1015 FORMAT(* LOCATIONS ON 3 ERBE GRID SYSTEMS*)
 1016 FORMAT(* RETURN MIDPOINTS FOR 3 ERBE GRIDS*)
 1017 FORMAT(1X2F10.3)
 9999 STOP
      END
```

```
      SUBROUTINE LLNEPH(XLON,XLAT,NEPHB,J1,I1)
C     CONVERTS LONGITUDE, LATITUDE TO 3DNEPH
C     BOX NUMBER AND ROW AND COLUMN INDICES.
C     CONVERTS ANY LONGITUDE TO -180<=LONG<=180
      ANGLE(X)=AMOD(X,360.)
      XL=ANGLE(XLON)
      IF(XL.GT.180.) XL=XL-360.
      IF(ABS(100.-XL).LT.1.E-4)XL=99.9999
      IF(ABS(-80.-XL).LT.1.E-1)XL=-79.9999
      X=SIGN(1.,COSD(350.+XL))*SQRT(62317.6272*(1.-SIND(ABS(XLAT)))/
     C(1.+SIND(ABS(XLAT)))/(1.+TAND(350.+XL)**2))
      Y=X*TAND(350.+XL)
      I=257+INT(X+.5)
      J=257-INT(Y+.5)
      IF(XLAT.LT.0.) J=257+INT(Y+.5)
      DO 100 K=1,64
      J1=(K-1)/8
      J1=J-J1*64
      IF(J1.GT.64)GOTO100
      IR=K
      GOTO110
  100 CONTINUE
  110 DO 120 LL=1,8
      IC=LL-1
      I1=I-IC*64
      IF(I1.GT.64) GOTO120
      GOTO130
  120 CONTINUE
  130 NEPHB=IR+IC
      RETURN
      END
```

```
      SUBROUTINE NEPHLL(IB,H,IROW,ICOL,XLON,XLAT)
C     CONVERTS 3DNEPH BOX, HEMISPHERE, ROW, AND COLUMN
C     BACK TO LONGITUDE AND LATITUDE
      INTEGER H,D
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      RD=57.29577951308
      D=(IB-1)/8
      I=MOD((IB-1),8)*64+ICOL
      J=D*64+IROW
      X=I-257
      Y=257-J
      IF(H.EQ.2) Y=J-257
      F=X*X+Y*Y
      F=(62317.6272-F)/(62317.6272+F)
      IF(F.LT.1)GOTO10
      XLAT=90.
      GOTO20
   10 XLAT=ASIN(F)*RD
   20 IF(H.EQ.2)XLAT=-XLAT
      IF(Y.NE.0.) GOTO33
      IF(X.LT.0.)XLON=-170.
      IF(X.EQ.0.) XLON=0.
      IF(X.GT.0.) XLON=10.
      GOTO34
   33 XLON=ATAN2(Y,X)*RD
      XLON=XLON-350.
   34 XLON=ANGLE(XLON)
      RETURN
      END


      SUBROUTINE LLERB(XLON,XLAT,S,IBOX,LAT,LON)
C     CONVERTS LONGITUDE, LATITUDE TO BOX AND ROW, COLUMN
C     INDICATORS IN 2.5, 5, OR 10 DEGREE ERBE GRIDS.
C     CONVERTS ANY LONGITUDE TO 0<=LONG<360
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      N=3600/IFIX(S*10.)
      XL=ANGLE(XLON)
      N1=N/4+1
      LAT=MIN0(N/2,INT(FLOAT(N1)-XLAT/S+1.E-9))
      LON=MIN0(N,INT(XL/S+1.+1.E-9))
      IBOX=(LAT-1)*N+LON
      RETURN
      END


      SUBROUTINE ERBLL(LAT,LON,S,CLON,CLAT)
C     GIVES CENTER LATITUDE AND LONGITUDE FOR ANY ERB GRID
      CLAT=90.-S/2.-FLOAT(LAT-1)*S
      CLON=FLOAT(LON-1)*S+S/2.
      RETURN
      END
```

```
      SUBROUTINE LLGOES(XLON,XLAT,IGOES,LAT,LON,IERR)
C     CONVERTS LONGITUDE TO BOXES IN A GOES-LANGLEY GRID.
C     CONVERTS ANY LONGITUDE TO -180<=LONG<=180.
      DIMENSION Z(3),D(3),XLO(3),XHI(3)
      ANGLE(X)=AMOD(X,360.)
      DATA Z/123.75,132.50,141.00/,D/2.25,2.50,3.00/
      DATA XLO/-121.5,-130.0,-138.0/,XHI/-31.5,-30.0,-18.0/
      I=INT(ABS(XLAT)/18.+1.)
      XL=ANGLE(XLON)
      IF((XL.LT.XLO(I)).OR.(XL.GE.XHI(I))) GOTO20
      IF(ABS(XLAT).GT.45.)GOTO20
      IERR=0
      LAT=MINO(40,(INT((47.25-XLAT)/2.25+1.E-9)))
      LON=MINO(40,INT((Z(I)+XLON)/D(I)+1.E-9))
      IGOES=(LAT-1)*40+LON
      GOTO100
   20 IERR=1
  100 RETURN
      END
```

```
      SUBROUTINE GOESLL(LAT,LON,CLON,CLAT)
C     CONVERTS GOES LAT,LON IDENTIFIERS TO CENTER LONG AND LAT.
      DIMENSION D(3),Z(3)
      DATA Z/123.75,132.50,141.00/, D/2.25,2.50,3.00/
      CLAT=46.125-2.25*FLOAT(LAT)
      K=LAT
      IF(LAT.GT.20) K=41-LAT
      I=INT(FLOAT(28-K)/8.+1.E-9)
      CLON=D(I)*FLOAT(LON)-(Z(I)-D(I)/2.)
      RETURN
      END
```

```
      SUBROUTINE LLNIM(XLON,XLAT,IB,LAT,LON)
C     CONVERTS LONGITUDE AND LATITUDE TO BOXES IN NIMBUS-ERB GRID
C     CONVERTS ANY LONGITUDE TO0<=LONG<360
      DIMENSION IG(40),A(20)
      COMMON/NIMDAT/IG,A
      DATA A/3.,9.,16.,20.,30.,36.,40.,45.,48.,3*60.,4*72.,4*80./
      DATA IG/   0,    3,   12,   28,   48,   78,  114,  154,
     C          199,  247,  307,  367,  427,  499,  571,  643,
     C          715,  795,  875,  955, 1035, 1115, 1195, 1275,
     C         1355, 1427, 1499, 1571, 1643, 1703, 1763, 1823,
     C         1871, 1916, 1956, 1992, 2022, 2042, 2058, 2067/
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      XL=ANGLE(XLON)
      LAT=MINO(40,(INT((XLAT+90.)/4.5+1.E-9)+1))
      I=LAT
      IF(LAT.GT.20) I=41-LAT
      C=360.-XL
      T=360./A(I)
      LON=INT(C/T+1.E-9)+1
      IF(XL.EQ.0.) LON=LON-INT(A(I))
      IB=IG(LAT)+LON
      RETURN
      END
```

```
      SUBROUTINE NIMLL(IB,CLON,CLAT)
C     GIVES CENTER LONGITUDE AND LATITUDE FOR ANY NIMBUS-ERB BOX
      DIMENSION IG(40),A(20)
      COMMON/NIMDAT/IG,A
      DO 10 I=1,40
      IF(IG(I).LT.IB) GOTO10
      LAT=I-1
      GOTO20
   10 CONTINUE
      LAT=40
   20 J=LAT
      IF(J.GT.20) J=41-LAT
      LON=IB-IG(LAT)
      T=360./A(J)
      CLON=360.-(T*FLOAT(LON-1)+T/2.)
      CLAT=-90.+FLOAT(LAT-1)*4.5+2.25
      IF(CLON.GT.180.)CLON=CLON-360.
      RETURN
      END
```

APPENDIX

Sample Output From Program LLGRID

```
INPUT LONGITUDE AND LATITUDE, FREE FORMAT
     0.000     0.000
3DNEPH LOCATION - BOX, ROW AND COL IDENTIFIERS
NORTHERN HEMISPHERE
   40    43    55
LOCATIONS ON 3 ERBE GRID SYSTEMS
  2.5 DEGREE ERBE GRID
 5185    37     1
  5.0 DEGREE ERBE GRID
 1297    19     1
 10.0 DEGREE ERBE GRID
  325    10     1
LONG. OR LAT. OUTSIDE OF RANGE FOR GOES GRID
NIMBUS-ERB TARGET AREA, LAT AND LON REFERENCE
 1036    21     1
RETURN LONGITUDE, LATITUDE FROM 3DNEPH BOX
LONGITUDE AND LATITUDE     .311    .017
RETURN MIDPOINTS FOR 3 ERBE GRIDS
  2.5 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE    1.250  -1.250
  5.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE    2.500  -2.500
 10.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE    5.000  -5.000
RETURN LONGITUDE, LATITUDE FROM NIMBUS-ERB BOX
LONGITUDE AND LATITUDE   -2.250   2.250


INPUT LONGITUDE AND LATITUDE, FREE FORMAT
     0.000    90.000
3DNEPH LOCATION - BOX, ROW AND COL IDENTIFIERS
NORTHERN HEMISPHERE
   37     1     1
LOCATIONS ON 3 ERBE GRID SYSTEMS
  2.5 DEGREE ERBE GRID
    1     1     1
  5.0 DEGREE ERBE GRID
    1     1     1
 10.0 DEGREE ERBE GRID
    1     1     1
LONG. OR LAT. OUTSIDE OF RANGE FOR GOES GRID
NIMBUS-ERB TARGET AREA, LAT AND LON REFERENCE
 2068    40     1
RETURN LONGITUDE, LATITUDE FROM 3DNEPH BOX
LONGITUDE AND LATITUDE    0.000  90.000
RETURN MIDPOINTS FOR 3 ERBE GRIDS
  2.5 DEGREE ERBE GRID
```

Sample Output From Program LLGRID - Continued

```
CENTER LONGITUDE AND LATITUDE    1.250  88.750
  5.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE    2.500  87.500
 10.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE    5.000  85.000
RETURN LONGITUDE, LATITUDE FROM NIMBUS-ERB BOX
LONGITUDE AND LATITUDE  -60.000  87.750


INPUT LONGITUDE AND LATITUDE, FREE FORMAT
     0.000   -90.000
3DNEPH LOCATION - BOX, ROW AND COL IDENTIFIERS
SOUTHERN HEMISPHERE
   37     1     1
LOCATIONS ON 3 ERBE GRID SYSTEMS
  2.5 DEGREE ERBE GRID
10225    72     1
  5.0 DEGREE ERBE GRID
 2521    36     1
 10.0 DEGREE ERBE GRID
  613    18     1
LONG. OR LAT. OUTSIDE OF RANGE FOR GOES GRID
NIMBUS-ERB TARGET AREA, LAT AND LON REFERENCE
    1     1     1
RETURN LONGITUDE, LATITUDE FROM 3DNEPH BOX
LONGITUDE AND LATITUDE    0.000 -90.000
RETURN MIDPOINTS FOR 3 ERBE GRIDS
  2.5 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE    1.250 -88.750
  5.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE    2.500 -87.500
 10.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE    5.000 -85.000
RETURN LONGITUDE, LATITUDE FROM NIMBUS-ERB BOX
LONGITUDE AND LATITUDE  -60.000 -87.750


INPUT LONGITUDE AND LATITUDE, FREE FORMAT
  -100.000    40.000
3DNEPH LOCATION - BOX, ROW AND COL IDENTIFIERS
NORTHERN HEMISPHERE
   44    45    26
LOCATIONS ON 3 ERBE GRID SYSTEMS
  2.5 DEGREE ERBE GRID
 2985    21   105
  5.0 DEGREE ERBE GRID
  773    11    53
 10.0 DEGREE ERBE GRID
  207     6    27
```

Sample Output From Program LLGRID - Concluded

```
GOES GRID BOX, ROW, COLUMN
   93     3    13
NIMBUS-ERB TARGET AREA, LAT AND LON REFERENCE
 1660    29    17
RETURN LONGITUDE, LATITUDE FROM GOES GRID
LONGITUDE AND LATITUDE -100.500  39.375
RETURN LONGITUDE, LATITUDE FROM 3DNEPH BOX
LONGITUDE AND LATITUDE  260.145  40.597
RETURN MIDPOINTS FOR 3 ERBE GRIDS
  2.5 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE  261.250  38.750
  5.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE  262.500  37.500
 10.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE  265.000  35.000
RETURN LONGITUDE, LATITUDE FROM NIMBUS-ERB BOX
LONGITUDE AND LATITUDE  -99.000  38.250


INPUT LONGITUDE AND LATITUDE, FREE FORMAT
   -75.000   -45.000
3DNEPH LOCATION - BOX, ROW AND COL IDENTIFIERS
SOUTHERN HEMISPHERE
   21    27    10
LOCATIONS ON 3 ERBE GRID SYSTEMS
  2.5 DEGREE ERBE GRID
 7891    55   115
  5.0 DEGREE ERBE GRID
 2002    28    58
 10.0 DEGREE ERBE GRID
  497    14    29
GOES GRID BOX, ROW, COLUMN
 1582    40    22
NIMBUS-ERB TARGET AREA, LAT AND LON REFERENCE
  320    11    13
RETURN LONGITUDE, LATITUDE FROM GOES GRID
LONGITUDE AND LATITUDE  -73.500 -43.875
RETURN LONGITUDE, LATITUDE FROM 3DNEPH BOX
LONGITUDE AND LATITUDE  285.042 -45.395
RETURN MIDPOINTS FOR 3 ERBE GRIDS
  2.5 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE  286.250 -46.250
  5.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE  287.500 -47.500
 10.0 DEGREE ERBE GRID
CENTER LONGITUDE AND LATITUDE  285.000 -45.000
RETURN LONGITUDE, LATITUDE FROM NIMBUS-ERB BOX
LONGITUDE AND LATITUDE  -75.000 -42.750
```

```
      PROGRAM LONGRA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     DETERMINES RIGHT ASCENSION FOR AN INPUT LONGITUDE AND
C     CALENDAR DATA.
      DIMENSION X(6)
      PRINT1001
    1 READ*,X,XLONG
      IF(X(1).EQ.99.) GOTO999
      PRINT1004,X,XLONG
      CALL CALJUL1(X,WJD,FJD)
      XJD=WJD+FJD
      CALL SIDTIM(WJD,FJD,XLONG,RA)
      PRINT1002
      PRINT1003,X,XJD,XLONG,RA
      GOTO1
 1001 FORMAT(//* INPUT CALENDAR DATE YEAR(LAST 2 DIGITS),MONTH,DAY,
     CHOUR,MINUTE,SEC, THEN LONGITUDE(DEG).  FREE FORMAT.*,
     C* ENTER X(1)=99 TO STOP.*)
 1002 FORMAT(* CALENDAR DATE, JULIAN DATE, LONGITUDE, RIGHT ASCENSION*)
 1003 FORMAT(1X6F3.0,F15.6,2F8.2)
 1004 FORMAT(1X6F4.0,F10.3)
  999 STOP
      END
```

```
      SUBROUTINE SIDTIM(WJD,FJD,ALONG,ST)
C              CALCULATES LOCAL SIDERAL TIME
C              A=JULIAN DATE AT 0 HRS. UNIVERSAL TIME
C              B=NUMBER OF MINUTES FROM 0 HRS. U.T. TO LOCAL TIME
      IF(FJD.LE..5) GO TO 1
      A=WJD+.5
      B=(FJD-.5)*1440.
      GO TO 2
    1 A=WJD-.5
      B=(FJD+.5)*1440.
    2 TU=(A-2415020.)/36525.
      TGO=99.6909833+36000.7689*TU+.00038708*TU**2+.25068447*B+ALONG
      ST=AMOD(TGO,360.)+180.-SIGN(180.,TGO)
      RETURN
      END
```

```
      SUBROUTINE CALJUL1(X,WJD,FJD)
C     INPUT : X=YR(LAST TWO DIGITS),MONTH,DAY,HR,MIN,SEC, UT
C     OUTPUT: WJD=WHOLE JULIAN DATE, FJD=FRACTIONAL DATA
      DIMENSION X(6),A(12)
      D50=2433282.
      YD=X(1)-48.
      YL=YD/4.
      KYL=YL
      CK=KYL
      IF(YL-CK)1,1,3
    1 IF(X(2)-2.)4,4,3
    3 DS=CK
      GOTO5
    4 DS=CK-1.
    5 DS=DS+365.*(YD-2.)
      DO 6 I=1,12
    6 A(I)=1.
      K=X(2)
      DO 7 I=K,12
    7 A(I)=0.
      DS=DS+31.*(A(1)+A(3)+A(5)+A(7)+A(8)+A(10)+A(12))
     C+30.*(A(4)+A(6)+A(9)+A(11))+28.*A(2)
      DS=DS+X(3)-1.
      WND=DS
      FD=X(4)/24.+X(5)/1440.+X(6)/86400.
      IF(FD-.4999999)9,8,8
    8 FJD=FD-.5
      WJD=1.
      GOTO10
    9 FJD=FD+.5
      WJD=0.
   10 WJD=D50+WJD+WND
      RETURN
      END
```

Sample Output From Program LONGRA

```
INPUT (UT) CALENDAR DATE YEAR(LAST 2 DIGITS),MONTH,DAY, HOUR,MINUTE,SEC,
THEN LONGITUDE(DEG).  FREE FORMAT. ENTER X(1)=99 TO STOP.
 81.  3. 21. 12.  0.  0.      0.000
CALENDAR DATE, JULIAN DATE, LONGITUDE, RIGHT ASCENSION
81. 3.21.12. 0. 0. 2444685.000000    0.00  358.92
 81.  1.  1.  0.  0.  0.      0.000
CALENDAR DATE, JULIAN DATE, LONGITUDE, RIGHT ASCENSION
81. 1. 1. 0. 0. 0. 2444605.500000    0.00  100.56
 81.  9. 29. 12.  0.  0.    -75.000
CALENDAR DATE, JULIAN DATE, LONGITUDE, RIGHT ASCENSION
81. 9.29.12. 0. 0. 2444877.000000  -75.00  113.16
 81.  9. 29. 13.  0.  0.    -75.000
CALENDAR DATE, JULIAN DATE, LONGITUDE, RIGHT ASCENSION
81. 9.29.13. 0. 0. 2444877.041667  -75.00  128.20
```

```
      PROGRAM ZENITH(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     CALCULATES COSINE OF SOLAR ZENITH ANGLE FOR INPUT OF
C     LONGITUDE, LATITUDE, GMT CALENDAR DATE.
      DIMENSION X(6)
      PRINT1001
    1 READ*,X,XLON,XLAT
      IF(X(1).EQ.0.) GOTO999
      PRINT1004,X,XLON,XLAT
      CALL CALJUL1(X,WJD,FJD)
      XJD=WJD+FJD
      CALL CZEN(XJD,XLAT,XLON,SOLZEN)
      CSZEN=COSD(SOLZEN)
      PRINT1002
      PRINT1003,X,XJD,XLON,XLAT,SOLZEN,CSZEN
      GOTO1
 1001 FORMAT(//* INPUT YR(LAST 2 DIGITS), MON,DAY,HR,MIN,SEC,LONG,LAT*/
     C* SET X(1)=0 TO TERMINATE*)
 1002 FORMAT(* CAL., JUL. DATE, LONG, LAT, SOLAR ZENITH ANGLE & COS*)
 1003 FORMAT(1X6F3.0,F15.5,2F7.2,F10.4,F7.4)
 1004 FORMAT(1X6F4.0,2F10.3)
  999 STOP
      END
```

```
      SUBROUTINE CZEN(XJD,XLAT,XLON,SOLZEN)
C     FINDS COSINE OF SOLAR ZENITH ANGLE FOR JULIAN DATE,
C     LATITUDE, AND LONGITUDE
      CALL EEARTH(XJD,X,Y,Z,DX,DY,DZ)
      R=SQRT(X*X+Y*Y+Z*Z)
      XS=-X/R
      YS=-Y/R
      ZS=-Z/R
      CALL RECEQ(XJD,XS,YS,ZS,XQ,YQ,ZQ)
      WJD=INT(XJD)
      FJD=XJD-WJD
      CALL SIDTIM(WJD,FJD,XLON,RA)
      XE=COSD(RA)*COSD(XLAT)
      YE=SIND(RA)*COSD(XLAT)
      ZE=SIND(XLAT)
C     COSZEN IS DOT PRODUCT OF UNIT SUN AND POSITION VECTORS
      SOLZEN=ACOS(XE*XQ+YE*YQ+ZE*ZQ)/.01745329252
      RETURN
      END
```

```
      SUBROUTINE CALJUL1(X,WJD,FJD)
```

(See subroutine listings for program LONGRA)

```
      SUBROUTINE EEARTH(JD,XHE,YHE,ZHE,DXHE,DYHE,DZHE)
C
C     THIS SUBROUTINE COMPUTES THE HELIOCENTRIC POSITION AND VELOCITY OF
C     THE EARTH IN MEAN EQUINOX AND ECLIPTIC OF DATE COORDINATE SYSTEM.
C     THIS ROUTINE CALLS SUBROUTINES TINVS AND CONCAR.
C
C     JD - JULIAN DATE
C     XHE,YHE,ZHE - POSITION OF EARTH
C     DXHE,DYHE,DZHE - VELOCITY OF EARTH
C
      REAL JD
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      AU=149598845. $ USUN=1.3271411E+11
      DR=.017453292519943
      RD=57.2957795130823
C
      D=JD-2415020.
      CD=D/10000.
      TE=D/36525.
C
      AE=1.00000023*AU
      EE=0.01675104-0.0000418O*TE-0.000000126*TE**2
      XIE=0.0
      WE=101.220833+0.000047068*D+0.0000339*CD**2+0.00000007*CD**3
      OE=0.0
      XME=ANGLE(358.475845+0.985600267*D-0.0000112*CD**2-0.00000007*CD**
     13)
C
      CALL TINVS(XME*DR,EE,ECE,FE)
      CALL CONCAR(AE,EE,XIE,WE,OE,FE*RD,XHE,YHE,ZHE,DXHE,DYHE,DZHE,USUN)
C
      RETURN
      END
```

```
      SUBROUTINE CONCAR(A,E,XI,W,O,F,X,Y,Z,DX,DY,DZ,U)
C     NOTE: THIS IS A SINGLE PRECISION VERSION OF CONCAR
      DR=.01745329251994З
      FR=DR*F
      WFR=DR*(W+F)
      OR=DR*O
      XIR=DR*XI
      R=A*(1.-E*E)/(1.+E*COS(FR))
      V=SQRT(U*(2./R-1./A))
      GAM=ATAN(E*SIN(FR)/(1.+E*COS(FR)))
      WFGR=WFR-GAM
      CWF=COS(WFR)
      SWF=SIN(WFR)
      SO=SIN(OR)
      CO=COS(OR)
      SI=SIN(XIR)
      CI=COS(XIR)
      SWFG=SIN(WFGR)
      CWFG=COS(WFGR)
      X=R*(CWF*CO-SWF*SO*CI)
      Y=R*(CWF*SO+SWF*CO*CI)
      Z=R*(SWF*SI)
      DX=V*(-SWFG*CO-CWFG*SO*CI)
      DY=V*(-SWFG*SO+CWFG*CO*CI)
      DZ=V*(CWFG*SI)
      RETURN
      END


      SUBROUTINE  TINVS(M,E,EC,F)
      REAL M,MO
      DATA PI/3.141592653589793/
      ASINH(X)=SIGN(ALOG(ABS(X)+SQRT(X**2+1.)),X)
      IF(E.GE.1.)GO TO 100
      EC=M
   10 MO=EC-E*SIN(EC)
      DM=M-MO
      DE=DM/(1.-E*COS(EC))
      EC=EC+DE
      IF(ABS(DE).GT.1.E-12  )GO TO 10
      HEC= EC/2.
      HF=ATAN(SQRT((1.+E)/(1.-E))*SIN(HEC)/COS(HEC))
      IF(HF.LT.0.)HF=HF+PI
      F=2.*HF
      GO TO 800
  100 CONTINUE
      EC=ASINH(M/E)
  101 MO=E*SINH(EC)-EC
      DM=M-MO
      DE=DM/(E*COSH(EC)-1.)
      EC=EC+DE
      IF(ABS(DE).GT.1.E-12  )GO TO 101
      F=2.*ATAN(SQRT((E+1.0)/(E-1.0))*TANH(EC/2.0))
  800 RETURN
      END
```

```
      SUBROUTINE RECEQ(JD,XEC,YEC,ZEC,XEQ,YEQ,ZEQ)
C
C     THIS SUBROUTINE ROTATES A VECTOR FROM GEOCENTRIC, ECLIPTIC, TO
C     THE GEOCENTRIC, EARTH EQUATORIAL COORDINATE SYSTEM
C
C     JD - JULIAN DATE
C     XEC,YEC,ZEC - COMPONENTS OF THE VECTOR IN THE GEOCENTRIC, ECLIPTIC
C                   COORDINATE SYSTEM
C     XEQ,YEQ,ZEQ - COMPONENTS OF THE VECTOR IN THE GEOCENTRIC, EARTH
C                   EQUATORIAL, COORDINATE SYSTEM
C
      REAL JD
      DR=.017453292519943
      TE=(JD-2415020.)/36525.
      XIE=23.452294-0.0130125*TE-0.00000164*TE**2+0.000000503*TE**3
      C=COS(XIE*DR)
      S=SIN(XIE*DR)
      XEQ=XEC
      YEQ=YEC*C-ZEC*S
      ZEQ=YEC*S+ZEC*C
      RETURN
      END
```

```
      SUBROUTINE SIDTIM(WJD,FJD,ALONG,ST)
```

(See subroutine listings for program LONGRA)

Sample Output From Program ZENITH

```
INPUT YR(LAST 2 DIGITS), MON,DAY,HR,MIN,SEC,LONG,LAT
SET X(1)=0 TO TERMINATE
 81.  9. 29. 12.  0.  0.     0.000    37.000
CAL., JUL. DATE, LONG, LAT, SOLAR ZENITH ANGLE & COS
81. 9.29.12. 0. 0.  2444877.00000   0.00  37.00   39.5511  .7711
 81.  9. 29. 13.  0.  0.   -75.000    37.000
CAL., JUL. DATE, LONG, LAT, SOLAR ZENITH ANGLE & COS
81. 9.29.13. 0. 0.  2444877.04167 -75.00  37.00   66.3318  .4014
 81.  3. 21. 12.  0.  0.     0.000     0.000
CAL., JUL. DATE, LONG, LAT, SOLAR ZENITH ANGLE & COS
81. 3.21.12. 0. 0.  2444685.00000   0.00   0.00    1.8376  .9995
```

REFERENCES

1. Woerner, Charles V.; Cooper, John E.; and Harrison, Edwin F.: The Earth Radiation Budget Satellite System for Climate Research. Remote Sounding of the Atmosphere From Space, H.-J. Bolle, ed., Pergamon Press, Inc., c.1979, pp. 201-215.

2. Minnis, Patrick; and Harrison, Edwin F.: Diurnal Variability of Regional Cloud Radiative Parameters From Geostationary Satellite Data. Fourth Conference on Atmospheric Radiation, American Meteorol. Soc., c.1981, pp. 219-226.

3. Smith, W. L.; Hickey, J.; Howell, H. B.; Jacobowitz, H.; Hilleary, D. T.; and Drummond, A. J.: Nimbus-6 Earth Radiation Budget Experiment. Appl. Opt., vol. 16, no. 2, Feb. 1977, pp. 306-318.

4. Fye, Falko K.: The AFGWC Automated Cloud Analysis Model. AFGWC-TM-78-002, U.S. Air Force, June 1978. (Available from DTIC as AD A057 176.)

5. Explanatory Supplement to the Astronomical Ephemeris and the American Ephemeris and Nautical Almanac. H.M. Naut. Alm. Off., 1961.

6. Brooks, David R.: An Introduction to Orbit Dynamics and Its Application to Satellite-Based Earth Monitoring Missions. NASA RP-1009, 1977.

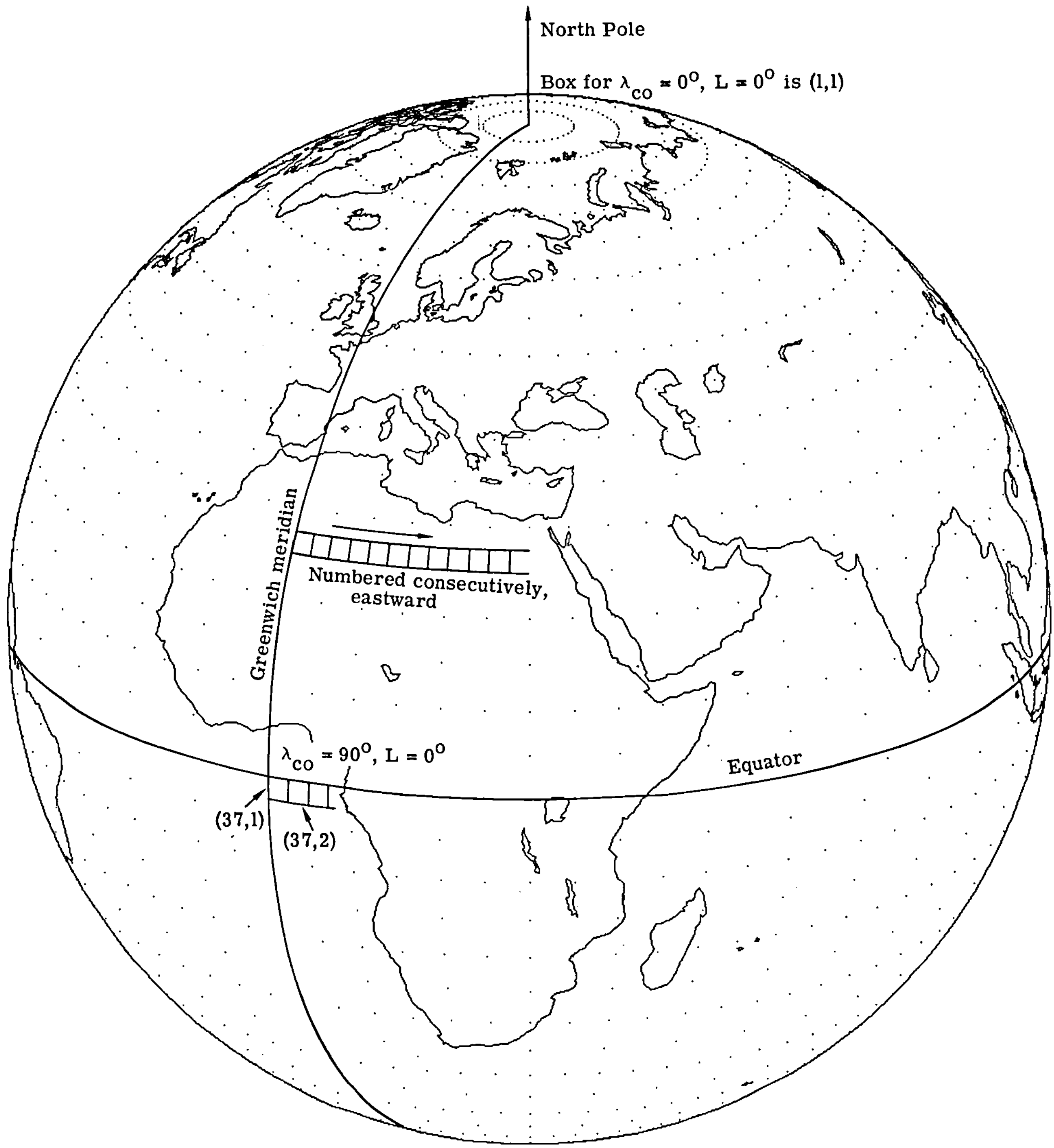


Figure 1.- The 2.5° ERBE grid system.

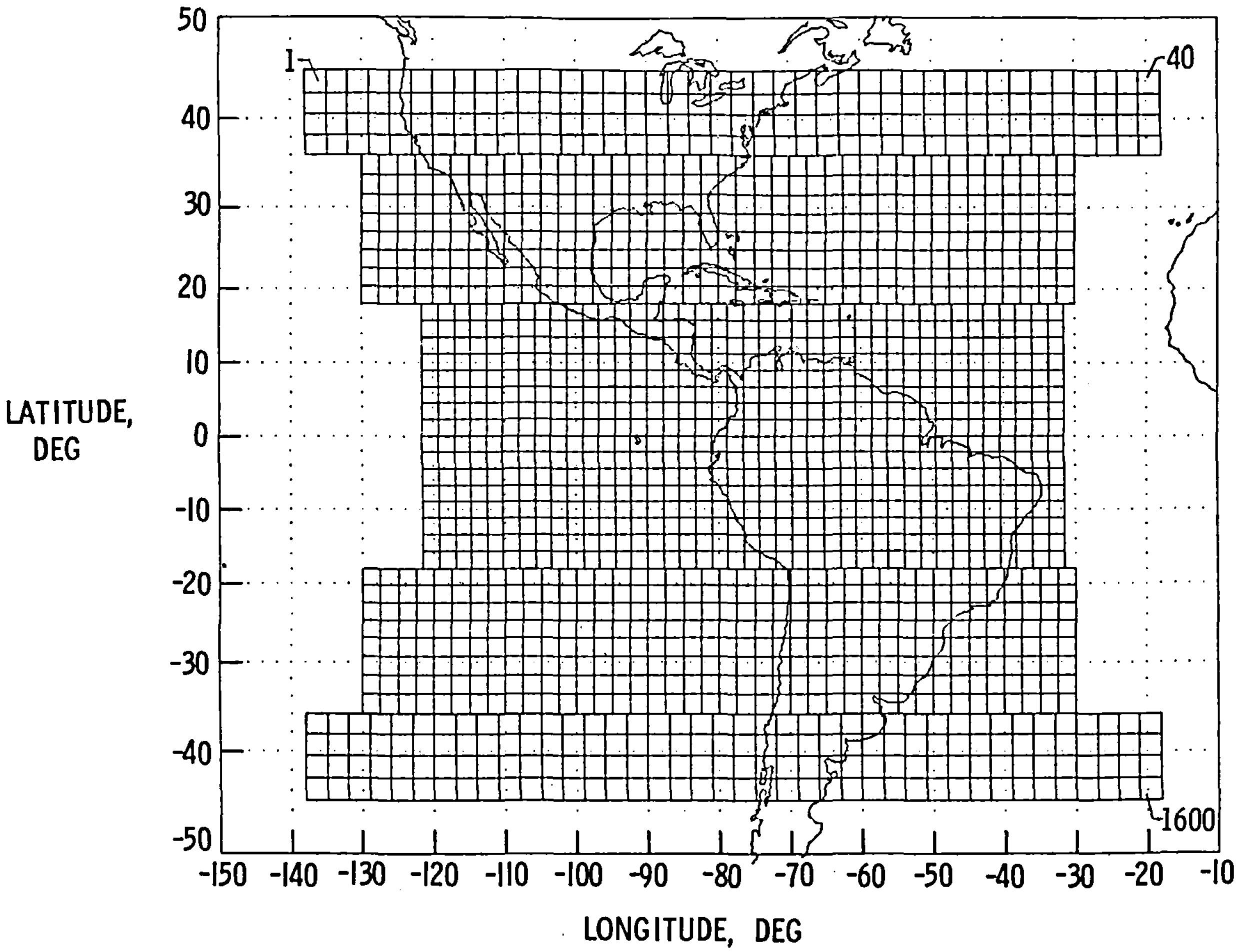


Figure 2.- The Langley GOES grid system.

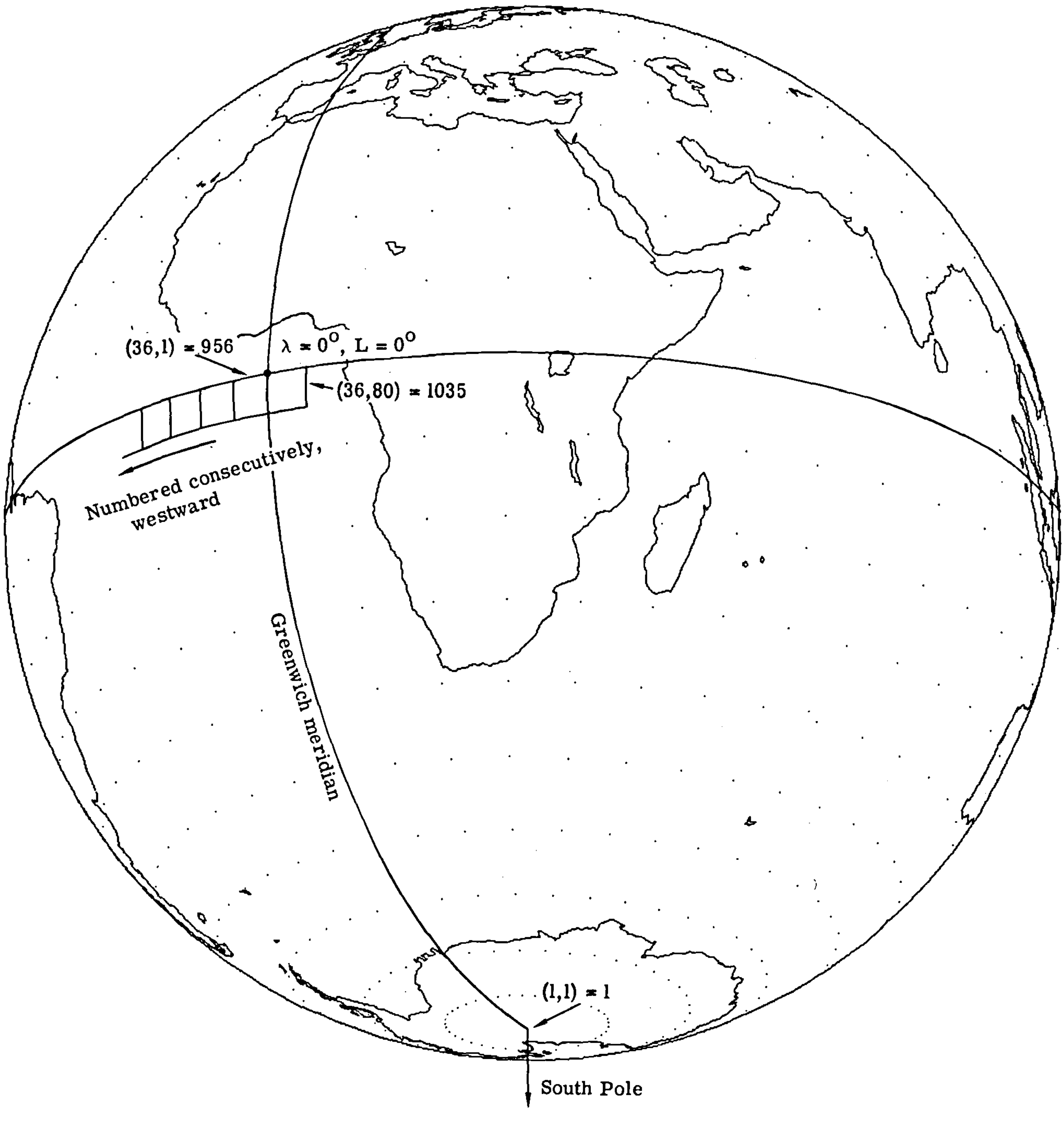
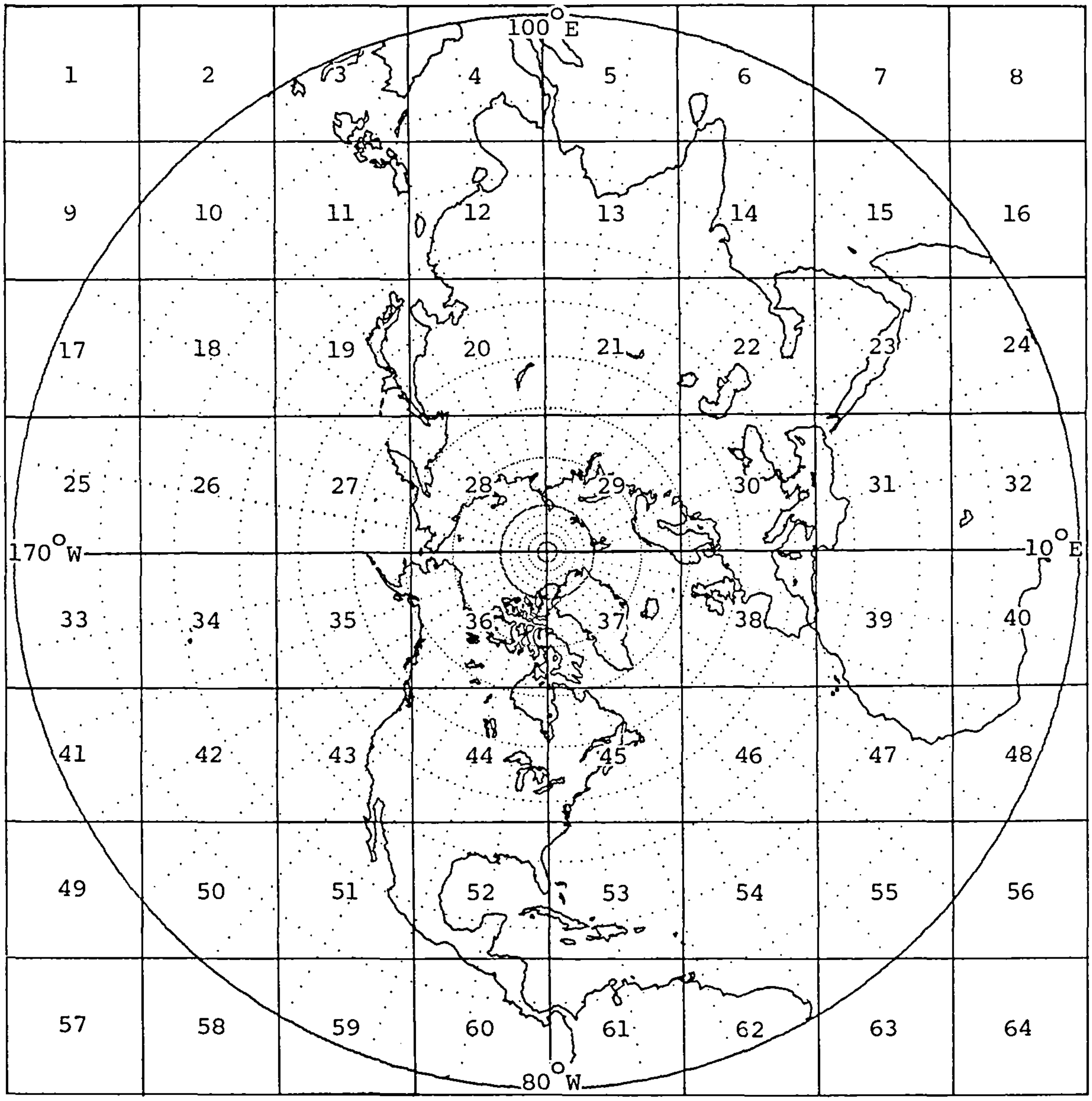
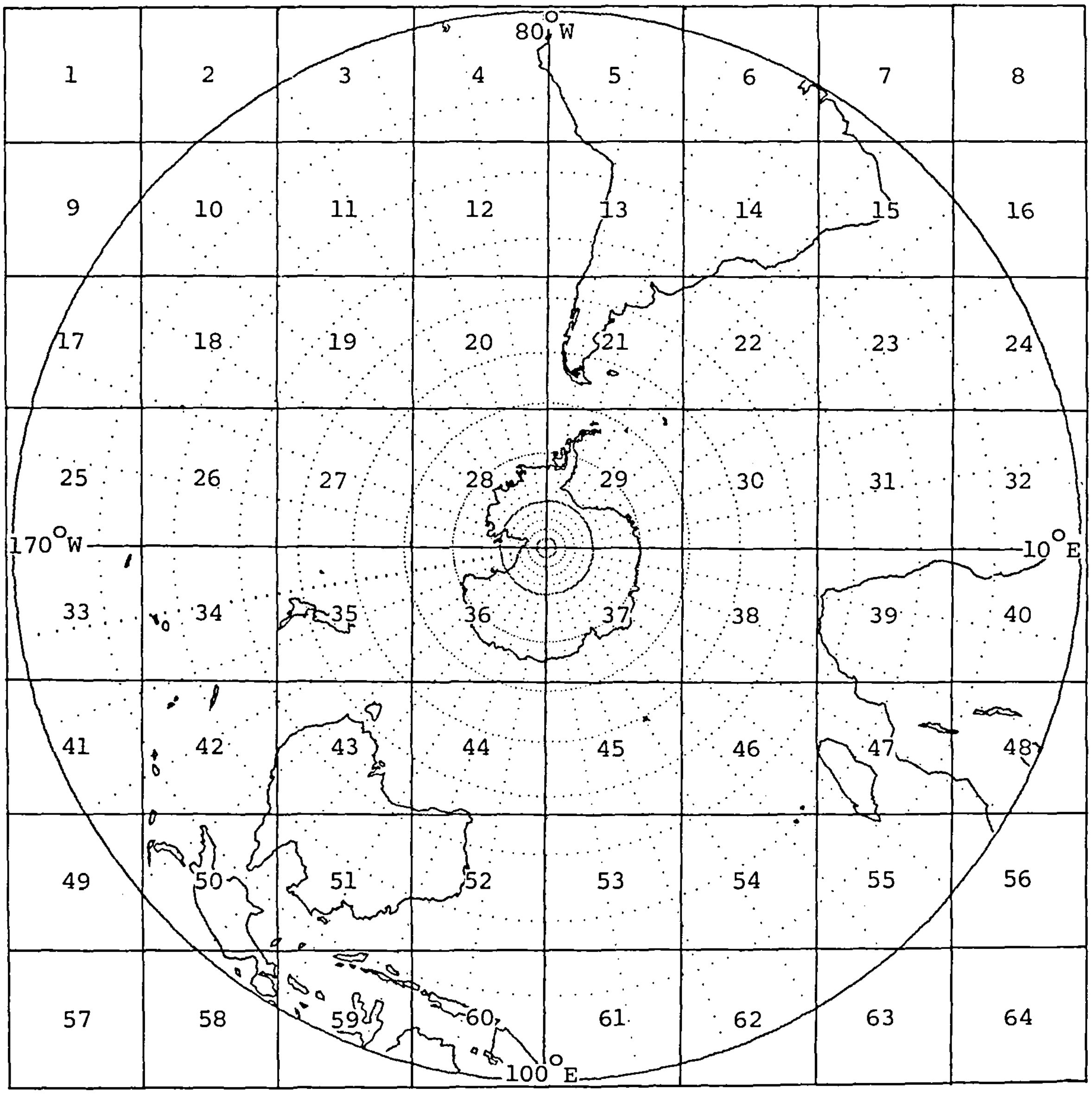


Figure 3.- The Nimbus-ERB grid system.



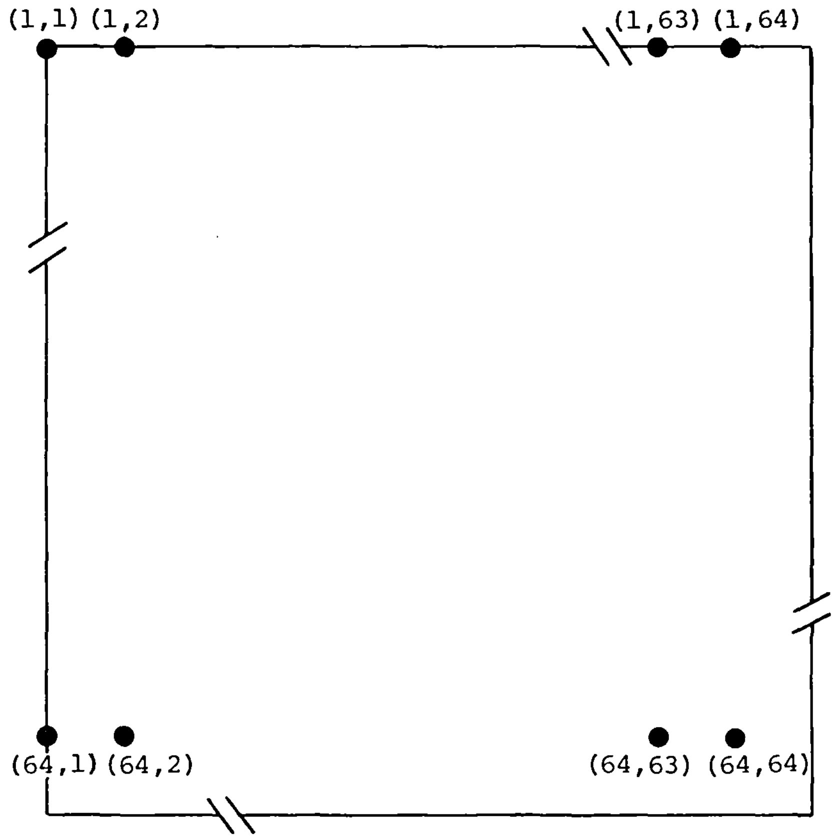
(a) Northern Hemisphere.

Figure 4.- The 3DNEPH grid system over Northern and Southern Hemisphere polar stereographic projection.



(b) Southern Hemisphere.

Figure 4.- Continued.



(c) Arrangement of grid points in a 3DNEPH box.

Figure 4.- Concluded.

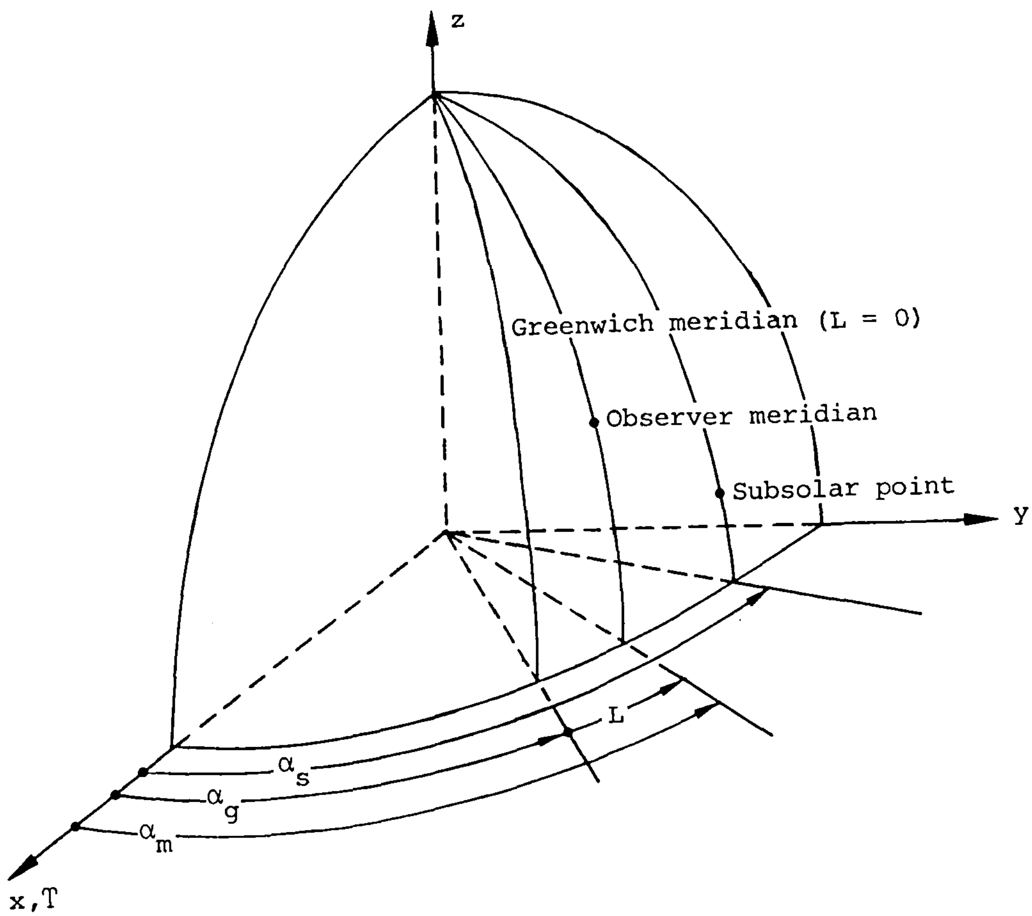


Figure 5.- Relationship between longitude and right ascension.

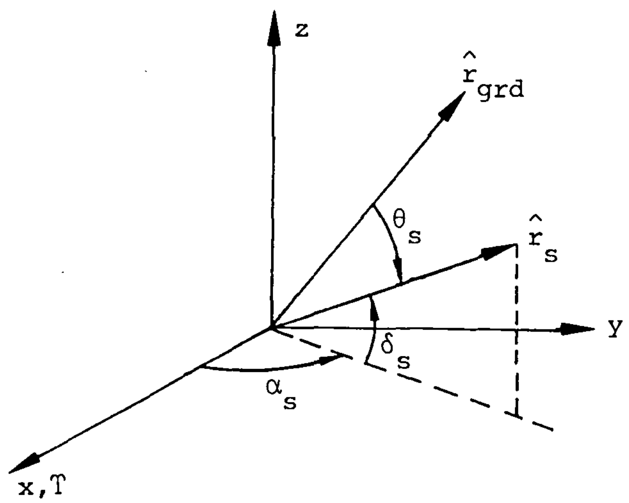


Figure 6.- Definition of solar right ascension, declination, and zenith angles in terms of solar and ground observer unit position vector.

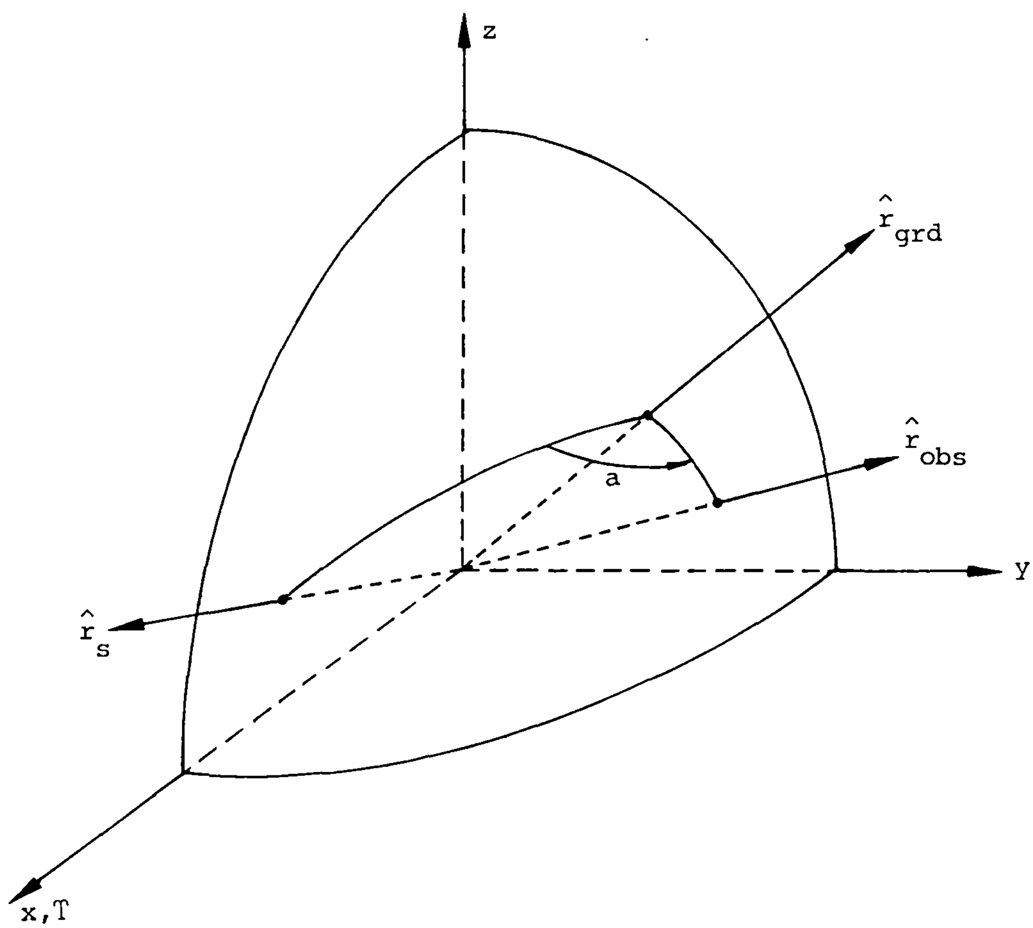


Figure 7.- Definition of relative azimuth angle between the Sun and an orbiting observer of a ground point.

1. Report No. NASA TM-83233	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle GRID SYSTEMS FOR EARTH RADIATION BUDGET EXPERIMENT APPLICATIONS		5. Report Date December 1981
		6. Performing Organization Code 619-12-30-02
7. Author(s) David R. Brooks		8. Performing Organization Report No. L-14843
		10. Work Unit No.
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Spatial coordinate transformations are developed for several global grid systems of interest to the Earth Radiation Budget Experiment. The grid boxes are defined in terms of a regional identifier and longitude-latitude indices. The transformations associate longitude and latitude with a particular grid box. The reverse transformations identify the center location of a given grid box. Transformations are given to relate the rotating (Earth-based) grid systems to solar position expressed in an inertial (nonrotating) coordinate system. FORTRAN implementations of the transformations are given in an appendix, along with sample input and output.

17. Key Words (Suggested by Author(s))		18. Distribution Statement	
Earth Radiation Budget Experiment (ERBE) Grid systems		Unclassified - Unlimited Subject Category 61	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 40	22. Price A03

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1981

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return