



FILE COPY

THE NEPRF SUNGLINT PATTERN CALCULATION AND DISPLAY SYSTEM

AD-A205 542

James R. Clark and Allan Caughey
Naval Environmental Prediction Research Facility

DTIC
ELECTE
MAR 23 1989
S H D

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION IS UNLIMITED

89 3 22 176

QUALIFIED REQUESTORS MAY OBTAIN ADDITIONAL COPIES
FROM THE DEFENSE TECHNICAL INFORMATION CENTER.
ALL OTHERS SHOULD APPLY TO THE NATIONAL TECHNICAL
INFORMATION SERVICE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR 89-01			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION (Naval Environmental Prediction Research Facility)		6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5006			7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Office of Naval Research		8b. OFFICE SYMBOL (if applicable) ONT, Code 22	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code) Arlington, VA 22217			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. 62435N	PROJECT NO. RM35G82	TASK NO. 4
			WORK UNIT ACCESSION NO. DN658752		
11. TITLE (Include Security Classification) The NEPRF Sunlight Pattern Calculation and Display System (U)					
12. PERSONAL AUTHOR(S) Clark, James R. and Caughey, Allan B.					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM 10/87 TO 9/88		14. DATE OF REPORT (Year, Month, Day) 1989, January	
15. PAGE COUNT 63					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD 04	GROUP 02	SUB-GROUP	Sunlight pattern Ruff model Reflectance.		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
The sunlight intensity model used at NEPRF is described. The model is based on the work of I. Ruff of NOAA-NESS with theory from the work of C. Cox and W. Munk. A detailed derivation of all equations used in the model is given, and graphic depictions are provided. NEPRF's system written on the Hewlett-Packard 9020 with Global Imaging operating system also is presented. Examples of sunlight patterns are shown. The computer code and operating instructions are provided as appendices.					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL James R. Clark			22b. TELEPHONE (Include Area Code) (408) 647-4727		22c. OFFICE SYMBOL NEPRF WU 6.2-9

CONTENTS

1. Introduction and Background	1
2. The Ruff Model	3
3. The NEPRF System	25
References	28
Appendix A -- Sunlint Program Listing	A-1
Appendix B -- Sunlint Program Running Instructions	B-1
Distribution	Dist-1



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Special
A-1	

1. INTRODUCTION AND BACKGROUND

One of the most interesting phenomena seen in satellite images in the visual wavelengths is sunglint, sometimes referred to as sun glitter. Over oceanic regions or even lakes and rivers in the tropic and subtropic regions, there appear to be areas of increased brightness or reflectance in the observed image where there are no obvious clouds. This is the phenomenon of sunglint, which is caused by the reflectance of the sun from a "shimmering" water surface at just the correct relative sun-observer position relationships. The slopes and directions of the wave facets are such that the reflected solar energy is directed into an observing system, such as a satellite sensor. The original studies of sunglint/wind relationships were done from an aircraft.

Locating sunglint patterns is a problem composed of two parts. The first is to locate the sun and satellite on a universal coordinate system, as well as to calculate all geometrical quantities and relationships as a function of date and time. This is basically a problem in spherical trigonometry. The second part is a determination of the relationships as a function of various angles of reflectance and the sea state conditions. The NEPRF system is based on the work of Cox and Munk (1954).

In 1979, NEPRF acquired a computer program to calculate the sunglint pattern. This program, coded to run on the Control Data Corporation (CDC) 6500, produced a hard copy on the Varian printer. Isopleths of reflectance were produced for a constant

wind speed in the sunglint region. The final graphic chart was placed over hard copy DMSP images using a light table. The image and graphic were co-registered by using the technique of Tsui and Fett (1980). The latitude and longitude of the primary spectral point (PSP) were determined by hand, and the scales of the graphics and the image were made the same. The system was completely manual. Input into the reflectance program was also done manually, with the required parameters such as satellite height, inclination angle, time and longitude of the ascending node, and the time of the image entered on punched cards.

With the advent of relatively inexpensive mini/micro computers and graphics, such as the Hewlett-Packard (HP) 9020 and the Zenith-248, it is now possible to display the satellite images, along with the overlays of reflected intensities and the navigation grids and geography, with relative ease. Input to the new system is automatic; an ephemeris model is a part of the input. Parameters may be obtained from regular messages or other computer files available to the user.

The original NEPRF sunglint program was developed from code written by Irwin Ruff of NOAA-NESS during the late 1970's. His code was modified for the HP-9020 using Fortran 77, along with the graphics processing and operating system developed by Global Imaging, Inc. NEPRF never received any program documentation, physical description or any other reference to Ruff's model. The in-line comments that he placed in the code were very good, however. Using these in-line comments and some of NEPRF's own modifications, this report will describe the inner workings of the Ruff (NEPRF) model.

2. THE RUFF MODEL

At this point in the model description, we will assume that all required sunglint input parameters are available. The input/output methodologies will be described in Section 3.

To perform field calculations of any geophysical parameter, a computational grid or coordinate system must be devised. Ruff chose to use the Satellite Normal Universal Grid System (SNUGS). In this system the positive X axis is directed along the sub-satellite track; the Y axis is directed along the satellite scan lines and is positive to the right of the track. Ruff used the names "orbital longitude" and "orbital latitude," respectively, to name his coordinates. The sun's position was the "solar" orbital latitude and longitude. The origin of the system is the equator and the longitude of the ascending node. The region for calculations is defined by the sun's position in the along-track path and by the satellite's across-the-track scan. The grid is illustrated in Figure 1. An increment of one degree is used along the track, while the increment across the track is five degrees of satellite scan angle. The array size is 61 by 17. With an orbital period of 100 minutes, the time increment represented is approximately 17 minutes.

The first task in the program is to determine the approximate solar hour angle (the sun's ground-point longitude at the zenith), the relative angle between the sun and satellite, and the solar declination. Given the date and the time of the satellite's ascending node, the declination (δ) is calculated as follows. A sine curve is fitted to the time differential,

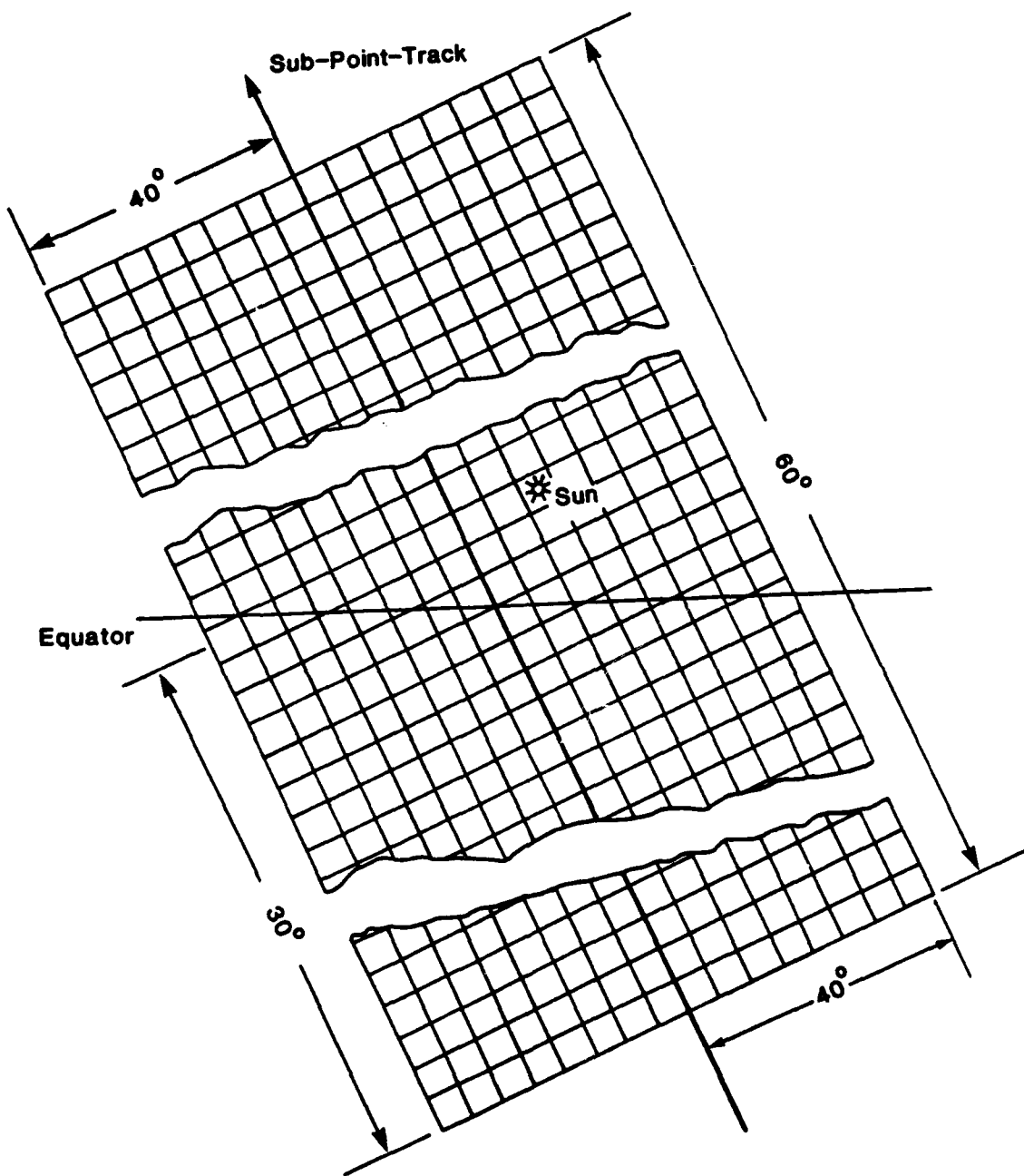


Figure 1. The computational grid.

Del(DTG), between 00Z on March 21 and the date time in hours (plus fractions) of the satellite's ascending node, using an amplitude of 23.5 degrees. The argument (ARG) of the sine curve is as follows:

$$\text{ARG} = \text{Del}(\text{DTG})/24.0/365.0*360.0*(\text{PI}/180.0), \text{ and}$$

$$\delta = 23.5*\text{sine}(\text{ARG})$$

The solar hour angle (HA) is calculated as follows:

$$\text{HA} = 540.0 - \text{Time}(\text{hours})*15.0,$$

where $\text{Time} = \text{Hours} + \text{Minutes}/60.0 + \text{Seconds}/3600.0$ and HA is limited to the range between 0 and 360 degrees. Remember that the hour angle is calculated from nadir; that is, at 1200Z, the sun is overhead at zero degrees longitude.

The solar hour angle is next converted to a local hour angle (LHA). The LHA is the angular difference between the longitude of the ascending node and the hour angle of the sun, measured eastward. All calculations, when referring to longitude, refer to this relative angle, the LHA.

Having the position of the sun in geocentric coordinates (declination and LHA), we now transform them to SNUGS coordinates to simplify all further calculations. Figure 2 illustrates the geometrical relationships.

Given the declination (δ), the LHA (λ), and the inclination of the satellites equatorial crossing (i), the solar orbital latitude (ϕ) and the solar orbital longitude (τ) may be calculated in the following manner. Following the geometry of Figure 2, given spherical triangles (Δ) EPS, ESO, ENS, ESA, use the law of sines for a spherical triangle:

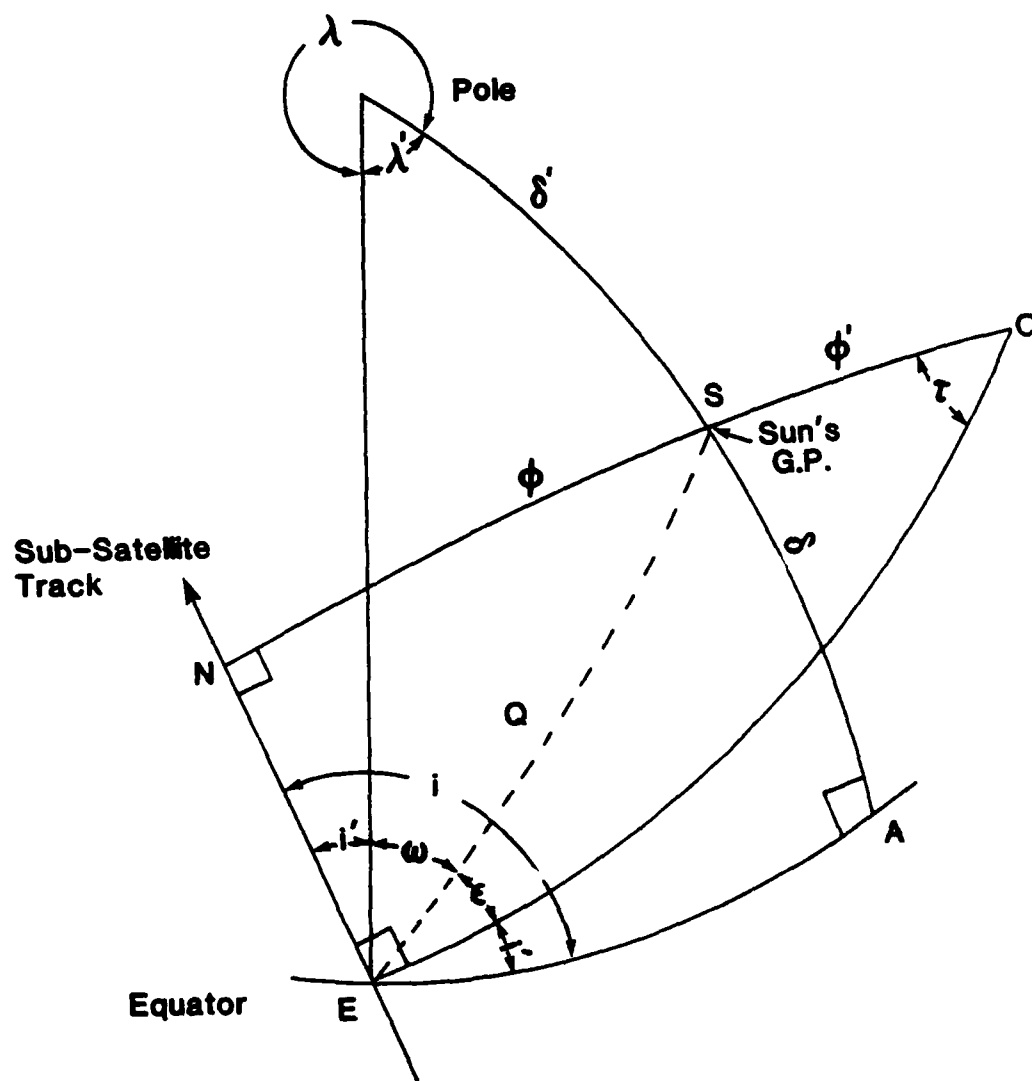


Figure 2. Geometry of the transformation from i, δ, λ to ϕ, τ .

$$\text{For } \Delta \text{ EPS, } \frac{\sin \lambda'}{\sin Q} = \frac{\sin \omega}{\sin \delta'}; \text{ For } \Delta \text{ EBA, } \frac{\sin(\epsilon + i')}{\sin \lambda'} = \frac{\sin 90^\circ}{\sin Q}$$

$$\text{For } \Delta \text{ ENS, } \frac{\sin(\omega + i')}{\sin \phi} = \frac{\sin 90^\circ}{\sin Q}; \text{ For } \Delta \text{ ESO, } \frac{\sin \epsilon}{\sin \phi} = \frac{\sin \tau}{\sin Q}.$$

Angles PAE, ENO, OEN, PEA are right angles. Further,

$$\begin{aligned} \omega + \epsilon + i' &= 90^\circ; i' = 90^\circ - i; \phi' = 90^\circ - \phi; \cos 90^\circ = 0.0; \\ \epsilon &= 90^\circ - (\omega + i'); \sin(\epsilon + i') = \cos \omega; \delta' = 90^\circ - \delta; \sin 90^\circ = 1.0; \\ \epsilon + i' &= 90^\circ - \omega; \sin \epsilon = \cos(\omega + i'); \sin \delta' = \cos \delta; \lambda' = \lambda - 360^\circ; \\ \cos i' &= \sin i; \sin \phi' = \cos \phi; \lambda' = -\lambda. \end{aligned}$$

For $\Delta \text{ ESA}$, $\cos \phi \sin \tau = \cos(\omega + i') \sin Q$. Substituting for $\sin Q$ in Δ 's EPS and EBA,

$$\frac{\sin \omega}{\sin \delta' \sin \lambda'} = \frac{\cos \omega}{\sin \delta}, \text{ or } \tan \omega = \frac{\cos \delta \sin \lambda'}{\sin \delta}.$$

Substituting for ϵ in $\Delta \text{ ESO}$ and expanding,

$$\cos \phi \sin \tau = (\cos \omega \cos i' - \sin \omega \sin i') \sin Q.$$

After substituting for $\sin(\epsilon + i')$ in $\Delta \text{ EBA}$, now substitute for $\sin Q$ in the expanded form of $\Delta \text{ ESO}$:

$$\begin{aligned} \cos \phi \sin \tau &= (\cos \omega \cos i' - \sin \omega \sin i') \sin \delta / \cos \omega, \text{ or} \\ \cos \phi \sin \tau &= (\cos i' - \tan \omega \sin i') \sin \delta. \end{aligned}$$

Substituting for $\tan \omega$, $\sin i'$ and $\cos i'$, we obtain

$$\sin \tau = (\sin i \sin \delta + \cos i \cos \delta \sin \lambda) / \cos \phi.$$

For the solar orbital latitude, expand $\Delta \text{ ENS}$:

$$\sin \phi = (\sin \omega \cos i' + \cos \omega \sin i') \sin Q.$$

Now substitute for $\sin Q$ from $\Delta \text{ EBA}$, and divide by $\cos \omega$; we then obtain,

$$\sin \phi = (\tan \omega \cos i' + \sin i') \sin \delta.$$

Now substitute for $\tan \omega$, $\cos i'$, and $\sin i'$, producing

$$\sin \phi = \cos i \sin \delta - \sin i \cos \delta \sin \lambda.$$

The next task of part 1 (see Introduction) in the initialization process is to calculate the geodetic latitude and longitude of the starting point of the grid. The starting point is 30.0 degrees of solar orbital longitude prior to the solar position at the time of the satellite's ascending node. This geodetic position is determined by an integration backwards from the equator and the longitude of the ascending node, with an azimuth equal to the satellite's inclination angle (i) + 90.0 degrees. The number of increments used in the integration is 30.0 minus the solar orbital longitude of the sun. For example, if the solar orbital longitude is 4.3, the integration is done in 25 one-degree steps, plus one step of .7 degrees. The purpose of this step is to be able to relate the calculations on the grid back to the image. This step is not necessary to the reflectance determination and was not part of the Ruff model. The geodetic latitude and longitude values are determined at the same time the SNUGS coordinates are calculated. A simple method is used, as a high degree of accuracy is not required. We are assuming that we are looking at a "snapshot" with only a limited time involved. The integration of position proceeds by providing a known latitude, longitude and azimuth, and then calculating a new latitude, longitude and azimuth, assuming a stationary earth. Figure 3 illustrates the geometry of the integration.

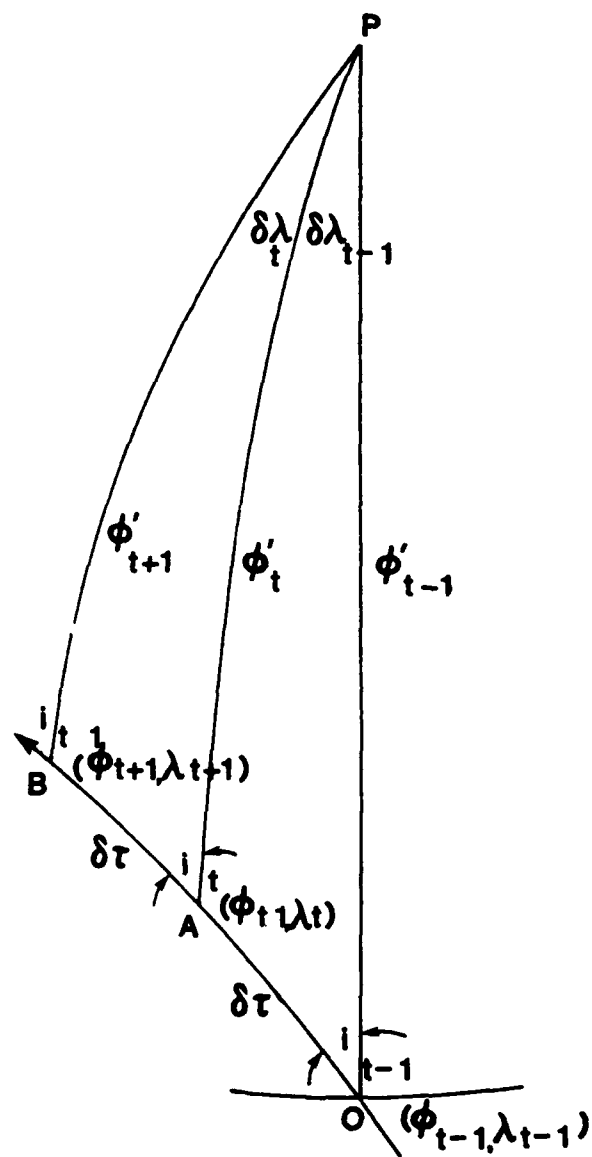


Figure 3. The integration of the satellite's position.

Using the law of cosines and sines for spherical triangles,

$$\cos \phi' = \cos \delta\tau \cos \phi' + \sin \delta\tau \sin \phi' \cos i_t, \text{ where}$$

$$\phi' = 90.0 - \phi; \cos \phi' = \sin Q; \sin \phi' = \cos \phi.$$

Substituting, we obtain

$$\sin \phi = \cos \delta\tau \sin \phi_t + \sin \delta\tau \cos \phi_t \cos i_t. \text{ Similarly,}$$

$$\cos \delta\lambda_{t+1} = \cos \phi_t \cos \phi_{t+1} = \cos \delta\tau - \sin \phi_t \sin \phi_{t+1},$$

or rearranging,

$$\cos \delta\lambda_{t+1} = (\cos \delta\tau - \sin \phi_t \sin \phi_{t+1}) / (\cos \phi_t \cos \phi_{t+1}).$$

And,

$$\sin i_{t+1} = \cos \phi_{t+1} \sin i_t / \cos \phi_t.$$

The program is now ready to make reflectance calculations on the grid, but additional geometrical relationships are needed first. As described previously, the scan line increment is five degrees, and the along-the-track increment is one degree of solar orbital longitude. We start 40.0 degrees to the left of the track and 30.0 degrees from the solar orbital longitude of the sun. The reflectances are first determined across the track, then along it. The first quantities needed at the different scan angles are the zenith angle of the satellite and its geocentric angle.

Figure 4 illustrates the geometry between the various angular measurements. The law of sines for a plane is used to relate them:

$$\sin \theta' = (r + h) \sin \eta / r, \text{ and}$$

$$\theta' = 180.0 - \theta, \sigma = \theta - \eta, \text{ where}$$

$$\theta = \text{the zenith angle of the satellite,}$$

$$\eta = \text{the scan angle of the satellite,}$$

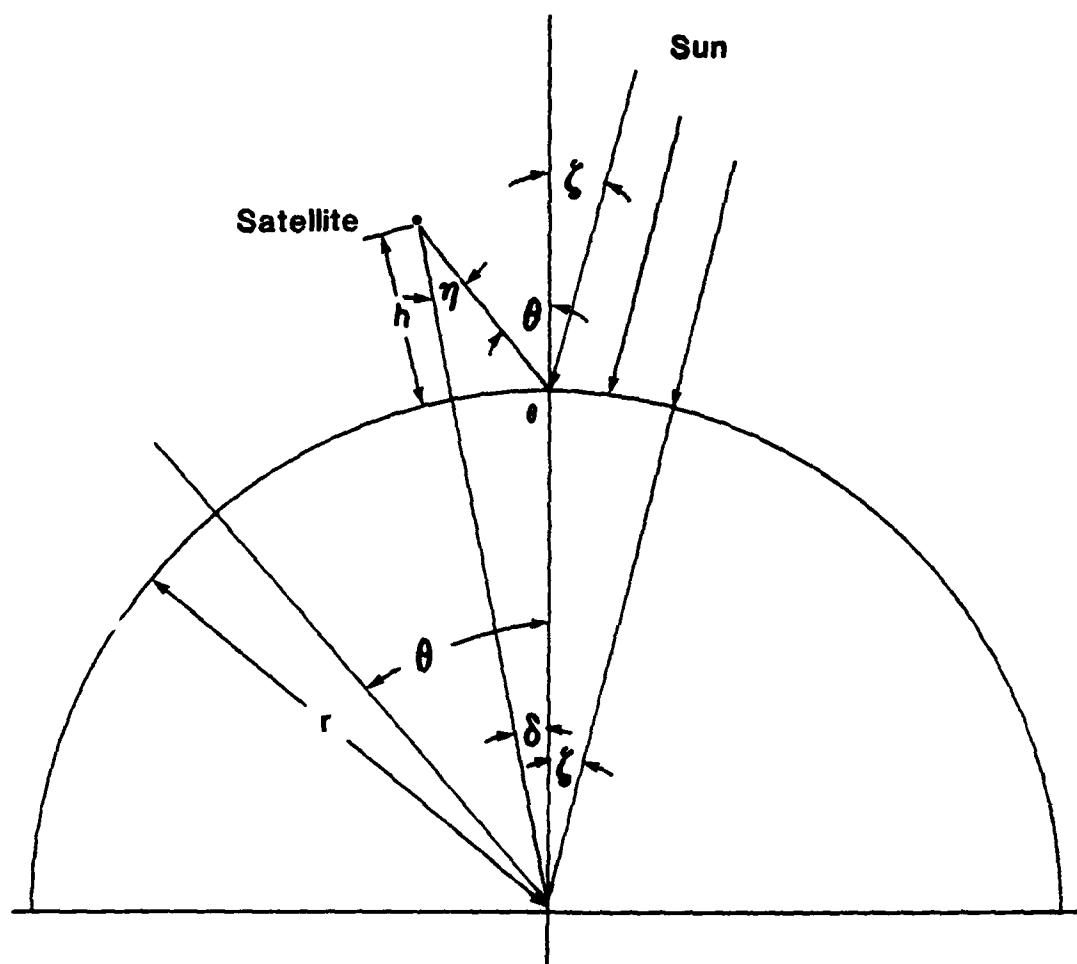


Figure 4. The angular measurements along a satellite scan line.

σ = the geocentric angle,
 h = the height of the satellite, and
 r = the radius of the earth.

The zenith angle of the sun (ζ) is shown for reference.

With the satellite geocentric angle (σ), along with the solar orbital latitude (ϕ) and longitude (τ), the zenith angle (ζ) and its azimuth (α) to the sun at the particular grid point are acquired. The geometry for the acquisition is shown in Figure 5.

From the spherical triangle VSO, the zenith angle can be written directly using the law of cosines:

$$\cos \zeta = \sin \phi \sin \sigma + \cos \phi \cos \sigma \cos \tau.$$

The azimuth requires a little more work.

As before, $\sigma = 90.0 - \sigma'$; $\alpha' = 180.0 - \alpha$; $\phi = 90.0^\circ - \phi'$.

For Δ VSE, $\cos Q = \cos \zeta \cos \sigma + \sin \zeta \sin \sigma \cos \alpha'$.

For Δ ESO, $\cos Q = \sin \phi \cos 90^\circ + \cos \phi \sin 90^\circ \cos \tau$ or,

$$\cos Q = \cos \phi \cos \tau.$$

Substituting for $\cos Q$ in Δ VSE,

$$\cos \zeta \cos \sigma + \sin \zeta \sin \sigma \cos \alpha' = \cos \zeta \cos \sigma / \sin \sigma.$$

Rearranging and dividing by $\sin \sigma$,

$$\cos \alpha' \sin \zeta = \cos \phi \cos \tau / \sin \sigma - \cos \zeta \cos \sigma / \sin \sigma.$$

Substituting for $\cos \zeta$,

$$\cos \alpha' \sin \zeta = \frac{\cos \phi \cos \tau}{\sin \sigma} - \sin \phi \cos \sigma - \frac{\cos \phi \cos^2 \sigma \cos \tau}{\sin \sigma},$$

or

$$\cos \alpha' \sin \zeta = \frac{\cos \phi \cos \tau}{\sin \sigma} (1.0 - \cos^2 \sigma) - \sin \phi \cos \sigma,$$

or

$$\cos \alpha = \sin \phi \cos \sigma - \sin \sigma \cos \phi \cos \tau.$$



Sun's Position

The geodetic coordinates are now generated. With these latitudes and longitudes the surface wind field can be extracted from the FNOC global marine data fields for use as a variable wind speed in the reflectance calculations; the graphics and the image can then be co-registered.

All the parameters necessary for the determination of the solar reflectance are now present.

There are three steps to determine the intensity of the reflected solar radiation. First, the reflectance coefficient (ρ) which is only a function of the incidence angle (reflected angle), and is an approximation of the Fresnel reflection coefficient, must be computed. The form of the Fresnel reflection coefficient is:

$$\rho = \cos \beta ((\exp(-k_1 \cos \beta)) + k_2 \cos \beta),$$

where $k_1 = 6.09$ and $k_2 = .019$.

The angle of reflection (the incidence angle), β is the angle necessary to reflect incoming solar radiation in the direction θ , with an azimuth angle α . Figures 6a and 6b illustrate the geometry of the problem. For spherical Δ TVS,

$$\cos 2\beta = \cos \theta \cos \zeta + \sin \theta \sin \zeta \cos \alpha, \text{ or}$$

$$\beta = \cos^{-1} ((0.5*(1.0 + \cos 2\beta))^{1/2}).$$

The second step is determining the probability that the slopes and directions of the wave facets of the sea surface will reflect the incoming solar radiation at the correct angles to give perfect reflection. This probability (P) is a function of the variance of the wind speed and the slopes of the sea surface:

$$P = (\exp(-\tan^2 \theta_1^N / \sigma^2) - \exp(-\tan^2 \theta_\mu^N / \sigma^2)) (\alpha_1^N - \alpha_\mu^N) / 2\pi, \text{ where}$$

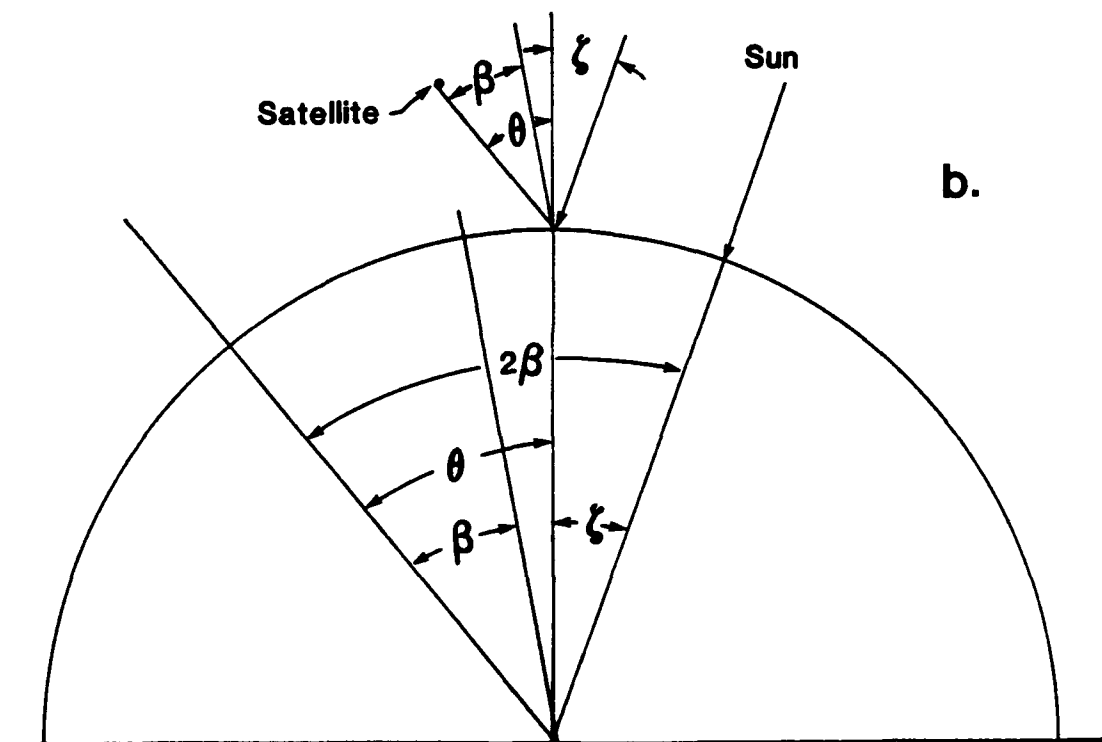
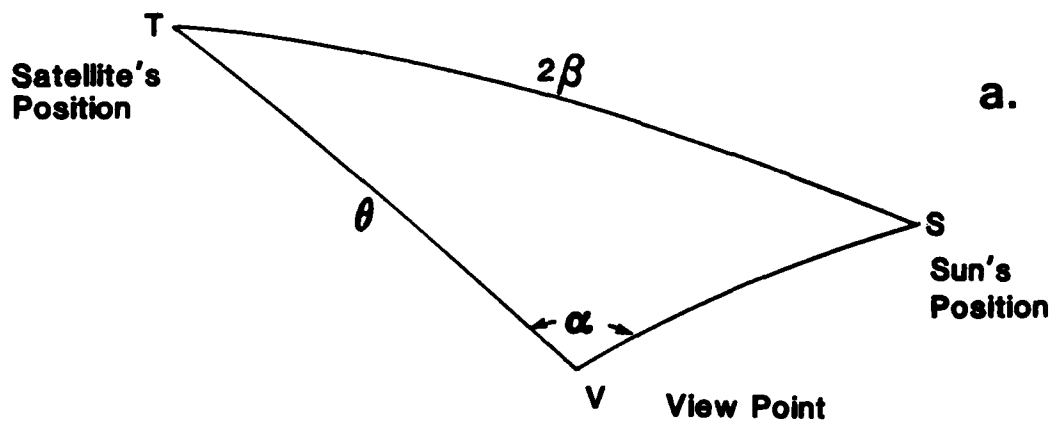


Figure 6. The relationship between satellite zenith angle, solar zenith angle, azimuth and incidence angle.
(a) Plane view, (b) Section view.

subscript l and μ refer to lower and upper limits, respectively, of the integration. Superscript N refers to the fact that we now are measuring angles relative to a normal to the sea surface.

This probability is an integration over a finite solid angle $\delta\omega$. The limits are between θ_l^N, α_l^N and $\theta_\mu^N, \alpha_\mu^N$ centered on θ_C^N, α_C^N . $\tan \theta_C^N$ is the slope of the sea surface (wave facet) or the tangent of the zenith angle of the surface normal required to reflect at a given θ and azimuth α . $\tan \theta_C^N$ is the ratio of the sine of the vector difference of θ, ζ and the cosine of the vector sum of θ, ζ :

$$\tan \theta_C^N = \frac{\sin \theta_C^N}{\cos \theta_C^N}$$

Using the law of cosines for a plane, the sine and cosine of the slope normal are calculated in the following manner:

$$\sin^2 \theta_C^N = \sin^2 \theta + \sin^2 \zeta + 2 \sin \theta \sin \zeta \cos \alpha_C^N, \text{ and}$$

$$\cos \theta_C^N = \cos \theta + \cos \zeta.$$

The corresponding azimuth, α_C^N is calculated as follows. Again, using the law of cosines for a plane,

$$\sin^2 \theta = \sin^2 \theta_C^N + \sin^2 \zeta - 2 \sin \theta_C^N \sin \zeta \cos \alpha_C^N.$$

Substituting for $\sin^2 \theta$,

$$\begin{aligned} \sin^2 \theta_C^N = \sin^2 \theta_C^N + \sin^2 \zeta - 2 \sin \theta_C^N \sin \zeta \cos \alpha_C^N \\ + \sin^2 \zeta + 2 \sin \theta \sin \zeta \cos \alpha_C^N, \text{ or} \end{aligned}$$

$$2 \sin^2 \zeta - 2 \sin \theta_C^N \sin \zeta \cos \alpha_C^N + 2 \sin \theta \sin \zeta \cos \alpha_C^N = 0.0.$$

Simplifying and rearranging,

$$\cos \alpha_C^N = (\sin \zeta + \sin \theta \cos \alpha) / \sin \theta_C^N.$$

Figure 7a and 7b show the geometrical relationships.

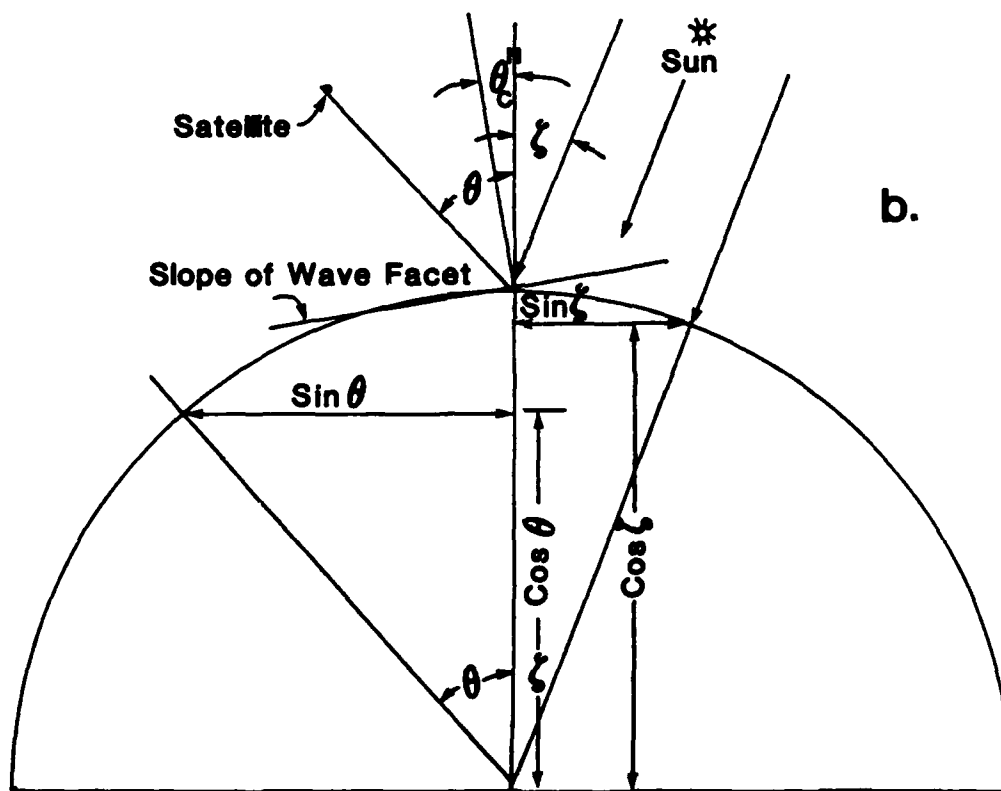
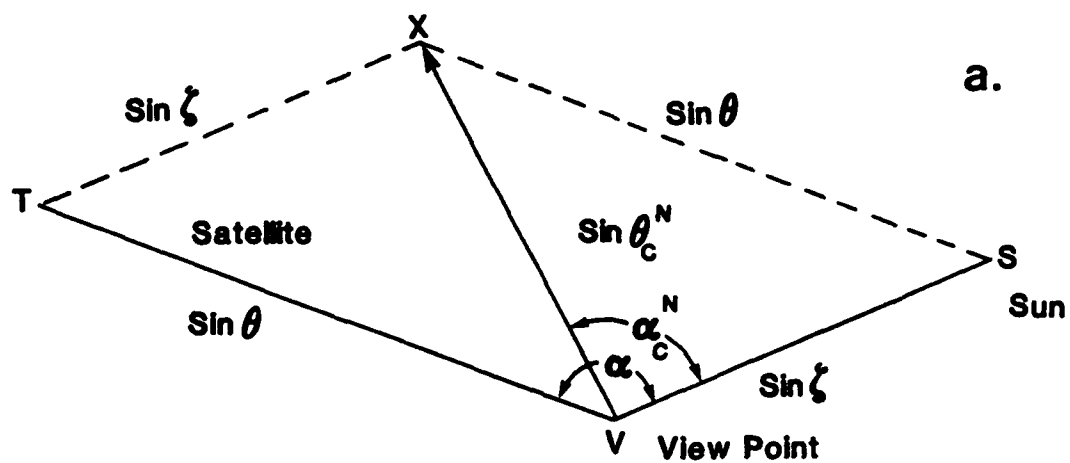


Figure 7. The wave facets slope and azimuth relationships.
(a) Plane view, (b) Section view.

For the solid angle integration and the probability, the limits are .01 radians on either side of the center angles θ_C^N , α_C^N which are the exact slope and azimuth needed to reflect the ray. Thus,

$$\theta_\mu^N = \theta_C^N + \delta\theta, \alpha_C^N = \alpha_C^N + \delta\alpha, \text{ and}$$

$$\theta_l^N = \theta_C^N - \delta\theta, \alpha_l^N = \alpha_C^N - \delta\alpha,$$

The wind dependence is introduced into the probability as the variance of a circular Gaussian distribution of the normals of the slopes of the sea surface (wave facets). This dependence is linear, and it is after Cox and Munk (1954):

$$\sigma^2 = A*w + B, \text{ where}$$

$A = .0051$, $B = .003$, w is the wind speed in m/sec, and σ^2 is the variance.

The third step for reflectance is to determine the elemental solid angle over the same limits that were used in the probability statement. The integration is in steps of .001 radians for both θ_1^N and α_1^N . The increment of solid angle is integrated on a Lambert azimuthal equal area projection of the upper hemisphere, centered on the solar direction $\theta = \zeta$; $\alpha = 0$.

Figures 8a, 8b, and 8c show the relationships between incidence angle and azimuth, as referred to the sphere and the projection. Using the law of cosines for a sphere, we may determine the incidence angle β_1 for Δ TVS in the solid angle integration (Fig. 8a):

$$\cos\beta_1 = \cos \zeta \cos \theta_1^N + \sin \zeta \sin \theta_1^N \cos \alpha_1^N .$$

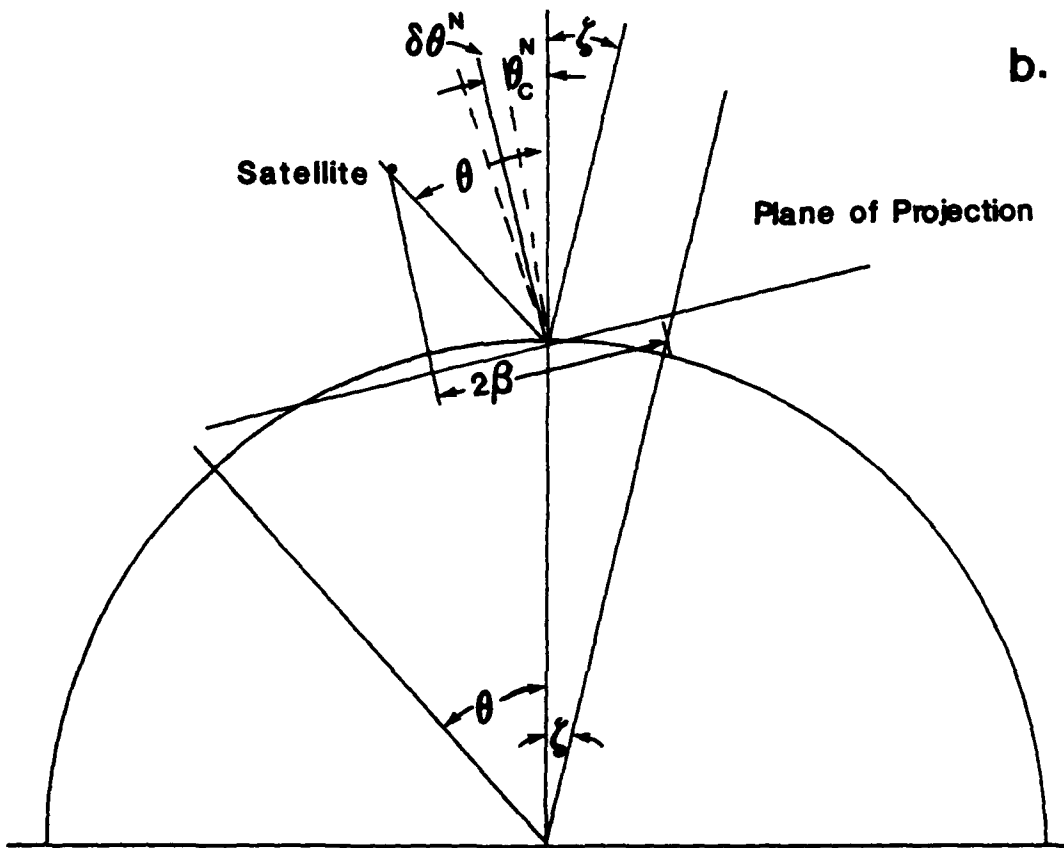
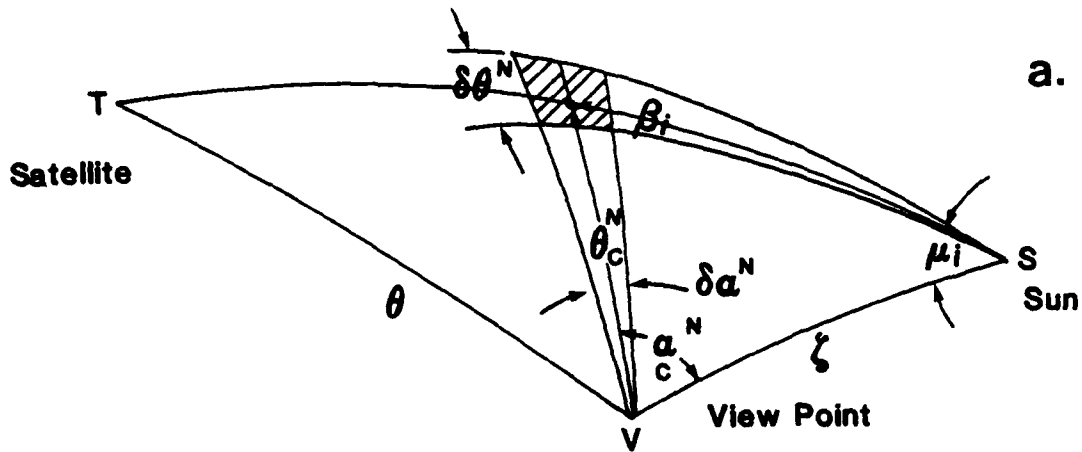


Figure 8. The geometrical relationships for solid angle calculations between the sphere and the projection. (a) Plane view of sphere, (b) Section view, (c) The projection.

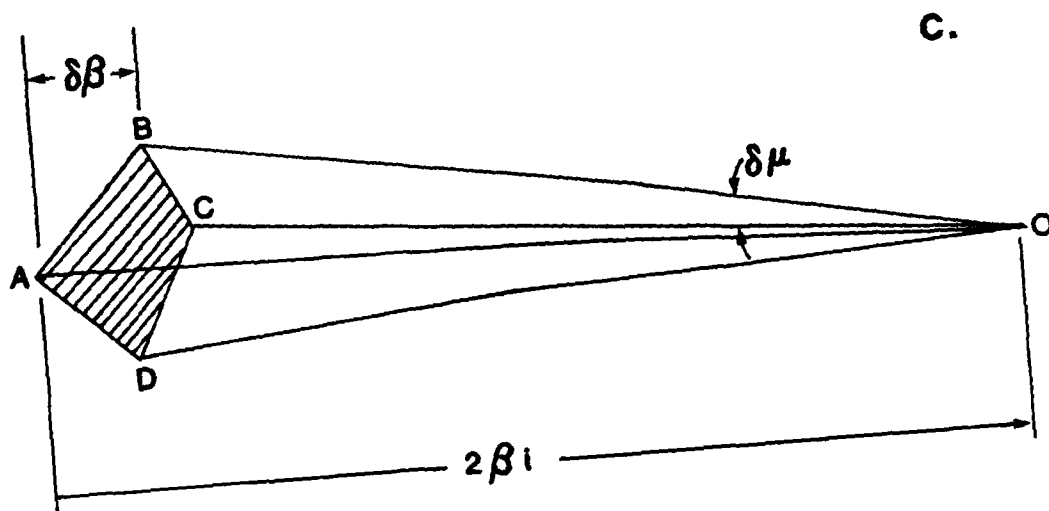


Figure 8(c) The projection.

The azimuth from the sun projected point (μ) may be determined as follows, again using the law of cosines:

$$\cos \theta_1^N = \cos \mu_1 \sin \zeta \sin \beta_1 + \cos \beta_1 \cos \zeta .$$

Substituting for $\cos \beta_1$, we obtain

$$\begin{aligned} \cos \theta_1^N &= \cos \mu_1 \sin \zeta \sin \beta_1 \\ &\quad + \cos \zeta (\cos \zeta \cos \theta_1^N + \sin \zeta \sin \theta_1^N \cos \alpha_1^N). \end{aligned}$$

Simplifying and dividing by $\sin \zeta$, we produce

$$\begin{aligned} \cos \mu_1 \sin \beta_1 &= \frac{\cos \theta_1^N}{\sin \zeta} - \cos \zeta \left[\cos \alpha_1^N \sin \theta_1^N \right. \\ &\quad \left. + \cos \theta_1^N \frac{\cos \zeta}{\sin \zeta} \right] \end{aligned}$$

$$\cos \mu_1 \sin \beta_1 = \cos \zeta \sin \theta_1^N \cos \alpha_1^N - \frac{\cos \theta_1^N}{\sin \zeta} (\cos^2 \zeta - 1.0),$$

or

$$\cos \mu_1 = (\cos \theta_1^N \sin \zeta - \cos \zeta \sin \theta_1^N \cos \alpha_1^N) / \sin \beta_1 .$$

The solid angle is calculated on the projection by determining the area for each sector and combining them in the correct sequence. $\delta\omega$ is the area of sectors ABO + ADO minus the area of sectors DCO + BCO. The area of each sector is an integration over β and μ where

$$\begin{array}{rcl} \text{Areas} & \begin{array}{l} \text{ABO} \quad Y \\ \text{ADO} \quad \neg \\ \text{DCO} = \int \quad .5*(2\beta)\delta\mu_1 , \\ \text{BCO} \quad \neg \\ \quad \quad x \end{array} & \end{array}$$

with the index limits, x and y , a function of each sector.

The integration of solid angle begins with the smallest values of α_1^N and θ_1^N . α_1^N is increased first, followed by θ_1^N . α_1^N is now decreased, followed by θ_1^N . In this way the proper signs are maintained during the calculations.

Throughout the program there are numerous instances where anomalous conditions exist. For example, the zenith angle θ^N may be equal to zero, or the azimuth angle α^N may be equal to zero. In addition, computations must be made using double precision.

Once the integration of the solid angle increment is complete, the intensity of the reflected energy for any point on the computational grid is calculated as the product of the probability (P) and the reflectance coefficient (ρ), normalized per unit solid angle ($\delta\omega$). Figure 9 shows the reflection coefficient as a function of incidence angle, and Figure 10 depicts the probability as a function of wave slope for different wind speeds. The reflectance coefficient and probability are the two terms in the reflectance (R) calculations, along with the normalization factor (the solid angle, delta omega):

$$R = \rho P / \delta\omega.$$

The reflectance is converted to percent, and along with the grid point's latitude and longitude, is written as an ASCII record. Each point is considered as a randomly spaced data point as required in the Global Imaging processing system. All data with reflectance less than 0.1% are set to zero. After the reflectances have been determined over the computational grid,

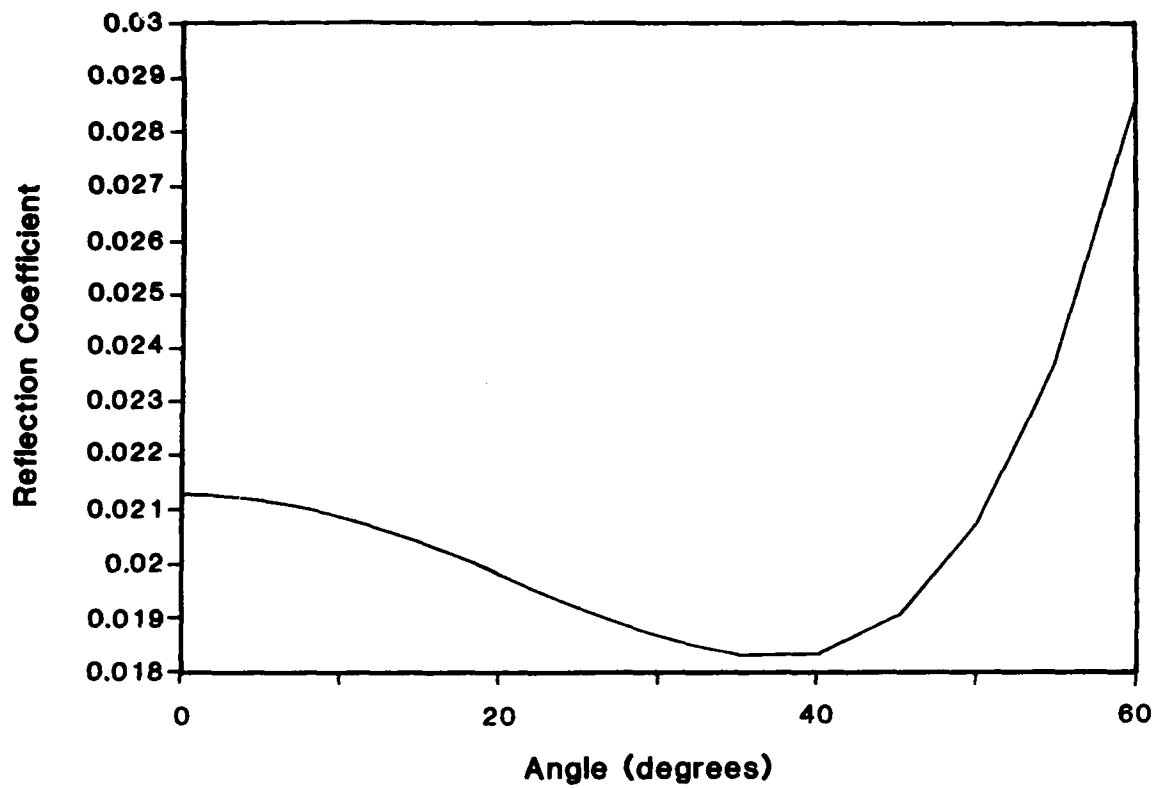


Figure 9. Reflectance coefficient as a function of incidence angle.

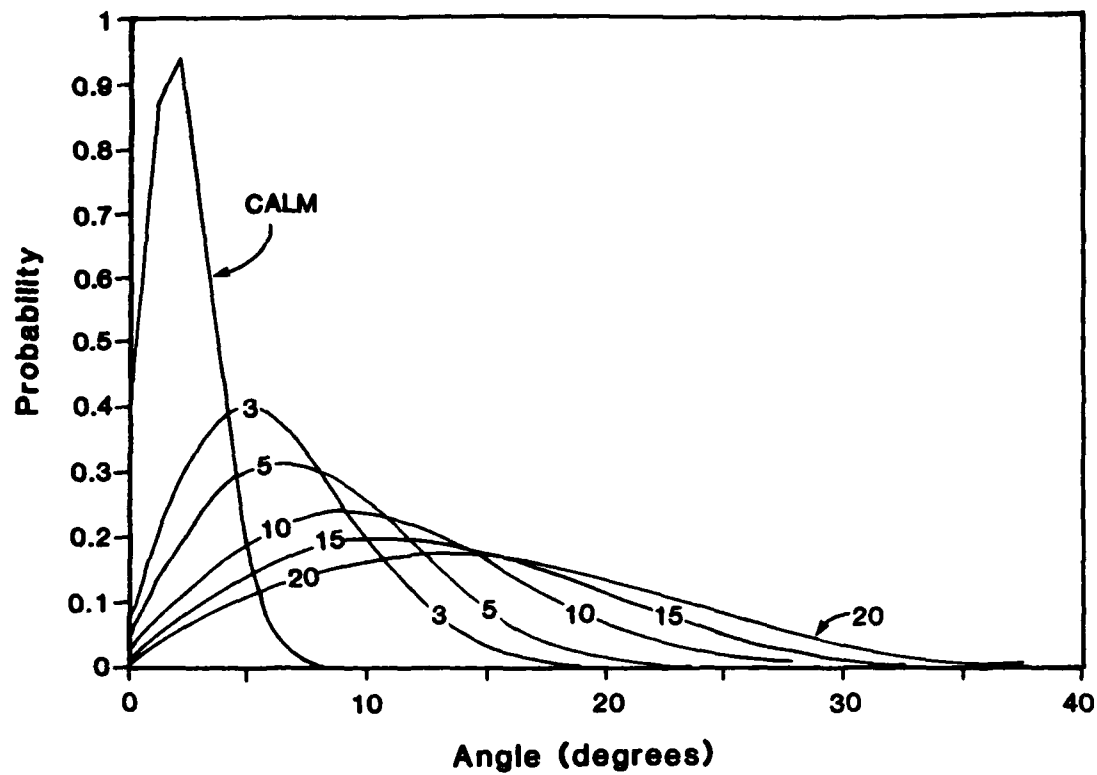


Figure 10. Probability as a function of wave slope (degrees) and wind speed (m/s).

the system will perform interpolations to place the reflectances to a regular grid, corresponding to the same lines and elements as the requested image.

3. THE NEPRF SYSTEM

The capability to examine images with the reflectance patterns superimposed is now part of the procedures of the HP 9020 graphics and image processing system. The only input to the system is a filename. In this file are all the parameters needed to display the isopleths of reflectance over this image and they are automatically obtained. A message containing the orbital elements is read and decoded. This message gives the epoch date, time and the inclination angle, the argument of perigee, the mean anomaly, the eccentricity, the right-ascension of the ascending node and the mean motion. The mean motion is converted to the semi-major axis. These parameters are fed to an ephemeris program that gives the latitude, longitude and height of the satellite as a function of time. The time interval is set to one minute. The ephemeris file is scanned, looking for the ascending nodes. The closest one to the time of the image is chosen. The longitude, time and height of the satellite are extracted on either side of the equator and then are linearly interpolated to the equatorial crossing of the satellite. There is one more parameter needed to run the sunlint code. This is the wind speed. There is an option to give the program one constant value of wind speed (as the old system did), or to extract the wind components u and v from the FNOC marine wind field analysis at the time closest to the image.

The system now has all parameters needed to make the sunglint reflectance calculations, navigate them and display the results. This information is passed to the program via a data file. After the sunglint module has been run, the desired image is displayed on the CRT, followed by an overlay of the contours of reflectance. The image/overlay display package is part of the Global Imaging, Inc. processing and display package written for the HP 9020. Their system manages all the files that have been transferred between the SPCU, the Zenith 248 and the HP 9020. At this time only data from NOAA AVHRR imagery can be processed on the Global Imaging System. However, DMSP images will soon be added to the system's capability. A difficulty arises from the navigation of the DMSP satellites. Another method of running the system, especially for DMSP images from the Fleet Numerical Oceanography Center (FNOC), is to perform the sunglint calculations for one 12-hour period (seven orbits). In this way, data from FNOC's specialized data base could be extracted and navigated for very large sectors of the globe. The corresponding reflectance overlays would be present for display. Figure 11 is an example of the display of the reflectance calculation for constant wind speeds (calm and 5 m/sec).

At this time, we have a system which will automatically display an image along with the corresponding sunglint patterns. The wind field may be variable or constant, at the user's discretion. In areas of the world where the sunglint is an observable phenomena, meteorologists under certain conditions

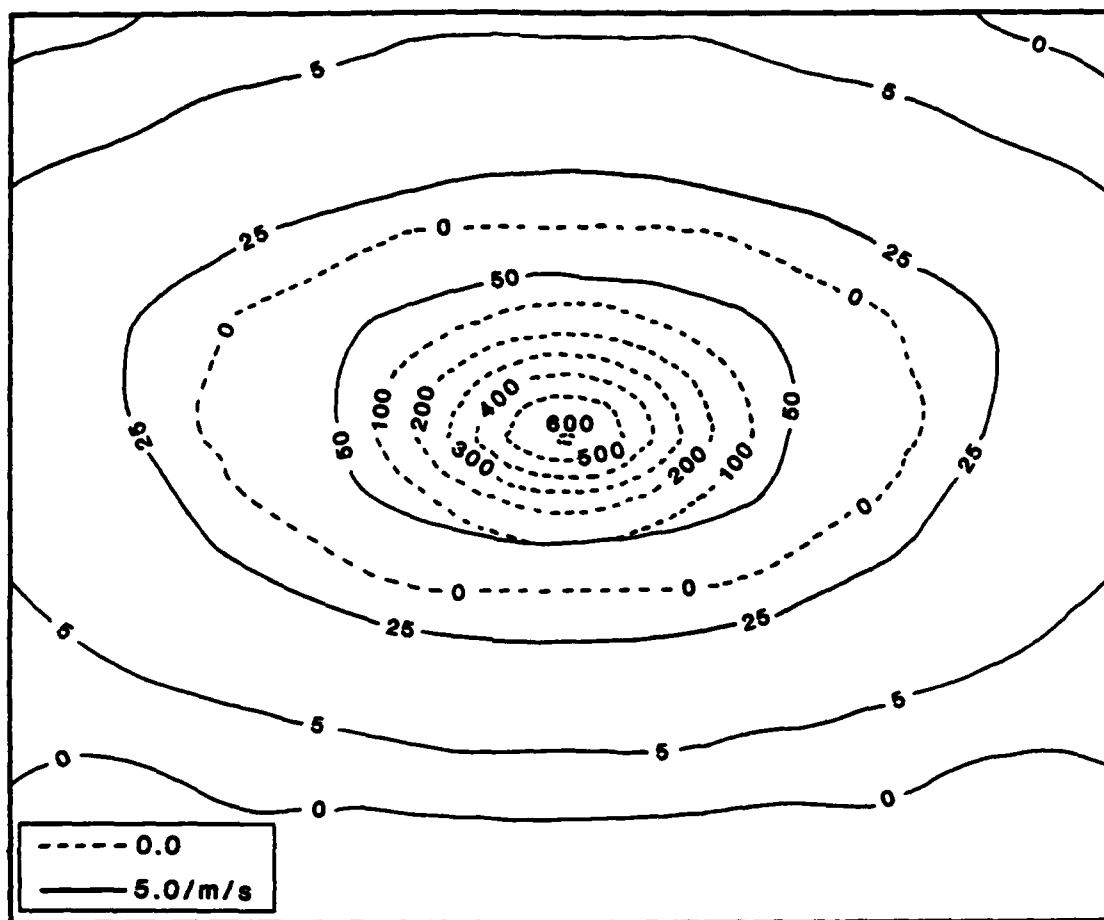


Figure 11. An example of the sunlint pattern for a constant field of wind speed (0.0 and 5.0 m/sec).

can determine in a qualitative sense the correctness of an analyzed wind field. Further work is needed to determine if a quantitative value of the wind may be determined from observations of sunglint.

Appendix A is a listing of the sunglint program, with associated subroutines; Appendix B details the running instructions for the sunglint program. Both appendices have been developed as HP 9020 using Global Imaging, Inc. support software.

REFERENCES

- Cox, C., and W. Munk, 1954: Statistics of the sea surface derived from sun glitter. J. Marine Res., 13, 198-227.
- Tsui, T., and R. W. Fett, 1980: Determination of the Primary Spectral Point in the Sunglint Pattern of a Polar Orbiting Satellite Image. Naval Environmental Prediction Research Facility, Technical Report TR 80-03, Monterey, CA 93943-5006, 26 pp.

APPENDIX A
SUNGLINT PROGRAM LISTING

```

CCC$option trace on
      PROGRAM SUNGLNT
C   PROGRAMMER  IRWIN RUFF
C   PROGRAMMER  T.L. TSUI
C
      IMPLICIT REAL *8(A-H), REAL *8(O-Z)
C
      COMMON /ABC/ IDATE(3), IPLOT, NUM1, NUM2, NUM3, NUM4, NUM5
      REAL*4 PLAT, PLON, PLONT, PLATMAX, PLONW, PLATMIN, PLONE
      REAL*4 W(73,144,2), WIND
      INTEGER TN, IQNX(3), LH
      REAL *8 LN
      REAL *4 DUMMY
      CHARACTER CDATE*12, CIQNX*12, CTN*4, CLH*12
      EQUIVALENCE (IDATE, CDATE), (IQNX, CIQNX), (TN, CTN)
      PI=DACOS(-1.000)
C
C   INITIALIZE GAE PARAMETER BLOCK "sunglint"
C
      CALL initblk("sunglint")
C
C*****
C INPUT DATA FROM GAE PARAMETER BLOCK
      call intgae(6h/YEAR/, IYR)
      call intgae(7h/MONTH/, IMO)
      call intgae(5h/DAY/, IDAY)
      call intgae(6h/HOUR/, IHR)
      call intgae(5h/MIN/, IMIN)
      call realgae(8h/EQLONG/, DUMMY)
      LN=DBLE(DUMMY)
      call strgae(7h/EQHEM/, CLH)
      read(clh, '(a1)') lh
      write(*, '( "EQHEM= ", a2, )') LH
      call realgae(9h/SATINCL/, DUMMY)
      SIA=DBLE(DUMMY)
      call realgae(8h/SATALT/, DUMMY)
      HT=DBLE(DUMMY)
      call fileop(6h/WIND/, 5)      ! open input file on unit 5
      REWIND 5
      NW=1
      IPLOT=2
      write(cdate, '(3i2.2)') iyr, imo, iday
      write(CTN, '(2i2.2)') ihr, imin
C
20  READ(5, 5000, END=9999) iend, jend
      READ(5, 5010, END=9999) (((W(I, J, k), J=1, jend), I=1, iend), k=1, 2)
C
C   COLUMN  FMT  DESCRIPTION
C   IDATE    1 - 6  A6  YYMMDD (DATE-TIME-GROUP)
C   SIA       7 - 10 F4.1 SATELLITE INCLINATION ANGLE
C              (E.G. DMSP F2: SIA=98.7)
C   HT       11 - 14 F4.0 AVERAGE SATELLITE HEIGHT IN KM
C              (E.G. DMSP F2: HT=833)
C   TN       16 - 19 A4  TIME OF LAST EQUATOR CROSSING IN ZULU
C              (TN=HHMM, HOURS AND MINUTES OF ZULU TIME)

```

```

C LN      20 - 24 F5.1  LONGITUDE OF EQUATOR CROSSING
C LH      25 A1      LN LOCATION INDICATOR, E - EASTERN HEMI.
C                               W - WESTERN HEMI.
C IPLOT   27 I1      >1 CONTOURS AND RADIANCE VALUES PLOTTED
C                               =1 CONTOURS PLOTTED ONLY
C                               <1 RADIANCE VALUES PLOTTED ONLY
C NW      30 I1      NUMBER OF CHARTS DESIRED, MAX. 6
C W       31 - 60 6F5.1 W(I) IS THE WIND SPEED (M/SEC) ASSOCIATED
C                               WITH EACH CHART.
C *****
C
C
C
C RADIUS OF EARTH
C   R=6378.16D0
C ORBITAL INCLINATION (EQUAL TO OR LESS THAN 90 DEGREES)
C   ORBINC = 180.D0-SIA
C ETA IS NADIR, DELETA IS INTERVAL IT IS READ
C   DELETA= 5.0D0
C STARTING POINT(LATITUDE) ON MAP ENTERED MANUALLY
C   NCNT=0
C STARTING POINT
C   TAU1=0.0D0
C INTERVAL REQUIRED
C   DELTAU=1.0D0
C
C
C
C FIND THE SOLAR DECLINATION
C
C   READ (CDATE,7001) IQNX(1)
C   WRITE (CIQNX,7002) IQNX(1)
C   WRITE (CDATE,7003) CDATE,TN
C   IQNX(1)=IDIFDTG(CIQNX,CDATE)
C   SOL= FLOAT(IQNX(1))/24.D0/365.D0*360.D0*PI/180.D0
C   SOLDEC=23.5D0*DSIN(SOL)
C
C FIND THE LONGITUDE OF EQUATOR CROSSING
C   EQCX=LN
C   IF(LH.EQ.1HW) EQCX=360.D0-EQCX
C
C FIND THE SOLAR LONGITUDE
C
C   READ (CTN,7004) SOL,AMDA
C   SOL=SOL+AMDA/60.D0
C   SOL=DMOD(540.D0-SOL*15.D0,360.D0)
C   AMDA=EQCX-SOL+360.D0
C   AMDA=DMOD(AMDA,360.D0)
C
C   WRITE (6,6000) IDATE
C   WRITE (6,6001) HT,SIA
C   IQNX(1)=1HE
C   IF(SOL.LT.180.) GO TO 12
C   SOL=360.D0-SOL
C   IQNX(1)=1HW

```

```

12  WRITE (6,6002) SOLDEC,SOL,IGNX(1)
    WRITE (6,6003) TN,LN,LH
    WRITE (6,6004) AMDA
C
C
    SNINC=SIN(ORBINC*PI/180.DO)
    CSINC=COS(ORBINC*PI/180.DO)
    SNDEC=SIN(SOLDEC*PI/180.DO)
    CSDEC=COS(SOLDEC*PI/180.DO)
    SNLMDA=SIN(AMDA*PI/180.DO)
    CSLMDA=COS(AMDA*PI/180.DO)
CCC  WRITE (6,6006) SNINC,CSINC,SNDEC,CSDEC,SNLMDA,CSLMDA
    CALL SLORBC(SNINC,CSINC,SNDEC,CSDEC,SNLMDA,CSLMDA,SNSLOT,CSSLLOT,
1SLOBLG)
CCC  WRITE (6,6006) SNSLOT,CSSLLOT,SLOBLG
C
C
C
    DO 11 K=1,NW
C
C
    WRITE (6,6005) K,WIND
    NUM1=0
    NUM2=0
    NUM3=0
    NUM4=0
    NUM5=0
C
C
    INITIALIZE PARAMETERS FOR SUBROUTINE SUBLTLN - - SUBTRACK LAT LO
C
    PHI = 0.0D0
    AZ = SIA - 90.0D0
    ALN = EQCX
    SNDTAU = DSIN(DELTAU*(dacos(-1.0D0)/180.0D0))
    CSDTAU = DCOS(DELTAU*(dacos(-1.0D0)/180.0D0))
C
C
    IF(NCNT.EQ.0)THEN
C
    TAU1=SLOBLG-30.0D0
    IF(TAU1.LT.0.0D0) AZ=AZ+180.0D0
    BCKFRCT=TAU1-INT(TAU1)
C
C
    STEP A FRACTION OF AN INCREMENT ALONG SATELLITE TRACK
C
    IF(ABS(BCKFRCT) .GT. 0.0) THEN
        SNDTAU = DSIN(BCKFRCT*(dacos(-1.0D0)/180.0D0))
        CSDTAU = DCOS(BCKFRCT*(dacos(-1.0D0)/180.0D0))
        CALL SUBLTLN(SNDTAU,CSDTAU,PHI,ALN,AZ,PHI2,ALN2,AZ2)
        SNDTAU = DSIN(DELTAU*(dacos(-1.0D0)/180.0D0))
        CSDTAU = DCOS(DELTAU*(dacos(-1.0D0)/180.0D0))
    ENDIF
    DO 100 I=0,INT(ABS(TAU1))
        CALL SUBLTLN(SNDTAU,CSDTAU,PHI,ALN,AZ,PHI2,ALN2,AZ2)
        PHI=PHI2
        ALN=ALN2

```

NG

```

        AZ=AZ2
100    CONTINUE
        ENDIF
        IF(TAU1.LT.0.0D0) AZ=AZ - 180.0D0
        IFIN=60.0D0/DELTAU+1.0D0
        TAU=TAUI
C
C      INITIALIZE IMAGE WINDOW TO MIN. CALC OF SUNGLINT AND RESTRICT CA
LC
C      TO AREA OF INTEREST.
        call realgae(9h/platmin/,platmin)
        call realgae(9h/platmax/,platmax)
        call realgae(7h/plone/,plone)
        call realgae(7h/plonw/,plonw)
        if(plone.lt.plonw) plone=plone+360.0
C
        DO 13 I=1,IFIN
C      ORGITAL LONG DIF BETWEEN SUN AND SATELLITE
        CSDOBG=DCOS((SLOBLG-TAU)*PI/180.D0)
CCC      write(6,'(3(f8.2,2x))')SNDTAU,CSDTAU,PHI
        CALL SUBLTLN(SNDTAU,CSDTAU,PHI,ALN,AZ,PHI2,ALN2,AZ2)
        ETA=-40.0D0
        JFIN=80.0D0/DELETA+1.0D0
C
C      INITIALIZE CALL ELOC
        JE=1
C
        DO 15 J=1,JFIN
        GETA=-ETA
        ETARD=ETA*PI/180.
        SNTHET=(R+HT)*DSIN(ETARD)/R
        THETRD=DASIN(SNTHET)
C      SATELLITE ZENITH ANGLE AT VP
        THETA=DABS(THETRD*180.D0/PI)
C      GEOCENTRIC ANGLE BETWEEN SATELLITE AND VP
        SIG=THETRD-ETARD
        SNSIG=DSIN(SIG)
        CSSIG=DCOS(SIG)
CCC      WRITE (6,6006) SNSLOT,CSSLT,CSDOBG,SNSIG,CSSIG,ZETA
        CALL GEOMET(SNSLOT,CSSLT,CSDOBG,SNSIG,CSSIG,ZETA,ALPHA)
CCC      WRITE (6,6006) ALPHA,THETA,WIND,RI
C
C      FIND LAT AND LONG FOR POINT BEING CONSIDERED
C
        CALL ELOC(JE,REAL(ETA),REAL(HT),REAL(PHI),REAL(PHI2),
+          REAL(ALN),REAL(ALN2),PLAT,PLON,IERR)      ! EARTH LOCA
TE
C
        if(plon.lt.plonw) then
            plont=plon+360.0
        else
            plont=plon
        endif
        IF(PLATMAX.GE.PLAT .AND. PLATMIN.LE.PLAT) THEN
            IF(PLONW.LE.PLONT .AND. PLONE.GE.PLONT) THEN
                CALL INTERP(W,PLAT,PLONT,PLATMAX,PLONW,WIND)
                CALL OCGLNT(ZETA,THETA,ALPHA,DFLOAT(WIND),RI)
            A-4

```

```

      RA=100.0D0*RI
C
C
CCC      WRITE (6,6006) GETA,TAU,RI,TAUI,ORBINC,SNINC
      IF(RI.GE.(.00100)) THEN
C=====C
      CALL PLACE(REAL(GETA),REAL(TAU),REAL(RI),1,REAL(TAUI),
+      REAL(ORBINC),REAL(SNINC),REAL(WIND),PLAT,PLON )
C=====C
      ENDIF
      ENDIF
C      ask jim about the placement of je=je+1 should it be just befor
C
C      call place inside if stmts or here???
      JE=JE+1
15      ETA=ETA+DELETA
      PHI=PHI2
      AZ = AZ2
      ALN = ALN2
13      TAU=TAU+DELTAU
C
CCC      WRITE (6,6006) GETA,TAU,RI,TAUI,ORBINC,SNINC
C
      CALL PLACE(REAL(GETA),REAL(TAU),REAL(RI),2,REAL(TAUI),
+      REAL(ORBINC),REAL(SNINC),REAL(WIND) )
11      CONTINUE
C
C
C
5000     FORMAT(2i3)
5010     FORMAT(61(13F6.1/))
C
6000     FORMAT(///' DATA = ',3A6/)
6001     FORMAT(' SATELLITE HEIGHT =',F6.0,'(KM),  SATELLITE'
1' INCLINATION ANGLE =',F6.1)
6002     FORMAT(' SOLAR POSITION:  DECLINATION =',F6.1,',  SOLAR ',
1' LONGITUDE =',F6.1,A1)
6003     FORMAT(' EQUATOR CROSSING:  TIME = ',A4,'Z,  LONGITUDE =',
1F6.1,A1)
6004     FORMAT(' SOLAR LONGITUDE RELATIVE TO LONGITUDE OF EQUATOR ',
1' CROSSING =',F6.1,' EASTWARD')
6005     FORMAT(' CHART NO.',I2,',  WIND SPEED =',F6.1,'(M/S)')
6006     FORMAT(' CHECK ',6F15.6)
C
7001     FORMAT(I2)
7002     FORMAT(I2,'032100')
7003     FORMAT(A6,A4)
7004     FORMAT(2F2.0)
C
9999     END

```

```

      SUBROUTINE OCGLNT(ZETA,THETA,GALPHA,W,RI)
      IMPLICIT REAL *8(A-H), REAL *8(O-Z)
      SAVE
C   CALCULATES ANGULAR REFLECTION FROM OCEAN SURFACE
C   INPUT (ALL IN DEGREES)-ZETA=SOLAR ZENITH ANGLE,
C   THETA=ZENITH ANGLE OF REFLECTED RAY, GALPHA=
C   AZIMUTH ANGLE OF REFLECTED RAY RELATIVE TO SUN(I.E.,
C   AZIMUTH OF SUN=0). GALPHA MAY BE EXPRESSED AS
C   0-360,-180-+180, ETC. **IF THETA=0, GALPHA MUST BE ZERO.**
C   W=WIND SPEED IN M/SEC
C   OUTPUT- RI=REFLECTED INTENSITY (/STERADIAN),
C   RELATIVE TO INCOMING SOLAR FLUX OF 1
C
      PI = DACOS(-1.000)
      HPI = PI/2.000
      RPD = PI/180.000
C
C   ANGINC=ANGULAR INCREMENT, IN RADIANS, USED IN CALCULATING
C   SLOPE PROBABILITY AND SOLID ANGLE OF REFLECTED BEAM
C
      ANGINC = 0.0100
C
C   STPNMB=NUMBER OF STEPS PER SIDE USED IN INTEGRATION
C   FOR SOLID ANGLE
C
      STPNMB = 20.000
C
C   IF SUN IS BELOW HORIZON, SKIP COMPUTATION
      IF(ZETA.LT.90.0)GO TO 8
      RI = 0.000
      GO TO 5
      8 SNZET = DSIN(ZETA*RPD)
      CSZET = DCOS(ZETA*RPD)
      SNSQZT=SNZET*SNZET
      SNTHET = DSIN(THETA*RPD)
      CSTHET = DACOS(THETA*RPD)
      SNSQTH=SNTHET*SNTHET
C
C   NEXT 2 STEPS CONVERT GALPHA TO ALPHA, A POSITIVE ANGLE
C   BETWEEN 0 AND 180 DEGREES. (PERMISSABLE SINCE REFLECTION
C   PATTERN IS SYMMETRICAL ABOUT 0-180 LINE).
      ALPHA=DABS(GALPHA)
C
      IF(ALPHA.GT.180.0) ALPHA = 360.000-ALPHA
      CSALF = DCOS(ALPHA *RPD)
C   SIGSQ=VARIANCE OF THE CIRCULAR GAUSSIAN DISTRIBUTION OF
C   THE NORMALS OF THE SLOPES OF THE OCEAN SURFACE
      SIGSQ = 0.0051200*W+0.00300
      CS2BET=CSZET*CSTHET+SNZET*SNTHET*CSALF
C
C   CSBET=COSINE OF ANGLE OF INCIDENCE (AND ANGLE OF REFLECTION)
C   NECESSARY TO REFLECT INCOMING RADIATION IN A DIRECTION
C   THETA,ALPHA
      CSBET = DSQRT(0.500*(1.000+CS2BET))
C   IF CSBET IS NEGATIVE,SLOPE IS SHADOWED

```

```

      IF(CSBET.GT.0.0) GO TO 9
      RI = 0.000
      GO TO 5
C
C   RHO=REFLECTION COEFFICIENT (APPROXIMATION TO FRESNEL LAW)
C   9 RHO =DEXP(-6.0900*CSBET)+0.01900*CSBET
      SRT = DSQRT(SNSQZT+SNSQTH+2.0*SNZET*SNTHET*CSALF)
C
C   TNTHN IS SEA SURFACE SLOPE (TANGENT OF ZENITH ANGLE OF SURFACE
C   NORMAL) REQUIRED TO REFLECT AT GIVEN THETA AND ALPHA
      TNTHN = SRT/(CSZET + CSTHET)
      THTNC = DATAN(TNTHN)
C   IF THE REFLECTED DIRECTION IS THE SPECULAR POINT, THE SURFACE
C   SLOPE IS ZERO, AND THE SURFACE NORMAL IS VERTICAL. THE
C   AZIMUTH OF THE NORMAL, ALFNC, IS UNDEFINED, AND SRT=0.
C   ALFNC IS DEFINED TO BE ZERO IN THIS CASE BY STATEMENT 10,
C   SINCE ALPHA FOR THE SPECULAR POINT IS 180 OR, FOR AN OVERHEAD
C   SUN,0.
      IF(SRT.EQ.0.0) GO TO 10
      CSALNC = (SNZET + SNTHET*CSALF)/SRT
C
C   10 IF(ALPHA.EQ.0.0.OR.ALPHA.EQ.180.0) CSALNC = 1.000
      IF(ALPHA.EQ.180.0.OR.THETA.GT.ZETA) CSALNC = -1.000
      IF(CSALNC.GT.1.0) CSALNC = 1.000
      IF(CSALNC.LT.-1.0) CSALNC = -1.000
C
      ALFNC = DACOS(CSALNC)
C   NEXT SECTION SETS UP THE ANGULAR REGION OVER WHICH
C   DETERMINATION OF SLOPE PROBABILITY AND SOLID ANGLE
C   ARE MADE
C
C   L AND U AT END OF NAMES REFER TO LOWER AND UPPER LIMITS
C   RESPECTIVELY. THTN=ZENITH ANGLE OF SLOPE NORMAL, AND
C   ALFN IS ITS AZIMUTH. THTNL AND ALFNL CANNOT BE NEGATIVE,
C   THTNU CANNOT EXCEED PI/2, AND ALFNU CANNOT EXCEED PI.
C
      THTNL = THTNC - ANGINC
      IF(THTNL.LT.0.0) THTNL = 0.000
      THTNU = THTNC + ANGINC
      IF(THTNU.GT.HPI) THTNU = HPI
      DTHTN = 0.500*(THTNU-THTNL)
      ALFNL = ALFNC - ANGINC
      IF(ALFNL.LT.0.0) ALFNL = 0.000
      ALFNU = ALFNC + ANGINC
      IF(ALFNU.GT.PI) ALFNU = PI
      DALFN = 0.5 *(ALFNU-ALFNL)
      TL = DTAN(THTNL)
      TU = DTAN(THTNU)
C   P=PROBABILITY OF HAVING SLOPES WITHIN THE DEFINED LIMITS
C
      P = (DEXP(-TL*TL/SIGSQ)-DEXP(-TU*TU/SIGSQ)) *DALFN/PI
C
C   THE REMAINDER OF THE PROGRAM, THROUGH STATEMENT 3, CALCULATES
C   THE SOLID ANGLE INTO WHICH RADIATION IS REFLECTED BY
C   SLOPES HAVING NORMALS LYING BETWEEN THTNL AND THTNU,

```

```

C   AND ALFNL AND ALFNU.  ** THIS SOLID ANGLE MUST BE
C   COMPUTED TO AGREE AS EXACTLY AS POSSIBLE WITH THE
C   RANGE OF NORMALS, SINCE LARGE ERRORS IN THE
C   FINAL RESULT MAY OTHERWISE OCCUR.**
C
C   NSTP=NUMBER OF VALUES PER SIDE FOR WHICH CALCULATED (INCLUDES POINT
C   ZERO)
C       NSTP = STPNMB + 1.000
C
C   SPTHN AND SPALN ARE STEP INTERVALS OF NORMAL THETA AND ALPHA
C       SPTHN = 2.000*DTHTN/STPNMB
C       SPALN = 2.000*DALFN/STPNMB
1421  FORMAT(' HERE')
C       SGN = 1.000
C
C   INTEGRATION ALWAYS STARTS AT SMALLEST VALUES OF NORMAL THETA AND
C   ALPHA, INCREASES ALPHA, THEN THETA, DECREASES ALPHA, THEN
C   THETA.  IN THIS WAY, THE PROPER SIGNS ARE PRESERVED.
C   SGN CONTROLS THE INCREASE OR DECREASE OF THE STEP INTERVALS.
C       THTST = THTNL
C       ALFST = ALFNL - SPALN
C   OMEGA=SOLID ANGLE INTO WHICH RADIATION IS REFLECTED
C       OMEGA = 0.000
C       CSTHST = DCOS(THTST)
C       SNTHST = DSIN(THTST)
C       DO 3 I=1,2
C
C   SOME CONDITIONS OF INTEGRATION RESULT IN ZERO CONTRIBUTION TO
C   THE SOLID ANGLE.  WHILE THE PRECISION OF THE CALCULATIONS
C   (DOUBLE PRECISION ON IBM 360) IS SUFFICIENT TO MAINTAIN
C   PRACTICAL SIGNIFICANCE IN THE SOLID ANGLE, IT IS POSSIBLE
C   IN CERTAIN CASES TO COMPUTE SINES OR COSINES WHICH ARE SLIGHTLY
C   GREATER THAN 1.  SINCE THESE WILL CAUSE PROGRAM FAILURE, THE CASES
C   FOR WHICH THIS MAY OCCUR ARE TREATED SEPARATELY, AND THE
C   INTEGRATION STEPS ARE OMITTED.  THESE ARE CONTROLLED BY THE TWO
C   GROUPS OF *IF* STATEMENTS BELOW, THE FIRST GROUP WHEN
C   THE NORMAL ZENITH ANGLE=0, AND THE SECOND GROUP
C   WHEN THE NORMAL AZITUTH IS 0 OR PI.
C
C       IF(THTNL.EQ.0.0.AND.I.EQ.1) ALFST = ALFNU
C       IF(THTNL.EQ.0.0.AND.I.EQ.1) CSALST =DCOS(ALFST)
C       IF(THTNL.EQ.0.0.AND.I.EQ.1) THTST = THTNL
C       IF(THTNL.EQ.0.0.AND.I.EQ.1) GO TO 11
C
C       DO 1 J=1,NSTP
C       ALFST = ALFST + SGN*SPALN
C       CSALST = DCOS(ALFST)
C
C       IF(I.EQ.1.AND.J.EQ.1.AND.ALFNL.EQ.0.0) CSALST = 1.000
C       IF(I.EQ.2.AND.J.EQ.NSTP.AND.ALFNL.EQ.0.0) CSALST = 1.000
C       IF(I.EQ.1.AND.J.EQ.NSTP.AND.ALFNU.EQ.PI) CSALST = -1.000
C       IF(I.EQ.2.AND.J.EQ.1.AND.ALFNU.EQ.PI) CSALST = -1.000
C
C   SLDANG CALCULATES THE INCREMENT OF SOLID ANGLE
ccc  PRINT *, ' CHECK SLDANG1 ',snzet,cszet,

```

```

ccc +      xnthst, csthst, csalst, j, omega
      CALL SLDANG(SNZET, CSZET, SNTHST, CSTHST, CSALST, J, OMEGA)
ccc PRINT *, ' CHECK SLDANG1 ', snzet, cszet,
ccc +      xnthst, csthst, csalst, j, omega
1 CONTINUE
C
  IF(ALFNL.EQ.0.0.AND.I.EQ.2) GO TO 4
  IF(ALFNU.EQ.PI.AND.I.EQ.1) THTST = THTNU
  IF(ALFNU.EQ.PI.AND.I.EQ.1) SNTHST = DSIN(THTST)
  IF(ALFNU.EQ.PI.AND.I.EQ.1) CSTHST = DCOS(THTST)
  IF(ALFNU.EQ.PI.AND.I.EQ.1) GO TO 21
11 THTST = THTST - SGN*SPTHN
  DO 2 J=1, NSTP
    THTST = THTST + SGN*SPTHN
    SNTHST = DSIN(THTST)
    CSTHST = DCOS(THTST)
ccc PRINT *, ' CHECK SLDANG2 ', snzet, cszet,
ccc +      xnthst, csthst, csalst, j, omega
      CALL SLDANG(SNZET, CSZET, SNTHST, CSTHST, CSALST, J, OMEGA)
ccc PRINT *, ' CHECK SLDANG1 ', snzet, cszet,
ccc +      xnthst, csthst, csalst, j, omega
2 CONTINUE
21 ALFST = ALFNU + SPALN
3 SGN = -1.0
4 RI = RHO*P*CSBET/OMEGA
5 RETURN
  END

```

```

      SUBROUTINE SLDANG(SNZET,CSZET,SNTHST,CSTHST,CSALST,J,OMEGA)
C   SLDANG CALCULATED THE INCREMENT OF SOLID ANGLE
C   OF REFLECTED ENERGY, OMEGA.  SEE OCGLNT FOR DISCUSSION
C   OF DEFINITIONS.  THE INTEGRATION IS PERFORMED ON A
C   LAMBERT AZIMUTHAL EQUAL AREA PROJECTION OF THE
C   UPWARD HEMISPHERE, CENTERED ON THE SOLAR DIRECTION
C   (I.E., THETA=ZETA,ALPHA=0).
C
C   2*BETA=THE ANGLE BETWEEN THE SUN AND THE REFLECTED RAY.
      IMPLICIT REAL *8(A-H), REAL *8(O-Z)
      SAVE STRB,STRM
      CSBT = CSZET*CSTHST + SNZET*SNTHST*CSALST
C   2*SIN(BETA)=2*SNBT=RADIUS OF REFLECTED RAY ON PROJECTION
      SNBT = DSQRT(1.0-CSBT*CSBT)
C
      IF(SNBT.NE.0.0) GO TO 2
      CSMU = CSALST
      GO TO 3
C
C   AMU=AZIMUTH AT PROJECTION CENTER FROM DIRECTION TO VERTICAL
C   (IF SUN IS VERTICAL, AMU MEASURED FROM LINE ALPHA=180)
C
      2  CSMU = (SNZET*CSTHST - CSZET*SNTHST*CSALST)/SNBT
C   *IF* STATEMENTS FORCE CSMU TO BE 1 OR -1 IN APPROPRIATE
C   CASES, IN ORDER TO AVOID PROGRAM ERROR DUE TO ROUNDOFF
C   WHEN ARC COS IS TAKEN
C
      3  IF(CSALST.EQ.-1.0) CSMU = 1.0D0
         IF(CSALST.EQ.1.0.AND.SNTHST.LE.SNZET) CSMU = 1.0D0
         IF(CSALST.EQ.1.0.AND.SNTHST.GT.SNZET) CSMU = -1.0D0
         AMU = DACOS(CSMU)
C
C   IF J=1, THE FIRST POINT ON A SIDE HAS BEEN COMPUTED, AND
C   AN INCREMENT OF AREA CANNOT BE FOUND.  RETURN
C   AND COMPUTE FOR 2ND POINT
C
      IF(J.EQ.1) GO TO 1
C
C   STRB=SNBT FOR J-1, STRM=AMU FOR J-1
C   IF 1ST POINT COMPUTED FOR SIDE WAS AT ORIGIN (OCCURS FOR SUN
C   AT OR NEAR ZENITH) FORCE AZIMUTH DIFFERENCE TO BE ZERO
C   BY SETTING AMU(1)=AMU(2).  THE GENERAL EQUATION MAY
C   RESULT IN A SUBSTANTIAL ERROR FOR AMU(1), SINCE IT IS
C   REALLY UNDEFINED
C
      IF(STRB.EQ.0.0) STRM = AMU
C   RAD=MEAN RADIUS FOR INCREMENT
      RAD = SNBT + STRB
C   DMU=AZIMUTH DIFFERENCE FOR INCREMENT
      DMU = AMU - STRM
C   ADD INCREMENT OF SOLID ANGLE TO PREVIOUS SUM
      OMEGA = OMEGA +0.5D0*RAD*RAD*DMU
C   STORE SNBT AND AMU FOR USE IN NEXT INCREMENT
C
      1  STRB = SNBT
         STRM = AMU
         RETURN
         END

```

```

      SUBROUTINE GEOMET(SNSLOT,CSSLT,CSDOBG,SNSIG,CSSIG,ZETA,ALPHA)
      IMPLICIT REAL *8(A-H), REAL *8(O-Z)
C   PREFIXED-SN=SINE, CS=COSINE
C   SIG=GEOCENTRIC ANGLE OF VIEWED POINT (VP) FROM ORBITAL TRACE
C   ZETA=SOLAR ZENITH ANGLE AT VP
C   ALPHA=AZIMUTH AT VP FROM SUN TO SATELLITE (GE.0,LE.180)
      PI=DACOS(-1.0D0)
      RTD=180.0D0/PI
      CSZET=SNSLOT*SNSIG+CSSLT*CSSIG*CSDOBG
      SNZET=DSQRT(1.0D0-CSZET*CSZET)
      ZETA=DACOS(CSZET)*RTD
      CSALF=(SNSLOT*CSSIG-CSSLT*SNSIG*CSDOBG)/SNZET
      IF(SNSIG.NE.0.0)GO TO 4
      CSALF=1.0D0
      GO TO 5
4     CSALF=-CSALF*SNSIG/ABS(SNSIG)
5     if(dabs(csalf).gt. 1.0d0) then
          print *, 'greater than one...'
          csalf = dint(csalf)
        endif
      ALPHA=DACOS(CSALF)*RTD
      RETURN
      END

```

```

      SUBROUTINE SLOBBC(SNINC,CSINC,SNDEC,CSDEC,SNLMDA,CSLMDA,SNSLOT,
1CSSLOT,SLOBLG)
      IMPLICIT REAL *8(A-H), REAL *8(O-Z)
      SAVE
C   PREFIXES-SN=SINE,CS=COSINE
C   INC=ORBITAL INCLINATION (LE.90), DEC=SOLAR DECLINATION,
C   LMDA=SOLAR LONGITUDE REFERRED TO ASCENDING NODE,
C   (0-360 DEG,W), SLOT=SOLAR ORBITAL LATITUDE, SLOBLG=SOLAR
C   ORBITAL LONGITUDE.
C
      RTD=180.0/DACOS(-1.0D0)
      SNSLOT=CSINC*SNDEC-SNINC*CSDEC*SNLMDA
      CSSLOT=DSQRT(1.0D0-SNSLOT*SNSLOT)
      CSSLOG=CSDEC*CSLMDA/CSSLOT
      SNSLOG=(SNINC*SNDEC+CSINC*CSDEC*SNLMDA)/CSSLOT
      SLOBLG=DACOS(CSSLOG)*RTD
      IF(SNSLOG.LT.0.0)SLOBLG=360.0D0-SLOBLG
      RETURN
      END

```

```

SUBROUTINE SUBLTLN(SNDAU,CSDAU,PHI,ALN,AZ,PHI2,ALN2,AZ2)
IMPLICIT REAL *8(A-H), REAL *8(O-Z)
PI=DACOS(-1.0D0)
D2R = PI/180.0D0
R2D = 180.0D0/PI
PHIR = PHI*D2R
AZR = AZ*D2R
CSPHI = DCOS(PHIR)
SNPHI = DSIN(PHIR)
PHI2 = DASIN(CSDAU*SNPHI + SNDAU*CSPHI*DCOS(AZR))
DELLN = DACOS((CSDAU - SNPHI*DSIN(PHI2))/(CSPHI*DCOS(PHI2)))*R2D
ccc AZ2 = DASIN(CSPHI*DSIN(AZR)/DCOS(PHI2))*R2D+quad
AZ2 = CSPHI*DSIN(AZR)/DCOS(PHI2)
C*****CHECK FOR QUADRANT*****
  if( (az.ge.0.0d0 .and. az.le.90.0d0) .or.
+    (az.gt.270.0d0 .and. az.lt.360.0d0) ) then
    az2=dasin(az2)*r2d
  else
    az2=180.0d0 - dasin(az2)*r2d
  endif
C*****
  PHI2 = PHI2*R2D
  if(phi.lt.phi2) then
    ALN2 = ALN - DELLN
  else
    ALN2 = ALN + DELLN
  endif
RETURN
END

```

```

subroutine initgae
c
c      written by a.b. caughey 18 aug 1988
c
c      subroutine to access gae parameter block
c
c
c      integer xnullstring,xprmdim,xcont,xsuccess,xci,xcs,xsave
c      include 'subdef.fin'
c      parameter (xprmdim=1000,xsuccess=1,xcont=2)
c      integer block(xprmdim)
c
c      character cmsg*72,creply*72
c      integer imsg(18)
c      integer reply(18)
c      equivalence (cmsg,msg),(creply,reply)
c
c      parameter (max_word=1024)
c      integer*2 ibuf(max_word)
c      integer cndblk(80)
c      integer factor,rln,band,type,x,name(6),strg(7)
c      character cname*24,cstrg*28
c      equivalence (cname,name)
c      real y
c      integer idsrn,odsrn
c      integer output
c      logical ok
c
c      integer cnm(3),unitno
c      character*28 blknam
c      integer ipackos(8)
c      save block
c
c      ***get parameter block from tm
c
c      call xrinim(block,xprmdim,xcont,istatus)
c      if(istatus.ne.xsuccess) then
13      format('xrinim error, status = ',i6)
c      write(cmsg,13) istatus
c      call printp(1,msg,72,0,0,0,0,0,0,0,0)
c      goto 1000
c      endif
c      return
c
c      ***get parameter block from disk file
c
c      entry initblk(blknam)
c      lun=12
c      call xsch2p(blknam,ipackos)
c      call xrrdb(ipackos,lun,block,xprmdim,1,istatus)
c      if(istatus.ne.xsuccess) then
c      write(cmsg,('xrrdb error, status = ',i6)) istatus
c      call printp(1,msg,72,0,0,0,0,0,0,0,0)
c      goto 1000
c      endif

```

```

        return
c
c ***write parameter block to disk file
c
    entry wrtblk(blknam)
    lun=12
    call xsch2p(blknam,ipackos)
    call xqwrtb(ipackos,lun,block,istatus)
    if(istatus.ne.xsuccess) then
        write(cmsg,'("xqwrtb error, status = ",i6)') istatus
        call printp(1,msg,72,0,0,0,0,0,0,0,0)
        goto 1000
    endif
    return
c
c *** open i/o files
c
    entry fileop(cnm,unitno)
    call xrstrp(block,cnm,10,name,istatus)
    if(istatus.ne.xsuccess) then
21    format('xrstrp error: parameter = name, status = ',i6)
        write(cmsg,21) istatus
        call printp(1,msg,72,0,0,0,0,0,0,0,0)
        goto 1000
    endif
    inquire(file=cname,exist=ok)
    if(.not.ok .and. (unitno.eq.5 .or. unitno.eq.1)) then
31    format('input file does not exist')
        write(cmsg,31)
        call printp(1,msg,25,0,0,0,0,0,0,0,0)
        goto 1000
    endif
    open(unit=unitno,file=cname)
c
    return
c
c *** get real parameter
c
    entry realgae(cnm,xx)
    call xrreal(block,cnm,1,xx,istatus)
    if(istatus.ne.xsuccess) then
19    format('xrreal error: parameter = x, status = ',i6)
        write(cmsg,19) istatus
        call printp(1,msg,72,0,0,0,0,0,0,0,0)
        goto 1000
    endif
    return
c
c *** get integer paramter
c
    entry intgae(cnm,ix)
    call xrntg(block,cnm,1,ix,istatus)
    if(istatus.ne.xsuccess) then
17    format('xrntg error: parameter = ix, status = ',i6)
        write(cmsg,17) istatus

```

```

        call printp(1,msg,72,0,0,0,0,0,0,0)
        goto 1000
    endif
    return
c
c *** get string paramter
c
    entry strgae(cnm,cstrg)
    call xrstrp(block,cnm,7,strg,istatus)
    if(istatus.ne.xsuccess) then
        write(cmsg,('xrstrp error: parameter = ",3a4," status = ",i6)')
+       (cnm(i),i=1,3),istatus
        call printp(1,msg,72,0,0,0,0,0,0,0)
        goto 1000
    endif
    call xsp2ch(strg,cstrg)
    return
c
c *** write a real parameter to parameter block
c
    entry wrtreal(cnm,xx)
    call xqreal(block,cnm,1,xx,103,istatus)
    if(istatus.ne.xsuccess) then
        write(cmsg,('xqreal error: parameter = xx, status = ",i6)')
+       istatus
        call printp(1,msg,72,0,0,0,0,0,0,0)
        goto 1000
    endif
    return
c
c *** write a integer parameter to parameter block
c
    entry writeint(cnm,ix)
    call xqintg(block,cnm,1,ix,103,istatus)
    if(istatus.ne.xsuccess) then
        write(cmsg,('xqintg error: parameter= ",3a4," status= ",i6)')
+       (cnm(i),i=1,3),istatus
        call printp(1,msg,72,0,0,0,0,0,0,0)
        goto 1000
    endif
    return
1000 stop
end

```

```

CCC$option trace on
SUBROUTINE PLACE(X,Y,RI,KEY,TAUI,ORBINC,SNINC,W,PLAT,PLON)
SAVE
C
COMMON /ABC/ IDATE(3),IPLOT,L(5)
C
DIMENSION VAL(5,800,4),XB(4),XT(4)
CHARACTER*12 refile ! output file for reflectance
DATA LL,ICNT/1,0/
C
MAXIMUM SIZE OF OUTPUT ARRAY
DATA MXMAX,NYMAX/61,17/
C
C
IF(LL.EQ.0)GO TO 101
C
XT(1)=TAUI
XB(1)=TAUI+15.0
DO 100 LP=2,4
LM=LP-1
XT(LP)=XT(LM)+15.0
100 XB(LP)=XB(LM)+15.0
C
C
LL=0
101 CONTINUE
IF(ABS(X).GT.40.0) RETURN
IF(KEY.EQ.2) GO TO 4
ICNT=ICNT+1
LCMP=5
IF(Y.GE.XT(1).AND.Y.LE.XB(1)) LCMP=1
IF(Y.GT.XT(2).AND.Y.LE.XB(2)) LCMP=2
IF(Y.GT.XT(3).AND.Y.LE.XB(3)) LCMP=3
IF(Y.GT.XT(4).AND.Y.LE.XB(4)) LCMP=4
IF(LCMP.EQ.5)RETURN
IF(L(LCMP).GE.800)RETURN
L(LCMP) = L(LCMP) + 1
VAL(1,L(LCMP),LCMP) = Y
VAL(2,L(LCMP),LCMP) = X+23
VAL(3,L(LCMP),LCMP) = RI*100
VAL(4,L(LCMP),LCMP) = PLAT*3600.
VAL(5,L(LCMP),LCMP) = PLON*3600.
RETURN
C
C
4
XM=0.
DO 16 KK=1,4
IF(L(KK).EQ.0)GO TO 16
XM=XM+1.
KL = 5 - KK
16 CONTINUE
C
C
C PLOT THE VALUES
C
MM=XM

```

```

      SIZE=0.14
C      output file opened in subroutine initgae
      call fileop(10h/reflfile/,9)
      rewind 9
      ICOUNT=1
C
C
      DO 36 KK=1,MM
      NUMV=L(KK)
      XK=KK-1
      A=XK*4.546875+1.0
      DO 36 I=1,NUMV
C
C      CONVERTS 65.680 TO 0. VALUES IN X DIRECTION TO 1. TO 4.546875*XK+1.
C      POSITION IN X DIRECTION = (VAL(1,I,KK)-XT(KK))*4.3786*0.06922769+A
      XNUM=(VAL(1,I,KK)-XT(KK))*0.30312+A
C
C      CONVERTS -40 TO +40 VALUES IN Y DIRECTION
      YNUM=(VAL(2,I,KK)+40.)*.096875 + 1.
C
      FVAL=VAL(3,I,KK)*10.
      MX=(XNUM-1.)/0.303125 + 1.2
      NY=(YNUM-1.)/0.484375 + 1.2
      LAT=NINT(VAL(4,I,KK))
      LON=NINT(VAL(5,I,KK)) - 39600
      IF(MX.GE.1 .OR. MX.LE.MXMAX .OR. NY.GE.1 .OR. NY.LE.NYMAX)then
        print *, 'we are finally printing something '
        WRITE(9,('(point',I3.3,2(1X,I3),2(1X,I7),3(1X,I6)))')
+      ICOUNT,MX,NY,LAT,LON,REAL(FVAL),real(fval),icount
        endif
      ICOUNT=ICOUNT+1
      IF(IPLOT.LT.1) GO TO 36
C
      ICHAR=FVAL+0.5
      FVAL=ICHR
      ICHAR=1
      IF(FVAL.GE. 9.5) ICHAR=2
      IF(FVAL.GE. 99.5) ICHAR=3
      IF(FVAL.GE.999.5) ICHAR=4
      XSHFT=SIZE*0.5
      YSHFT=SIZE*ICHR*0.84*0.5
C
C 36 CONTINUE
      CLOSE (9)
C
C      CLEAR OUTPUT ARRAY
C
      DO 3000 IX=1,3
        DO 2000 IY=1,800
          DO 1000 IZ=1,4
            VAL(IX,IY,IZ)=0.0D0
          1000 CONTINUE
        2000 CONTINUE
      3000 CONTINUE
      RETURN
C
C 6000 FORMAT(6H *****, 'NOT ENOUGH POINTS TO PLOT', 5H*****)
C
      END

```

ccc*option trace on

SUBROUTINE ELOC(IEPH, SCANG, DALT, PHI, PHI2, ALN, ALN2,

+ PLAT, PLON, IERR)

!EARTH LOCATE

SAVE

C=====C

C (FORMERLY NAMED ELOCAT)

C ELOCAT GIVEN THE SCAN TIME, THE SCAN ANGLE, AND THE THREE ATTITU

DE

C ANGLES, COMPUTES THE LATITUDE AND LONGITUDE OF THE PRINCIPAL PO

INT,

C THE ALTITUDE OF THE SATELLITE ABOVE THE EARTH, AND THE ZENITH A

NGLE

C ABOVE THE PRINCIPAL POINT.

C ELOCAT WAS BASED ON ROUGHLY THE SECOND HALF OF THE PROGRAM CELCA

L OB

C FROM M. CHALFANT OF NOAA/NESS

C MODIFICATIONS BY G. COLLINS, ODSI APRIL 1975

C MODIFICATIONS BY STEVE REYNOLDS, TELOS COMPUTING, NOV. 1977

C MODIFIED BY DONALD C. SCHERTZ 1978-1982 NEPRF MONTEREY, CALIF

C INPUT

C IEPH - INDEX INTO EPHEMERIS TABLES FOR SUBTRACK POINT

C ISP - INDEX INTO SCAN POINT TABLES FOR SCAN ANGLE

C OUTPUT

C PLAT - LATITUDE OF PRINCIPAL POINT (PRINCIPAL POINT IS
C THE ASSUMED CENTER OF THE SCAN AREA ON THE EARTH.
C IN DEGREES, + NORTHERN HEMISPHERE, - SOUTHERN)

C PLON - LONGITUDE OF PRINCIPAL POINT. (DEGREES, + EASTWARD
C FROM GREENWICH)

C IERR - ERROR FLAG

C =0 - NO ERRORS

C =2 - NEW EPHEMERIS TABLE FOR NEW DAY NEEDED

C =3 - SPOT OFF EARTH

C =4 - BAD ORBITAL DATA

C =5 - UNKNOWN TYPE ERROR

C COMMON /EPHE2/ CONTAINS THE EPHEMERIS CALCULATIONS

C NEPH - NUMBER OF EPHEMERIS POINTS

C STIME - STARTING TIME OF EPHEMERIS DATA (SECONDS AFTER EPOCH)

C FTIME - ENDING TIME OF EPHEMERIS DATA (SECONDS AFTER EPOCH)

C ETIM - ELAPSED SECONDS SINCE EPOCH TIME

C ELAT - SUBSATELLITE LATITUDE, DEGREES, + NORTHERN
C HEMISPHERE, - SOUTHERN.

C ELON - SUBSATELLITE LONGITUDE, DEGREES, + EASTWARD
C FROM GREENWICH RANGING 0 TO 360

C EALT - SATELLITE ALTITUDE ABOVE ELLIPSOID (KM)

C (THE LAST FOUR VALUES ARE PROVIDED BY

C CALLS TO DATA PAGING ROUTINES)

C=====C

C

PARAMETER (NSP=17)

PI=ACOS(-1.)

PIOV2=PI/2.0

TWOPI=2.0 *PI

RADDEG=180.0/PI

DEGRAD=PI/180.0

C

C=====C

C IERR=0 !INIT RETURN ERR INDICATOR

C SET SCANG FROM SCAN ANGLE TABLE

C CHECK IF SUBTRACK POINT IS SAME AS LAST CALL,

```

C      SKIP CALCULATION OF TRACKLINE AZIMUTH (A) IF SO.
      IF(IEPH.GT.1)GO TO 80
C-----C
40     CONTINUE
      SPLAT=PHI*DEGRAD      !CONVERT TO RADIANS, 1 = ELAT
      SPLON=ALN*DEGRAD      !2 = ELON
      SPALT=DALT            !SATELLITE ALTITUDE IN KM, 3 = EALT
C     CHECK FOR QUICK BYPASS IF SCAN IS ALONG NADIR (I.E., SUM OF SCAN
C     AND ROLL ANGLES IS ZERO AND PITCH ANGLE IS ZERO.)
      IF(SCANG.EQ.0.0)GO TO 280
C     PICK UP NEXT POINT FOR TRACKLINE AZIMUTH CALCULATIONS
      SLON2=ALN2*DEGRAD     !2 = ELON
      SLAT2=PHI2*DEGRAD     !1 = ELAT
C     THE NEXT SECTION SOLVES THE SPHERICAL TRIANGLE
C     BETWEEN THE SUBPOINT, A FUTURE SUBPOINT, AND THE NORTH POLE
C     TO DETERMINE THE TRACKLINE AZIMUTH(A).
      SA=PIOV2-SLAT2
      CSA=COS(SA)
      SSA=SIN(SA)
ccc    print *, ' piov2 = ',piov2,' splat = ',splat,' sb = ',sb
      SB=PIOV2-SPLAT
      AC=SLON2-SPLON
      CSB=COS(SB)
ccc    print *, ' sb = ',sb
      SSB=SIN(SB)
      CAC=COS(AC)
      CKON=SSA*SSB*CAC
      CSAB=CSA*CSB
      AKON=CSAB+CKON
      SC=ACOS(AKON)
      IF(SC.EQ.0.)GO TO 350
      CSC=COS(SC)
      SSC=SIN(SC)
      ACSB=CSB*CSC
      ASBC=SSB*SSC
ccc    print *, ' csa = ',csa,' acsb = ',acsb,' asbc = ',asbc
ccc    print *, ' ssb = ',ssb,' ssc = ',ssc
      A=(CSA-ACSB)/ASBC
      IF(A.GE.1.)GO TO 350
      IF(A.LE.-1.)GO TO 350
      A=ACOS(A)*RADDEG      ! *****THIS WAS ALTERED PER JIM CLARK*****
50     IF(A.LT.360.)GO TO 60
      A=A-360.
      GO TO 50
60     IF(A.GE.0.)GO TO 70
      A=A+360.
      GO TO 60
70     A=A*DEGRAD
C
C
C
80     SCANG=SCANG*DEGRAD
      PANAD=SCANG
C     THIS SECTION CALCULATES THE AZIMUTH FROM THE SUB-SATELLITE POINT
C     TO THE PRINCIPAL POINT (AA).

```

```

          IF(SCANG)90,280,100
90      TANAD=-PANAD
          AA=A-PIOV2
          GO TO 110
100     TANAD=PANAD
          AA=A+PIOV2
110     IF(AA)120,140,130
120     AA=AA+TWOPI
          GO TO 110
130     IF(AA.LT.TWOPI)GO TO 140
          AA=AA-TWOPI
          GO TO 130
C      CALCULATION OF THE ZENITH ANGLE.
140     RKON=.0001569612*SPALT+1.
          ZANG=ASIN(RKON*SIN(TANAD))
C      TEST FOR EARTH TANGENCY
          CNAL=ASIN (1./RKON)
ccc     print *, ' cnal = ',cnal,' tanad = ',tanad
          IF(CNAL.LE.TANAD)GO TO 360
C      THIS SECTION SOLVES THE SPHERICAL TRIANGLE BETWEEN THE SUBPOINT,
C      THE PRINCIPAL POINT AND THE NORTH POLE.  REMEMBER IF
C      A B+C=180 THEN A=(SUPPLEMENT OF B) -C.
          SIDEA=ZANG-TANAD
          IF(SPLAT.LE.0.)GO TO 150
          CNOS=0.
          SIDEB=PIOV2-SPLAT
          GO TO 160
150     CNOS=1.
          SIDEB=PIOV2+SPLAT
160     IF (SIDEB.LT.0.)GO TO 370
          IF(AA)370,320,170
170     IF(AA-PI)180,290,200
180     CAPC=AA
          WOE=1.
          IF(CNOS)370,220,190
190     CAPC=PI-CAPC
          GO TO 220
200     WOE=0.
          CAPC=TWOPI-AA
          IF(CNOS)370,220,210
210     CAPC=AA-PI
220     SINA=SIN(SIDEA)
          SINB=SIN(SIDEB)
          COSC=COS(CAPC)
          TEMP2=SINB*SINA*COSC
          COSB=COS(SIDEB)
          COSA=COS(SIDEA)
          SIDEC=ACOS(TEMP2+COSB*COSA)
          SINC=SIN(CAPC)
          SINEC=SIN(SIDEC)
          CAPA=ASIN(SINA*SINC/SINEC)
C      SEE IF CAPA IS IN THE SECOND QUADRANT.
          COST=COSB*COS(SIDEC)
          IF(COST.GE.COSA)CAPA=PI-CAPA
C      NOW DETERMINE PRINCIPAL POINT LATITUDE AND LONGITUDE.

```

```

      PLAT=PIOV2-SIDEC
      IF(CNDS)370,240,230
230    PLAT=-PLAT
240    IF(WOE)370,250,260
250    PLON=SPLON+CAPA
      IF(PLON.GT.TWOPI)PLON=PLON-TWOPI
      GO TO 270
260    PLON=SPLON-CAPA
      IF(PLON.LT.0.)PLON=PLON+TWOPI
C      FINAL CLEAN-UP AND CONVERSION
270    PLAT=PLAT*RADDEG
      PLON=PLON*RADDEG
      GO TO 380 !RETURN
C      SPECIAL CASES
C      SPECIAL CASE.  THE TRUE NADIR ANGLE IS ZERO AND THE PRINCIPAL
C      POINT AND THE SUBPOINT COINCIDE.
280    TANAD=0.
      ZANG=0.
      PLAT=SPLAT
      PLON=SPLON
      GO TO 270
C      SPECIAL CASE (LINE TRIANGLE).
290    PLON=SPLON
      PLAT=SPLAT-SIDEA+PIOV2
      IF(PLAT.GE.0.)GO TO 300
      TLAT=PLAT+PIOV2
      IF(TLAT.LT.0.)TLAT=-TLAT
      PLAT=-TLAT
      GO TO 310
300    PLAT=PLAT-PIOV2
      GO TO 270
C      SPOT IS OVER THE POLE, REVERSE THE LONGITUDE.
310    PLON=SPLON+PI
      IF(PLON.GT.TWOPI)PLON=PLON-TWOPI
      GO TO 270
C      SPECIAL CASE (LINE TRIANGLE).
320    PLON=SPLON
      PLAT=SPLAT+SIDEA-PIOV2
      IF(PLAT.GT.0.)GO TO 330
      PLAT=PLAT+PIOV2
      GO TO 270
330    TLAT=PLAT-PIOV2
      IF(TLAT.LT.0.)TLAT=-TLAT
      PLAT=TLAT
      GO TO 310
C===== ERROR RETURNS =====C
C      EPHEMERIS TABLES FOR NEW DAY NEEDED
340    IERR=2
C      RESET INDEX FOR NEW EPHEMERIS TABLE ON NEXT ENTRY
      GO TO 380
C      IMPOSSIBLE SITUATION, MUST BE CAUSED BY BAD ORBITAL DATA.
350    IERR=4
      GO TO 380
C      OFF THE EARTH RETURN.
360    IERR=3

```

```
C      SET DUMMY VALUES
      GO TO 380
C      MISCELLANEOUS ERRORS
370    IERR=5
380    RETURN
      END
```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C SUBROUTINE IDIFDTG
C

```

```

C
C PURPOSE: TO COMPUTE THE TIME DIFFERENCE BETWEEN TWO DISPLAY CODE
C LEFT-JUSTIFIED DATE-TIME GROUPS (DTGS).
C

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C

```

```

C INPUT BY THE USER: NONE
C

```

```

C OUTPUT: IDIFDGT
C

```

```

C THIS SUBROUTINE IS CALLED BY PROGRAM SUNGLINT
C

```

```

C THIS SUBROUTINE CALLS: NONE
C

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C

```

```

C VARIABLES TYPE ALLOWED
C INPUT
C DTG1 FIRST LEFT-JUSTIFIED DATE-TIME GROUP CHAR 00000000 -
C 99123123
C DTG2 SECOND LEFT-JUSTIFIED DATE-TIME GRP CHAR 00000000 -
C 99123123
C

```

```

C OUTPUT
C IDIFDTG TIME DIFFERENCE IN HOURS BETWEEN DTG2
C AND DTG1 INTEGER 0 - 876767
C

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C

```

```

C METHOD: THIS FUNCTION CONVERTS DTG INTO HOURS SINCE 1900
C THEN SUBTRACTS 2ND DTG FROM 1ST DTG. THE DIFFERENCE
C IS IN HOURS.
C

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C

```

```

C FUNCTION IDIFDTG(DTG1,DTG2)
C

```

```

C CHARACTER *12 CDTG(2), DTG1, DTG2
C INTEGER IMOS(12),IHRS(2)
C

```

```

C NUMBER OF DAYS IN EACH MONTH
C

```

```

C DATA IMOS/31,28,31,30,31,30,31,31,30,31,30,31/
C

```

```

C CALCULATE NUMBER OF HOURS FROM 1900phcNZ9JT64491uvLhL44h8Gu
C

```

```

X6zo!jT3L444h8JIKk#QG2
C

```

```

C DO 100 I=1,2
C READ(CDTG(I),9001) IYR,IMO,IDAY,IHR
C IF(MOD(IYR,4).EQ.0) THEN

```

```

C          ADD IN YEARS
          IHRS(1)=IYR*366*24
          IMOS(2)=29
        ELSE
          IHRS(1)=IYR*365*24
          IMOS(2)=28
        ENDIF
C          ADD IN MOS
        DO 50 J=1,IMO
          IHRS(1)=IHRS(1) + IMOS(J)*24
50      CONTINUE
C          ADD IN DAYS AND HOURS
        IHRS(1)=IHRS(1) + (IDAY*24) + IHR
100 CONTINUE
C
C          TIME DIFFERENCE IN HOURS BETWEEN IDTG1 & IDTG2
C
          IDIFDTG=IHRS(2)-IHRS(1)
          RETURN
9001  FORMAT(4I2)
          END

```

```

SUBROUTINE INTERP(UVFLD,WLAT,WLON,FLAT,FLON,GSPD)

C
C      UVFLD  --  U FIELD IN UVFLD(II,JJ,1)          M/S
C              V FIELD IN UVFLD(II,JJ,2)          M/S
C      WLAT   --  LAT OF POINT TO BE INTERPOLATED TO  DEGREES
C      WLON   --  LONG OF POINT TO BE INTERPOLATED TO  DEGREES
C      FLAT   --  MAXIMUM LAT OF UV GRID FIELD         DEGREES
C      FLON   --  MAXIMUM LONG OF UV GRID FIELD        DEGREES
C      GSPD   --  WIND SPEED                           M/S
C
      REAL UVFLD(73,144,2),GOBS(2)
      PARAMETER (NREP=2,GRDLONSP=2.5,GRDLATSP=2.5)
      PARAMETER (DLT=2.5,DLN=2.5)
C      DLT & GRDLATSP and DLN & GRDLONSP should be equivalenced!!!!
CCC  EQUIVALENCE (DLT,GRDLATSP),(DLN,GRDLONSP)
C
C      INTERPOLATE TO OBSERVATION POINT
C
      DX=MOD(WLAT,GRDLATSP)
      DY=MOD(WLON,GRDLONSP)
      GLAT=WLAT-DX+GRDLATSP
      GLON=WLON-DY+GRDLONSP
      A=ABS(DY/DLT)
      B=ABS(DX/DLN)
      I=ABS(GLAT-FLAT)/2.5+1
      J=ABS(GLON-FLON)/2.5+1
      DO 100 K=1, NREP
          VIJ  = UVFLD(I,J,K)
          VIJP = UVFLD(I,J+1,K)
          VIPJ = UVFLD(I+1,J,K)
          VPP  = UVFLD(I+1,J+1,K)
C
C              PERFORM BILINEAR INTERPOLATION
C
          GOBS(K) = VIJ +(VIJP -VIJ)*A +(VIPJ -VIJ)*B
          &      +(VPP +VIJ -VIJP -VIPJ)*A*B
100  CONTINUE
      GSPD = (SQRT(GOBS(1)**2 + GOBS(2)**2))
C
      RETURN
      END

```

APPENDIX B

SUNGLINT PROGRAM RUNNING INSTRUCTIONS

The Sunlint algorithm is implemented in Global Imaging's executive, operating on NEPRF's HP9020. There are two modes in which to run the sunlint algorithm. The first mode requires an input image with a navigational file. The second mode requires the user to supply the required navigation parameters.

To plot sunlint isopleths on a navigated image, from Global Imaging's executive, type "tutor SUN". The following parameters must be assigned values:

PARAMETERS

IMAGE	Input image name. Type = string Length = 12 Count = 1 Default= "NONE"
Gfile	Flag "Y" is set if a navigational file is available for the input image. Type = string Length = 1 Default= "Y" Valid = ("Y", "N")
WIND	Wind field file name. Type = string Length = 12 Count = 1 Default= "INPUT.DAT"

To calculate reflectance for a given area and time, type "tutor SUN". The output will be a file of reflectance values and their location. Values must be assigned to the following parameters:

PARAMETERS

IMAGE	Input image name. Type = string Length = 12
-------	---

	Count = 1 Default= "NONE"
Gfile	Flag "Y" is set if a navigational file is available for the input image. Type = string Length = 1 Default= "Y" Valid = ("Y","N")
YEAR	Year of image. Type = integer Count = 1 Default= 0
MONTH	Month of image. Type = integer Count = 1 Default= 0
DAY	Day of image. Type = integer Count = 1 Default= 0
HOURL	Hour of image. Type = integer Count = 1 Default= 0
MIN	Minute of image. Type = integer Count = 1 Default= 0
EQLONG	Longitude of Equator crossing. Type = real Count = 1 Default= 0
EQHEM	Hemisphere of Equator crossing. Type = string Length = 1 Count = 1 Default= "W"
SATINCL	Satellite inclination angle.

	Type = real Count = 1 Default= 98.7
SATALT	Average satellite height in km. Type = real Count = 1 Default= 0 Count= 1
WIND	Wind field file name. Type = string Length = 12 Count = 1 Default= "INPUT.DAT"
platmin	Latitude of southern most point on image. Type = real Default = 12.5 Count = 1
platmax	Latitude of northern most point on image. Type = real Default = 30.0 Count = 1
plone	Longitude of eastern most point on image. Type = real Default = 227.5 Count = 1
plonw	Longitude of western most point on image. Type = real Default = 252.5 Count = 1

DISTRIBUTION LIST

ASST. FOR ENV. SCIENCES
ASST. SEC. OF THE NAVY (R&D)
ROOM 5E731, THE PENTAGON
WASHINGTON, DC 20350

OFFICE OF NAVAL RESEARCH
CODE 1122AT, ATMOS. SCIENCES
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL RESEARCH
ATTN: PROG. MANAGER, 1122CS
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL RESEARCH
CODE 1122 PO, PHYS. OCEANO.
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL TECHNOLOGY
ONR (CODE 22)
800 N. QUINCY ST.
ARLINGTON, VA 22217-5000

CHIEF OF NAVAL RESEARCH
LIBRARY CODE 01232L
800 QUINCY ST.
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL RESEARCH
ENV. SCI. PROGRAM, CODE 112
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL RESEARCH
ATTN: HEAD, OCEAN SCI. DIV
CODE 1122
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL RESEARCH
CODE 1122 MM, MARINE METEO.
ARLINGTON, VA 22217-5000

CHIEF OF NAVAL OPERATIONS
OP-082
U.S. NAVAL OBSERVATORY
WASHINGTON, DC 20390

OFFICER IN CHARGE
NAVOCEANCOMDET
AFGWC
OFFUTT AFB, NE 68113

COMMANDING OFFICER
NAVAL RESEARCH LAB
ATTN: LIBRARY, CODE 2620
WASHINGTON, DC 20390

OFFICE OF NAVAL RESEARCH
SCRIPPS INSTITUTION OF
OCEANOGRAPHY
LA JOLLA, CA 92037

COMMANDING OFFICER
NAVAL OCEAN RESCH & DEV ACT
JCSSC, MS 39529-5004

COMMANDER, NAVAL
OCEANOGRAPHY COMMAND
STENNIS SPACE CENTER
JCSSC, MS 39529

COMNAVOCEANCOM
ATTN: CODE N5
JCSSC, MS 39529-5000

COMMANDING OFFICER
NAVAL OCEANOGR. OFFICE
JCSSC, MS 39522-5001

COMMANDING OFFICER
FLENUMOCEANCEN
MONTEREY, CA 93943-4005

SUPERINTENDENT
LIBRARY REPORTS
U.S. NAVAL ACADEMY
ANNAPOLIS, MD 21402

DIRECTOR OF RESEARCH
U.S. NAVAL ACADEMY
ANNAPOLIS, MD 21402

NAVPGSCOL
METEOROLOGY DEPT.
MONTEREY, CA 93943-5000

COMMANDER, SPAWARSSCOM
ATTN: CODE PMW 141-81
WASHINGTON, DC 20363-5100

CHAIRMAN
OCEANOGRAPHY DEPT.
U.S. NAVAL ACADEMY
ANNAPOLIS, MD 21402

NAVAL POSTGRADUATE SCHOOL
OCEANOGRAPHY DEPT.
MONTEREY, CA 93943-5000

LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5002

COMMANDER
NAVAIRSSCOM
ATTN: LIBRARY (AIR-723D)
WASHINGTON, DC 20381-0001

COMMANDER
NAVAIRSSCOM, CODE 526W
WASHINGTON, DC 20381-0001

COMMANDER, NAVAL SURFACE
WARFARE CENTER
DAHLGREN, VA 22448-5000

DIRECTOR
NAVSURFWARCEN, WHITE OAKS
NAVY SCI ASSIST. PROGRAM
SILVER SPRING, MD 20903-5000

COMMANDER
NAVOCEANSSCEN
DR. J. RICHTER, CODE 54
SAN DIEGO, CA 92152-5000

COMMANDING OFFICER
AMPHIBIOUS WARFARE LIBRARY
NAVAL AMPHIBIOUS SCHOOL
NORFOLK, VA 23521

AFGLLY
HANSCOM AFB, MA 01731

USAFETAC/TS
SCOTT AFB, IL 62225

DIRECTOR (10)
DEFENSE TECH. INFO. CENTER
CAMERON STATION
ALEXANDRIA, VA 22314

CHIEF
MARINE & EARTH SCI. LIBRARY
NOAA, DEPT. OF COMMERCE
ROCKVILLE, MD 20852

DIRECTOR
NHC, NOAA, GABLES ONE TOWER
1320 S. DIXIE HWY
CORAL GABLES, FL 33146

DIRECTOR, NOAA
ATLANTIC MARINE CENTER
COAST & GEODETIC SURVEY,
439 W. YORK ST.
NORFOLK, VA 235110

FISHERIES LIBRARY
NOAA
75 VIRGINIA BEACH DRIVE
MIAMI, FL 33149

DIRECTOR, ATLANTIC OCEANO.
& METEOROLOGY LABS
15 RICKENBACKER CAUSEWAY
VIRGINIA KEY
MIAMI, FL 33149

DIRECTOR, NOAA
PACIFIC MARINE CENTER
NATIONAL OCEAN SURVEY,
1801 FAIRVIEW AVE., EAST
SEATTLE, WA 98102

NATIONAL MARINE FISH. SERV.
OCEAN CLIMATOLOGY PROJECT
P.O. BOX 271
LA JOLLA, CA 92037

LABORATORY DIRECTOR
NATIONAL MARINE FISH. SERV.
NOAA, P.O. BOX 6
WOODS HOLE, MA 02543

DIRECTOR
TECHNIQUES DEVELO. LAB
GRAMAX BLDG.
8060 13TH ST.
SILVER SPRING, MD 20910

HEAD, ATMOS. SCIENCES DIV.
NATIONAL SCI. FOUNDATION
1800 G STREET, NW
WASHINGTON, DC 20550

SCRIPPS INSTITUTION OF
OCEANOGRAPHY, LIBRARY
DOC/REPORTS SECTION
LA JOLLA, CA 92037

WOODS HOLE OCEANO. INST.
DOCUMENT LIBRARY LO-206
WOODS HOLE, MA 02543

DIRECTOR
COASTAL STUDIES INSTITUTE
LOUISIANA STATE UNIVERSITY
ATTN: O. HUH
BATON ROUGE, LA 70803

AMERICAN METEORO. SOCIETY
MET. & GEOASTRO. ABSTRACTS
P.O. BOX 1736
WASHINGTON, DC 20013

DIRECTOR, METEO. & OCEANO.
NATIONAL DEFENSE HDQ.
OTTAWA, ONTARIO, KIA 0K2
CANADA

DIRECTOR OF NAVAL OCEANO.
& METEOROLOGY
MINISTRY OF DEFENCE
LACON HOUSE, THEOBOLD RD.
LONDON WC 1X8RY, ENGLAND

LIBRARY, INSTITUTE OF
OCEAN SCIENCES
ATTN: DIRECTOR
WORMLEY, GODALMING
SURRY GU8 5UB, ENGLAND