THE NEPRF SUNGLINT PATTERN CALCULATION AND DISPLAY SYSTEM

James R. Clark and Allan Caughey
Naval Environmental Prediction Research Facility
QUALIFIED REQUESTORS MAY OBTAIN ADDITIONAL COPIES FROM THE DEFENSE TECHNICAL INFORMATION CENTER.
ALL OTHERS SHOULD APPLY TO THE NATIONAL TECHNICAL INFORMATION SERVICE.
The NEPRF Sunglint Pattern Calculation and Display System (U)

Clark, James R. and Caughey, Allan B.

Final

From 10/87 to 9/88

1989, January

63

The sunglint 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 sunglint patterns are shown. The computer code and operating instructions are provided as appendices.
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:

\[
\text{ARG} = \frac{\text{Del(DTG)}}{24.0/365.0*360.0*(\pi/180.0)}, \text{ and}
\]
\[
\delta = 23.5\times \sin(\text{ARG})
\]

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

\[
\text{HA} = 540.0 - \text{Time(hrs)}\times 15.0,
\]

where Time = Hours + Minutes/60.0 + 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 (\(\delta\)), the LHA (\(\lambda\)), and the inclination of the satellites equatorial crossing (\(i\)), the solar orbital latitude (\(\phi\)) and the solar orbital longitude (\(\tau\)) may be calculated in the following manner. Following the geometry of Figure 2, given spherical triangles (\(\Delta\)) EPS, ESO, ENS, ESA, use the law of sines for a spherical triangle:
Figure 2. Geometry of the transformation from $i, \delta, \lambda$ to $\phi, \tau$. 
\[
\sin \lambda' = \sin \omega \\
\sin \delta' = \sin \delta; \quad \text{for } \Delta \text{EPS, } \sin \phi = \sin \phi' = \sin \phi
\]
\[
\sin \left(\varepsilon + i'\right) = \sin 90^\circ \\
\sin \lambda' = \sin \phi
\]
\[
\sin \phi = \sin \phi
\]

Angles PAE, ENO, OEN, PEA are right angles. Further,
\[
\omega + \varepsilon + i' = 90^\circ; \quad \phi = 90^\circ - i; \quad \phi' = 90^\circ - \varepsilon; \quad \cos 90^\circ = 0.0;
\]
\[
\varepsilon = 90^\circ-(\omega + i'); \quad \sin \left(\varepsilon + i'\right) = \cos \omega; \quad \delta' = 90^\circ - \delta; \quad \sin 90^\circ = 1.0;
\]
\[
\varepsilon + i' = 90^\circ - \omega; \quad \sin \varepsilon = \cos \left(\omega + i'\right); \quad \sin \phi' = \cos \phi; \quad \lambda' = \lambda - 360^\circ;
\]
\[
\cos i' = \sin i; \quad \sin \phi = \cos \phi; \quad \lambda' = -\lambda.
\]

For \(\Delta \text{ESA}\),
\[
\cos \phi \sin \tau = \cos \left(\omega + i'\right) \sin Q.
\]
Substituting for \(\sin Q\) in \(\Delta\)'s EPS and EBA,
\[
\frac{\sin \omega}{\sin \delta'} = \frac{\cos \omega}{\sin \delta'}, \quad \text{or } \tan \omega = \frac{\cos \delta \sin \lambda'}{\sin \delta}.
\]

Substituting for \(\varepsilon\) in \(\Delta \text{ESO}\) and expanding,
\[
\cos \phi \sin \tau = (\cos \omega \cos i' - \sin \omega \sin i') \sin Q.
\]

After substituting for \(\sin (\varepsilon + i')\) in \(\Delta \text{EBA}\), now substitute for \(\sin Q\) in the expanded form of \(\Delta \text{ESO}\):
\[
\cos \phi \sin \tau = (\cos \omega \cos i' - \sin \omega \sin i') \sin \delta / \cos \omega, \quad \text{or}
\]
\[
\cos \phi \sin \tau = (\cos i' - \tan \omega \sin i') \sin \delta.
\]

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 $\tan \omega$, $\cos \iota'$, and $\sin \iota'$, producing
\[ \sin \phi = \cos \iota \sin \delta - \sin \iota \cos \delta \sin \lambda. \]

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 ($\iota$) + 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,
\[
\cos \phi' = \cos \delta \cos \phi' + \sin \delta \sin \phi' \cos \iota_t, \quad \text{where}
\]
\[
\phi' = 90.0 - \phi; \quad \cos \phi' = \sin \delta; \quad \sin \phi' = \cos \phi.
\]
Substituting, we obtain
\[
\sin \phi = \cos \delta \sin \phi_t + \sin \delta \cos \phi \cos \iota_t. \quad \text{Similarly,}
\]
\[
\cos \delta \lambda_{t+1} = \cos \phi_t \cos \phi_{t+1} = \cos \delta - \sin \phi \sin \phi_{t+1},
\]
or rearranging,
\[
\cos \delta \lambda_{t+1} = (\cos \delta - \sin \phi \sin \phi_{t+1})/(\cos \phi \cos \phi_{t+1}).
\]
And,
\[
\sin \iota_{t+1} = \cos \phi_{t+1} \sin \iota_t / \cos \phi_t.
\]

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:
\[
\sin \theta' = (r + h) \sin \eta / r, \quad \text{and}
\]
\[
\theta' = 180.0 - \theta, \quad \sigma = \phi - \eta, \quad \text{where}
\]
\[
\theta = \text{the zenith angle of the satellite,}
\quad \eta = \text{the scan angle of the satellite,}
\]
Figure 4. The angular measurements along a satellite scan line.
\( \sigma \) = the geocentric angle,  
\( h \) = the height of the satellite, and  
\( r \) = the radius of the earth.

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

With the satellite geocentric angle (\( \sigma \)), along with the solar orbital latitude (\( \phi \)) and longitude (\( \tau \)), the zenith angle (\( \zeta \)) and its azimuth (\( \alpha \)) 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 \( \Delta VSE \), \( \cos Q = \cos \zeta \cos \sigma + \sin \zeta \sin \sigma \cos \alpha' \).

For \( \Delta ESO \), \( \cos Q = \sin \phi \cos 90^\circ + \cos \phi \sin 90^\circ \cos \tau \) or,

\[
\cos Q = \cos \phi \cos \tau.
\]

Substituting for \( \cos Q \) in \( \Delta VSE \),

\[
\cos \zeta \cos \sigma + \sin \zeta \sin \sigma \cos \alpha' = \cos \zeta \cos \sigma / \sin \sigma.
\]

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

\[
\cos \alpha = \sin \phi \cos \sigma - \sin \sigma \cos \phi \cos \tau.
\]

12
Figure 5. The relationship of azimuth and solar zenith to SNUGS coordinates.
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 ($\rho$) 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:

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

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

The angle of reflection (the incidence angle), $\beta$ is the angle necessary to reflect incoming solar radiation in the direction theta, with an azimuth angle alpha. Figures 6a and 6b illustrate the geometry of the problem. For spherical $\Delta TVS$,

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

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

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:

$$
P = \frac{\exp(-\tan^2 \theta^N/\sigma^2) - \exp(-\tan^2 \theta^M/\sigma^2)}{\sigma^N - \sigma^M}/2\pi, \text{ where}
$$
Figure 6. The relationship between satellite zenith angle, solar zenith angle, azimuth and incidence angle. (a) Plane view, (b) Section view.
subscript \( l \) and \( \mu \) 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 \( d\omega \). The limits are between \( \theta^N_l, \phi^N_l \) and \( \theta^N_\mu, \phi^N_\mu \) centered on \( \theta^N_c, \phi^N_c \). \( \tan \theta^N_c \) 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 \( \theta \) and azimuth \( \alpha \). \( \tan \theta^N_c \) is the ratio of the sine of the vector difference of \( \theta, \phi \) and the cosine of the vector sum of \( \theta, \phi \):

\[
\tan \theta^N_c = \frac{\sin \theta^N_c}{\cos \theta^N_c}
\]

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

\[
\sin^2 \theta^N_c = \sin^2 \theta + \sin^2 \phi - 2 \sin \theta \sin \phi \cos \alpha^N_c, \quad \text{and}
\cos \theta^N_c = \cos \theta + \cos \phi.
\]

The corresponding azimuth, \( \alpha^N_c \) is calculated as follows. Again, using the law of cosines for a plane,

\[
\sin^2 \phi = \sin^2 \phi^N_c + \sin^2 \phi - 2 \sin \phi \sin \phi^N_c \cos \alpha^N_c.
\]

Substituting for \( \sin^2 \phi_c \),

\[
\sin^2 \phi^N_c = \sin^2 \phi^N_c + \sin^2 \phi - 2 \sin \phi^N_c \sin \phi \cos \alpha^N_c
+ \sin^2 \phi + 2 \sin \theta \sin \phi \cos \alpha^N_c, \quad \text{or}
\]

\[
2 \sin^2 \phi - 2 \sin \phi^N_c \sin \phi \cos \alpha^N_c + 2 \sin \theta \sin \phi \cos \alpha^N_c = 0.0.
\]

Simplifying and rearranging,

\[
\cos \alpha^N_c = (\sin \phi + \sin \theta \cos \alpha) / \sin \phi^N_c.
\]

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 $\theta_C^N$, $\alpha_C^N$ which are the exact slope and azimuth needed to reflect the ray. Thus,

$$\theta_\mu^N = \theta_C^N + \delta\theta, \quad \alpha_\mu^N = \alpha_C^N + \delta\alpha,$$

$$\theta_\lambda^N = \theta_C^N - \delta\theta, \quad \alpha_\lambda^N = \alpha_C^N - \delta\alpha,$$

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):

$$\sigma^2 = A\cdot w + B,$$

where

$$A = .0051, \quad B = .003, \quad w$$

is the wind speed in m/sec, and $\sigma^2$ 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 $\theta_1^N$ and $\alpha_1^N$. The increment of solid angle is integrated on a Lambert azimuthal equal area projection of the upper hemisphere, centered on the solar direction $\theta = \gamma; \quad \alpha = 0$.

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 $\beta_1$ for $\Delta TVS$ in the solid angle integration (Fig. 8a):

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

\[ \cos \theta_1^N = \cos \mu_1 \sin \gamma \sin \beta_1 + \cos \beta_1 \cos \gamma. \]

Substituting for \( \cos \beta_1 \), we obtain

\[ \cos \theta_1^N = \cos \mu_1 \sin \gamma \sin \beta_1 \\
+ \cos \gamma (\cos \gamma \cos \theta_1^N + \sin \gamma \sin \theta_1^N \cos \alpha_1^N). \]

Simplifying and dividing by \( \sin \gamma \), we produce

\[ \cos \mu_1 \sin \beta_1 = \frac{\cos \theta_1^N}{\sin \gamma} - \cos \gamma \left[ \cos \alpha_1^N \sin \theta_1^N \right. \\
+ \cos \theta_1^N \left. \frac{\cos \gamma}{\sin \gamma} \right] \]

\[ \cos \mu_1 \sin \beta_1 = \cos \gamma \sin \theta_1^N \cos \alpha_1^N - \frac{\cos \theta_1^N}{\sin \gamma} (\cos^2 \gamma - 1.0), \]

or

\[ \cos \mu_1 = \frac{(\cos \theta_1^N \sin \gamma - \cos \gamma \sin \theta_1^N \cos \alpha_1^N)}{\sin \beta_1}. \]

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 \( \beta \) and \( \mu \) where

\[
\begin{align*}
\text{Areas:} & \\
ABO & = \gamma \\
ADO & = \psi \\
DCO & = \frac{1}{2} \pi (2\beta) \delta \mu_1 \\
BCO & = x
\end{align*}
\]

with the index limits, \( x \) and \( y \), a function of each sector.
The integration of solid angle begins with the smallest values of $a_T^N$ and $\theta_I^N$. $a_T^N$ is increased first, followed by $\theta_I^N$. $a_T^N$ is now decreased, followed by $\theta_I^N$. 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 $\theta^N$ may be equal to zero, or the azimuth angle $a^N$ 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 ($\rho$), 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):

$$R = \rho \frac{P}{\delta \omega}.$$  

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 sunglint 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 sunglint 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


APPENDIX A

SUNGLINT PROGRAM LISTING

CCC*option trace on

PROGRAM SUNGLINT
C PROGRAMMER IRWIN RUFF
C PROGRAMMER T.L. TSUI
C
IMPLICIT REAL *8(A-H), REAL *8(O-Z)
C
COMMON /ABC/ IDATE(3),IPILOT,NUM1,NUM2,NUM3,NUM4,NUM5
REAL*4 PLAT,PLON,PLONT,PLATMAX,PLONW,PLATMN,PLONE
INTEGER TN,IGNX(3),LH
REAL *8 LN
REAL *4 DUMMY
CHARACTER CDATE*12,CIONX*12,CTN*4,CLH*12
EQUIVALENCE (IDATE,CDATE),(IQNX,CIONX),(TN,CTN)
PI=3.141592653589793/180
C
C INITIALIZE GAE PARAMETER BLOCK "sunglint"
C
CALL initblk("sunglint")
C
C*******************************************
C INPUT DATA FROM GAE PARAMETER BLOCK
C 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,'(a12)') lh
write(*,'(EDHEM= !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
IPLD=2
write(cdate,'(3i2)') iyr,imo,iday
write(CTN,'(2i2)') ihr,imin
C
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
<table>
<thead>
<tr>
<th>COLUMN</th>
<th>FMT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDATE</td>
<td>1-6</td>
<td>A6</td>
</tr>
<tr>
<td>SIA</td>
<td>7-10</td>
<td>F4.1</td>
</tr>
<tr>
<td>HT</td>
<td>11-14</td>
<td>F4.0</td>
</tr>
<tr>
<td>TN</td>
<td>16-19</td>
<td>A4</td>
</tr>
</tbody>
</table>
LONGITUDE OF EQUATOR CROSSING

LN 20 - 24 F5.1

LN LOCATION INDICATOR, E - EASTERN HEMI.
W - WESTERN HEMI.

LH 25 A1

LOCATION INDICATOR,
E - EASTERN HEMI.
W - WESTERN HEMI.

IPLOT 27 I1

>1 CONTOURS AND RADIANCE VALUES PLOTTED
=1 CONTOURS PLOTTED ONLY
<1 RADIANCE VALUES PLOTTED ONLY

NW 30 I1

NUMBER OF CHARTS DESIRED, MAX. 6

W 31 - 60 E5.1

W(I) IS THE WIND SPEED (M/SEC) ASSOCIATED
WITH EACH CHART.

RADIUS OF EARTH

R=6378.16DO

ORBITAL INCLINATION (EQUAL TO OR LESS THAN 90 DEGREES)

ORBINC = 180.DO-SIA

ETA IS NADIR, DELETA IS INTERVAL IT IS READ

DELETA= 5.0DO

STARTING POINT(LATITUDE) ON MAP ENTERED MANUALLY

NCNT=0

STARTING POINT

TAUI=0.0DO

INTERVAL REQUIRED

DELTAU=1.0DO

FIND THE SOLAR DECLINATION

READ (CDATE,7001) IQNX(1)
WRITE (CIQNX,7002) IQNX(1)
WRITE (CDATE,7003) CDATE,TN
IQNX(1)=IDIFDTG(CIQNX,CDATE)
SOL= FLOAT(IQNX(1))/24.DO/365.DO*360.DO*PI/180.DO
SOLDEC=23.5DO*DSIN(SOL)

FIND THE LONGITUDE OF EQUATOR CROSSING

EQCX=LN
IF(LH.EQ.1HW) EQCX=360.DO-EQCX

FIND THE SOLAR LONGITUDE

READ (CTN,7004) SOL,AMDA
SOL=SOL+AMDA/60.DO
SOL=DMOD(540.DO-SOL*15.DO,360.DO)
AMDA=EQCX-SOL+360.DO
AMDA=DMOD(AMDA,360.DO)

WRITE (6,6000) IDATE
WRITE (6,6001) HT,SIA
IQNX(1)=1HE
IF(SOL.LT.180.) GO TO 12
SOL=360.DO-SOL
IQNX(1)=1HW
WRITE (6,6002) SOLDEC,SOL,IGNX(1)
WRITE (6,6003) TN, LN, LH
WRITE (6,6004) AMDA

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)

WRITE (6,6006) SNINC,CSINC,SNDEC,CSDEC,SNLMDA,CSLMDA
CALL SLORBC(SNINC,CSINC,SNDEC,CSDEC,SNLMDA,CSLMDA,SNSLOT,CSSLLOT,
1SLOBLG)

WRITE (6,6006) SNSLOT,CSSLLOT,SLOBLG

DO 11 K=1,NW

WRITE (6,6005) K,WIND
NUM1=0
NUM2=0
NUM3=0
NUM4=0
NUM5=0

INITIALIZE PARAMETERS FOR SUBROUTINE SUBLTN - - SUBTRACT LAT LOAD

PHI = 0.0DO
AZ = SIA - 90.0DO
ALN = EQCX
SNDTAU = DSIN(DELTAU*(dacos(-1.0dO)/180.0dO))
CSTDUA = DCOS(DELTAU*(dacos(-1.0dO)/180.0dO))

IF(NCNT.EQ.0) THEN

TAUI=SLOBLG-30.0DO
IF(TAUI.LT.0.0DO) AZ=AZ+180.0DO
BCKFRCT=TAUI-INT(TAUI)

STEP A FRACTION OF AN INCREMENT ALONG SATELLITE TRACK

IF(ABS(BCKFRCT) .GT. 0.0) THEN

SNDTAU = DSIN(BCKFRCT*(dacos(-1.0dO)/180.0dO))
CSTDUA = DCOS(BCKFRCT*(dacos(-1.0dO)/180.0dO))
CALL SUBLTN(SNDTAU,CSTDUA,PHI,ALN,AZ,PHI2,ALN2,AZ2)
SNDTAU = DSIN(DELTAU*(dacos(-1.0dO)/180.0dO))
CSTDUA = DCOS(DELTAU*(dacos(-1.0dO)/180.0dO))
ENDIF
DO 100 I=0,INT(ABS(TAUI))
CALL SUBLTN(SNDTAU,CSTDUA,PHI,ALN,AZ,PHI2,ALN2,AZ2)
PHI=PHI2
ALN=ALN2
AZ=AZ2

100 CONTINUE
END IF
IF(TAUI.LT.0.0) AZ=AZ - 180.0
IFIN=60.0/DELTAU+1.0
TAU=TAUI
C C INITIALIZE IMAGE WINDOW TO MIN. CALC OF SUNGLINT AND RESTRICT CA
LC 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 C DO 13 I=1,IFIN
C ORIGITAL LONG DIF BETWEEN SUN AND SATELLITE
CSDOBG=DCOS((SLOBLG-TAU)*PI/180.DO)
CCC write(6,'(3(f8.2,2x))')SNDTAU,CSDTAU,PHI
CALL SUBLTN(SNDTAU,CSDTAU,PHI,ALN,AZ,PHI2,ALN2,AZ2)
ETA=-40.0
JFIN=80.0/DELETA+1.0
C C INITIALIZE CALL ELOC
JE=1
C 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.DO/PI)
C GEOCENTRIC ANGLE BETWEEN SATELLITE AND VP
SIG=THETRD-ETARD
SNSIG=DSIN(SIG)
CSSIG=DCOS(SIG)
CCC write(6,6006) SNSLOT,CSSLOT,CSDOOG,SNSIG,CSSIG,ZETA
CALL GEOMET(SNSLOT,CSSLOT,CSDOOG,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)
C
if(plon.lt.plonw) then
  plon=plon+360.0
else
  plon=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=100.000*RI

WRITE (6,6006) GETA,TAU,RI,TAUI,ORBINC,SNINC

IF(RI.GE.(.00100)) THEN
   CALL PLACE(REAL(GETA),REAL(TAU),REAL(RI),I,REAL(TAUI),
   REAL(ORBINC),REAL(SNINC),REAL(WIND),PLAT,PLON)
ENDIF

ask jim about the placement of je=je+1 should it be just before
call place inside if stmts or here???

JE=JE+1
ETA=ETA+DELETA
PHI=PHI2
AZ = AZ2
ALN = ALN2
TAU=TAU+DELTAU

WRITE (6,6006) GETA,TAU,RI,TAUI,ORBINC,SNINC

CALL PLACE(REAL(GETA),REAL(TAU),REAL(RI),2,REAL(TAUI),
REAL(ORBINC),REAL(SNINC),REAL(WIND))

CONTINUE

5000 FORMAT(2i3)
5010 FORMAT(61(13F6.1/))

6000 FORMAT(//' DATA = ',3A6/)
6001 FORMAT(//' SATELLITE HEIGHT = ',F6.0,' (KM), SATELLITE'\n1' 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 =',
1'F6.1,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)

7001 FORMAT(I2)
7002 FORMAT(I2,'032100')
7003 FORMAT(A6,A4)
7004 FORMAT(2F2.0)

END
SUBROUTINE OCGLNT(ZETA, THETA, GALPHA, W, RI)
IMPLICIT REAL *B(A-H), REAL *B(D-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(-1.0DO)
HPI = PI/2.0DO
RPD = PI/180.0DO
C ANGINC = ANGULAR INCREMENT, IN RADIANS, USED IN CALCULATING
C SLOPE PROBABILITY AND SOLID ANGLE OF REFLECTED BEAM
C
ANGINC = 0.01DO
C STPNMB = NUMBER OF STEPS PER SIDE USED IN INTEGRATION
C FOR SOLID ANGLE
C
STPNMB = 20.0DO
C IF SUN IS BELOW HORIZON, SKIP COMPUTATION
IF(ZETA.LT.90.0)GO TO 8
RI = 0.0DO
GO TO 5
8 SNZET = DSIN(ZETA*RPD)
CSZET = DCOS(ZETA*RPD)
SNSOZT = SNZET*SNZET
SNTHET = DSIN(THETA*RPD)
CSTHET = DACOS(THETA*RPD)
SNSTHET = SNTHET*SNTHET
C NEXT 2 STEPS CONVERT GALPHA TO ALPHA, A POSITIVE ANGLE
C BETWEEN 0 AND 180 DEGREES. (PERMISSIBLE SINCE REFLECTION
C PATTERN IS SYMMETRICAL ABOUT 0-180 LINE).
ALPHA = DABS(GALPHA)
C
IF(ALPHA.GT.180.0) ALPHA = 360.0DO-ALPHA
CSALF = DCOS(ALPHA *RPD)
C SIGSO = VARIANCE OF THE CIRCULAR GAUSSIAN DISTRIBUTION OF
C THE NORMALS OF THE SLOPES OF THE OCEAN SURFACE
SIGSO = 0.00512DO*W+0.003DO
CS2BET = CSZET*CSTHET+SNZET*SNTHET*CSALF
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(0.5DO*(1.0DO+CS2BET))
C IF CSBET IS NEGATIVE, SLOPE IS SHADOWED
IF(CSBET.GT.0.0) GO TO 9
RI = 0.0DO
GO TO 5

C RHO = REFLECTION COEFFICIENT (APPROXIMATION TO FRESNEL LAW)
9 RHO = DEXP(-6.09DO*CSBET)+0.019DO*CSBET
SRT = DSQRT(SNSGZT+SNSGTH+2.0*SNZET*SNTHET*CSALF)

C TNTHN = SEA SURFACE SLOPE (TANGENT OF ZENITH ANGLE OF SURFACE
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
SLOPE IS ZERO, AND THE SURFACE NORMAL IS VERTICAL. THE
AZIMUTH OF THE NORMAL, ALFNC, IS UNDEFINED, AND SRT=0.
ALFNC IS DEFINED TO BE ZERO IN THIS CASE BY STATEMENT 10,
SINCE ALPHA FOR THE SPECULAR POINT IS 180 OR, FOR AN OVERHEAD
SUN, 0.
IF(SRT.EQ.0.0) GO TO 10
CSALNC = (SNZET + SNTHET*CSALF)/SRT

C 10 IF(ALPHA.EQ.0.0.OR.ALPHA.EQ.180.0) CSALNC = 1.0DO
IF(ALPHA.EQ.180.0.OR.THETA.GT.ZETA) CSALNC = -1.0DO
IF(ALFNC.GT.1.0) CSALNC = 1.0DO
IF(ALFNC.LT.-1.0) CSALNC = -1.0DO
ALFNC = DACOS(CSALNC)

C NEXT SECTION SETS UP THE ANGULAR REGION OVER WHICH
DETERMINATION OF SLOPE PROBABILITY AND SOLID ANGLE
ARE MADE
C L AND U AT END OF NAMES REFER TO LOWER AND UPPER LIMITS
RESPECTIVELY. THTN=ZENITH ANGLE OF SLOPE NORMAL, AND
ALFN IS ITS AZIMUTH. THTNL AND ALFNL CANNOT BE NEGATIVE,
THTNU CANNOT EXCEED PI/2, AND ALFNU CANNOT EXCEED PI.

C THTNL = THTNC - ANGINC
IF(THTNL.LT.0.0) THTNL = 0.0DO
THTNU = THTNC + ANGINC
IF(THTNU.GT.HPI) THTNU = HPI
DTHTN = 0.5DO*(THTNU-THTNL)
ALFNL = ALFNC - ANGINC
IF(ALFNL.LT.0.0) ALFNL = 0.0DO
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
P = (DEXP(-TL*TL/SIGSO)-DEXP(-TU*TU/SIGSQ)) *DALFN/PI

C THE REMAINDER OF THE PROGRAM, THROUGH STATEMENT 3, CALCULATES
THE SOLID ANGLE INTO WHICH RADIATION IS REFLECTED BY
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 + 1.0D0
C
C SPThN AND SPALN ARE STEP INTERVALS OF NORMAL THETA AND ALPHA
C SPThN = 2.0D0*DTHTN/STPNMB
C SPALN = 2.0D0*DALFN/STPNMB
1421 FORMAT(’ HERE’)
C SGN = 1.0D0
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
C OMEGA=SOLID ANGLE INTO WHICH RADIATION IS REFLECTED
C OMEGA = 0.0D0
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 = 1.0D0
C IF(I.EQ.2.AND.J.EQ.NSTP.AND.ALFNL.EQ.0.0) CSALST = 1.0D0
C IF(I.EQ.1.AND.J.EQ.NSTP.AND.ALFNU.EQ.PI) CSALST = -1.0D0
C IF(I.EQ.2.AND.J.EQ.1.AND.ALFNU.EQ.PI) CSALST = -1.0D0
C
C SLDANG CALCULATES THE INCREMENT OF SOLID ANGLE
C PRINT *,’ CHECK SLDANG1 ’,SNZET,CSZET,
CALL SLDANG(SNZET,CSZET,SNTHST,CSTHST,CSALST,J,OMEGA)
PRINT *,' CHECK SLDANG1 ',SNZET,CSZET,
xnths,t,csthy,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) THST = THTN
IF(ALFNU.EQ.PI.AND.I.EQ.1) SNTHST = DSIN(THST)
IF(ALFNU.EQ.PI.AND.I.EQ.1) CSTHST = DCOS(THST)
IF(ALFNU.EQ.PI.AND.I.EQ.1) GO TO 21

11 THST = THST- SGN*SPTHN
DO 2 J=1,NSTP
THST = THST + SGN*SPTHN
SNTHST = DSIN(THST)
CSTHST = DCOS(THST)
PRINT *,' CHECK SLDANG2 ',SNZET,CSZET,
xnths,t,csthy,csalst,j,omega
CALL SLDANG(SNZET,CSZET,SNTHST,CSTHST,CSALST,J,OMEGA)
PRINT *,' CHECK SLDANG1 ',SNZET,CSZET,
xnths,t,csthy,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 AMU=AZIMUTH AT PROJECTION CENTER FROM DIRECTION TO VERTICAL
C (IF SUN IS VERTICAL, AMU MEASURED FROM LINE ALPHA=180)
C 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 = 1.0DO
IF(CSALST.EQ.1.0.AND.SNTHST.LE.SNZET) CSMU = 1.0DO
IF(CSALST.EQ.1.0.AND.SNTHST.GT.SNZET) CSMU = -1.0DO
AMU = DACOS(CSMU)
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 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 +0.5DO*RAD*RAD*DMU
C STORE SNBT AND AMU FOR USE IN NEXT INCREMENT
C 1 STRB = SNBT
STRM = AMU
RETURN
END

A-10
SUBROUTINE GEOMET(SNSLOT, CSSLOT, CSDOOG, 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+CSSLOT*CSSIG*CSDOOG
     SNZET=DSQRT(1.0D0-CSZET*CSZET)
     ZETA=DACOS(CSZET)*RTD
     CSALF=(SNSLOT*CSSIG-CSSLOT*SNSIG*CSDOOG)/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 SLORBC(SNINC, CSINC, SNDEC, CSDEC, SNLMDA, CSLMDA, SNSLOT, CSSLOT, SLOBLG)
IMPLICIT REAL *B(A-H), REAL *B(O-Z)
SAVE
C PRELIES-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/ACOS(-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=ACOS(CSSLOG)*RTD
IF(SNSLOG.LT.0.0)SLOBLG=360.0D0-SLOBLG
RETURN
END
SUBROUTINE SUBLTLN(SNDTAU, CSDTAU, PHI, ALN, AZ, PHI2, ALN2, AZ2)
IMPLICIT REAL *8(A-H), REAL *8(D-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(CSDTAU*SNPHI + SNDTAU*CSPHI*DCOS(AZR))
DELLN = DACOS((CSDTAU - SNPHI*DSIN(PHI2))/(CSPHI*DCOS(PHI2)))*R2D
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

written by a.b. caughey 18 aug 1988

subroutine to access gae parameter block

integer xnullstring, xprmdim, xcont, xsuccess, xci, xcs, xsave
include 'subdef.fin'
parameter (xprmdim=1000, xsuccess=1, xcont=2)
integer block(xprmdim)

character cmsg*72, creply*72
integer imsg(18)
integer reply(18)
equivalence (cmsg, imsg), (creply, reply)

parameter (max_word=1024)
integer*2 ibuf(max_word)
integer cmdblk(80)
integer factor, rlen, band, type, x, name(6), strg(7)
character cname*24, cstrg*28
equivalence (cname, name)
real y
integer idsrn, odsrn
integer output
logical ok

integer cnm(3), unitno
character*28 blknam
integer ipackos(8)
save block

***get parameter block from tm

call xrinim(block, xprmdim, xcont, istatus)
if(istatus.ne.xsuccess) then
13 format('xrinim error, status = ', i6)
write(cmsg, 13) istatus
call printp(1, imsg, 72, 0, 0, 0, 0, 0, 0)
goto 1000
endif return

***get parameter block from disk file

c entry initblk(blknam)
lun=12
call xsch2p(blknam, ipackos)
call xrrdb(ipackos, lun, block, xprmdim, 1, istatus)
if(istatus.ne.xsuccess) then
write(cmsg, ('xrrdb error, status = ', i6)) istatus
call printp(1, imsg, 72, 0, 0, 0, 0, 0, 0)
goto 1000
endif
***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,imsg,72,0,0,0,0,0,0,0)
    goto 1000
  endif
  return
```

*** open i/o files

```c
entry fileop(cnm,unitno)
  call xrstrp(block,cnm,10,name,istatus)
  if(istatus.ne.xsuccess) then
    format('xrstrp error: parameter = name, status = ',i6)
    write(cmsg,21) istatus
    call printp(1,imsg,72,0,0,0,0,0,0,0)
    goto 1000
  endif
  inquire(file=cname,exist=ok)
  if(.not.ok .and. (unitno.eq.5 .or. unitno.eq.1)) then
    format('input file does not exist')
    write(cmsg,31)
    call printp(1,imsg,25,0,0,0,0,0,0,0)
    goto 1000
  endif
  open(unit=unitno,file=cname)
```

*** get real parameter

```c
entry realgae(cnm,xx)
  call xrreal(block,cnm,1,xx,istatus)
  if(istatus.ne.xsuccess) then
    format('xrreal error: parameter = x, status = ',i6)
    write(cmsg,19) istatus
    call printp(1,imsg,72,0,0,0,0,0,0,0)
    goto 1000
  endif
  return
```

*** get integer parameter

```c
entry intgae(cnm,ix)
  call xrintg(block,cnm,1,ix,istatus)
  if(istatus.ne.xsuccess) then
    format('xrintg error: parameter = ix, status = ',i6)
    write(cmsg,17) istatus
```
call printp(1,imsg,72,0,0,0,0,0,0,0,0)
goto 1000
endif
return

*** get string parameter

entry strgae(cnm,cstrg)
call xrstrp(block,cnm,7,strg,istatus)
if (istatus.ne.xsuccess) then
    write(cmsg,'(xrstrp error: parameter = ',3a4,' status = ',i6)')
    (cmn(i),i=1,3),istatus
    call printp(1,imsg,72,0,0,0,0,0,0,0,0)
goto 1000
endif
call xsp2ch(strg,cstrg)
return

*** write a real parameter to parameter block

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,imsg,72,0,0,0,0,0,0,0,0)
goto 1000
endif
return

*** write a integer parameter to parameter block

entry wrtint(cnm,ix)
call xqintg(block,cnm,1,ix,103,istatus)
if (istatus.ne.xsuccess) then
    write(cmsg,'(xqintg error: parameter = ',3a4,', status= ',i6)')
    (cmn(i),i=1,3),istatus
    call printp(1,imsg,72,0,0,0,0,0,0,0,0)
goto 1000
endif
return

1000 stop
end
SUBROUTINE PLACE(X,Y,RI,KEY,TAUI,ORBINC,SNINC,W,PLAT,PLON)
SAVE
COMMON /ABC/ IDATE(3),IPlot,L(5)
DIMENSION VAL(5,800,4),XB(4),XT(4)
CHARACTER*12 refifile output file for reflectance
DATA LL,ICNT/1,0/
MAXIMUM SIZE OF OUTPUT ARRAY
DATA MXMAX,MYMAX/61,17/

IF(LL.EQ.0)GO TO 101
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
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

4 XM=0.
DO 16 KK=1,4
IF(L(KK).EQ.0)GO TO 16
XM=XJM+1.
KL = 5 - KK
16 CONTINUE

PLOT THE VALUES
MM=XMJ

A-17
SIZE=0.14

output file opened in subroutine initgae
call fileop(10h/reflfile/,9)
rewind 9
ICOUNT=1

DO 36 KK=1,MM
NUMV=L(KK)
XK=KK-1
A=XK*.546875+.5

DO 36 I=1,NUMV

CONVERTS 65.68 TO 0. VALUES IN X DIRECTION TO 1. TO 4.546875*XK+1.

POSITION IN X DIRECTION = (VAL(1,I,KK)-XT(KK))*4.3786*0.0692269*A

CONVERTS -40 TO +40 VALUES IN Y DIRECTION

YNUM=(VAL(2,IKK)+40.)*.096875 + 1.

FVAL=VAL(3,I,KK)*10.
MX=(XNUM-1.)/.303125 + 1.2
NY=(YNUM-1.)/.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'',3,2(I,13),2(I,17:3d1X,I16))':
        ICOUNT,MX,NY,LAT,LON,REAL(FVAL),REAL(FVAL),icount
endif
ICOUNT=ICOUNT+1
IF(IPLT.LT.1) GO TO 36

ICHAR=FVAL+.5
FVAL=ICHAR
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*ICHAR*0.84*0.5

CONTINUE
CLOSE (9)
CLEAR OUTPUT ARRAY

DO 3000 IX=1,3
   DO 2000 IY=1,800
      DO 1000 IZ=1,4
         VAL(IX,IY,IZ)=0.000
   1000 CONTINUE
   2000 CONTINUE
3000 CONTINUE
RETURN

FORM(6H *****,'NOT ENOUGH POINTS TO PLOT',5H*****)

END
SUBROUTINE ELOC(IEPH, SCANG, DALT, PHI, PHI2, ALN, ALN2,
   + PLAT, PLON, IERR)

!EARTH LOCATE

SAVE

C==========================================================================================================
C (FORMERLY NAMED ELOCAT)
C ELOCAT WAS BASED ON ROUGHLY THE SECOND HALF OF THE PROGRAM CELCAL 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 THE ASSUMED CENTER OF THE SCAN AREA ON THE EARTH.
C          IN DEGREES, + NORTHERN HEMISPHERE, - SOUTHERN)
C PLON - LONGITUDE OF PRINCIPAL POINT. (DEGREES, + EASTWARD 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 HEMISPHERE, - SOUTHERN.
C ELON - SUBSATELLITE LONGITUDE, DEGREES, + EASTWARD FROM GREENWICH RANGING 0 TO 360
C EALT - SATELLITE ALTITUDE ABOVE ELLIPSOID (KM)
C (THE LAST FOUR VALUES ARE PROVIDED BY CALLS TO DATA PAGING ROUTINES)
C==========================================================================================================
C
PARAMETER (NSP=17)
PI=ACOS(-1.)
PI0V2=PI/2.0
TWOPI=2.0*PI
RADDEG=180.0/PI
DEGRAD=PI/180.0
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=SINCSA)
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*SC
A=ACOS(A)*RADDEG !*****THIS WAS ALTERED PER JIM CLARK*****
50 IF(A.LE.360.)GO TO 60
A=A-360.
GO TO 50
60 IF(A.GE.360.)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).
A-29
IF(SCANG)90,280,100
    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    RKDN=.0001569612*SPALT+1.
    ZANG=ASIN(RKDN*SIN(TANAD))
C  TEST FOR EARTH TANGENCY
CNCAL=ASIN((1./RKDN))
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
150    IF(SPLAT.LE.0.)GO TO 150
    CNOS=0.
    SIDEB=PIOV2-SPLAT
    GO TO 160
170    IF(AA-PI)180,290,200
180    CAPC=AA
    WOE=1.
190    IF(CNOS)370,220,190
    CAPC=PI-CAPC
    GO TO 220
200    WOE=0.
210    CAPC=TWOPI-AA
    IF(CNOS)370,220,210
    SINA=SIN(SIDEA)
220    SINB=SIN(SIDEB)
    COSC=COS(CAPC)
    TEMP2=SINB*SINA*COSC
    COSB=COS(SIDEB)
    COSA=COS(SIDEA)
    SIDE=ACOS(TEMP2+COSB*COSA)
    SINC=SIN(CAPC)
    SINEC=SIN(SIDE)
    CAPA=ASIN(SINA*SINC/SINEC)
C  SEE IF CAPA IS IN THE SECOND QUADRANT.
    COST=COSB*COS(SIDE)
    IF(COST.GE.COSA)CAPA=PI-CAPA
C  NOW DETERMINE PRINCIPAL POINT LATITUDE AND LONGITUDE.
PLAT=PIOV2-SIDEC
IF(CNOS)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    A-22
C       SET DUMMY VALUES
       GO TO 380
C       MISCELLANEOUS ERRORS
 370    IERR=5
 380    RETURN
       END
SUBROUTINE IDIFDTG

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

INPUT BY THE USER: NONE

OUTPUT: IDIFDTG

THIS SUBROUTINE IS CALLED BY PROGRAM SUNGLINT

THIS SUBROUTINE CALLS: NONE

VARIABLES

INPUT
  DTG1  FIRST LEFT-JUSTIFIED DATE-TIME GROUP  CHAR 00000000 -
        99123123
  DTG2  SECOND LEFT-JUSTIFIED DATE-TIME GRP  CHAR 00000000 -
        99123123

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

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

FUNCTION IDIFDTG(DTG1,DTG2)

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

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

DO 100 I=1,2
  READ(CDTG(I),*001) IYR, IMO, IDAY, IHR
  IF(MOD(IYR,4).EQO) THEN
    A-24
ADD IN YEARS
IHRS(I)=IYR*366*24
IMOS(2)=29
ELSE
IHRS(I)=IYR*365*24
IMOS(2)=28
ENDIF
ADD IN MOS
DO 50 J=1,IMO
   IHRS(I)=IHRS(I) + IMOS(J)*24
50 CONTINUE
ADD IN DAYS AND HOURS
IHRS(I)=IHRS(I) + (IDAY*24) + IHR
100 CONTINUE

TIME DIFFERENCE IN HOURS BETWEEN IDTG1 & IDTG2
IDIFDTG=IHRS(2)-IHRS(1)
RETURN
9001 FORMAT(4I2)
END
SUBROUTINE INTERP(UVFLD, WLAT, WLON, FLAT, FLON, GSPD)

UVFLD -- U FIELD IN UVFLD(I,J,1) M/S
V FIELD IN UVFLD(I,J,2) M/S
WLAT -- LAT OF POINT TO BE INTERPOLATED TO DEGREES
WLON -- LONG OF POINT TO BE INTERPOLATED TO DEGREES
FLAT -- MAXIMUM LAT OF UV GRID FIELD DEGREES
FLON -- MAXIMUM LONG OF UV GRID FIELD DEGREES
GSPD -- WIND SPEED M/S

REAL UVFLD(73,144,2), GOBS(2)
PARAMETER (NREP=2, GRDLONSP=2.5, GRDLATSP=2.5)
PARAMETER (DLT=2.5, DLN=2.5)

DLT & GRDLATSP and DLN & GRDLONSP should be equivalenced!!!
EQUIVALENCE (DLT,GRDLATSP), (DLN,GRDLONSP)

INTERPOLATE TO OBSERVATION POINT

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)

   PERFORM BILINEAR INTERPOLATION

   GOBS(K) = VIJ + (VIJP - VIJ)*A + (VIPJ - VIJ)*B
& + (VPP - VIJ - VIPJ - VPP)*A*B

100 CONTINUE

GSPD = (SQRT(GOBS(1)**2 + GOBS(2)**2))

RETURN
END
APPENDIX B

SUNGLINT PROGRAM RUNNING INSTRUCTIONS

The Sunglint algorithm is implemented in Global Imaging's executive, operating on NEPRF's HP9020. There are two modes in which to run the sunglint 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 sunglint isopleths on a navigated image, from Global Imaging's executive, type "tutor SUN". The following parameters must be assigned values:

PARAMETERS

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>Input image name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>Length</td>
<td>12</td>
</tr>
<tr>
<td>Count</td>
<td>1</td>
</tr>
<tr>
<td>Default</td>
<td>&quot;NONE&quot;</td>
</tr>
</tbody>
</table>

Gfile Flag "Y" is set if a navigational file is available for the input image.

<table>
<thead>
<tr>
<th>Type</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>1</td>
</tr>
<tr>
<td>Default</td>
<td>&quot;Y&quot;</td>
</tr>
<tr>
<td>Valid</td>
<td>(&quot;Y&quot;,&quot;N&quot;)</td>
</tr>
</tbody>
</table>

WIND Wind field file name.

<table>
<thead>
<tr>
<th>Type</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>12</td>
</tr>
<tr>
<td>Count</td>
<td>1</td>
</tr>
<tr>
<td>Default</td>
<td>&quot;INPUT.DAT&quot;</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>Input image name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>Length</td>
<td>12</td>
</tr>
</tbody>
</table>
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

HOUR

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
<table>
<thead>
<tr>
<th>ASST. FOR ENV. SCIENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASST. SEC. OF THE NAVY (R&amp;D)</td>
</tr>
<tr>
<td>ROOM 8731, THE PENTAGON</td>
</tr>
<tr>
<td>WASHINGTON, DC 20350</td>
</tr>
<tr>
<td>OFFICE OF NAVAL RESEARCH</td>
</tr>
<tr>
<td>CODE 1122AT, ATMOS. SCIENCES</td>
</tr>
<tr>
<td>ARLINGTON, VA 22217-5000</td>
</tr>
<tr>
<td>OFFICE OF NAVAL RESEARCH</td>
</tr>
<tr>
<td>ATTN: PROG. MANAGER, 1122CS</td>
</tr>
<tr>
<td>ARLINGTON, VA 22217-5000</td>
</tr>
<tr>
<td>OFFICE OF NAVAL RESEARCH</td>
</tr>
<tr>
<td>CODE 1122 PO, PHYS. OCEANO.</td>
</tr>
<tr>
<td>ARLINGTON, VA 22217-5000</td>
</tr>
<tr>
<td>OFFICE OF NAVAL TECHNOLOGY</td>
</tr>
<tr>
<td>ONR (CODE 22)</td>
</tr>
<tr>
<td>800 N. QUINCY ST.</td>
</tr>
<tr>
<td>ARLINGTON, VA 22217-5000</td>
</tr>
<tr>
<td>CHIEF OF NAVAL RESEARCH</td>
</tr>
<tr>
<td>LIBRARY CODE 0122L</td>
</tr>
<tr>
<td>800 QUINCY ST.</td>
</tr>
<tr>
<td>ARLINGTON, VA 22217-5000</td>
</tr>
<tr>
<td>OFFICE OF NAVAL RESEARCH</td>
</tr>
<tr>
<td>ENV. SCI. PROGRAM, CODE 112</td>
</tr>
<tr>
<td>ARLINGTON, VA 22217-5000</td>
</tr>
<tr>
<td>OFFICE OF NAVAL RESEARCH</td>
</tr>
<tr>
<td>ATTN: HEAD, OCEAN SCI. DIV</td>
</tr>
<tr>
<td>CODE 112</td>
</tr>
<tr>
<td>ARLINGTON, VA 22217-5000</td>
</tr>
<tr>
<td>OFFICE OF NAVAL RESEARCH</td>
</tr>
<tr>
<td>CODE 1122 MM, MARINE METEO.</td>
</tr>
<tr>
<td>ARLINGTON, VA 22217-5000</td>
</tr>
<tr>
<td>CHIEF OF NAVAL OPERATIONS</td>
</tr>
<tr>
<td>OP-928</td>
</tr>
<tr>
<td>U.S. NAVAL OBSERVATORY</td>
</tr>
<tr>
<td>WASHINGTON, DC 20330</td>
</tr>
<tr>
<td>OFFICER IN CHARGE</td>
</tr>
<tr>
<td>NAVOCEANCOMDET</td>
</tr>
<tr>
<td>AFPWC</td>
</tr>
<tr>
<td>OFFUTT AFB, NE 68131</td>
</tr>
<tr>
<td>COMMANDING OFFICER</td>
</tr>
<tr>
<td>NAVAL RESEARCH LAB</td>
</tr>
<tr>
<td>ATTN: LIBRARY, CODE 2820</td>
</tr>
<tr>
<td>WASHINGTON, DC 20360</td>
</tr>
<tr>
<td>OFFICE OF NAVAL RESEARCH</td>
</tr>
<tr>
<td>SCRIPPS INSTITUTION OF OCEANOGRAPHY</td>
</tr>
<tr>
<td>LA JOLLA, CA 92037</td>
</tr>
<tr>
<td>COMMANDING OFFICER</td>
</tr>
<tr>
<td>NAVAL OCEAN RBCH &amp; DEV ACT</td>
</tr>
<tr>
<td>JCSSC, MS 36828-5004</td>
</tr>
</tbody>
</table>

| COMMANDER, NAVAL OCEANOGRAPHY COMMAND |
| STEENHUIS SPACE CENTER |
| JCSSC, MS 36829-5000 |
| COMMANDING OFFICER |
| NAVOCEANSCEN |
| ATTN: CODE NS |
| JCSSC, MS 36829-5000 |
| COMMANDING OFFICER |
| NAVAL OCEANOGR. OFFICE |
| JCSSC, MS 36822-5001 |
| COMMANDING OFFICER |
| FLENUMOCEANEN |
| MONTEREY, CA 93943-4005 |
| SUPERINTENDENT |
| LIBRARY REPORTS |
| U.S. NAVAL ACADEMY |
| ANNAPOLIS, MD 21402 |
| DIRECTOR OF RESEARCH |
| U.S. NAVAL ACADEMY |
| ANNAPOLIS, MD 21402 |
| NAVPOSQSCOL |
| METEOROLOGY DEPT. |
| MONTEREY, CA 93943-5000 |
| COMMANDER, SPAWARSYSNCOM |
| ATTN: CODE PMW 141-B1 |
| 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 |
| NAVYBSYSCOM |
| ATTN: LIBRARY (AIR-7220) |
| WASHINGTON, DC 20351-0001 |
| COMMANDER |
| NAVYBSYSCOM, CODE 526W |
| WASHINGTON, DC 20351-0001 |
| COMMANDER, NAVAL SURFACE WARFARE CENTER |
| DAHLGREN, VA 22448-5000 |
| DIRECTOR |
| NAVSURFWARREN, WHITE OAKS |
| NAVY SCI. ASSIST. PROGRAM |
| SILVER SPRING, MD 20930-5000 |

| COMMANDER |
| NAVOCEANSCEN |
| DR. J. RICHTER, CODE 54 |
| SAN DIEGO, CA 92152-5000 |
| COMMANDING OFFICER |
| AMPHIBIOUS WARFARE LIBRARY |
| NAVAL AMPHIBIOUS SCHOOL |
| NORFOLK, VA 23521 |
| APOFFY |
| HANSCOM AFB, MA 01731 |
| USAFACUS/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 |
| 1200 S. DIXIE HWY |
| CORAL GABLES, FL 33146 |
| DIRECTOR, NOAA |
| ATLANTIC MARINE CENTER |
| COAST & GEODNETIC SURVEY |
| 499 W. YORK ST. |
| NORFOLK, VA 23510 |
| FISHERIES LIBRARY |
| NOAA |
| 75 VIRGINIA BEACH DRIVE |
| MIAMI, FL 33140 |
| DIRECTOR, ATLANTIC OCEANO. & METEOROLOGY LABS |
| 15 ROKENBACKER CAUSEWAY |
| VIRGINIA KEY |
| MIAMI, FL 33140 |
| 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 |
| TECHNOLOGIES DEVELOP. LAB |
| GRAMAX BLDG. |
| 8080 15TH 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. RUH |
| BATON ROUGE, LA 70803 |
| AMERICAN METEORO. SOCIETY |
| MET. & GEOASTRO. ABSTRACTS |
| P.O. BOX 1708 |
| WASHINGTON, DC 20013 |
| DIRECTOR, METEO. & OCEANO. |
| NATIONAL DEFENSE HQD. |
| OTTAWA, ONTARIO, KIA OK2 |
| CANADA |
| DIRECTOR OF NAVAL OCEANO. & METEOROLOGY |
| MINISTRY OF DEFENCE |
| LACON HOUSE, THEOBOLD RD. |
| LONDON WC 1B9RY, ENGLAND |
| LIBRARY, INSTITUTE OF OCEAN SCIENCES |
| ATTN: DIRECTOR |
| WORMLEY, GODALMING |
| SURRY GU5 9UB, ENGLAND |