NASA CR-143809

(NASA-CR-143809) DETERMINATION OF THE CENTER OF BRIGHTNESS OF THE LUNAR CRESCENT Final Report (Ball Bros. Research Corp.) 98 p HC $4.75 CSCL 22A

N75-24785

Unclas

G3/13 24220

BALL BROTHERS

RESEARCH CORPORATION

LABORATORIES IN BOULDER, COLORADO AND MUNCIE, INDIANA

Ball

BR 139
(2-64)

LUNAR SENSOR STUDY

FINAL TECHNICAL REPORT FOR
ASSIGNMENT 014
Contract NAS5-23185

20 January 1975

PREPARED BY:

Ball Brothers Research Corporation
P.O. Box 1062
Boulder, Colorado

PREPARED FOR:

Goddard Space Flight Center
Greenbelt, Maryland

1.0 INTRODUCTION

This task is the fourteenth in the series accomplished within the scope of the Work Statement, Attachment A, Item 5 of Contract NAS5-23185.

2.0 SCOPE

This task was issued to perform the following analysis and associated effort:

Perform suitable analysis, assuming Lambertian surface, to determine required sensor offset in a lunar tracking rocket flight. The offset is defined as the angular difference, in arc minutes, between the geometrical center of the Moon and the sensor null. The following shall be furnished:

- A computer program which will provide the offset, in arc minutes, as a function of the following lunar variables.

  a) Fraction illuminated

  b) Apparent size of Moon

- A technical note documenting the analysis and describing the computer program.

3.0 FINANCIAL SUMMARY

The following is the final manhour and cost totals for this assignment.

	Manhours	Labor Dollars (Unburdened)	MODC (Unburdened)	Total Costs W/O Fee
:al Budget	295	2,120	459	6,482
:ual Total	361	2,654	433	8,549

4.0 CONCLUSIONS

The analysis and results are summarized in the Final Technical Report F74-13 dated 31 December 1974 attached as Appendix A.

The overrun in the manhours and total cost were primarily due to more than estimated time to complete the final report and the tests requested by GSFC that were not included in the original scope of the program. The tests consisted of moon intensity measurements made using the lunar sensor to observe the moon. This data was used to calibrate the sensor for the actual launch.

BALL BROTHERS RESEARCH CORPORATION

BOULDER, COLORADO

Final Report

F 74-13
31 December 1974

Determination of the Center of Brightness of the Lunar Crescent

Contract NAS5-23185 Task 014

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Goddard Space Flight Center
Greenbelt, Maryland

Prepared by:

Rocket and Balloon Pointing Systems
Ball Brothers Research Corporation
Boulder, Colorado

Charlie Rose
Charlie Rose (Author)
Systems Analyst

Marv Greeb
Marv Greeb
Project Manager

marda barthuli
Marda Barthuli (Author)
Programmer

FORM BR-72A

Table of Contents

Section I

INTRODUCTION

This final report is submitted to Goddard Space Flight Center in fulfillment of a task intended to allow the center of brightness of the lunar crescent to be computed. This capability was required to support sounding rocket launches from Kauai, Hawaii (3 November 1974) and White Sands Missile Range (28 December 1974).

This paper discusses, briefly, the operational characteristics of the lunar sensor which was used to point the sounding rocket, develops the associated mathematical model of the system, describes the computer programs which were written to implement the model, and presents data pertinent to the two launches.

The work was performed under NASA Contract NAS5-23185, Task Assignment 014.

Section II

DESCRIPTION OF THE PROBLEM

An angular error sensor has been built by Ball Brothers Research Corporation for the purpose of pointing a sounding rocket payload at the moon. The sensor has an essentially linear transfer characteristic in the vicinity of null and angular position-errors of the vehicle are manifested as proportional error signals which can be used by the rocket control system to reduce these errors to zero.

The application for which the sensor was designed necessitated pointing the payload at the center of the moon; that is, the center of the disk which would be visible from the earth at full moon.

The sensor is an electro-optical device of the "energy-balance" variety. Such sensors split their fields of view into two parts and generate error signals which are proportional to the difference of energies gathered by each half. Thus, such a sensor produces a zero error signal when it is pointed at the center of intensity of the source.

For the case at hand the source is, of course, the moon and for conditions other than full moon, the center of intensity will not coincide with the selenographic center. It is necessary to know just where the sensor will be pointing when it is nulled so appropriate biases can be applied to correct the situation.

The problem, then, is to determine where the center of intensity of the visible crescent of the moon is with respect to the selenographic center. In order to compute this quantity, it is necessary to know the realtive positions of the sun, moon, and sensor, the reflective characteristics of the lunar surface, and the optic-operational basis of the sensor.

This report is devoted to constructing a mathematical model of the situation, discussing the computer program which implements the model, and presenting results applicable to actual flights.

Section III

MATHEMATICAL MODEL

3.0 ASSUMPTIONS

The simplifying assumptions upon which the mathematical model is based are as follows:

1. The moon's albedo is <u>not</u> a function of wavelength. That is, the incident and reflected energies have the same spectral distribution.

2. The moon's albedo is <u>not</u> a function of selenographic latitude and longitude.

3. The sun is considered, for illumination purposes, to be an infinitely distant point source. Thus, the radiant energy is constant in the vicinity of the moon and, since there is no penumbra, the terminator is sharply defined.

3.1 Basic Concepts

The lunar sensor is a two axis device and, as such, produces error signals about its yaw and pitch axes. To see how these signals are generated, consider the operation of just one of the channels; e.g., yaw.

A plane may be associated, ficticiously, with the yaw channel. This plane is defined by the yaw axis and optical axis of the sensor and it contains these axes. For want of a better term, we can call it the "yaw-null plane". When the sensor is pointed at a source of radiant energy, this plane effectively divides the source into two parts. The part lying on the "left" side of the plane is converted into one proportional voltage while the part lying on the "right" side of the plane is converted into another proportional voltage. These voltages are differenced to produce the final error signal which is used by the control system.

Exactly the same situation applies to the orthogonal pitch axis and the origin of the term "energy balance sensor" becomes obvious. This concept of the sensor's operation is essential to understanding why the mathematical model is constructed the way it is.

3.2 Coordinate Systems

Figure 3-1 shows the basic elements of the system; the sensor together with its yaw-null plane, the moon, a coordinate system (with axes labeled 1, 2, 3) with its origin at the center of the moon, and a vector, S, pointing towards the sun.

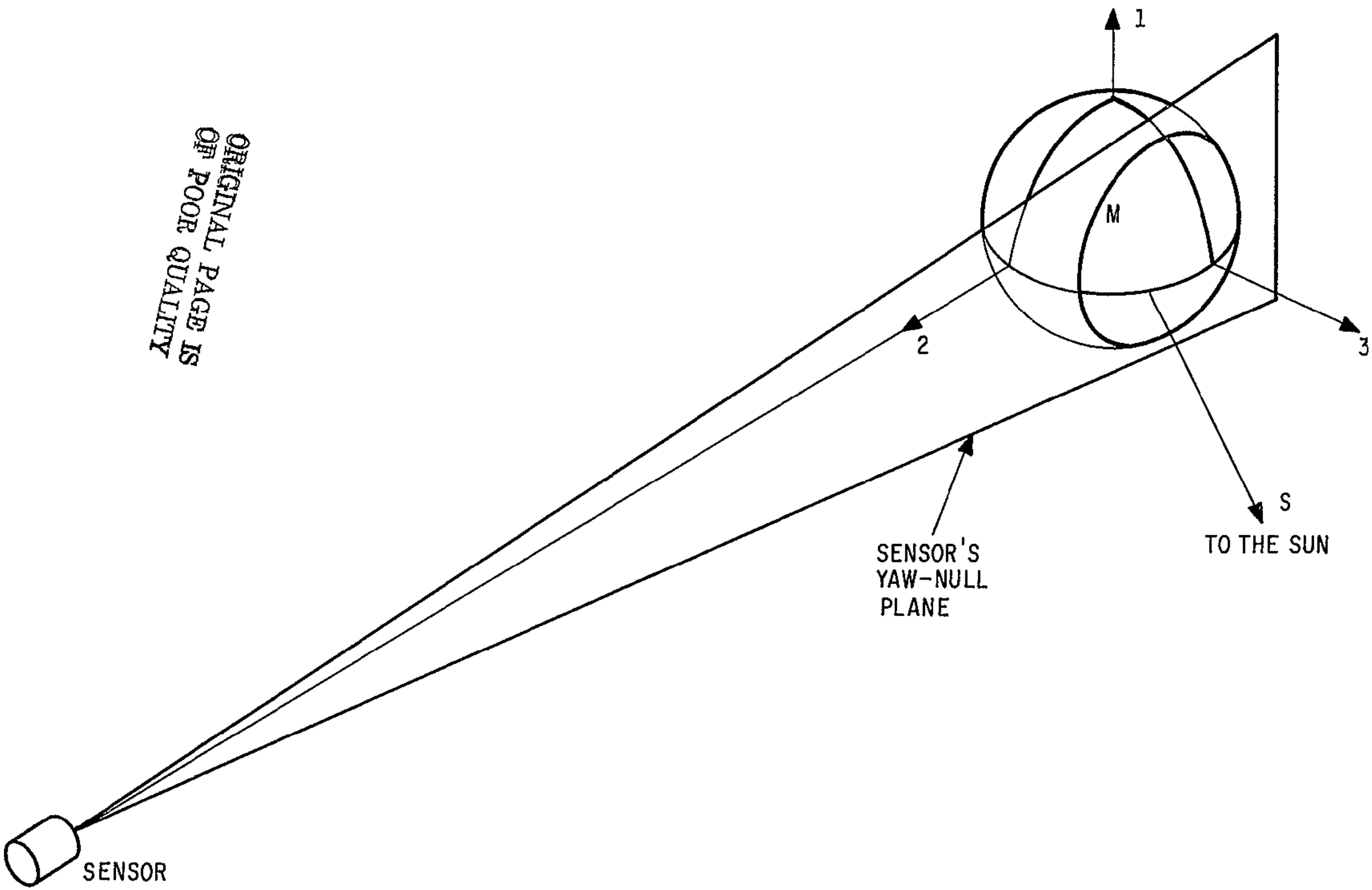


Figure 3-1

Pictorial View of Sensor-Moon Coordinate System

The sensor is located on the 2-axis and the sun lies in the 2-3 plane. These two facts serve to define the orientation of the coordinate system. Although it will be convenient to refer to the 2-3 plane as the "equator" and the 1-axis as the "north pole", it must be clearly understood that the true lunar equator and pole are something else entirely and are of no concern in the present development due to assumption 2 (Section 3.0). Thus we shall use the terms equator and pole freely in the context just described.

Figure 3-2 is a top view of Figure 3-1, includes no additional information, and is included only to clarify the situation.

The sensor's optical axis lies in the 2-3 plane at all times and, as shown in Figure 3-1, the yaw-null plane intersects the moon somewhere between the pole and its eastern limb. This line of intersection, which lies on the moon's surface, is called M.

Nestled within the positive 1, 2, and 3 axes is octant 1 of the lunar sphere; it is outlined in Figure 3-1. This octant, the coordinate system, and some quantitative terms are shown in Figure 3-3. For instance, there is the vector S again, still lying in the 2-3 plane, and pointing at the sun. Note that

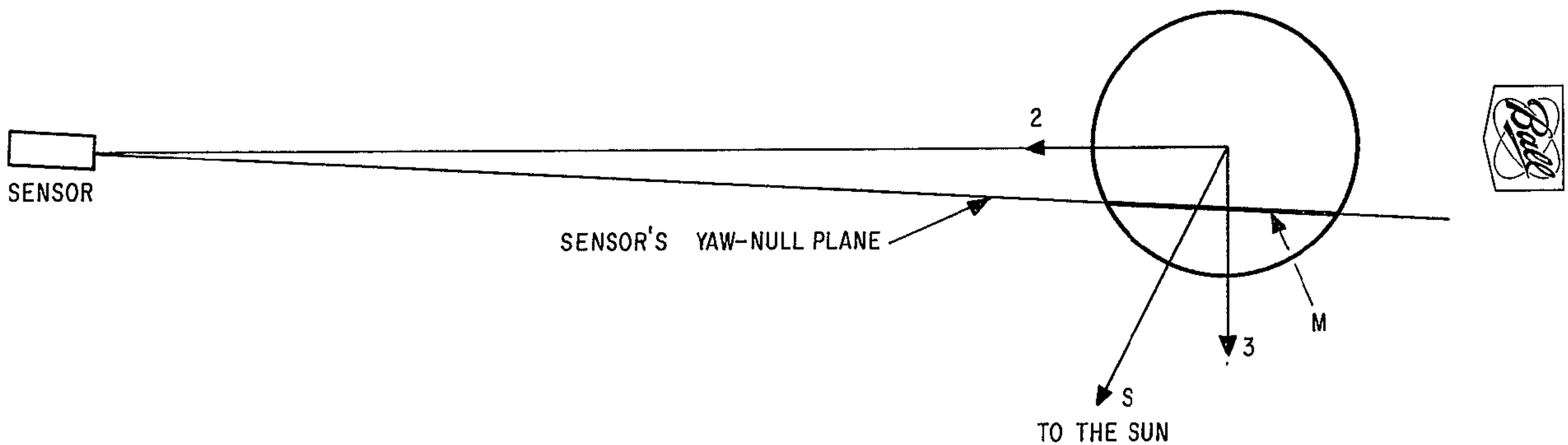


Figure 3-2

Top View of Sensor-Moon Coordinate System it is displaced an angle σ from the 2-axis. As shown, σ is positive, as are all the angles shown in Figure 3-3.

The intersection line, M, also appeareth. It intersects the equator an angle θ_0 from the 2-axis. It is drawn so that M is parallel to the 1-2 plane and this is a departure from reality in that it results from a plane which makes a small non-zero angle with the 1-2 plane. This angle is never greater than 16 arc-minutes however (or the yaw-null plane would fail to intersect the moon at all) so it has been ignored. The resultant errors are very small compared to those introduced by the assumptions discussed earlier.

3.3 Construction of the Mathematical Model

Our ultimate goal is to determine just where the lunar sensor will attain a null condition. This is equivalent to calculating, for a given value of σ (sun position) where the intersection line, M, must be located so that the energies reaching the sensor from both sides of it are equal.

To this end, we subdivide the portion of the moon visible from the sensor into differential areas, dA, compute the relative energy reaching the sensor from each of them, and then sum these differential energies to obtain the net effect. By com-

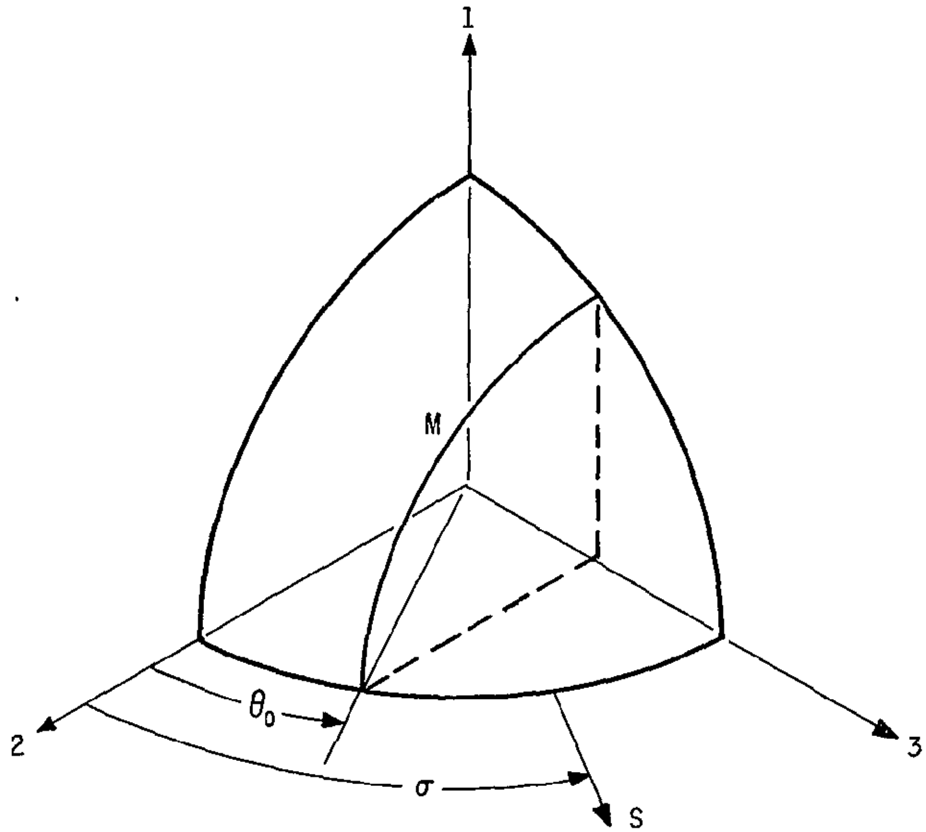


Figure 3-3

First Octant of Lunar Surface

paring the energy contributed by the area to the west of M to the total energy, we can determine the value of θ_0 which makes this ratio equal one-half. This, then, will be the location on the equator where the sensor will be nulled.

It is now time to get down to specifics. If you are not interested in getting involved in the mathematics, this is the time to read elsewhere. Appendix B comes highly recommended.

Figures 3-4a and 3-4b are somewhat more detailed versions of Figure 3-3 in that they locate a typical differential area, dA, on the lunar surface. The manner in which dA is defined in each case is different and two figures are presented to unclutter the artwork.

In Figure 3-4a, dA is defined in terms of its normal vector, N. This vector is situated an angle ϕ above the equatorial plane and its projection in that plane makes an angle θ with the 2-axis. Thus, N can be written[1] as

$$N = \begin{bmatrix} s\phi \\ c\theta c\phi \\ s\theta c\phi \end{bmatrix} \qquad 3.1$$

[1]For an exposition of the perhaps unfamiliar notation used in this paper, see Appendix A.

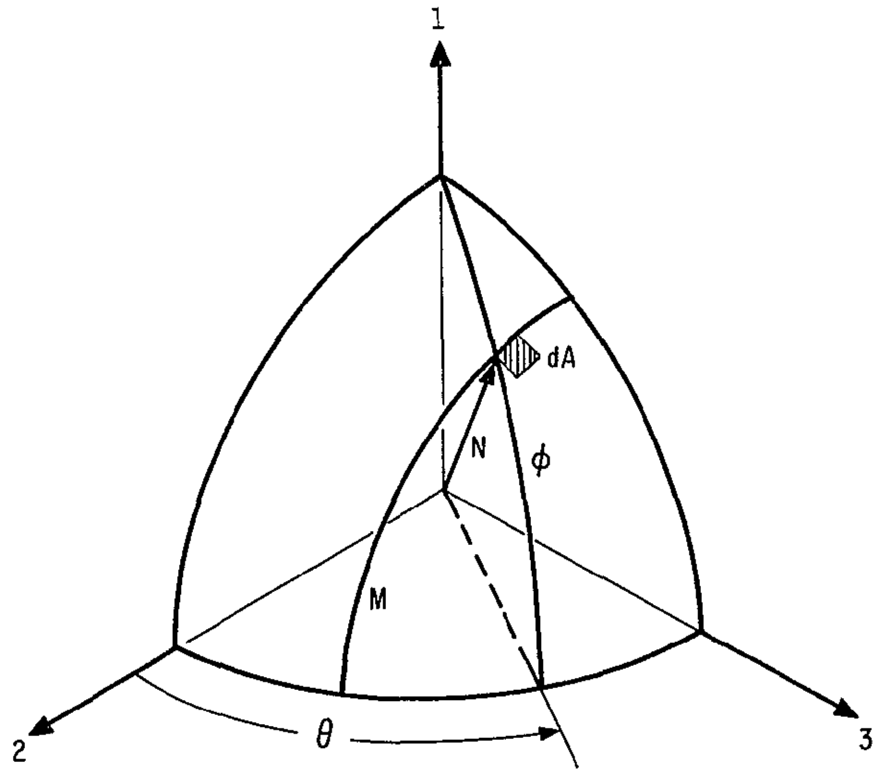


Figure 3-4a

Definition of dA in Terms of its Normal Vector

Note that the moon is assumed to be spherical and for now, of radius 1.

Since the sensor responds to energies which are distributed symmetrically about the line M, we must also locate dA in a way that will mathematically reflect this symmetry. This is done in Figure 3-4b. Our differential area is now defined by the vector V which has its "tail" located on the 3-axis, lies in the sensor's yaw-null plane, and is an angle α above the equatorial plane. It is written as

$$V = \begin{bmatrix} c\theta_0 s\alpha \\ c\theta_0 c\alpha \\ 0 \end{bmatrix} \qquad 3.2$$

Note that V is not, in general, a unit vector. V and N are related by the following equation:

$$N = \begin{bmatrix} 0 \\ 0 \\ s\theta_0 \end{bmatrix} + V \qquad 3.3$$

Some of the sunlight incident upon dA will ultimately be received by the lunar sensor. Let us now compute just what the intensity of this reflected sunlight will be.

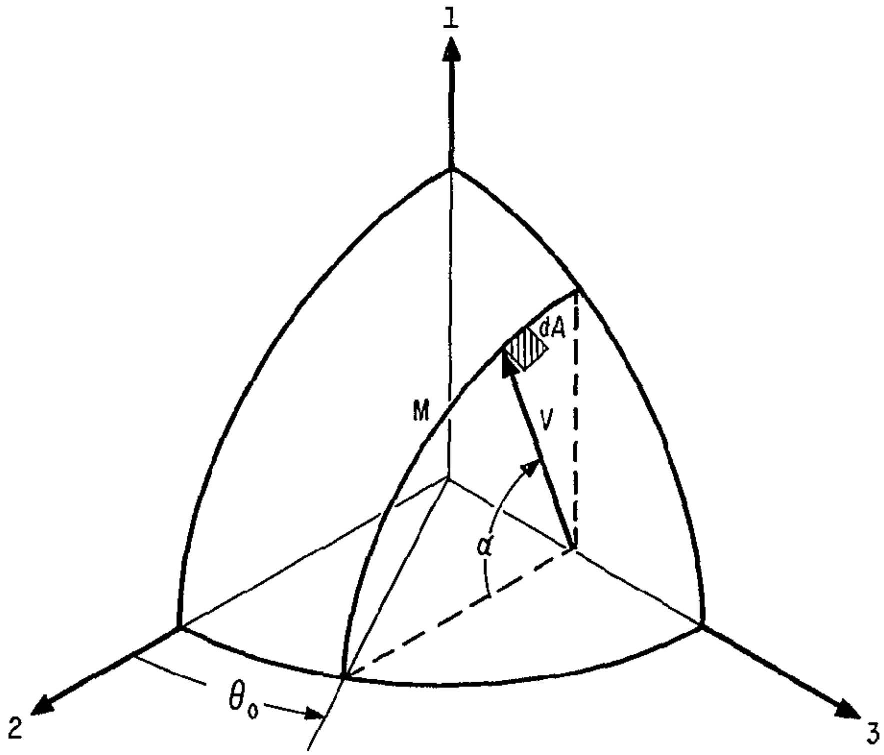


Figure 3-4b

Definition of dA in Terms of a Symetric Distribution of Area

Toward this end we let:

B_o = Power density of sunlight in the vicinity of the moon. (watts-cm^{-2}).

B = Power density of the reflected sunlight at the point of observation.

γ = Angle between the solar vector, S, and N.

ξ = Angle between the 2-axis (also called E), and N.

R = Distance from dA to the point of observation.

$f(\gamma,\xi)$ = a "bidirectional reflectance function" which is the ratio of reflected-to-incident power densities <u>at</u> <u>dA</u>. The following two attributes of f should be noted:

1. It is a function of the incident angle γ, and the reflectance angle, ξ, only and does not depend on the location of dA. This is a result of assumption 2, Section 3.0.

2. The reflectance angle is taken to be ξ rather than the angle between N and the line con-

necting dA and the observation point. This will produce, at worst, a 16 arc-minute error in the reflectance angle, the implications of which are so small so as to make the simplification more than reasonable.

The differential power density of sunlight, dB, which emanates from dA and is received by the sensor is then

$$dB = kB_o \frac{f(\gamma,\xi)}{R^2} dA \qquad 3.4$$

where k is simply a proportionality constant that will disappear shortly when we come to compare energies reciived from two areas on the moon.

The problem now is to find, for a given value of σ, the θ_0 which causes line M to divide the total energy received from the moon into two equal parts. That is, if $B(\theta_0)$ is defined to be the result of sunlight reflected from the visible region of the moon to the west of M, then we seek that particular θ_0, $\overset{*}{\theta}_0$, that will give

$$\frac{B(\overset{*}{\theta}_0)}{B(\pi/2)} = 1/2 \qquad 3.5$$

To accomplish this, we will have to integrate dB over α and θ_0.

From Figure 3-4b, we see that

$$dA = r^2 \, c\theta_0 \, d\alpha \, d\theta_0 \tag{3.6}$$

where r is the radius of the moon.

Let d_m be the distance from the center of the moon to the observation point. It is measured along the 2-axis. From the law of cosines we have

$$R^2 = r^2 + d_m^2 - 2rd_m \, c\xi \tag{3.7}$$

Now

$$c\xi = N \cdot E \tag{3.8}$$

$$= c\phi \, c\theta \tag{3.9}$$

so 3.7 becomes

$$R^2 = r^2 + d_m^2 - 2rd_m \, c\phi \, c\theta \tag{3.10}$$

Combining 3.4, 3.6, and 3.10 together with the definition

$$q \triangleq r/d_m \qquad 3.11$$

we have

$$dB = kB_o q^2 \frac{f(\gamma,\xi)c\theta_0}{1-2q\ c\phi\ c\theta + q^2}\ d\alpha d\theta_0 \qquad 3.12$$

To actually perform the calculations, we must have γ, ξ, ϕ, and θ in terms of α and θ_0. By expanding 3.3 we get

$$\begin{bmatrix} s\phi \\ c\phi c\theta \\ c\phi c\theta \end{bmatrix} = \begin{bmatrix} c\theta_0 s\alpha \\ c\theta_0 c\alpha \\ s\theta_0 \end{bmatrix} \qquad 3.13$$

so

$$\phi = \sin^{-1}\left\{ c\theta_0\ s\alpha \right\} \qquad 3.14$$

and

$$\theta = \tan^{-1}\left\{ \frac{\tan\ \theta_0}{c\alpha} \right\} \qquad 3.15$$

The reflectance angle, ξ, can be obtained by combining 3.9, 3.14, and 3.15. Finally, we get the incidence angle, γ, by noting that

$$c\gamma = N \cdot S \tag{3.16}$$

Now

$$S = \begin{bmatrix} 0 \\ c\sigma \\ s\sigma \end{bmatrix} \tag{3.17}$$

so

$$N \cdot S = c\sigma \; c\theta \; c\phi + s\sigma \; s\theta \; c\phi \tag{3.18}$$

and

$$c\gamma = c\phi \; c(\theta - \sigma) \tag{3.19}$$

which, when combined with 3.14 and 3.15 will yield γ.

We now have only to settle on what the limits of integration shall be. We can choose between two approaches.

1. Integrate over the region

$$-\pi/2 \leq \sigma \leq + \pi/2$$

$$-\pi/2 \leq \theta_0 \leq \theta_0^*$$

and construct f so it will be zero in those regions not illuminated by the sun; or

2. Integrate only over the region which is visible from the observation point <u>and</u> illuminated by the sun.

The latter approach is definately preferable because it involves significantly less computation time; an important consideration since numercial integration on a digital computer is the only reasonable way to integrate 3.12.

Before proceeding, we should note that because of the inherent symmetry of the problem, we can confine our calculations to the northern hemisphere and then simply double the results. Even the doubling can be omitted because of the ratioing (equation 3.5) that will eventually be done.

In order to integrate only over the illuminated protion of the moon's surface, we must be able to mathematically define the terminator in terms of θ_0 and α. Assumption 3 permits us to do this with a minimum of fuss.

The terminator is defined to be the locus of points on the surface of the moon where S is tangent to the surface; equivalently, where S and N are perpendicular. From 3.19, we see that this condition is met when

$$c\phi\ c(\theta-\alpha) = 0 \qquad 3.20$$

Now this equation is satisfied when either

$$\phi = \pm\ \pi/2 \qquad 3.21a$$

or

$$\theta = \sigma \pm \pi/2 \qquad 3.21b$$

All 3.21a states is that the terminator passes through the north and south poles; a true thing but of no particular interest. It is equation 3.21b that will provide a useful relationship.

Before developing 3.21b, let us agree to restrict the analysis to values of σ in the range

$$0 \le \sigma \le \pi \qquad 3.22$$

This will cover the range from full moon to eclipse but will always produce a center of brightness in the eastern half of the moon. This does not cause a problem, however, since the basic symmetry of the problem allows us to simply negate our result (placing the center of brightness in the western half) if the original σ should happen to lie in the range

$$-\pi \le \sigma < 0 \qquad 3.23$$

Now, with σ restricted according to 3.22, we can remove the amgiguity of 3.21b and rewrite it as

$$\theta = \sigma - \pi/2 \qquad 3.24$$

Combining 3.24 and 3.15 gives

$$\tan^{-1}\left\{\frac{\tan\theta_0}{c\alpha}\right\} = \sigma - \pi/2 \qquad 3.25$$

Solving 3.25 for α gives the particular value of α which will place V <u>on</u> the terminator for the specified θ_0. This special value of α will be called α_0. Thus

$$\alpha_0 = \cos^{-1}\left\{-\tan\sigma\ \tan\theta_0\right\} \qquad 3.26$$

Notice that 3.26 is not defined for some combinations of σ and θ_0. This does not worry us, however, since those cases for which

$$|\tan\sigma\ \tan\theta_0| > 1 \qquad 3.27$$

are precisely those for which V is in the dark portion of the moon regardless of α and we will not be integrating over those regions anyway.

The problem must now be separated into two basic cases depending on the location of the sun;

Case I $0 \leq \sigma < \pi/2$

Case II $\pi/2 \leq \sigma \leq \pi$

The reason for this can be best understood by referring to Figures 3-5a and 3-5b which depict typical situations for Case I and Case II respectively.

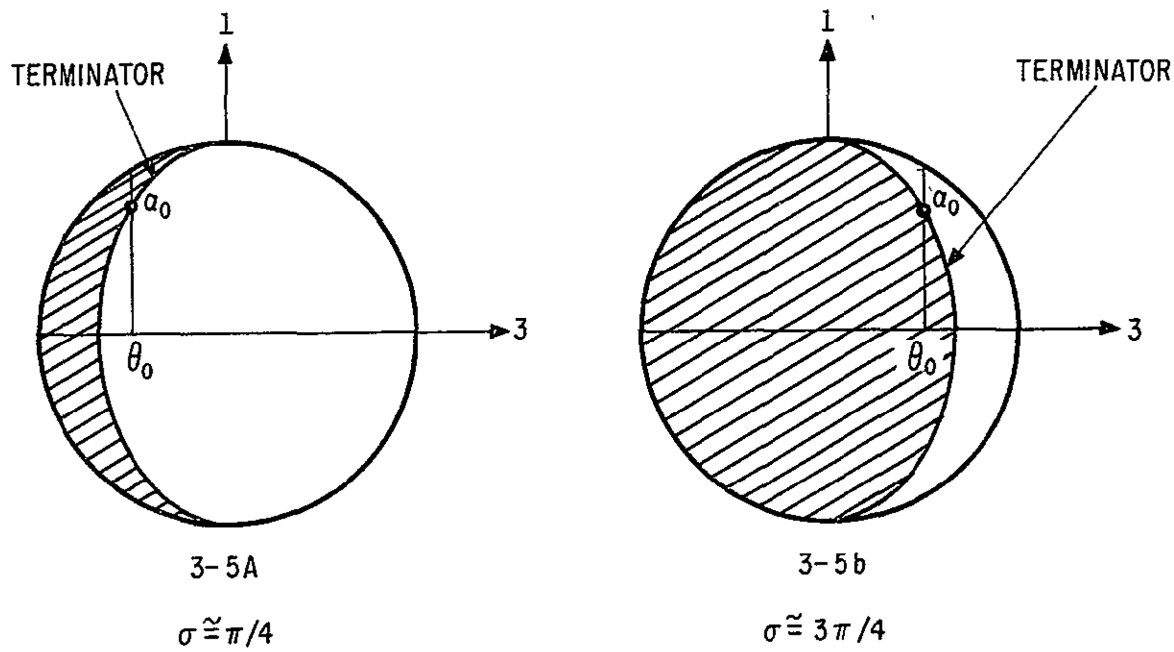


Figure 3-5

Views of the Moon as Seen From the Sensor

First consider 3-5a which corresponds to Case I. The sketch is drawn for a σ of approximately $\pi/4$. The terminator intersects the equator at $\theta_0 = \sigma-\pi/2$ so we begin integrating there and let θ_0 cover the range

$$\sigma-\pi/2 \leq \theta_0 \leq \pi/2 \tag{3.28}$$

Now, over the semi-range $\sigma-\pi/2 \leq \theta_0 \leq 0$, the terminator is visible so α will take on values $0 \leq \alpha \leq \alpha_0$ where α_0 is calculated from 3.26. However, for positive values of θ_0 (θ_0 in the eastern hemisphere) the terminator disappears onto the back side of the moon so α must cover the region from equator to limb; i.e., $0 \leq \alpha \leq \pi/2$.

For Case II (Figure 3-5b) the range of θ_0 is constant and independent of σ; $0 \leq \theta_0 \leq \pi/2$. Again we have two subcases for the α interval. From Figure 3-5b they can be seen to be:

$$\text{for} \quad 0 \leq \theta_0 \leq \sigma-\pi/2 \; , \quad \alpha_0 \leq \alpha \leq \pi/2 \tag{3.29}$$

$$\text{for} \quad \sigma-\pi/2 \leq \theta_0 \leq \pi/2 \; , \quad 0 \leq \alpha \leq \pi/2 \tag{3.30}$$

So there we have it. If we define the function A to mean "the applicable range of...", then we can write our basic integral as

$$B(\sigma) = k \int_{A(\theta_0)} \int_{A(\alpha)} \frac{f(\gamma,\xi)}{1-2q\ c\phi\ c\theta + q^2}\, d\alpha\ d\theta_0$$

where k is a factor of proportionality, and $A(\alpha)$ and $A(\theta_0)$ are defined in the preceding paragraphs.

We have yet to get specific about $f(\gamma,\xi)$ and come up with a means of calculating σ. Read on.

3.4 The "Reflectance Function", $f(\gamma,\xi)$

At the beginning of this study, the author (Rose) indicated to GSFC his intent to model the reflective properties of the lunar surface after a Lamertian reflector; that is

$$f(\gamma,\xi) = c\gamma\ c\xi \qquad 3.31$$

It was common "knowledge" by workers in the field that this was a reasonable thing to do. Nevertheless, as it was decided to perform some calculations using this model and compare the results to empirical data, thereby verifying the model's applicability.

The inappropriateness of the model was demonstrated instead.

The analytical test which produced these disturbing results was a computation of the ratio of half moon brightness to full moon brightness. The Lambertian model produced (see Appendix C) a value of 0.32 while actual measurement[1] of the ratio gives of value of 0.089.

The wide discrepancy between theoretical and measured results initiated a search for a better reflectance function. This led to discussions with personnel at the Lunar and Planetary Laboratory, University of Arizona, in Tucson. They suggested that we model the lunar surface as a Lommel-Seeliger reflector; that is, have

[1] C.W. Allen, 'Astrophsical Quantities', 2-nd Edition University of London, the Athlone Press 1964; Table of Moon's Phase Law - Page 146

$$f(\gamma,\xi) = \frac{c\gamma}{c\gamma + c\xi} \qquad 3.32$$

This did not help the situation, however, for when the half-to-full moon brightness ratio was computed, we obtained a value of 0.5. This was worse than the Lambertian model! It was then suspected that the University of Arizona information was being misinterpreted and that equation 3.32 gave the ratio of reflected to <u>actual</u> incident energies. If that were true, 3.32 would have to multiplied by cos γ to account for the non-perpendicularity of dA and S. This would give

$$f(\gamma,\xi) = \frac{c^2\gamma}{c\gamma + c\xi} \qquad 3.33$$

But alas. When the half-to-full moon brightness ratio was calculated using 3.33 the situation became even worse; the value was 0.62.

By now, the sounding rocket launch was becoming imminent and a usable reflectance function had to be settled upon soon. We therefore returned to the best function we had (the Lambertian model, equation 3.31) in hopes that even though it produced invalid brightness ratios, it might permit the significantly different problem of locating sensor null points to be solved.

To see that this might be possible, consider the following. Suppose we sketch the lunar albedo energy falling to the west of the sensor yaw null plane as the plane is scanned from west to east across the moon. Two such plots are shown in Figure 3-6 and they correspond to two different values of σ.

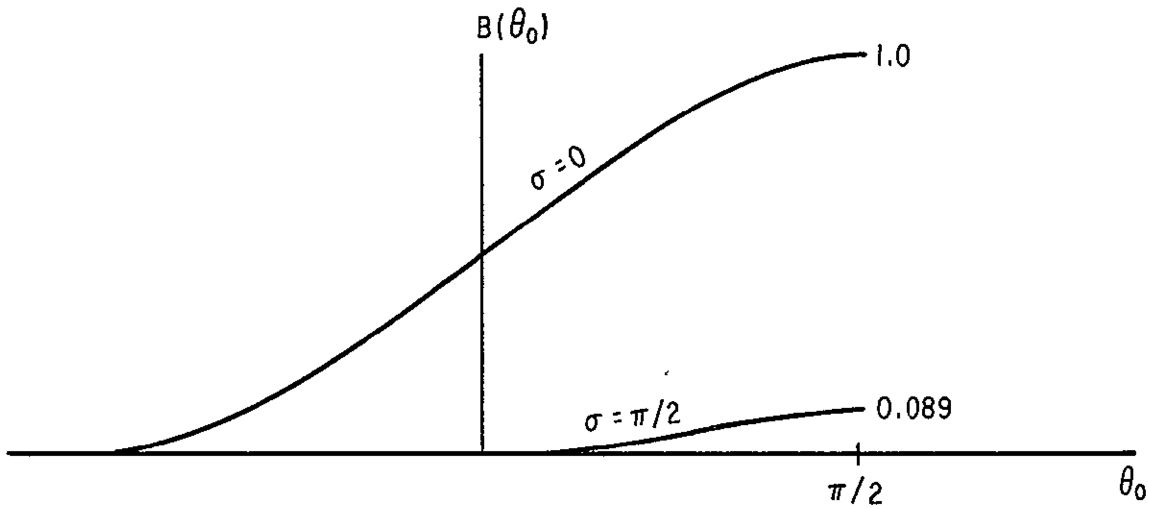


Figure 3-6

Sketch of $B(\theta_0)$ for Full and Half Moons

Let us now sketch another pair of simular curves only this time we hold σ constant at $\pi/2$ (half moon) and plot the real $B(\theta_0)$ and the $B(\theta_0)$ corresponding to a Lambertian moon.

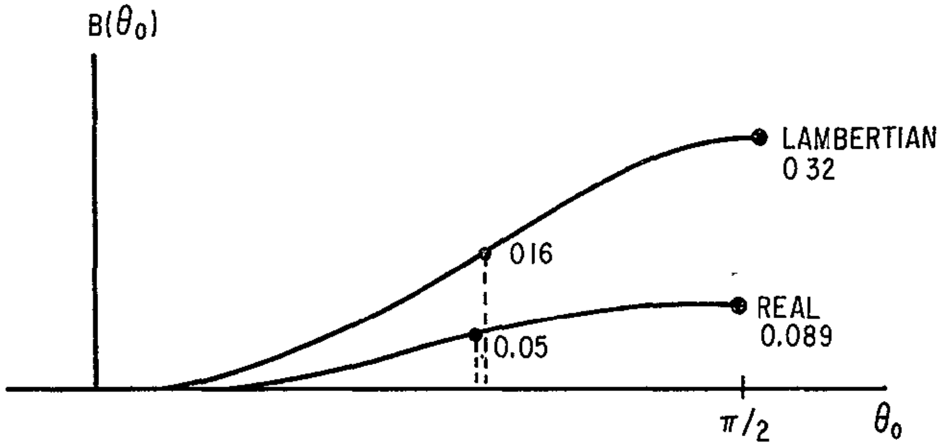


Figure 3-7

$B(\theta_0)$ Curves for Real and Lambertian Moons

The point is that the sensor will point to that value of θ_0 where the $B(\theta_0)$ curve has half its end point ($\theta_0 = \pi/2$) value and that even though the two curves shown in 3-7 differ in magnitude by more than a factor of 3, their mid-points could occur at almost identical places.

It was on the basis of such reasoning that the sensor offset was measured one night (8^h U.T., 8 September 1974) by using the sensor itself and the results compared to theoretical predictions. There was agreement to within better than 11 arc-seconds; much better than the goal of ±1 arc-minute for which we were striving.

Thus, as a result of this very close agreement between theory and experiment, it was decided to use the Lambertian model in all subsequent calculations. It is nevertheless most annoying that we cannot calculate absolute brightnesses and we plan to continue the search for a good reflectance function. Someday we may even do it.

3.5 Computation of Sun Position (σ) From Ephemeris Data

In order to compute the sun's position angle, σ, we must know:

1. where the sun is,
2. where the moon is,
3. where the sensor is,

all with respect to the center of the earth. To accomplish this, we introduce another coordinate system, E^c, which is tied to the celestial sphere and has its origin at the center of the earth. E_1^c points to the north celestial pole and E_2^c is aligned to the First Point of Aries. Thus, the celestial sphere right ascension variable has 0^h aligned to E_2^c and 6^h

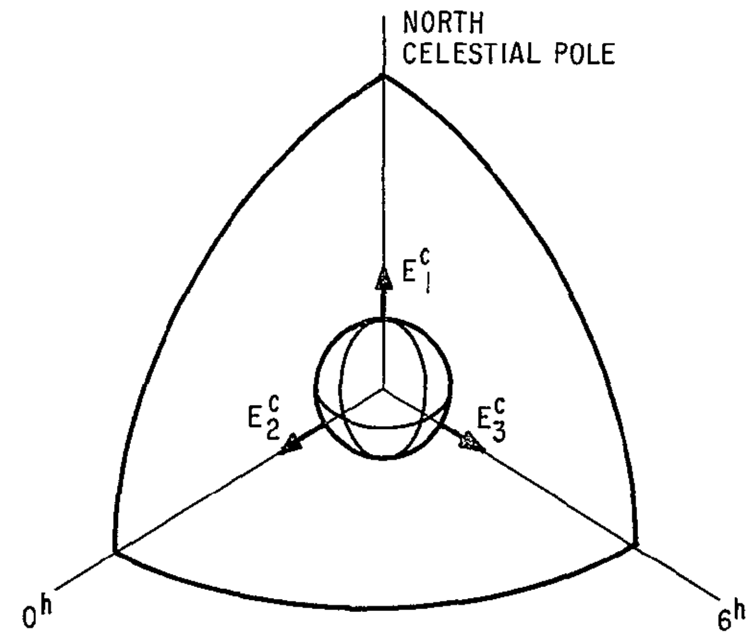


Figure 3-8

Relationship of Coordinate System
E^c to the Celestial Sphere

aligned to E_3^c; ref. Figure 3-8. The earth rotates about E_1^c and Greenwich lies in the positive E_1^c - E_2^c plane at 0^h U.T.

Suppose a point on the celestial sphere has a right ascension and declination of ρ and δ respectively. The unit vector aligned to this point will have the form

$$\begin{bmatrix} s\delta \\ c\delta c\rho \\ c\delta s\rho \end{bmatrix} \qquad 3.34$$

A point on the earth having an east longitude of λ and a latitude of ϕ can be represented by

$$\begin{bmatrix} s\phi \\ c\phi c\lambda \\ c\phi s\lambda \end{bmatrix} \qquad 3.35$$

in an earth-based coordinate system. To compute its coordinates in E^c we must account for the fact that the vector is rotating about E_1^c at an angular rate ω and has been doing so for some time t. The transformation is performed as follows

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\omega t & -s\omega t \\ 0 & s\omega t & c\omega t \end{bmatrix} \begin{bmatrix} s\phi \\ c\phi c\lambda \\ c\phi s\lambda \end{bmatrix} \qquad 3.36$$

Note that this 3 x 3 matrix rotates <u>vectors</u> about the 1-axis in a <u>fixed coordinate frame</u>.

Now we can proceed. Look at Figure 3-9. It depicts the pertinent elements of the analysis; earth, moon, lunar sensor, and sun. The view is from above the plane defined by the center of the moon, the center of the sun, and the lunar sensor. Note that this plane is none other than the 2-3 plane used in earlier sections; e.g., look at Figure 3-3. The vectors G, L, and H extend from the center of the earth to the sensor, the lunar center, and sun's center respectively. As such, they do not necessarily lie in the plane defined by D and S* (our familiar 2-3 plane). One point of clarification; the vector S*, appearing in Figure 3-9, differs from the vector S, Figure 3-3, only in magnitude. S is a unit vector.

Our problem is to find σ and we can do this if we know D and S* for then

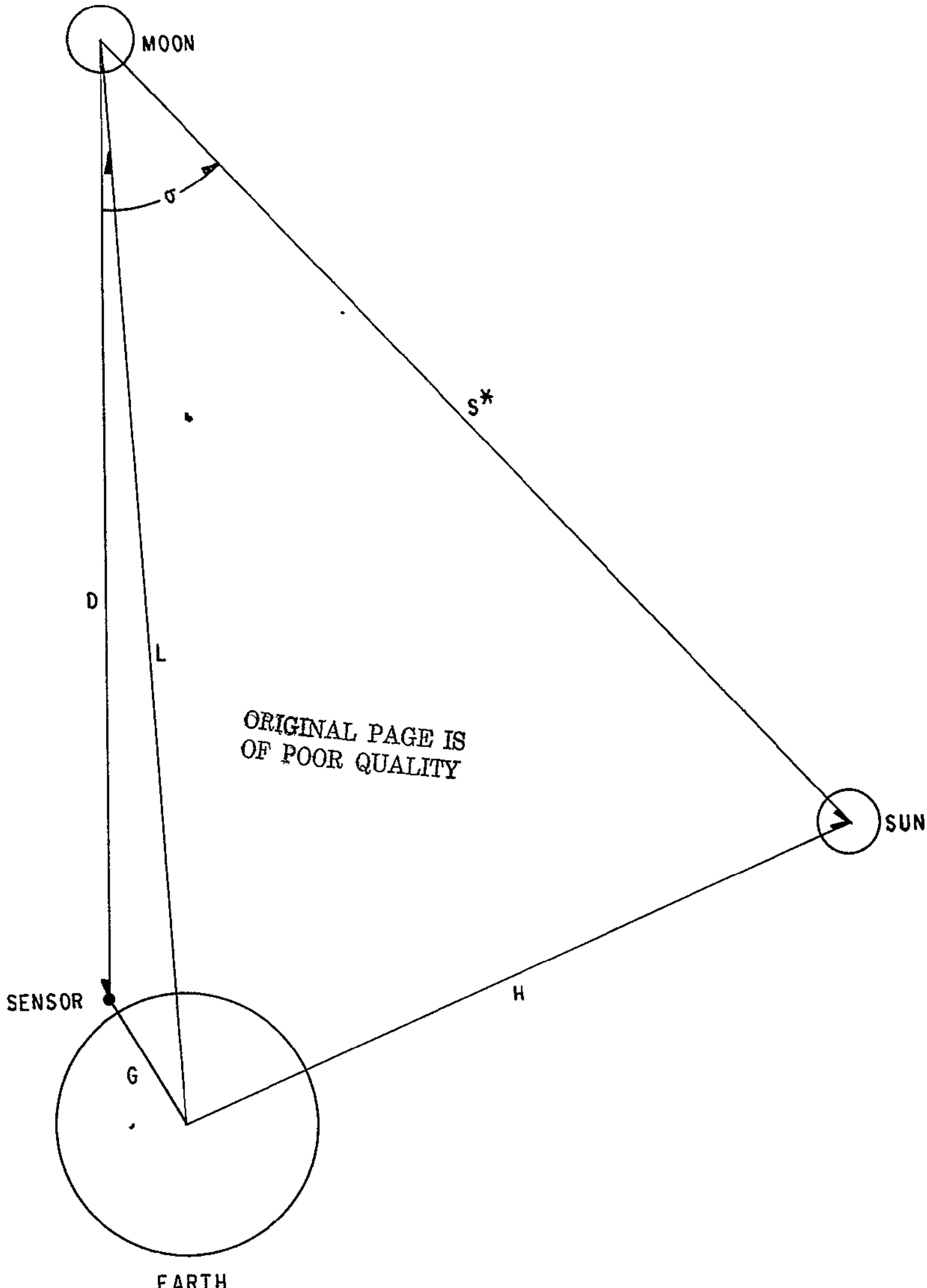


Figure 3-9

Orientation of Earth, Moon, Sensor and Sun for Vector - Definition Purposes

$$\sigma = \cos^{-1} \frac{D \cdot S^*}{|D| \; |S^*|} \qquad 3.37$$

To calculate D, we first need G and L. If h is the distance from the center of the earth to the sensor, then

$$G = h \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\omega t & -s\omega t \\ 0 & s\omega t & c\omega t \end{bmatrix} \begin{bmatrix} s\phi \\ c\phi c\lambda \\ c\phi s\lambda \end{bmatrix} \qquad 3.38$$

If d_m is the earth-moon distance and ρ_m and δ_m are the moon's right ascension and declination, then

$$L = d_m \begin{bmatrix} s\delta_m \\ c\delta_m \; c\rho_m \\ c\delta_m \; s\rho_m \end{bmatrix} \qquad 3.39$$

From Figure 3-9 we see that

$$D = G - L \qquad 3.40$$

Next, to compute S*, we need H in addition to L. Let d_s be the earth-sun distance; ρ_s and δ_s are the sun's right ascension and declination. Then

$$H = d_s \begin{bmatrix} s\delta_s \\ c\delta_s \; c\rho_s \\ c\xi_s \; s\rho_s \end{bmatrix} \qquad 3.41$$

Again, from Figure 3-9, we see that

$$S^* = H - L \qquad 3.42$$

Thus, given the sensor's geographic position, the sun and moon's celestial coordinates, and the time, equations 3.38 through 3.42 can be used to compute D and S* which in turn, via equation 3.37, produce σ. The sun and moon data can be obtained from an ephemeris and the particulars of doing so are discussed in Section 4.

3.6 Computation of Sensor Offset Angle

Once we have computed all the pertinent quantities developed in the preceding sections we will know where the sensor will point...in terms of a point on the lunar equator which is displaced an angle θ_0^* from the 2-axis. All that is left is to convert this into an angular offset of the sensor about its yaw axis.

Consider Figure 3-10. It is a top view of the 2-3 plane and shows the moon, the vector D (ref. Section 3.5), and the edge-view of the sensor's yaw-null plane. The sensor is nulled and, as such, its yaw-null plane intersects the lunar equator at a point which is displaced an angle θ_0^* from the 2-axis.

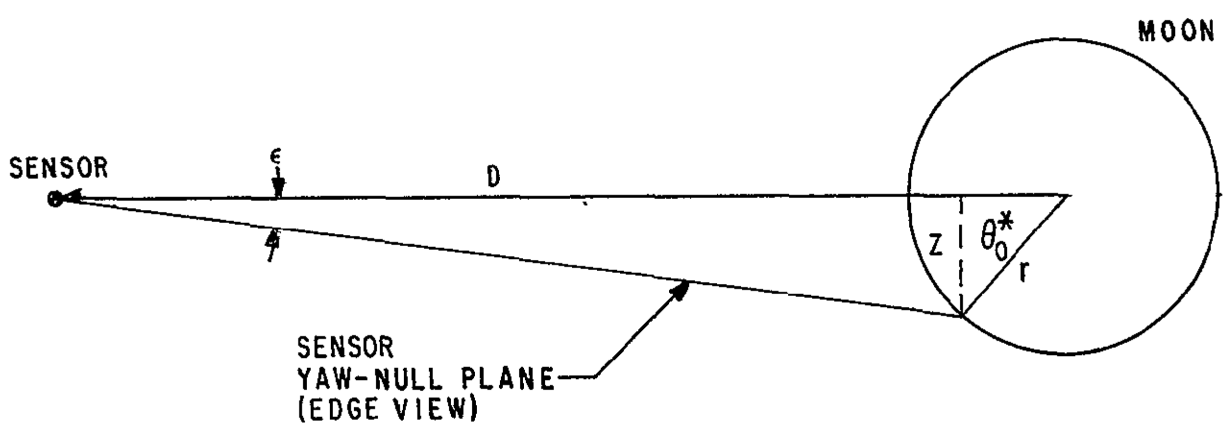


Figure 3-10

Geometry for Calculating Sensor Offset Angles

Recall; D is the vector from the center of the moon to the sensor and has magnitude d_m, r is the radius of the moon. From Figure 3-10 we have

$$r\, s\theta_0^* = z = (d_m - r\, c\theta_0^*) \tan \varepsilon \qquad 3.43$$

or

$$\varepsilon = \tan^{-1} \left\{ \frac{r\, s\theta_0^*}{d_m - r\, c\theta_0^*} \right\} \qquad 3.44$$

This quantity, ε, is the angle the sensor will be displaced from the center of the lunar disk when the sensor is nulled.

Section IV

COMPUTER PROGRAM DESCRIPTIONS

4.0 INTRODUCTION

The purpose of the computer programs written for the lunar radiometric center problem was, in a nutshell, to evaluate the predicted vehicle sensor offset angle when pointed precisely at the moon's center of brightness rather than its physical center. The solution to the problem involved, primarily, determination of required variables to evaluate the integral of the power density function for a specified time and location of launch, and from the results of that integration, to calculate the radiometric center of the moon and thereby determine the predicted offset angle.

These calculations were accomplished in one mainline program, assisted by three subroutines, written in FORTRAN IV language for the BBRC engineering computer, a General Automation 18/30. Descriptions of the general logic flow of each of the programs constitute Section 4.2 and are preceeded by some explanation of features common to all of the programs in Section 4.1. Section 4.3 provides listings and examples of output for each of the programs.

4.1 General Notes

4.1.1 Accuracy of Calculations and 18/30 Computer Precision

Because of the relatively complex nature of and the many iterative calculations required by the lunar radiometric center problem, the computer programs were very susceptible to error propagation. The use of the extended precision feature significantly reduced error build-up problems, but a substantial effort was also made through coding and numerical analysis techniques to avoid inaccurate results due to error pitfalls. This effort was particularly applied in the task of incrementing loop variables when the increment values were fractions which could not be exactly represented in binary. In this case simple accumulation of the variable value ($x = x + \Delta x$) also results in accumulated error, but the problem was avoided completely by always returning the variable to a base value before adjusting it for the next iteration.

All the programs performing the lunar radiometric center calculations were run in G.A. 18/30 Extended Precision, which utilizes 3 16-bit words to represent real variables. Mention is made of this fact to differentiate it from the more commonly known "double precision" which uses four-word floating point representation. The 18/30 is not provided with the double precision feature, but the use of extended precision yields between nine and ten significant digits of accuracy which was adequate for these calculations.

4.1.2 Computer Program Execution Time

Again because of the iterative nature of these calculations, it was necessary to reduce, as much as possible, the computer run time required to complete the computations. The problem was aggrevated by the fact that the 18/30 is not supplied with floating point hardware, but rather must rely entirely on software subroutines to perform all floating point operations. The use of extended precision intensifies the run time problem even further. The resulting calls to floating point subroutines, especially in the long integration loops, greatly increased run time requirements. As a result, extensive efforts were made in the programming of the lunar radiometric center calculations to simplify equations in loops in such a way as to reduce the number of redundant subroutine calls and thereby decrease program execution time. Another solution was to integrate only over half of symmetrical regions. Because the majority of program calculations were involved with the integration of the power density function, use of this technique reduced execution time by a factor of two. (See

Subroutine BIGNT description for further discussion of this integration technique).

4.2 Program Descriptions

4.2.1 Mainline Program, JAKE

The primary function of the mainline program, named JAKE, was to initialize parameters required by the power density function integration subroutines. Upon return from those subroutines, JAKE operated on the integration results to calculate the anticipated sensor offset angle.

4.2.1.1 Parameter Initialization

Five parameters were required as input to the integration subroutines: the lunar phase angle, earth-moon distance for the specified launch time, radius of the moon, and increment values for the two variables of integration. The lunar radius was constant and so was initialized as data. The two integration variable increments values were constant throughout the calculations and were read from card input. The other two parameters were calculated using ephemeris data for the launch time and latitude and longitude of the launch site.

a. Calculation of Earth-Moon Distance

The American Ephemeris provides the necessary equation for calculation of the true geocentric distance for given time of day:

$$D = (a_o + a_1P + a_2P^2 + a_3P^3)6378.16 \qquad 4.1$$

where

p is a number between zero and one representing the fractional part of the half-day in which the GMT lies.

a_o, a_1 a_2, a_3 are polynomial coefficients for true geocentric distance, tabulated in the ephemeris for each half day of the year (expressed in units of Earth's equatorial radius).

The constant 6378.16 is the earth's equatorial radius in kilometers and is used to convert the earth-moon distance into kilometers for consistency with the other calculations.

b. Calculation of the Lunar Phase Angle

The vector equations required to calculate the lunar phase angle were described in Section 3.5 of this report. Parameters required for these calculations are:

Sidereal time for 0 hours universal time of the launch date; right ascension, declination, and geocentric distance of the sun; right ascension, declination, and geocentric distance of the moon; and latitude and longitude on the launch site.

Latitude and longitude of the launch site and sidereal time were read as card input requiring no further adjustment and the calculation of lunar geocentric distance was described above. Right ascension, declination, and geocentric distance of the sun were obtained from the American Ephemeris for 0 hours Ephemeris Time of the days preceeding and succeeding the launch, and right ascension and declination of the moon were obtained from the same source for the hours of Ephemeris Time preceeding and succeeding the launch. These values then needed to be adjusted to reflect the exact time of the launch, which was accomplished via the following equation for general linear interpolation:

$$y = (x(y_2 - y_1) + x_2y_1 - x_1y_2)/(x_2 - x_1) \qquad 4.2$$

where y = the interpolated right ascension, declination or distance

x = the Universal Time of the launch (hours and minutes for solar variables, minutes for lunar variables)

x_1 and x_2 = 0 and 24 hours, respectively for solar variables, and 0 and 60 minutes, respectively, for lunar variables.

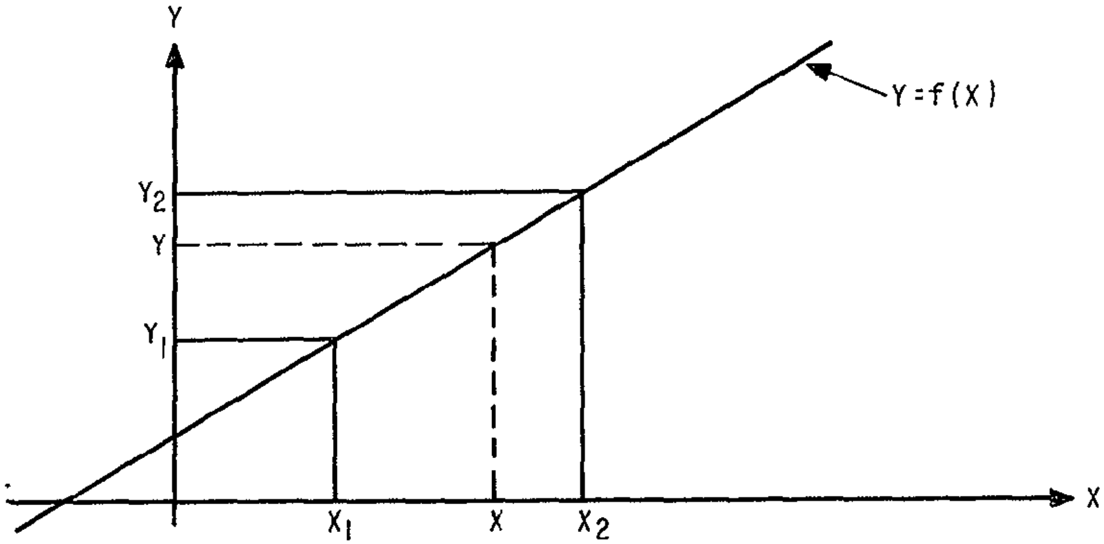
y_1 and y_2 = right ascensions, declinations, or distances as obtained from the ephemeris for the times bracketing the time of launch.

See Figure 4-1 for a graphic illustration of lunar interpolation.

Once these interpolated values were computed, calculation of the lunar phase angle was accomplished as described in Section 3.5.

4.2.1.2 Calculation of the Sensor Offset Angle

The calculation of the sensor offset angle is described in Section 3.6 of this report, and of necessity, some of the variable names mentioned there will also be used here. In order to



TO INTERPOLATE, ASSUMING Y = f(x) IS LINEAR:

$Y = (X(Y_2 - Y_1) + X_2 Y_1 - X_1 Y_2) / (X_2 - X_1)$

Figure 4-1

Generalized Linear Interpolation

determine the Sensor Error Angle, it was first necessary to determine the value for θ_0^* which would correspond to the lunar radiometric center, as shown by the following ratio:

$$\frac{B(\theta_0^*)}{B(\pi/2)} = 1/2 \qquad 4.3$$

The power density function integration subroutine returns to the mainline an array of tabulated integration values ($B(\theta_0)$'s) corresponding to the range of θ_0 angles. Dividing the end point of this array ($B(\pi/2)$) by two yields the center of brightness ($B(\theta_0^*)$ value. By finding the two elements in the array which bracketed that point, it was then only necessary to interpolate between the two corresponding θ_0 values to determine the required θ_0^* angle. This interpolation was accomplished using the linear equation described above. Having found θ_0^*, the Sensor Offset Angle was calculated as shown in Section 3.6.

4.2.2 Subroutine BIGNT

The purpose of subroutine BIGNT was to perform the required integration of the power density function, given the necessary parameters provided by mainline program JAKE. BIGNT's first task was to determine the limits of the integration variables, α and θ_0. The criteria for this determination are described in

Section 3.3 of this report. Initially, the incremental values of integration, $\Delta\alpha$ and $\Delta\theta_0$, were fixed at 1°, and since the upper and lower bounds of θ_0 were always integral amounts of degrees, the 1° restriction on $\Delta\theta_0$ was sufficient. The integration limits of α, however, often included fractional parts of degrees, so the $\Delta\alpha$ value was subsequently adjusted slightly to fit exactly within the range of integration.

It is also necessary to note that the initial values for the variables of integration were not set exactly to their respective lower limits, but rather were set a half increment above those values (i.e., lower limit + $\Delta\alpha/2$ or +$\Delta\theta_0/2$). The reason for this initialization was twofold: 1) Since only half of the full symmetrical region was being integrated. (See Section 3.3), the variables of integration were set up a half step to fully cover the illuminated region of the moon and to avoid duplication of the region's mid-area; and 2) the most meaningful result of the point-wise integration could be obtained by evaluating the integrand at the mid-point, rather than an end-point, of the differential area. See Figure 4-2 for an illustration of this method of area mid-point integrand evaluation.

Having accomplished all of this initialization, BIGNT was then free to attack the problem at hand, the evaluation and integra-

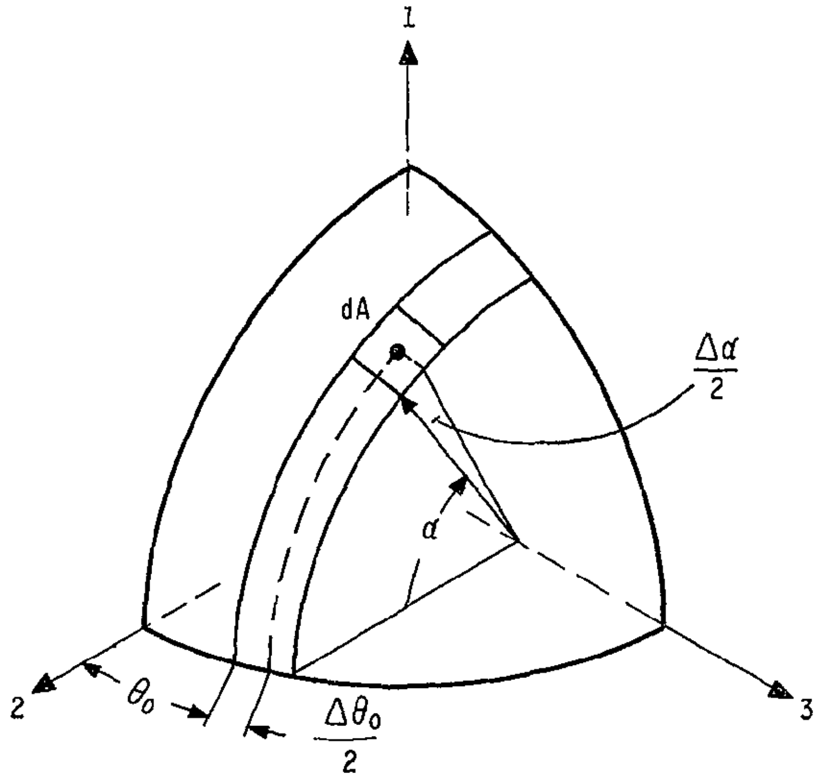


Figure 4-2

Illustration of Mid-Point Integrand Evaluation

tion of the lunar power density function. The task was accomplished in two nested DO loops, the outer loop being concerned with θ_0, and the inner loop with α. For each value of θ_0, the power density function was evaluated for every value in the range of α and the resultant function was then integrated over α. The actual integration was performed by an 18/30 Scientific Subroutine Library program, QSF, which uses a combination of Simpson's Rule and Newton's 3/8 Rule. QSF operates on a tabulated array of function values, and returns a tabulated array of step-wise integration values, the endpoint of which is the final result of the integration.

As each α loop is completed for each θ_0 value, the endpoint of the resulting integration vector was used to form an element of the array which, when the θ_0 loop was complete, would again be integrated by the QSF subroutine, this time over θ_0. The final array of integration values, thus obtained by BIGNT, was then returned to the mainline program along with an integer variable indicating the number of elements in the array. Having completed its appointed rounds, BIGNT then goes to sleep awaiting another call from the mainline program and a new set of parameters.

4.2.3 Function Subprograms FLUX and RDFCN

The actual evaluation of the power density function at a single point was performed by two small function subprograms, FLUX and RDFCN. Subprogram FLUX was responsible for evaluating the entire power density function, and it calls RDFCN for definition of the lunar reflectance function. During the development of the math model, when experimentation with Lambertian, Lommel-Seeliger, and other radiance functions was going on, it was most convenient to have the function evaluation and reflectance function itself separated from the larger programs so that small changes to the equations could be easily and quickly incorporated. Since that experimentation continued up to the very last calculations, RDFCN and FLUX remained in existance and were never merged with BIGNT.

4.2.4 Additional Programs

Using BIGNT, FLUX and RDFCN as a calculating base for the lunar radiometric center problem, several other mainline programs, besides JAKE, were written. Since these programs did not constitute part of the final product, but only contributed to its formulation, they will not be described in any detail here. They should, however, be mentioned because of the functions they provided as constructive tools.

Primarily, the other mainline programs differed from JAKE in that they did not actually calculate the lunar phase angle and earth-moon distance parameters required for the integration of the power density function. Instead, they operated on constant values for these variables. Neither were those programs concerned with the determination of the sensor offset angle, but rather centered around the calculation of power density ratios for different lunar phase conditions. No consideration was given to an actual launch situation, but instead hypothetical, and therefore somewhat predictable, circumstances were assumed. In this sense, these programs provided a valuable tool for evaluating the performance of different lunar reflectance functions when comparing, for instance, full to half-moon brightnesses. These programs also provided a wealth of plotted output which served as a visual aid in the trouble-shooting of both the programs and the math model. So although these programs did not constitute a part of the final calculations required by the lunar radiometric center problem, their contribution to the end item cannot be overlooked. The listing for mainline program HERB is included in the next section to serve as an example of the type of function provided by these intermediate mainline programs.

4.3 Program Listings and Sample Output

Included in this section are listings and sample output, where applicable, for each of the programs and subroutines which contributed to the solution of the lunar radiometric center problem. The order of appearance is as follows:

1. Mainline Program JAKE

   Sample teletype output
   Sample printer output

2. Subroutine BIGNT

   Sample teletype output

3. Function Subprograms FLUX and RDFCN

4. Intermediate Mainline Program HERB

   Sample plotted output.

LISTING OF

MAINLINE PROGRAM JAKE

C-ERRS...STNO.C..... F O R T R A N S O U R C E S T A T E M E N T S .... IDENTFCN **COMPILER MESSAGES**

```
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C     PROGRAM NAME - JAKE                                               C
C                                                                       C
C     UTILIZATION - DETERMINES LUNAR PHASE ANGLE, SIGMA, FROM LUNAR     C
C                   AND SOLAR RIGHT ASCENSIONS, DECLINATIONS, EARTH     C
C                   DISTANCES, ETC. FOR THE SPECIFIED TIME AND LOCA-    C
C                   TION OF THE LAUNC.H  SIGMA IS THEN OUTPUT TO SUB-   C
C                   ROUTINE 'BIGNT', WHERE THE LUNAR BRIGHTNESS FUNC-   C
C                   TION IS EVALUATED.  UPON RETURN TO 'JAKE', THE      C
C                   LUNAR CENTER OF BRIGHTNESS ANGLE, THETA-0 IS DE-    C
C                   TERMINED, FROM WHICH THE TRACKER OFFSET ANGLE, EP-  C
C                   SILON IS CALCULATED.                                C
C                                                                       C
C     SUBROUTINES REQUIRED - 'BIGNT', BRIGHTNESS FUNCTION INTEGRATOR    C
C                            'ROT', THREE-DIMENSIONAL COORDINATE SYS-   C
C                              TEM ROTATION SUBROUTINE                  C
C                            'DMS2R' AND 'HMS2R', CONVERSION ROUTINES   C
C                              FOR EPHEMERIS HOURS AND DEGREES TO       C
C                              RADIANS                                  C
C                                                                       C
C     CARD INPUT - LAUNCH DESCRIPTION HEADER CARD                       C
C                  DAY OF YEAR AND TIME IN HOURS AND MINUTES OF LAUNCH  C
C                  RIGHT ASCENSIONS, DECLINATIONS AND EARTH DIS-        C
C                    TANCES OF SUN FOR DOY AND DOY+1                    C
C                  SIDEREAL TIME FOR 0 HOURS ON DOY                     C
C                  RIGHT ASCENSIONS, DECLINATIONS AND EARTH DIS-        C
C                    TANCES OF MOON FOR HOURS AND HOURS+1               C
C                  MOON-EARTH DISTANCE POLYNOMIAL COEFFICIENTS          C
C                  LATITUDE AND LONGITUDE OF LAUNCH SITE                C
C                  DELTA ALPHA AND DELTA THETA FOR BRIGHTNESS FUNC-     C
C                    TION INTEGRATION                                   C
C                                                                       C
C     MARDA BARTHULI                                                    C
C     BALL BROTHERS RESEARCH CORPORATION                                C
C     BOULDER, COLORADO         SEPTEMBER, 1974                         C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
      COMMON SIGMA, RAD, DM
      REAL INTRP, LAMDA
      INTEGER DOY
C
      DATA AU/1.496E8/, SF/1.0E-6/
      DATA OMEGA/15.04107/
      DATA PI/3.141592654/
C
      DIMENSION ISTUF(36),A(4),BM(3),BS(3),BV(3),BD(3),EO(3),BE(3)
      DIMENSION B(180)
C
      R2D(X) = X*180.0/PI
      D2R(X) = X*PI/180.0
      INTRP(X1,X2,Y1,Y2,X) = (X*(Y2-Y1) + X2*Y1 - X1*Y2)/(X2-X1)
      ROUND(X) = IFIX(X*100.0 + 0.5)/100.0
C                                                                       C
```

C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S .... IDENTFCN **COMPILER MESSAGES**

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
C     DO SOME INITIALIZING                                          C
C                                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
      CALL TIMEX(0,LL,KK)
C
      RAD = 1738.0
C
C
5     READ(2,4000,END=500) ISTUF
      READ(2,4001) DOY, IH, M
      WRITE(1,8000)
      WRITE(5,8001)
      WRITE(5,4007) ISTUF
      WRITE(1,4000) ISTUF
      WRITE(5,4100) DOY, IH, M
C
      UT = IH + M/60.0
      P = UT/12.0
      IF(IH .GT. 12) P = P-1.0
C                                                                   C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
C     DO SUN'S R.A., DEC., AND DISTANCE INTERPOLATIONS...READ THETO  C
C                                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
1     READ(2,4002) IDAY,IRH1,IRM1,RS1,IDH1,IDM1,DS1,SDS1
      READ(2,4002) LDAY,IRH2,IRM2,RS2,IDH2,IDM2,DS2,SDS2
C
      IF(IDAY .EQ. DOY .AND. LDAY .EQ. DOY+1) GO TO 2
      WRITE(5,4003)
      PAUSE 1234
      GO TO 1
C
2     READ(2,4009) ITH,ITM,TS
      THETO = (ITH + (ITM + TS/60.0)/60.0)*15.0
C
      WRITE(5,4200)
      WRITE(5,4300) IRH1,IRM1,RS1,IDH1,IDM1,DS1,SDS1
      WRITE(5,4300) IRH2,IRM2,RS2,IDH2,IDM2,DS2,SDS2
      WRITE(5,4400) ITH,ITM,TS
C
      SRA = INTRP(0.0,24.0,HMS2R(IRH1,IRM1,RS1),HMS2R(IRH2,IRM2,RS2),UT)
C
      SDEC = INTRP(0.0,24.,DMS2R(IDH1,IDM1,DS1),DMS2R(IDH2,IDM2,DS2),UT)
C
      DS = INTRP(0.0,24.0,SDS1,SDS2,UT)
      DS = DS*AU
C                                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
C     DO INTERPOLATIONS FOR MOON'S R.A. AND DEC., AND COMPUTE DISTANCE C
```

C-ERRS...STNO.C..... F O R T R A N S O U R C E S T A T E M E N T S .... IDENTFCN **COMPILER MESSAGES**

```
      C     FROM MOON TO EARTH USING A-COEFFICIENTS AND P                   C
      C                                                                     C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                     C
      3     READ(2,4002) IM, IRH1, IRM1, RS1, IDH1, IDM1, DS1
            READ(2,4002) LM, IRH2, IRM2, RS2, IDH2, IDM2, DS2
      C
            IF(IM .EQ. IH .AND. LM .EQ. IH+1) GO TO 4
            WRITE(5,4005)
            PAUSE 2345
            GO TO 3
      C
      4     READ(2,4004) A
      C
            WRITE(5,4500)
            WRITE(5,4600) IRH1,IRM1,RS1,IDH1,IDM1,DS1
            WRITE(5,4600) IRH2,IRM2,RS2,IDH2,IDM2,DS2
            WRITE(5,4700) A
      C
            RAM = INTRP(0.0,60.0,HMS2R(IRH1,IRM1,RS1),HMS2R(IRH2,IRM2,RS2),FLO
           1AT(M))
      C
            DECM = INTRP(0.0,60.0,DMS2R(IDH1,IDM1,DS1),DMS2R(IDH2,IDM2,DS2),FL
           1OAT(M))
      C
            DO 10 J=2,4
            A(J) = A(J)*SF
      10    CONTINUE
      C
            DM = (((A(4)*P + A(3))*P + A(2))*P + A(1))*6378.16
      C
            WRITE(5,4006) SRA, SDEC, DS, RAM, DECM, DM
      C                                                                     C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                     C
      C     CALCULATE VECTORS M, S, EO, E, V AND D                          C
      C                                                                     C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                     C
            BM(1) = SIN(DECM)*DM
            BM(2) = COS(DECM)*COS(RAM)*DM
            BM(3) = COS(DECM)*SIN(RAM)*DM
      C
            BS(1) = SIN(SDEC)*DS
            BS(2) = COS(SDEC)*COS(SRA)*DS
            BS(3) = COS(SDEC)*SIN(SRA)*DS
      C
            READ(2,4008) LAMDA, PHI
      C
            EO(1) = SIN(D2R(PHI))
            EO(2) = COS(D2R(PHI))*COS(D2R(LAMDA))
            EO(3) = COS(D2R(PHI))*SIN(D2R(LAMDA))
      C
            THETA = D2R(OMEGA*UT + THETO)
            CALL ROT(-THETA, 1, EO, BE, IER)
      C
```

C-ERRS...STNO.C..... F O R T R A N S O U R C E S T A T E M E N T S .... IDENTFCN **COMPILER MESSAGES**

```
            DO 30 J=1,3
            BD(J) = BM(J) - BE(J)
            BV(J) = BM(J) - BS(J)
    30      CONTINUE
    C                                                                  C
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    C                                                                  C
    C       CALCULATE PHASE ANGLE SIGMA                                C
    C                                                                  C
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    C                                                                  C
            DEN1 = 0.0
            DEN2 = 0.0
            PROD = 0.0
    C
            DO 40 J=1,3
            DEN1 = DEN1 + BV(J)**2
            DEN2 = DEN2 + BD(J)**2
            PROD = PROD + BV(J)*BD(J)
    40      CONTINUE
            DENOM = SQRT(DEN1*DEN2)
    C
            SIGMA = R2D(EACOS(PROD/DENOM))
    C                                                                  C
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    C                                                                  C
    C       PRINT ALL THIS STUFF OUT ON THE LINE PRINTER               C
    C                                                                  C
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    C                                                                  C
            WRITE(5,5004) PHI, LAMDA
            WRITE(5,5000) BM(1), BS(1), EO(1)
            WRITE(5,5001) BM(2), BS(2), EO(2), BM(3), BS(3), EO(3)
            WRITE(5,5002) BE(1), BV(1), BD(1)
            WRITE(5,5001) BE(2), BV(2), BD(2), BE(3), BV(3), BD(3)
    C
            WRITE(5,5003) SIGMA
    C                                                                  C
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    C                                                                  C
    C       DO THE INTEGRATING OF THE BRIGHTNESS FUNCTION FOR THIS SIGMA C
    C                                                                  C
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    C                                                                  C
            READ(2,4008) DALPH, DTHET
    C
            CALL BIGNT(DALPH, DTHET, B, J)
            BB1 = B(J)
    C                                                                  C
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    C                                                                  C
    C       CALCULATE THE EPSILON ANGLE FROM THE HALF BRIGHTNESS ANGLE THETO C
    C                                                                  C
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    C                                                                  C
            HALF = B(J)/2.0
```

ORIGINAL PAGE IS OF POOR QUALITY

C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S .... IDENTFCN   **COMPILER MESSAGES**

```
      C
              THET = DTHET/2.0
              IF(SIGMA .LE. 90.0) THET = THET + SIGMA - 90.0
              THETA = THET
      C
              DO 60 I=1,J
              IF(B(I) - HALF) 11,22,23
      11      A1 = THETA
              THETA = THET + I*DTHET
      60      CONTINUE
      C
      22      A0 = THETA
              GO TO 70
      C
      23      A2 = THETA
              A0 = (HALF*(A2-A1) + B(I)*A1 - B(I-1)*A2)/(B(I) - B(I-1))
      C
      70      EPS = ATAN(RAD*SIN(D2R(A0))/(DM - RAD*COS(D2R(A0))))
              EPS = R2D(EPS)*60.0
              EPSI = ROUND(EPS)
      C
              WRITE(5,6200) A0
              WRITE(5,6300) EPS
              WRITE(1,6900) DOY, IH, M
              WRITE(1,6800) PHI, LAMDA
              WRITE(1,3000) SIGMA
              WRITE(1,6000) EPSI
      C
              GO TO 5
      C
      500     CALL TIMEX(1,LL,KK)
              XK = KK/100.0
              WRITE(5,6100) LL, XK
      C
              CALL EXIT
      C
      C                                                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                   C
      C       FORMAT STATEMENTS                                           C
      C                                                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                   C
      C
      C
      3000    FORMAT(//'LUNAR PHASE ANGLE = ',F8.4,' DEGREES')
      C
      4000    FORMAT(36A2)
      C
      4001    FORMAT(I3,2(1X,I2))
      C
      4002    FORMAT(I3,1X,2(I4,I3,1X,F6.3),F10.7)
      C
      4003    FORMAT('0SCREW UP ON DOY FOR SUN DATA...TRY AGAIN.')
      C
      4004    FORMAT(4F10.0)
```

C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S .... IDENTFCN   **COMPILER MESSAGES**

```
      C
      4005  FORMAT('0SCREW UP ON HOURS FOR MOON DATA...TRY AGAIN')
      C
      4006  FORMAT(//'0','SUN'/' ',5X,'R.A. = ',E14.7,' DEC. = ',E14.7,' AND E
           1ARTH DISTANCE = ',F14.7/'0','MOON'/' ',5X,'R.A. = ',E14.7,' DEC. =
           2 ',E14.7,' AND EARTH DISTANCE = ',E14.7/)
      C
      4007  FORMAT('1', 36A2)
      C
      4008  FORMAT(2F10.5)
      C
      4009  FORMAT(I4,I3,1X,F6.3)
      C
      4100  FORMAT('0','DOY' I3,5X,'  TIME = ',I2,'H',2X,I2,'M, UNIVERSAL TIME
           2')
      C
      4200  FORMAT(//'0','EPHEMERIS DATA, SUN')
      C
      4300  FORMAT('0',5X,'RIGHT ASCENSION = ',I4,I3,1X,F6.3,2X,', DECLINATION
           1 = ',I4,I3,1X,F6.3,' AND DISTANCE = ',F10.5)
      C
      4400  FORMAT(//'0','EPHEMERIS THETA-0 ANGLE = ',I4,I3,1X,F6.3)
      C
      4500  FORMAT(//'0','EPHEMERIS DATA, MOON')
      C
      4600  FORMAT('0',5X,'RIGHT ASCENSION = ',I4,I3,1X,F6.3,2X,' AND DECLINAT
           1ION = ',I4,I3,1X,F6.3)
      C
      4700  FORMAT('0',5X,'MOON DISTANCE POLYNOMIAL COEFFICIENTS = ',F9.6,3(2X
           1,F10.1))
      C
      5000  FORMAT(//'0',5X,'M VECTOR = ',E14.7,5X,'S VECTOR = ',E14.7,5X,'EO
           1VECTOR = ',E14.7)
      C
      5001  FORMAT(2('0',16X,E14.7,16X,E14.7,17X,E14.7/))
      C
      5002  FORMAT(//'0',5X,'E VECTOR = ',E14.7,5X,'V VECTOR = ',E14.7,6X,'D V
           1ECTOR = ',E14.7)
      C
      5003  FORMAT(//'0','SIGMA = ',E14.7)
      C
      5004  FORMAT(//'0','LATITUDE AND LONGITTUDE OF LAUNCH SITE = ',E14.7,'
           1AND ',E14.7,', RESPECTIVELY')
      C
      6000  FORMAT('TRACKER OFFSET ANGLE = ',F5.2,' ARC-MINUTES')
      C
      6100  FORMAT(//'0','EXECUTION TIME = ',I3,1X,F5.2)
      C
      6200  FORMAT('0','THETO, CENTER OF INTENSITY ANGLE = ',E14.7)
      C
      6300  FORMAT('0','EPSILON, LOS ANGLE = ',E14.7,' ARC-MINUTES')
      C
      6800  FORMAT(//'LATITUDE OF THE LAUNCH SITE = ',F7.3,' DEGREES' /'LONGIT
           1UDE OF THE LAUNCH SITE = ',F8.3,' DEGREES')
      C
      6900  FORMAT(//'DOY = ',I3,' AND LAUNCH TIME = ',I2,'H',1X,I2,'M, UNIVER
```

```
C-ERRS...STNO.C.....  F O R T R A N   S O U R C E   S T A T E M E N T S  ........  IDENTFCN   **COMPILER MESSAGES**

         1SAL TIME')
      C
      8000  FORMAT(/////)
      C
      8001  FORMAT('1')
      C
      C
            END
VARIABLE ALLOCATIONS
  BLANK COMMON BLOCK
  SIGMA(R*6 C)=7FFD          RAD(R*6 C)=7FFA          DM(R*6 C)=7FF7

 EQUIVALENCES & INTERNAL VARIABLES
  LAMDA(R*6  )=0200      DOY(I*2  )=0203       PI(R*6  )=0204       LL(I*2  )=0207       KK(I*2  )=0208
     IH(I*2  )=0209        M(I*2  )=020A       UT(R*6  )=0208        P(R*6  )=020E     IDAY(I*2  )=0211
   IRH1(I*2  )=0212     IRM1(I*2  )=0213      RS1(R*6  )=0214     IDH1(I*2  )=0217     IDM1(I*2  )=0218
    DS1(R*6  )=0219     SDS1(R*6  )=021C     LDAY(I*2  )=021F     IRH2(I*2  )=0220     IRM2(I*2  )=0221
    RS2(R*6  )=0222     IDH2(I*2  )=0225     IDM2(I*2  )=0226      DS2(R*6  )=0227     SDS2(R*6  )=022A
    ITH(I*2  )=022D      ITM(I*2  )=022E       TS(R*6  )=022F    THETO(R*6  )=0232      SRA(R*6  )=0235
   SDEC(R*6  )=0238       DS(R*6  )=023B       AU(R*6  )=023E       IM(I*2  )=0241       LM(I*2  )=0242
    RAM(R*6  )=0243     DECM(R*6  )=0246        J(I*2  )=0249       SF(R*6  )=024A      PHI(R*6  )=024D
  THETA(R*6  )=0250    OMEGA(R*6  )=0253      IER(I*2  )=0256     DEN1(R*6  )=0257     DEN2(R*6  )=025A
   PROD(R*6  )=025D    DENOM(R*6  )=0260    DALPH(R*6  )=0263    DTHET(R*6  )=0266      BB1(R*6  )=0269
   HALF(R*6  )=026C     THET(R*6  )=026F        I(I*2  )=0272       A1(R*6  )=0273       AU(R*6  )=0276
     A2(R*6  )=0279      EPS(R*6  )=027C     EPSI(R*6  )=027F       XK(R*6  )=0282    ISTUF(I*2  )=02D0-02AD
      A(R*6  )=02DA-02D1  BM(R*6  )=02E3-02DD   BS(R*6  )=02EC-02E6   BV(R*6  )=02F5-02EF   BD(R*6  )=02FE-02F8
     EO(R*6  )=0307-0301  BE(R*6  )=0310-030A    B(R*6  )=052C-0313

STATEMENT ALLOCATIONS
  3000=0562  4000=0576  4001=0579  4002=057F  4003=0589  4004=05A0  4005=05A3  4006=05BB  4007=0604  4008=0609
  4009=060C  4100=0611  4200=062E  4300=063E  4400=0668  4500=067F  4600=068F  4700=06B1  5000=06CF  5001=06EF
  5002=06FB  5003=071B  5004=0726  6000=074F  6100=0765  6200=0777  6300=078E  6800=07A5  6900=07D6  8000=07F8
  8001=07FE   R2D=0801   D2R=080E INTRP=081B ROUND=084C     5=0883     1=08D6     2=0916     3=09A6     4=09E2
    10=0A5A    30=0B38    40=0B72    11=0C21    60=0C2E    22=0C37    23=0C3D    70=0C6F   500=0CC8

FEATURES SUPPORTED
 ONE WORD INTEGERS
 EXTENDED PRECISION
 ORIGIN
 IOCS=
   PLOTTER
   1403 PRINTER
   TYPEWRITER
   CARD

CALLED SUBPROGRAMS
 EIFIX    TIMEX    HMS2R    DMS2R    EFLOT    ESIN     ECOS     ROT      ESQRT    EACOS    BIGNT    EATAN    INITS    EADD     ESUB
 ESUBX    EMPY     EMPYX    EDIV     ELD      ELDX     ESTO     ESTOX    ESBR     EDVR     EAXI     WRTYZ    CARDZ    VCHRI    SRED
 SWRT     SCOMP    SFIO     SIOAI    SIOAF    SIOF     SIOI     SUBSC    PRNZ     PAUSE    SNR      SUBIN    SEOF     FLOAT    ISUB
 LRGT     LRLE     LREQ

REAL CONSTANTS
  .180000000E 03=0530   .100000000E 03=0533   .500000000E 00=0536   .173800000E 04=0539   .600000000E 02=053C
  .120000000E 02=053F   .100000000E 01=0542   .150000000E 02=0545   .000000000E 00=0548   .240000000E 02=054B
  .637816000E 04=054E   .200000000E 01=0551   .900000000E 02=0554
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
LAUNCH AT KAUAI, HAWAII, 3 NØV. 1974, 10H 38M U.T.

DØY = 307 AND LAUNCH TIME = 10H 38M, UNIVERSAL TIME

LATITUDE ØF THE LAUNCH SITE =  22.065 DEGREES
LØNGITUDE ØF THE LAUNCH SITE = -159.781 DEGREES

LUNAR PHASE ANGLE =  42.1534 DEGREES
TRACKER ØFFSET ANGLE =  4.88 ARC-MINUTES
```

EXAMPLE OF FINAL TELETYPE OUTPUT
FROM MAINLINE PROGRAM JAKE

EXAMPLE OF PRINTED OUTPUT

FROM MAINLINE PROGRAM JAKE

LAUNCH AT KAUAI, HAWAII, 3 NOV. 1974, 10H 38M U.T.

DOY307 TIME = 10H 38M, UNIVERSAL TIME

EPHEMERIS DATA, SUN

RIGHT ASCENSION = 14 31 4.990 , DECLINATION = -14 52 24.700 AND DISTANCE = 0.99203

RIGHT ASCENSION = 14 35 1.370 , DECLINATION = -15 11 10.800 AND DISTANCE = 0.99178

EPHEMERIS THETA-0 ANGLE = 2 47 29.308

EPHEMERIS DATA, MOON

RIGHT ASCENSION = 5 27 36.739 AND DECLINATION = 22 8 15.320

RIGHT ASCENSION = 5 30 5.930 AND DECLINATION = 22 7 2.450

MOON DISTANCE POLYNOMIAL COEFFICIENTS = 58.992645 -196552.0 10632.0 -39.0

ORIGINAL PAGE IS OF POOR QUALITY

SUN
R.A. = 0.3808433E 01 DEC. = -0.2620108E 00 AND EARTH DISTANCE = 0.1483921E 09

MOON
R.A. = 0.1436349E 01 DEC. = 0.3861500E 00 AND EARTH DISTANCE = 0.3752067E 06

LATITUDE AND LONGITUDE OF LAUNCH SITE = 0.2206553E 02 AND -0.1597818E 03, RESPECTIVELY

M VECTOR =	0.1413121E 06	S VECTOR =	-0.3843702E 08	EO VECTOR =	0.3756667E 00
	0.4659011E 05		-0.1126239E 09		-0.8696515E 00
	0.3444420E 06		-0.8864911E 08		-0.3202821E 00

E VECTOR =	0.3756667E 00	V VECTOR =	0.3857833E 08	D VECTOR =	0.1413117E 06
	0.6884210E 00		0.1126705E 09		0.4658943E 05
	0.6204440E 00		0.8899356E 08		0.3444414E 06

SIGMA = 0.4215347E 02

B MATRIX BEFORE INTEGRATION

0.52741E-01	0.29119E 00	0.64657E 00	0.10990E 01	0.16405E 01	0.22639E 01	0.29669E 01	0.37448E 01	0.45989E 01	0.55229E 01
0.65169E 01	0.75789E 01	0.87068E 01	0.98954E 01	0.11148E 02	0.12461E 02	0.13826E 02	0.15246E 02	0.16721E 02	0.18240E 02
0.19809E 02	0.21417E 02	0.23064E 02	0.24746E 02	0.26461E 02	0.28211E 02	0.29981E 02	0.31772E 02	0.33581E 02	0.35404E 02
0.37237E 02	0.39077E 02	0.40918E 02	0.42757E 02	0.44590E 02	0.46413E 02	0.48222E 02	0.50013E 02	0.51781E 02	0.53522E 02
0.55233E 02	0.56910E 02	0.58548E 02	0.60144E 02	0.61687E 02	0.63187E 02	0.64635E 02	0.66026E 02	0.67363E 02	0.68631E 02
0.69833E 02	0.70968E 02	0.72032E 02	0.73023E 02	0.73939E 02	0.74779E 02	0.75540E 02	0.76221E 02	0.76820E 02	0.77337E 02
0.77770E 02	0.78119E 02	0.78382E 02	0.78559E 02	0.78651E 02	0.78657E 02	0.78577E 02	0.78411E 02	0.78160E 02	0.77825E 02
0.77407E 02	0.76907E 02	0.76326E 02	0.75665E 02	0.74926E 02	0.74112E 02	0.73223E 02	0.72263E 02	0.71233E 02	0.70137E 02
0.68976E 02	0.67753E 02	0.66471E 02	0.65134E 02	0.63744E 02	0.62305E 02	0.60820E 02	0.59292E 02	0.57725E 02	0.56122E 02
0.54487E 02	0.52824E 02	0.51167E 02	0.49428E 02	0.47702E 02	0.45963E 02	0.44215E 02	0.42460E 02	0.40704E 02	0.38949E 02
0.37199E 02	0.35459E 02	0.33730E 02	0.32018E 02	0.30325E 02	0.28655E 02	0.27011E 02	0.25396E 02	0.23813E 02	0.22265E 02
0.20755E 02	0.19285E 02	0.17858E 02	0.16476E 02	0.15142E 02	0.13857E 02	0.12623E 02	0.11443E 02	0.10317E 02	0.92476E 01
0.82348E 01	0.72801E 01	0.63844E 01	0.55482E 01	0.47720E 01	0.40560E 01	0.34003E 01	0.28048E 01	0.22691E 01	0.17929E 01
0.13754E 01	0.10157E 01	0.71282E 00	0.46557E 00	0.27255E 00	0.13224E 00	0.42946E-01	0.28029E-02	0.00000E 00	-0.13366E-38

B MATRIX AFTER INTEGRATION

0.00000E 00	0.16091E 00	0.62135E 00	0.14864E 01	0.28491E 01	0.47948E 01	0.74036E 01	0.10753E 02	0.14918E 02	0.19974E 02
0.25987E 02	0.33031E 02	0.41167E 02	0.50465E 02	0.60980E 02	0.72782E 02	0.85920E 02	0.10045E 03	0.11643E 03	0.13391E 03
0.15292E 03	0.17354E 03	0.19577E 03	0.21968E 03	0.24528E 03	0.27261E 03	0.30171E 03	0.33258E 03	0.36526E 03	0.39975E 03
0.43607E 03	0.47423E 03	0.51422E 03	0.55607E 03	0.59974E 03	0.64524E 03	0.69256E 03	0.74168E 03	0.79258E 03	0.84524E 03
0.89961E 03	0.95569E 03	0.10134E 04	0.10727E 04	0.11336E 04	0.11961E 04	0.12600E 04	0.13253E 04	0.13920E 04	0.14600E 04
0.15293E 04	0.15997E 04	0.16712E 04	0.17437E 04	0.18172E 04	0.18916E 04	0.19667E 04	0.20426E 04	0.21192E 04	0.21963E 04
0.22738E 04	0.23518E 04	0.24300E 04	0.25085E 04	0.25871E 04	0.26658E 04	0.27444E 04	0.28229E 04	0.29012E 04	0.29792E 04
0.30568E 04	0.31340E 04	0.32106E 04	0.32866E 04	0.33619E 04	0.34364E 04	0.35101E 04	0.35829E 04	0.36546E 04	0.37253E 04
0.37949E 04	0.38632E 04	0.39304E 04	0.39962E 04	0.40606E 04	0.41236E 04	0.41852E 04	0.42453E 04	0.43038E 04	0.43607E 04
0.44160E 04	0.44697E 04	0.45217E 04	0.45719E 04	0.46205E 04	0.46673E 04	0.47124E 04	0.47558E 04	0.47973E 04	0.48372E 04
0.48752E 04	0.49116E 04	0.49462E 04	0.49790E 04	0.50102E 04	0.50397E 04	0.50675E 04	0.50937E 04	0.51183E 04	0.51414E 04
0.51629E 04	0.51829E 04	0.52015E 04	0.52186E 04	0.52344E 04	0.52489E 04	0.52622E 04	0.52742E 04	0.52851E 04	0.52949E 04
0.53036E 04	0.53113E 04	0.53182E 04	0.53241E 04	0.53293E 04	0.53337E 04	0.53374E 04	0.53405E 04	0.53430E 04	0.53451E 04
0.53466E 04	0.53478E 04	0.53487E 04	0.53493E 04	0.53496E 04	0.53498E 04	0.53499E 04	0.53500E 04	0.00000E 00	-0.13366E-38

THETO, CENTER OF INTENSITY ANGLE = 0.1777014E 02

EPSILON, LOS ANGLE = 0.4881530E 01 ARC-MINUTES

EXECUTION TIME = 10 31.91

LISTING OF SUBROUTINE BIGNT

C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S .... IDENTFCN **COMPILER MESSAGES**

```
C                                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                           C
C     SUBROUTINE NAME - BIGNT                                               C
C                                                                           C
C     UTILIZATION - DETERMINES INTEGRATION RANGES AND EVALUATES            C
C          DOUBLE INTEGRAL FOR LUNAR RADIOMETRIC CENTER FUNCTION            C
C                                                                           C
C     PRINTED OUTPUT - DATA SWITCH 13 ON PRINTS THETA-0, DELTA ALPHA        C
C                        AND DELTA THETA VALUES.                            C
C                      DATA SWITCH 14 ON PRINTS ALPHA VALUES.               C
C                      DATA SWITCH 0 ON PRINTS INTEGRATION ARRAYS BE-       C
C                        FORE AND AFTER FIRST AND SECOND INTEGRATIONS.      C
C                                                                           C
C     VALUES INPUT BY MAINLINE - DELTA VALUES FOR ALPHA AND THETA,          C
C          DALP AND DTHET.                                                  C
C                                                                           C
C     VALUES RETURNED TO MAINLINE - ARRAY OF STEP-WISE SECOND INTEGRA-      C
C          TION RESULTS, B, AND NUMBER OF ELEMENTS IN ARRAY, J.             C
C                                                                           C
C     ANGLES INPUT TO THIS SUBROUTINE ARE EXPECTED TO BE TO BE IN DE-       C
C          GREES.  VALUES INPUT TO FUNCTION 'FLUX' ARE IN RADIANS.          C
C                                                                           C
C     SUBROUTINES REQUIRED - 'FLUX', INTEGRAND EVALUATOR                    C
C                            'QSF', SIMPSON'S RULE INTEGRATOR               C
C                            'LOGSW', DATA SWITCH TESTER                    C
C                                                                           C
C     MARDA BARTHULI                                                        C
C     BALL BROTHERS RESEARCH CORPORATION                                    C
C     BOULDER, COLORADO          SEPTEMBER 1974                             C
C                                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                           C
      SUBROUTINE BIGNT(DALP, DTHET, B, J)
      LOGICAL LOGSW
      COMMON SIGMA, RAD, DIAM
      DIMENSION B(181), X(90)
      DATA PI/3.141592654/, DELT/1.0E-06/
C
      SIND(X) = SIN(X*PI/180.0)
      COSD(X) = COS(X*PI/180.0)
      TAND(X) = SIND(X)/COSD(X)
C                                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                           C
C     DETERMINE INTEGRATION RANGE FOR THETA-0 DEPENDENT ON LUNAR            C
C     PHASE ANGLE, SIGMA.  THEN SET UP MAIN THETA LOOP TEST VALUE, J.       C
C                                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                           C
      SIG = SIGMA*PI/180.0
      IF(SIGMA - 90.0) 10,10,11
10    THET = (SIGMA-90.0) + DTHET/2.0
      GO TO 12
C
11    THET = DTHET/2.0
```

4-29

C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S .... IDENTFCN **COMPILER MESSAGES**

```
      C
      12    THETO = THET
            J = (90.0-THET)/DTHET +1
      C
            DO 500 K=1,J
            IF(LOGSW(13)) WRITE(1,1000) THETO
            THEO = THETO*PI/180.0
            DALPH = DALP
      C                                                                 C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                 C
      C     DETERMINE INTEGRATION RANGE FOR ALPHA, DEPENDENT ON SIGMA AND   C
      C     THETA-0.  THEN SET UP INNER ALPHA LOOP TEST VALUE, I.  ALSO     C
      C     ADJUST DELTA ALPHA TO FIT WITHIN THE RANGE EVENLY.              C
      C                                                                 C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                 C
            A = -TAND(SIGMA)*TAND(THETO)
            IF(ABS(A) - 1.0) 13,13,14
      C
      13    A = EACOS(A)
            IF((A+DELT) - PI/2.0) 15,14,14
      C
      14    ALPH = 0.0
            Q = 90.0
            GO TO 18
      C
      15    IF(SIND(SIGMA)*SIND(THETO)) 16,17,17
      C
      16    ALPH = 0.0
            Q = A*180.0/PI
            GO TO 18
      C
      17    ALPH = A*180.0/PI
            Q = 90.0
      C
      18    ALPHA = ALPH
            I = (Q-ALPHA+2*DELT)/DALPH
            DALPH = (Q-ALPHA)/I
            ALPH = ALPH + DALPH/2.0
            ALPHA = ALPH
            IF(LOGSW(13)) WRITE(1,1100) DALPH, DTHET
      C                                                                 C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                 C
      C     EVALUATE INTEGRAND FOR THE RANGE OF ALPHA.  THEN INTEGRATE      C
      C     USING 'QSF', SIMPSON'S RULE SUBROUTINE.                         C
      C                                                                 C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                 C
            DO 100 L=1,I
            PHI = EASIN(COSD(THETO)*SIND(ALPHA))
            THETA = ATAN(TAND(THETO)/COSD(ALPHA))
            X(L) = FLUX(PHI,THETA,SIG,THEO)
            IF(LOGSW(14)) WRITE(1,3000) ALPHA
            ALPHA = ALPH + DALPH*L
```

```
C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S ........ IDENTFCN   **COMPILER MESSAGES**

         100   CONTINUE
         C
               CALL QSF(DALPH,X,X,I)
               B(K) = X(I)*2.0
               THETO = THET + K*DTHET
         500   CONTINUE
         C
               IF(.NOT. LOGSW(0)) GO TO 650
               WRITE(5,4000)
               DO 600 KK=1,J,10
               MM = KK+9
               WRITE(5,4100) (B(LL), LL=KK,MM)
         600   CONTINUE
         C                                                                       C
         CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
         C                                                                       C
         C     PERFORM SECOND INTEGRATION ON DELTA THETA.  BUMP THETA-0 AND GO   C
         C     DO IT AGAIN.  RETURN TO MAINLINE AFTER COMPLETING RANGE OF        C
         C     THETA-0.                                                          C
         C                                                                       C
         CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
         C                                                                       C
         650   CALL QSF(DTHET,B,B,J)
         C
               IF(.NOT. LOGSW(0)) RETURN
               WRITE(5,5000)
               DO 700 KK=1,J,10
               MM = KK+9
               WRITE(5,4100) (B(LL), LL=KK,MM)
         700   CONTINUE
         C
               RETURN
         C
         C
         C
         C
         1000  FORMAT('THETO = ',E14.7)
         1100  FORMAT('DALPH = ',E14.7,' AND DTHET = ',E14.7)
         2000  FORMAT('I = ',I4, ' AND J = ',I4)
         3000  FORMAT(5X,'ALPHA = ',E14.7)
         4000  FORMAT('1','B MATRIX BEFORE INTEGRATION'//)
         4100  FORMAT(' ',10(E12.5,1X))
         5000  FORMAT('0','B MATRIX AFTER INTEGRATION'//)
         C
         C
         C
         C
               END
VARIABLE ALLOCATIONS
  BLANK COMMON BLOCK
  SIGMA(R*6 C)=7FFD        RAD(R*6 C)=7FFA        DIAM(R*6 C)=7FF7

 EQUIVALENCES & INTERNAL VARIABLES
     PI(R*6  )=0000       SIG(R*6  )=0003      THET(R*6  )=0006     THETO(R*6  )=0009       K(I*2  )=000C
   THEO(R*6  )=000D     DALPH(R*6  )=0010         A(R*6  )=0013      DELT(R*6  )=0016    ALPH(R*6  )=0019
      Q(R*6  )=001C     ALPHA(R*6  )=001F         I(I*2  )=0022         L(I*2  )=0023     PHI(R*6  )=0024
```

```
 THETA(R*6  )=0027          KK(I*2  )=002A          MM(I*2  )=002B          LL(I*2  )=002C          X(R*6  )=0152-0047

UNREFERENCED STATEMENTS
 2000

STATEMENT ALLOCATIONS
  1000=016D   1100=0174   2000=0184   3000=0190   4000=0198   4100=01AC   5000=01B3   SIND=01C6   COSD=01D8   TAND=01EA
    10=0223     11=0233     12=0239     13=027D     14=0291     15=029B     16=02A8     17=02B6     18=02C2    100=0347
   500=0371    600=03A5    650=03AF    700=03E2

FEATURES SUPPORTED
 ONE WORD INTEGERS
 EXTENDED PRECISION

CALLED SUBPROGRAMS
 LOGSW   ESIN    ECOS    EABS    EACOS   EASIN   EATAN   FLUX    QSF     EADD    ESUB    EMPY    EDIV    ELD     ELDX
 ESTO    ESTOX   ESBR    EDVR    SWRT    SCOMP   SIOFX   SIOF    SUBSC   SNR     SUBIN   IFIX    FLOAT   LNOT

REAL CONSTANTS
   .180000000E 03=0156     .900000000E 02=0159     .200000000E 01=015C     .100000000E 01=015F     .000000000E 00=0162

INTEGER CONSTANTS
     1=0165     13=0166     2=0167     14=0168     0=0169     5=016A     10=016B     9=016C

CORE REQUIREMENTS FOR - BIGNT
 BLANK COMMON-    10,   VARIABLES AND TEMPORARIES-    342,   CONSTANTS AND PROGRAM-    664

RELATIVE ENTRY POINT ADDRESS IS 01FC (HEX)

END OF SUCCESSFUL COMPILATION
```

```
LAUNCH AT KAUAI, HAWAII, 3 NØV. 1974, 10H 38M U.T.
THETØ = -0.4734652E 02
DALPH =  0.1069554E 01 AND DTHET =  0.1000000E 01
     ALPHA =  0.5347773E 00
     ALPHA =  0.1604332E 01
     ALPHA =  0.2673886E 01
     ALPHA =  0.3743441E 01
     ALPHA =  0.4812996E 01
     ALPHA =  0.5882550E 01
     ALPHA =  0.6952105E 01
THETØ = -0.4634652E 02
DALPH =  0.1022502E 01 AND DTHET =  0.1000000E 01
     ALPHA =  0.5112513E 00
     ALPHA =  0.1533754E 01
     ALPHA =  0.2556256E 01
THETØ = -0.4534652E 02
DALPH =  0.1026583E 01 AND DTHET =  0.1000000E 01
     ALPHA =  0.5132916E 00
     ALPHA =  0.1539874E 01
     ALPHA =  0.2566458E 01
     ALPHA =  0.3593041E 01
THETØ = -0.4434652E 02
DALPH =  0.1028412E 01 AND DTHET =  0.1000000E 01
     ALPHA =  0.5142060E 00
     ALPHA =  0.1542618E 01
     ALPHA =  0.2571030E 01
     ALPHA =  0.2108244E 02
     ALPHA =  0.2211085E 02
THETØ = -0.4334652E 02
DALPH =  0.1009642E 01 AND DTHET =  0.1000000E 01
     ALPHA =  0.5048214E 00
THETØ = -0.4234652E 02
DALPH =  0.1011884E 01 AND DTHET =  0.1000000E 01
     ALPHA =  0.5059420E 00
     ALPHA =  0.1517826E 01
     ALPHA =  0.2529710E 01
     ALPHA =  0.1264855E 02
     ALPHA =  0.1366043E 02
     ALPHA =  0.1467231E 02
     ALPHA =  0.2074362E 02
     ALPHA =  0.2175550E 02
     ALPHA =  0.2276739E 02
     ALPHA =  0.3187434E 02
     ALPHA =  0.3288623E 02
```

EXAMPLE OF INTERMEDIATE TELETYPE OUTPUT
FROM SUBROUTINE BIGNT

LISTINGS OF FUNCTION SUBPROGRAMS

FLUX AND RDFCN

C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S .... IDENTFCN   **COMPILER MESSAGES**

```
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C     SUBROUTINE NAME - FLUX                                              C
C                                                                         C
C     UTILIZATION - EVALUATES LUNAR BRIGHTNESS FUNCTION AT A SINGLE      C
C                   POINT                                                 C
C                                                                         C
C     SUBROUTINES REQUIRED - 'RDFCN', LUNAR ALBEDO FUNCTION               C
C                                                                         C
C     MARDA BARTHULI                                                      C
C     BALL BROTHERS RESEARCH CORPORATION                                  C
C     BOULDER, COLORADO          SEPTEMBER 1974                           C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
      FUNCTION FLUX(PHI,THETA,SIG,THEO)
C
      COMMON SIGMA, RAD, DIAM
C
      CPHI = COS(PHI)
      CTHET = COS(THETA)
      CXI = CPHI*CTHET
      CGAM = CPHI*COS(THETA-SIG)
      Q = RAD/DIAM
C
      FLUX = (RDFCN(CGAM,CXI)*COS(THEO))/(1.0 + Q**2-2.0*CPHI*CTHET*Q)
C
      RETURN
C
      END
```

ORIGINAL PAGE IS OF POOR QUALITY

```
VARIABLE ALLOCATIONS
  BLANK COMMON BLOCK
  SIGMA(R*6 C)=7FFD        RAD(R*6 C)=7FFA        DIAM(R*6 C)=7FF7

 EQUIVALENCES & INTERNAL VARIABLES
   FLUX(R*6  )=0000      CPHI(R*6  )=0003      CTHET(R*6  )=0006      CXI(R*6  )=0009      CGAM(R*6  )=000C
      Q(R*6  )=000F

FEATURES SUPPORTED
 ONE WORD INTEGERS
 EXTENDED PRECISION

CALLED SUBPROGRAMS
 ECOS    RDFCN   EADD    ESUB    EMPY    EDIV    ELD    ESTO    ESBR    EDVR    EAXI    SUBIN

REAL CONSTANTS
     .100000000E 01=001E    .200000000E 01=0021

INTEGER CONSTANTS
     2=0024

CORE REQUIREMENTS FOR - FLUX
 BLANK COMMON=   10,   VARIABLES AND TEMPORARIES=    30,   CONSTANTS AND PROGRAM=    94
```

C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S .... IDENTFCN **COMPILER MESSAGES**

```
      C                                                                      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                      C
      C     SUBROUTINE NAME - RDFCN                                          C
      C                                                                      C
      C     UTILIZATION - SETS UP LUNAR ALBEDO FUNCTION.  THIS FUNCTION MAY  C
      C                   ASSUME THE MOON TO BE ANY KIND OF RADIATOR DESIRED C
      C                   SUCH AS LAMBERTIAN, LOMMEL-SELLIGER, OR COMBI-     C
      C                   NATIONS OF THE TWO.                                C
      C                                                                      C
      C     MARDA BARTHULI                                                   C
      C     BALL BROTHERS RESEARCH CORPORATION                               C
      C     BOULDER, COLORADO          SEPTEMBER 1974                        C
      C                                                                      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                      C
            FUNCTION RDFCN(CGAM,CXI)
      C
            COMMON SIGMA, RAD, DIAM
      C
            RDFCN = CGAM*CXI
            RETURN
      C
            END
```

VARIABLE ALLOCATIONS
  BLANK COMMON BLOCK
  SIGMA(R*6 C)=7FFD   RAD(R*6 C)=7FFA   DIAM(R*6 C)=7FF7

EQUIVALENCES & INTERNAL VARIABLES
  RDFCN(R*6)=0000

FEATURES SUPPORTED
 ONE WORD INTEGERS
 EXTENDED PRECISION

CALLED SUBPROGRAMS
 EMPY   ELD   ESTO   SUBIN

CORE REQUIREMENTS FOR - RDFCN
 BLANK COMMON-   10,   VARIABLES AND TEMPORARIES-   4,   CONSTANTS AND PROGRAM-   18

RELATIVE ENTRY POINT ADDRESS IS 0004 (HEX)

END OF SUCCESSFUL COMPILATION

LISTING OF

INTERMEDIATE MAINLINE PROGRAM HERB

C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S .... IDENTFCN   **COMPILER MESSAGES**

```
      C                                                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                   C
      C     PROGRAM NAME - HERB                                           C
      C                                                                   C
      C     PROGRAM UTILIZATION - MAINLINE FOR LUNAR RADIOMETRIC CENTER   C
      C          PROBLEM.                                                 C
      C                                                                   C
      C          AFTER READING IN A DELTA SIGMA VALUE (SIGMA = PHASE ANGLE), C
      C          HERB RUNS SIGMA FROM 0 TO 180 DEGREES, INTEGRATING THE   C
      C          BRIGHTNESS FUNCTION FOR EACH VALUE AND PLOTTING THE RE-  C
      C          SULTING CURVE IN TERMS OF ANALYTICAL ANGLE PSI AND BRIGHT- C
      C          NESS.                                                    C
      C                                                                   C
      C     THE CENTER OF BRIGHTNESS ANGLE IS CALCULATED FOR EACH VALUE OF C
      C     SIGMA AND PRINTED ALONG WITH THE BRIGHTNESS RATIO OF FULL TO  C
      C     HALF MOON.                                                    C
      C                                                                   C
      C     MARDA BARTHULI                                                C
      C     BALL BROTHERS RESEARCH CORPORATION                            C
      C     BOULDER, COLORADO       10 SEPTEMBER 1974                     C
      C                                                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                   C
            COMMON SIGMA, RAD, DIAM
            DIMENSION B(180), BB1(19)
            DATA PI/3.141592654/
      C
      C                                                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                   C
      C     INITIALIZATION SECTION                                        C
      C     INITIALIZE LOOP VALUES AND SET UP PLOTTING FORMAT             C
      C                                                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                   C
      C
            RAD = 1738.0
            DIAM = 384403.0
            READ(2,1100) DALPH, DTHET, DSIG
      C
            KK = 180.0/DSIG + 1
            JJ = (180.0/DSIG)/2.0 + 1
      C
            WRITE(1,4100)
            CALL SCALE(.05,9.0,-90.0,0.0)
            CALL EGRID(0,-90.0,0.0,10.0,18)
            CALL EGRID(3,0.0,1.0,.1,10)
      C                                                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                   C
      C     MAIN SIGMA LOOP . . . DO INTEGRATIONS                         C
      C                                                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                   C
            SIGMA = 0.0
```

4-38

C-ERRS...STNO.C..... FORTRAN SOURCE STATEMENTS .... IDENTFCN **COMPILER MESSAGES**

```
      DO 300 II=1,KK
      CALL BIGNT(DALPH,DTHET,B,J)
      BB1(II) = B(J)
      IF(II .EQ. JJ) RATIO = (BB1(II)/BB1(1))*100.0
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C     EVALUATE CENTER OF BRIGHTNESS ANGLE FOR ONE VALUE OF SIGMA        C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
      HALF = B(J)/2.0
      THET = SIGMA - 90.0
      IF(SIGMA .GT. 90.0) THET = 0.0
      THETA = THET + DTHET/2.0
C
      DO 200 I=1,J
      IF(B(I) - HALF) 101,202,203
C
101   A1 = THETA
      THETA = THET + DTHET/2.0 + I*DTHET
C
200   CONTINUE
C
202   A0 = THETA
      GO TO 210
C
203   A2 = THETA
      A0 = (HALF*(A2-A1) + B(I)*A1 - B(I-1)*A2)/(B(I) - B(I-1))
C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C     PLOT ONE CURVE FOR ONE VALUE OF SIGMA                             C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
210   WRITE(1,4200) SIGMA, BB1(II), HALF, A0
      THET = SIGMA - 90.0
      IF(SIGMA .GT. 90.0) THET = 0.0
      THETA = THET + DTHET/2.0
C
      DO 250 JJJ = 1,J
      CALL EPLOT(-2,THETA, B(JJJ)/BB1(1))
      THETA = THET + DTHET/2.0 + JJJ*DTHET
250   CONTINUE
      CALL UP
C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C     GO DO INTEGRATIONS FOR ANOTHER VALUE OF SIGMA                     C
C     WHEN DONE WITH ALL THE SIGMAS EXCEPT SIGMA = 180, WRITE THE HALF  C
C          TO FULL MOON BRIGHTNESS RATIO AND PRETTY THE PLOT.  THEN GO  C
C          AWAY.                                                        C
C                                                                       C
```

ORIGINAL PAGE IS OF POOR QUALITY

C-ERRS...STNO.C..... F O R T R A N S O U R C E S T A T E M E N T S .... IDENTFCN **COMPILER MESSAGES**

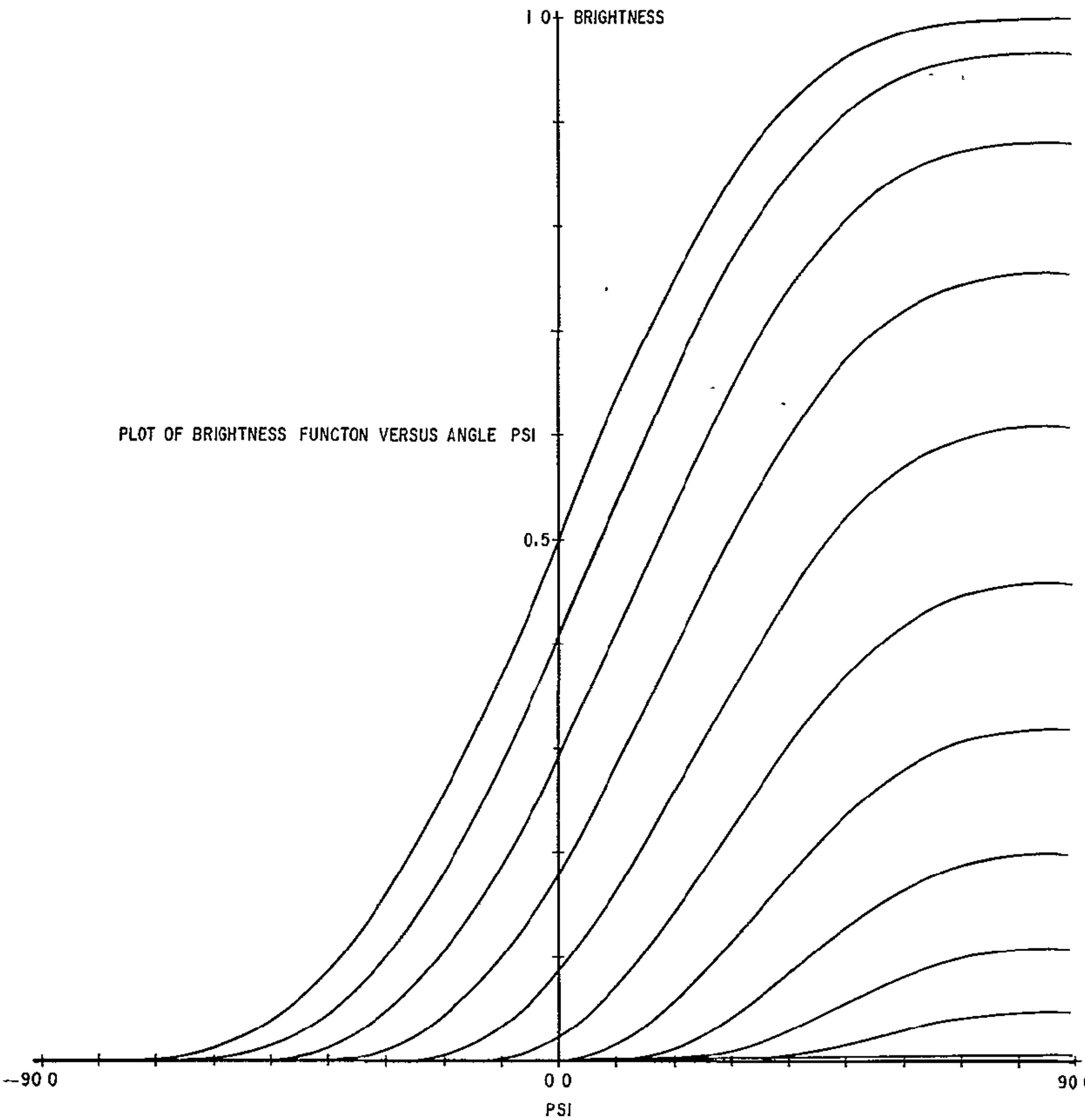
```
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                    C
            SIGMA = 0.0 + DSIG*II
            IF(SIGMA .EQ. 180.0) GO TO 400
      300   CONTINUE
      C
      400   WRITE(1,4300) RATIO
      C
            CALL ECHAR(-90.0,0.6,.1,.1,0.0)
            WRITE(7,5000)
            CALL ECHAR(-3.0,-.1,.1,.1,0.0)
            WRITE(7,5100)
      C
            XX = -90.0
            Y = -.06
            X = -89.0
            DO 205 I=1,3
            CALL ECHAR(X,Y,.1,.1,PI/2.0)
            WRITE(7,5200) XX
            XX = XX + 90.0
            X = X + 90.0
      205   CONTINUE
      C
            CALL ECHAR(5.0,.99,.1,.1,0.0)
            WRITE(7,5300)
            X = -7.0
            Y = 1.0
            XX = 1.0
            DO 206 I=1,2
            CALL ECHAR(X,Y,.1,.1,0.0)
            WRITE(7,5400) XX
            XX = XX - 0.5
            Y = Y - 0.5
      206   CONTINUE
      C
            CALL EXIT
      C                                                                    C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                    C
      C     FORMAT STATEMENTS                                              C
      C                                                                    C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                    C
      C
      1100  FORMAT(3F10.5)
      C
      4100  FORMAT(5X,'SIGMA',9X,'ENDPOINT',9X,'CENTER',10X,'CENTER'/34X,'BRIG
           1HTNESS',9X,'ANGLE')
      C
      4200  FORMAT(/4(E14.7,2X))
      C
      4300  FORMAT('BRIGHTNESS RATIO, FULL TO HALF MOON = ',E14.7)
      C
      5000  FORMAT('PLOT OF BRIGHTNESS FUNCTION VERSUS ANGLE PSI')
      C
      5100  FORMAT('PSI')
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S ........ IDENTFCN    **COMPILER MESSAGES**

          C
          5200  FORMAT(F5.1)
          C
          5300  FORMAT('BRIGHTNESS')
          C
          5400  FORMAT(F3.1)
          C
                END
VARIABLE ALLOCATIONS
  BLANK COMMON BLOCK
  SIGMA(R*6 C)=7FFD        RAD(R*6 C)=7FFA        DIAM(R*6 C)=7FF7

 EQUIVALENCES & INTERNAL VARIABLES
  DALPH(R*6  )=0200      DTHET(R*6  )=0203      DSIG(R*6  )=0206      KK(I*2  )=0209      JJ(I*2  )=020A
     II(I*2  )=020B          J(I*2  )=020C     RATIO(R*6  )=020D    HALF(R*6  )=0210    THET(R*6  )=0213
  THETA(R*6  )=0216          I(I*2  )=0219        A1(R*6  )=021A      A0(R*6  )=021D      A2(R*6  )=0220
    JJJ(I*2  )=0223         XX(R*6  )=0224         Y(R*6  )=0227       X(R*6  )=022A      PI(R*6  )=022D
      B(R*6  )=045A-0241    BB1(R*6  )=0493-045D

STATEMENT ALLOCATIONS
  1100=04D9  4100=04DC  4200=04FF  4300=0505  5000=0518  5100=0533  5200=0537  5300=0539  5400=0540  101=0614
   200=0629   202=0632   204=0638   210=066A   250=068E   300=06D9   400=06E2   205=073F   206=077D

FEATURES SUPPORTED
 ONE WORD INTEGERS
 EXTENDED PRECISION
 ORIGIN
 IOCS-
   PLOTTER
   1403 PRINTER
   TYPEWRITER
   CARD

CALLED SUBPROGRAMS
 SCALE  EGRID  BIGNT  EPLOT  UP     ECHAR  INIT$  EADD   ESUB   ESUBX  EMPY   EDIV   ELD    ELDX   ESTO
 ESTOX  ESBR   EDVR   WRTYZ  CARDZ  VCHRI  SRED   SWRT   SCOMP  SFIO   SIOFX  SIOF   SUBSC  PRNZ   SNR
 IFIX   FLOAT  ISUB   LRGT   LREQ

REAL CONSTANTS
     .173800000E 04=0496   .384403000E 06=0499   .180000000E 03=049C   .200000000E 01=049F   .500000000E-01=04A2
     .900000000E 01=04A5   .900000000E 02=04A8   .000000000E 00=04AB   .100000000E 02=04AE   .100000000E 01=04B1
     .100000000E 00=04B4   .100000000E 03=04B7   .600000000E 00=04BA   .300000000E 01=04BD   .600000000E-01=04C0
     .890000000E 02=04C3   .500000000E 01=04C6   .990000000E 00=04C9   .700000000E 01=04CC   .500000000E 00=04CF

INTEGER CONSTANTS
     2=0402      1=0403      0=0404      18=0405      3=0406      10=0407      7=0408

CORE REQUIREMENTS FOR -
 BLANK COMMON-    10,   VARIABLES AND TEMPORARIES-    662,   CONSTANTS AND PROGRAM-    754

END OF SUCCESSFUL COMPILATION                                                  COPYRIGHT UNA SYSTEMS, INC.
```

EXAMPLE OF PLOTTED OUTPUT

FROM MAINLINE PROGRAM HERB


1 0 BRIGHTNESS
PLOT OF BRIGHTNESS FUNCTON VERSUS ANGLE PSI
0.5
-90 0
0 0
90 0
PSI

Section V

SYNOPSIS OF OPERATIONS

The following steps gather, in logical order, the rather rambling developments of the preceding sections. By knowing the location of the sensor (three space coordinates and the time) one can, by following the steps given below, compute the angular offset between the center of the moon and the lunar sensor's optical axis.

These steps are, incidentally, precisely those performed by the computer program which is discussed in Section 4.

1. Compute the right ascension and declination of the sun and moon using ephemeris data and Equation 4.2.

2. Compute vectors G, H, and L (see Figure 3-9) using Equations 3.38, 3.41, and 3.39. The magnitudes of H and L are obtained from ephemeris data and the application of Equations 4.1 and 4.2.

3. From G, H, and L, compute vector D using 3.40 and vector S* using 3.42.

4. Compute the phase angle, σ, from D and S* using 3.37. If $\sigma<0$, note that the center of brightness will be to the west of the selenographic center and use the absolute value of σ in all subsequent calculations.

5. Now compute limits of integration for θ_0.

   5a. If $\sigma<\pi/2$, then $\sigma-\pi/2 \leq \theta_0 \leq \pi/2$

   5b. If $\sigma>\pi/2$, then $0 \leq \theta_0 \leq \pi/2$

6. Now compute limits of integration for α. Evaluate 3.26 to get α_0. The range of integration over α is as follows:

   6a. If 5a (above) applies, then

   - $0 \leq \alpha \leq \alpha_0$ if $\theta_0 < 0$
   - $0 \leq \alpha \leq \pi/2$ if $\theta_0 > 0$

   6b. If 5b (above) applies, then

   - $\alpha_0 \leq \alpha \leq \pi/2$ if $\theta_0 < \sigma-\pi/2$
   - $0 \leq \alpha \leq \pi/2$ if $\theta_0 \geq \sigma-\pi/2$

7. Evaluate the integral appearing on page 3-23 over the ranges of θ_0 and α obtained in steps 5 and 6. The result is a table of values of this integral as θ_0 spans its range.

8. Using this table of values, determine (using 4.2) what value of θ_0 produces half the value produced by $\theta_0=\pi/2$. Call this quantity θ_0^*.

9. Compute the sensor offset angle, ε, using θ_0^* in Equation 3.44. If the original phase angle, σ, (computed in step 4) was negative, negate ε.

Appendix A

Mathematical Notation

Coordinate systems are designated by the letter E and are distinguished from each other by superscripts. Thus, a system aligned to the celestial sphere could be called E^c while one attached to the earth could be E^e. Base vectors of coordinate systems are not termed x, y, and z but rather 1, 2, and 3 and form a right-handed set in that order. If confusion between systems is likely to occur, base vectors are referred to by subscripts on the designator of the system of which they are a part. Thus, the 2-axis of the earth-based coordinate system is termed E_2^c.

Vectors are invariably represented as 3 x 1 matrices. Thus, a vector which would, in the notation of elementary vector analysis, be written as

$$V = 3\bar{i} - 4\bar{j} + 7\bar{k}$$

is now expressed as

$$V = \begin{bmatrix} 3 \\ -4 \\ 7 \end{bmatrix}$$

The reason for this is that virtually all the vector operations that will be performed are scalar products and orthonormal rotations and the matrix formulation is just more convenient.

The scalar, or "dot", product of two vectors (for example, A and B) is written synonomously as A•B or A^TB where T denotes matrix transpose. Thus

$$A^TB = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \sum_{j=1}^{3} a_j b_j$$

Since sines and cosines are used so extensively in the definition of vectors and rotation matrices, these functions have been abbreviated to:

$$\text{sine } (\Psi) = s\Psi$$

$$\text{cosine } (\Psi) = c\Psi$$

Appendix B

Something to Read if the Mathematics are Dull

ANOTHER TIME, WE WENT TO MANNHEIM AND ATTENDED A SHIVAREE - OTHERWISE AN OPERA - THE ONE CALLED "LOHENGRIN". THE BANGING AND SLAMMING AND BOOMING AND CRASHING WERE SOMETHING BEYOND BELIEF. THE RACKING AND PITILESS PAIN OF IT REMAINS STORED UP IN MY MEMORY ALONGSIDE THE MEMORY OF THE TIME THAT I HAD MY TEETH FIXED. THERE WERE CIRCUMSTANCES WHICH MADE IT NECESSARY FOR ME TO STAY THROUGH THE FOUR HOURS TO THE END, AND I STAYED; BUT THE RECOLLECTION OF THAT LONG, DRAGGING, RELENTLESS SEASON OF SUFFERING IS INDESTRUCTIBLE. TO HAVE TO ENDURE IT IN SILENCE, AND SITTING STILL, MADE IT ALL THE HARDER. I WAS IN A RAILED COMPARTMENT WITH EIGHT OR TEN STRANGERS, OF THE TWO SEXES, AND THIS COMPELLED REPRESSION; YET AT TIMES THE PAIN WAS SO EXQUISITE THAT I COULD HARDLY KEEP THE TEARS BACK. AT THOSE TIMES, AS THE HOWLINGS AND WAILINGS AND SHRIEKINGS OF THE SINGERS, AND THE RAGINGS AND ROARINGS AND EXPLOSIONS OF THE VAST ORCHESTRA ROSE HIGHER AND HIGHER, AND WILDER AND WILDER, AND FIERCER AND FIERCER, I COULD HAVE CRIED IF I HAD BEEN ALONE. THOSE STRANGERS WOULD NOT HAVE BEEN SURPRISED TO SEE A MAN DO SUCH A THING WHO WAS BEING GRADUALLY SKINNED, BUT THEY WOULD HAVE MARVELLED AT IT HERE, AND MADE REMARKS ABOUT IT NO DOUBT, WHEREAS THERE WAS NOTHING IN THE PRESENT CASE WHICH WAS AN ADVANTAGE OVER BEING SKINNED.

EACH SANG HIS INDICTIVE NARRATIVE IN TURN, ACCOMPANIED BY THE WHOLE ORCHESTRA OF SIXTY INSTRUMENTS; AND WHEN THIS HAD CONTINUED FOR SOME TIME, AND ONE WAS HOPING THEY MIGHT COME TO AN UNDERSTANDING AND MODIFY THE NOISE, A GREAT CHORUS COMPOSED ENTIRELY OF MANIACS WOULD SUDDENLY BREAK FORTH, AND THEN DURING TWO MINUTES, AND SOMETIMES THREE, I LIVED OVER AGAIN ALL THAT I HAD SUFFERED THE TIME THE ORPHAN ASYLUM BURNED DOWN.

I HAVE SINCE FOUND OUT THAT THERE IS NOTHING THE GERMANS LIKE SO MUCH AS AN OPERA. THEY LIKE IT, NOT IN A MILD AND MODERATE WAY, BUT WITH THEIR WHOLE HEARTS. THIS IS A LEGITIMATE RESULT OF HABIT AND EDUCATION. OUR NATION WILL LIKE THE OPERA, TOO, BY-AND-BY, NO DOUBT. ONE IN FIFTY OF THOSE WHO ATTEND OUR OPERA LIKES IT ALREADY, PERHAPS, BUT I THINK A GOOD MANY OF THE OTHER FORTY-NINE GO IN ORDER TO LEARN TO LIKE IT, AND THE REST IN ORDER TO BE ABLE TO TALK KNOWINGLY ABOUT IT. THE LATTER USUALLY HUM THE AIRS WHILE THEY ARE BEING SUNG, SO THAT THEIR NEIGHBORS MAY PERCEIVE THAT THEY HAVE BEEN TO OPERAS BEFORE. THE FUNERALS OF THESE DO NOT OCCUR OFTEN ENOUGH.

Mark Twain

Appendix C

Evaluation of Lambertian and Lommel-Seeliger Reflectance Functions

In this appendix, we wish to compute the half-moon to full-moon brightness ratios for three reflectance functions.

The power density of the moon-reflected sunlight at some observation point is

$$B = k \int_A \frac{f(\gamma, \xi)}{R^2} \, dA \qquad \text{C.1}$$

where k = a proportionality factor

R = distance from the observation point to dA

A = illuminated portion of the moon visible from the observation point

f = lunar-surface reflectance function

γ = incidence angle of sunlight at dA

ξ = reflectance angle of sunlight from dA to the observation point

If we make the assumption that the observation point to moon distance is very large compared to the lunar radius, we can treat R as constant and the visible region of the moon becomes a complete hemisphere.

γ

Under these simplified condition, equation C.1 can be written (in terms of normal spherical coordinates ϕ and θ) as

$$B(\sigma) = \frac{2kr^2}{R^2} \int_{\sigma-\pi/2}^{\pi/2} \int_{0}^{\pi/2} f(\gamma,\xi)\ s\phi\ d\phi\ d\theta \qquad \text{C.2}$$

where r = lunar radius

σ = lunar phase angle

Our approach will be to evaluate C.2 for $\sigma = 0$ (full moon) and $\sigma = \pi/2$ (half moon) with a particular function f, ratio the two terms and compare the result to the experimentally obtained value of 0.089 (ref. "Astronomical Quantities"; Section 3.4).

Lambertain Reflector

The reflectance function for a Lambertain reflector has the form

$$f(\gamma,\xi) = c\gamma\ c\xi \qquad \text{C.3}$$

Now equations 3.8 and 3.19 give as

$$c\xi = c\phi \; c\theta \tag{C.4}$$

$$c\gamma = c\phi \; c(\theta-\sigma) \tag{C.5}$$

and C.2 becomes

$$B(\sigma) = 2k\left(\frac{r}{R}\right)^2 \int_{\sigma-\pi/2}^{\pi/2} \int_{0}^{\pi/2} c^2\phi \; c\theta \; c(\theta-\sigma) \; s\phi \; d\phi \; d\theta \tag{C.6}$$

which can be separated into

$$B(\sigma) = 2k\left(\frac{r}{R}\right)^2 \int_{0}^{\pi/2} c^2\phi \; s\phi \; d\phi \int_{\sigma-\pi/2}^{\pi/2} c\theta \; c(\theta-\sigma) \; d\theta \tag{C.7}$$

Since the ϕ - integral is simply a constant, it can be absorbed, together with the $2k(r/R)^2$ term into a single super-constant, K, which leaves

$$B(\sigma) = K \int_{\sigma-\pi/2}^{\pi/2} c\theta \; \dot{c}(\theta-\sigma) \; d\theta \qquad \text{C.8}$$

This can be integrated directly to give

$$B(\sigma) = \frac{K}{2} \left\{ (\pi-\sigma) c\sigma + s\sigma \right\} \qquad \text{C.9}$$

Thus

$$\frac{B(\pi/2)}{B(0)} = \frac{1}{\pi} = 0.32 \qquad \text{C.10}$$

Lommel-Seeliger Reflector

The reflectance function for a Lommel-Seeliger reflector has the form

$$f(\gamma,\xi) = \frac{c\gamma}{c\gamma + c\xi} \qquad \text{C.11}$$

which, by using C.4 and C.5, transform into

$$f(\gamma,\xi) = \frac{c(\theta-\sigma)}{c(\theta-\sigma) + c\theta} \qquad \text{C.12}$$

and

$$B(\sigma) = 2k \left(\frac{r}{R}\right)^2 \int_{\sigma-\pi/2}^{\pi/2} \int_{0}^{\pi/2} \frac{s\phi \ c(\theta-\sigma)}{c(\theta-\sigma) + c\theta} \, d\phi \, d\theta \qquad \text{C.13}$$

As before, the ϕ - integral can be extracted and constant terms combined. Doing so yields

$$B(\sigma) = K \int_{\sigma-\pi/2}^{\pi/2} \frac{c(\theta-\sigma)}{c(\theta-\sigma) + c\theta} \, d\theta \qquad \text{C.14}$$

Rather than attempt to integrate C.13 as it stands, let us instead perform the integration twice; once with $\sigma = 0$ and once with $\sigma = \pi/2$.

$$B(0) = K\int_{-\pi/2}^{\pi/2} \frac{d\theta}{2} = K\pi/2 \qquad \text{C.15}$$

$$B(\pi/2) = K\int_{0}^{\pi/2} \frac{s\theta}{s\theta + c\theta}\, d\theta \qquad \text{C.16}$$

$$= \frac{K}{2}\left\{\theta - \ln|c\theta - s\theta|\right\}\Bigg|_{0}^{\pi/2} \qquad \text{C.17}$$

$$= K\pi/4 \qquad \text{C.18}$$

Thus,

$$\frac{B(\pi/2)}{B(0)} = 0.5 \qquad \text{C.19}$$

Something Else

In the discussions of Section 3.4, it was stated that a "modified Lommel-Seeliger" reflectance function was also investigated. It had the form

$$f(\gamma,\xi) = \frac{c^2\gamma}{c\gamma + c\xi} \qquad \text{C.20}$$

Putting this in terms of ϕ and θ gives

$$f(\gamma,\xi) = \frac{c\phi c^2(\theta-\sigma)}{c(\theta-\sigma) + c\theta} \qquad \text{C.21}$$

and the resultant brightness function , $B(\sigma)$, becomes

$$B(\sigma) = K\int_{\sigma-\pi/2}^{\pi/2} \frac{c^2(\theta-\sigma)}{c(\theta-\sigma) + c\theta}\, d\theta \qquad \text{C.22}$$

As, before, we compute B(0) and B(π/2) separately:

$$B(0) = \frac{K}{2}\int_{-\pi/2}^{\pi/2} c\theta \, d\theta \qquad \text{C.23}$$

$$= K \qquad \text{C.24}$$

$$B(\pi/2) = K\int_{0}^{\pi/2} \frac{s^2\theta}{s\theta + c\theta} \, d\theta \qquad \text{C.25}$$

The author spent his normal, self-imposed time allotment of 5 minutes trying to integrate C.25 before turning to his best beloved table of integrals. After another 10 minute search, he abandoned that approach as well in favor of a numerical solution using Simpson's rule. The integrand was evaluated at eleven points and an error of less than 10^{-5} was expected.

The result of this numerical integration gave

$$B(\pi/2) = 0.62\ k \tag{C.26}$$

so

$$\frac{B(\pi/2)}{B(0)} = 0.62 \tag{C.27}$$

Appendix D

Predicted Sensor Offsets
for
Kauai and White Sands Launches

Taken from the letter:

"Radiometric Center of the Moon Study"
Charlie Rose to Morris Gisser
20 September 1974

The sensor offset angle, ε, has been calculated for two launches; Kauai, Hawaii and White Sands, New Mexico. The offsets were calculated for the opening, center, and closing of the launch window in each case.

Location	Time (U.T.)	Offset Angle, ε
Kauai, Hawaii 3 November 1974	$10^h\ 38^m$	4.88 arc minutes
	$11^h\ 20^m$	4.92 " "
	$12^h\ 3^m$	4.98 " "
WSMR 28 December 1974	$5^h\ 3^m$	1.55 arc minutes
	$5^h\ 41^m$	1.51 " "
	$6^h\ 19^m$	1.47 " "