Implementation of Scene Shadows in the Target Acquisition TDA (TARGAC)

by Ernest A. Carroll, Melanie J. Gouveia, Daniel A. DeBenedictis, Donald J. Hamann, Paul F. Hilton, Donald B. Hodges, Leigh N. Matheson, Steven J. McKay, Michael J. Oberlatz, Kevin J. Rappoport, and Jerry C. VanWert
Pacific-Sierra Research Corporation
Hughes-STX Corporation

ARL-CR-194
November 1994

Approved for public release; distribution is unlimited.
The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

DESTRUCTION NOTICE — For classified documents, follow the procedures in DoD 5200.22-M. Industrial Security Manual, Section 11-19 or 5200.1-R, Information Security Program Regulation, Chapter IX. For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.
Implementation of Scene Shadows in the Target Acquisition TDA (TARGAC)

The performance of electro-optical weapon systems operating the visible and near-infrared (IR) spectra depend on the illumination of the target and background. Scene shadows can decrease the illumination level of the target scene and alter the contrast characteristics between the target and background. Shadows are also an important source of clutter in a target scene. The primary objective of the first year of this Phase II SBIR is to (1) develop the partly cloudy shadowing model, (2) deliver the updated TARGAC software for the PC that implements the partly cloudy shadowing model, (3) determine the relationships between target acquisition, small-scale features, shadows, and clutter, and (4) develop a design approach for the incorporation into TARGAC of the temporal and spatial characteristics of clutter.
ACKNOWLEDGMENTS

The authors would like to acknowledge and thank the following people for their contributions.

Dr. Patti Gillespie, ARL/BED, for many helpful suggestions and for supporting the technical efforts described here.

Mr. Max Lorenzo, NVESD, for supporting this effort by supplying needed unclassified electro-optical and IR imagery.
# TABLE OF CONTENTS

## LIST OF FIGURES

iv

## LIST OF TABLES

v

## 1. INTRODUCTION

1-1

1.1. PROJECT OVERVIEW ................................. 1-1

1.2. BACKGROUND .......................................... 1-2

1.3. REPORT ORGANIZATION ............................ 1-4

## 2. TECHNICAL OBJECTIVES

2-1

2.1. TASK 1: SHADOWING BY CLOUDS ................. 2-1

2.1.1. Subtask 1.1: Incorporation of Revised FASCAT Model .......... 2-1

2.1.2. Subtask 1.2: Modification of Input Requirements .......... 2-1

2.1.3. Subtask 1.3: Provision of Radiance in Cloud Shadow and Direct Light ........................................ 2-2


2.1.5. Subtask 1.5: Provision of Bracketing Ranges ............ 2-2

2.2. TASK 2: SHADOWING BY SMALL-SCALE FEATURES ....... 2-2

2.2.1. Subtask 2.1: Target Acquisition Relationship: Small Scale Features, Shadows, and Clutter ............ 2-3

2.2.2. Subtask 2.2: Algorithm Development and Software Design ...... 2-3

## 3. TECHNICAL APPROACH

3-1

3.1. TASK 1: SHADOWING BY CLOUDS ................. 3-1

3.1.1. Input ............................................... 3-3

3.1.2. FASCAT Calculations ................................ 3-4

3.1.3. Output ............................................. 3-10

3.2. TASK 2: SHADOWING BY SMALL-SCALE FEATURES ....... 3-11

3.2.1. Target Acquisition Relationships ................. 3-11

3.2.1.1. Theory of Detection Overview .................. 3-14

3.2.1.2. The Congestion/Complexity Factor ................ 3-17

3.2.2. Algorithm Development and Software Design Approach .... 3-17

3.2.2.1. Calculating the Clutter Index .................. 3-18

3.2.2.2. Calculating the Signal-to-Clutter Ratio ............ 3-21

3.2.2.3. Calculating the Probability of Detection & Recognition 3-22

## 4. IMPACT OF THE REVISIONS

4-1

4.1. TASK 1: IMPACTS ON USERS ....................... 4-1

4.1.1. Loading and Compiling ............................ 4-1

4.1.2. Input ............................................... 4-3

4.1.2.1. Interactive Mode .................................. 4-3

4.1.2.2. Batch Mode ..................................... 4-5

4.1.3. Output ............................................. 4-13

4.2. TASK 1: IMPACTS ON ANALYSTS .................. 4-15

4.2.1. Delta-Eddington Radiative Transfer Model ............ 4-15

4.2.2. Inherent Contrast .................................. 4-25

4.2.3. Contrast Transmittance ............................ 4-26

4.2.4. Sensor Performance Model .......................... 4-27
LIST OF FIGURES

Figure 1. The shadowing problem for partly cloudy condition ........................................ 3-2
Figure 2. Assignment of clouds to atmospheric layers .................................................... 3-7
Figure 3. Diffuse radiance calculation for two partly cloudy layers .............................. 3-9
Figure 4. Probability factors ......................................................................................... 3-16
Figure 5. Clutter Index Overview .................................................................................. 3-19
Figure 6. Plate 1 ........................................................................................................... 3-21
Figure 7. Plate 2 .......................................................................................................... 3-22
Figure 8. Targets and target shadow ............................................................................ 3-24
Figure 9. Sensor-target geometry ................................................................................. 4-28
Figure 10. The PSR Test Discrepancy Report Form ...................................................... 5-10
Figure 11. TARGAC small-scale feature shadowing testing schedule ......................... 5-12
LIST OF TABLES

TABLE 1. Average Optical Depths for Various Clouds .......................................... 3-4
TABLE 2. Delta-Eddington Optical Depths for Average Clouds ................................ 3-5
TABLE 3. Geometric Thicknesses for Various Clouds ............................................ 3-5
TABLE 4. Background clutter/complexity modeling approaches ............................... 3-12
TABLE 5. Man-made and natural clutter matrix ...................................................... 3-20
TABLE 6. Assignment of Cloud Type for Climatology Data .................................... 4-5
TABLE 7. Input Files for TARGAC Sample Runs .................................................... 5-4
1. INTRODUCTION

1.1. PROJECT OVERVIEW

The performance of electro-optical weapon systems operating in the visible and near-infrared spectra depends on the illumination of the target and background. Scene shadows from clouds, large-scale features, and small-scale features can decrease the illumination level of the target scene and alter the contrast characteristics between the target and background. Thus, scene shadows can have an important effect on system performance.

Pacific-Sierra Research Corporation and Hughes STX Corporation collaborated on a Phase I Small Business Innovative Research (SBIR) project to study the feasibility of adding the effects of scene shadows to the Army's target acquisition range prediction model, TARGAC (Snapp et al., 1991). The Phase I work primarily resulted in the development of algorithms to implement scene shadow modifications in the TARGAC source code. It was assumed that actual software coding, algorithm verification, and validation would occur in a Phase II effort.

The Phase II effort is divided into a basic year and an option year. The primary objectives of the basic year were to implement the shadowing effects of partial cloud cover in the visible and near-IR sections of TARGAC and to develop a detailed approach for the implementation of small-scale feature shadows. Under this plan, the actual software implementation of small- and large-scale feature shadows and the integration of the three modules will occur in the second year.

This report focuses on the work completed by Pacific-Sierra Research Corporation and Hughes STX Corporation during the first year of the Phase II project. The work had these major objectives:

- Develop the partly cloudy shadowing model.
- Deliver the updated TARGAC software for the PC that implements the partly cloudy shadowing model. (It is assumed that updates to the UNIX version will occur in the second year of the project.)
- Determine the relationships between target acquisition, small-scale features, shadows, and clutter. (In this task we examined alternative approaches to the basic 2-d clutter and 3-d clutter caused by small-scale features and the shadows that these features produce.)
- Develop a design approach for the incorporation into TARGAC of the temporal and spatial characteristics of shadows.

The report describes the physics of the updates to TARGAC in detail, provides guidance for using the updated PC software, and provides the reader with a detailed review of the design of the planned small-scale feature shadowing implementation. PSR has included a formal software design document as appendix D of the report. A copy of the "beta" version of the software developed by Hughes STX is being shipped under separate cover to Dr. Patti Gillespie, the ARL program COTR, White Sands Missile Range, NM. For easy reference, the original version of TARGAC is referred to in this report as "TARGAC-2" and the modified version as "TARGAC-3."

1.2. BACKGROUND

Environmental conditions have played an important role in military operations throughout history. Weather can affect such wide-ranging areas as trafficability, chemical dispersion, and target acquisition. The U.S. Army uses electro-optical devices for detection and recognition of targets. These devices, which depend on the contrast between a target and its surroundings, are adversely affected by weather conditions such as rain, fog, and poor visibility.

The Army relies on tactical decision aids (TDAs) to help in understanding environmental effects on military operations. It is important that these models be as realistic as possible, while remaining easy to use. To this end, a target acquisition TDA for visual and near-infrared systems should have provisions for treating the effects of scene shadows. Shadows can decrease acquisition range by decreasing scene illumination and altering the contrast characteristics between the target and its background. Shadows cast by clouds, small-scale features, and large-scale features are important contributors to the shadowing problem.
Seagraves and Davis of the Atmospheric Sciences Laboratory (now the Army Research Laboratory) developed the TARGAC model to predict the performance of electro-optical devices for various weather conditions (1989). TARGAC includes detection and recognition range prediction models for direct view optics (DVO), image intensifiers (II), silicon television systems (SiTV), and thermal imagers (TI). This SBIR project considered only the DVO, II, and SiTV systems, which operate in the visible and near-IR wavelength region (0.4 - 1.1 \( \mu m \)).

In this wavelength region, TARGAC-2 accounted only for clear or overcast conditions. However, it is necessary to account for partly cloudy conditions because they cause a variance in irradiance that can play havoc with visual and near-IR sensor performance.

TARGAC-2 employed the methodology used in Hering's second Fast Atmospheric Scattering (FASCAT) model (1983) to determine downwelling illumination for clear or overcast situations. FASCAT was updated to account for partly cloudy situations (Hering and Johnson, 1984), and the update was included in the U.S. Air Force Electro-Optical Tactical Decision Aid (EOTDA) (Higgins et al., 1987). For more realistic modeling of scene shadows, this update has now been incorporated into TARGAC-3.

Under partly cloudy conditions, cloud shadows can cover a target and its background or have no effect at all. Since no person or model can predict the exact location of a cloud, the recommended approach is to calculate scene radiation for the target and background in cloud shadow and for the target and background in direct light. The acquisition range can then be bracketed by values for both cases. In addition, this approach includes an estimate of the probability that the target and background are in cloud shadow.

As with clouds, small- and large-scale features also create shadows. In addition, small-scale 3-d features (features of approximately target size or smaller) interact with the background to cause variances in the apparent 2-d clutter. Large-scale features can probably be best modeled using available Defense Mapping Agency (DMA) data such as Digital Terrain Elevation Data (DTED) and Digital Feature Analysis Data (DFAD). These products are available for most of the world with data sampled on an approximate 100 meter grid. A set of less than a dozen locations would provide representative terrain types that would meet most user needs. Large-scale feature shadows will be addressed in the second year of this effort.
One could envision extending the above referenced large-scale features approach for use with small-scale features by obtaining extremely detailed terrain/scene knowledge. However, the knowledge level required, the size of the databases, and the computational power required would be prohibitive. After examination of several approaches, we selected a statistical approach based on Schmeider et al. (1982) which will provide an output bracketed by values of acquisition ranges as a function of sensor-to-sun azimuth angle.

1.3. REPORT ORGANIZATION

This report contains information to assist TARGAC analysts, users, and maintenance programmers. The material is organized into two sections: a main report and a series of appendices.

The main body of the report should be useful to TARGAC users and analysts. Section 2 outlines the objectives of the first year of the Phase II SBIR. Section 3 describes the approach taken to implement partly cloudy situations and the design of the implementation of shadowing by small scale features. Section 4 details the impact of the implemented and planned revisions to the TARGAC code. Section 5 describes tests that were conducted to validate the revised code and planned testing of the small-scale shadowing code. Section 6 provides a summary of work completed and section 7 contains references.

The first four appendices to the report should be useful to those responsible for maintaining the PC version of the TARGAC code. Appendix A contains a tree diagram of TARGAC-3. Appendix B describes the modified FORTRAN routines and COMMON blocks. Appendix C contains the details of each change made. Each change is accompanied by an Engineering Change Report (ECR) and in-line documentation of the source code. Appendix D is a formal design document of the changes needed to implement shadowing by small-scale features. The implementation presented in appendix D is focused on the UNIX version of TARGAC; once operational in UNIX, no obstacles to porting a modified version of this software to the PC are seen.

The main report contains material generated by both PSR and Hughes STX. Hughes STX prepared appendices A, B, and C and PSR prepared appendix D.
2. TECHNICAL OBJECTIVES

The first year of the Phase II SBIR scene shadows project emphasized the incorporation of cloud shadows in the visible and near-IR sections of TARGAC and the design of the software to implement shadowing by small-scale features. Both small- and large-scale feature shadows will be incorporated during the second year. The technical objectives of the cloud shadows and small-scale feature shadows tasks are outlined below.

2.1. TASK 1: SHADOWING BY CLOUDS

The objective for Task 1 was to implement cloud shadowing effects in the TARGAC code. Since TARGAC-2 was already capable of handling overcast conditions, the work for this task centered on upgrading the software to include partial cloud cover for the 0.4 - 1.1 μm region. The upgrades recommended for this task in the Phase I effort (Snapp et al., 1991) were made to the PC version of the code. This involved the following subtasks.

2.1.1. Subtask 1.1: Incorporation of Revised FASCAT Model

This subtask involved updating TARGAC's delta-Eddington radiance model from the clear/overcast FASCAT model (Hering, 1983) to the partly cloudy FASCAT model (Hering and Johnson, 1984).

2.1.2. Subtask 1.2: Modification of Input Requirements

TARGAC-2 restricted the user to just one cloud layer for the visible devices. This subtask involved updating the input requirements to allow up to three cloud layers. For each layer, an amount, a type, and a base height may be specified. In addition, cloud inputs were made the same for the visible and thermal imager devices.
2.1.3. Subtask 1.3: Provision of Radiance in Cloud Shadow and Direct Light

This subtask involved using the revised FASCAT computations to produce two values of radiance, one for the target scene in cloud shadow and the other for the scene in direct illumination. Direct and diffuse radiance for the possible sky conditions are computed based on the cloud cover. "Bracketing" values for total radiance are then computed.

2.1.4. Subtask 1.4: Estimation of Cloud Shadow Probability

This subtask involved using the cloud cover information to estimate the probability that the target and its immediate background are in cloud shadow at any given time.

2.1.5. Subtask 1.5: Provision of Bracketing Ranges

Bracketing values of radiance at the target scene were used to provide values of sky-to-ground ratio, detection range, and recognition range for the scene in and out of cloud shadow. These bounding values can be used with the probability value to estimate sensor performance at any given time.

2.2. TASK 2: SHADOWING BY SMALL-SCALE FEATURES

The objective for Task 2 was to develop an approach to the inclusion of the effects of shadowing by small-scale features into TARGAC. Since TARGAC-2 does not handle small-scale feature shadows, the work for this task included: (1) the study and analysis of the relationship between small-scale feature shadows and target detection and recognition, (2) the development of efficient algorithmic approaches to the inclusion into TARGAC of small-scale feature shadows, and (3) the preparation of a software design document for upgrading TARGAC to include small-scale feature shadows in the 0.4 - 1.1 μm region. This involved the following subtasks.
2.2.1. Subtask 2.1: Target Acquisition Relationship: Small Scale Features, Shadows, and Clutter

Determine the relationship between small-scale features, shadows and clutter with respect to target acquisition. Perform static observer testing in order to validate the relationship between target acquisition and shadow-caused clutter.

2.2.2. Subtask 2.2: Algorithm Development and Software Design

Develop an approach to incorporate into TARGAC temporal and spatial characteristics of shadows, including diurnal, latitudinal, seasonal, terrain effects, and clutter.
3. TECHNICAL APPROACH

3.1. TASK 1: SHADOWING BY CLOUDS

A cloud shadow appears on the target scene whenever a cloud shields the scene from direct illumination. Only an estimate of cloud base height, thickness, type, and amount (fractional cloud cover for the target area) is practical in a forecast. Any approach to cloud shadow modeling must take this limitation into consideration.

TARGAC-2 constrained the user to a single cloud layer for the visible and near-IR wavelength region. If the cloud fraction was less than 0.7, clear conditions were assumed. If the cloud fraction was greater than or equal to 0.7, overcast conditions were assumed. The FASCAT model (Hering, 1983) was used to compute direct and diffuse illumination at the scene.

TARGAC-3 allows the user to enter up to three cloud layers (low, middle, and high) and accounts for partly cloudy conditions. This models the radiative effects of clouds more effectively. If the fractional cloud cover for all three cloud layers is 0.0, clear skies are assumed. The target scene is not shadowed by clouds. If the fractional cloud cover is 1.0 for at least one of the three cloud layers, overcast skies are assumed. The target scene is shadowed by clouds. In all other cases, partly cloudy skies are assumed.

For clear or overcast conditions, the decision as to whether the target scene is shadowed by clouds is straightforward. Under partly cloudy conditions, the target and its background may be in direct light or in cloud shadow at any particular time—it is impossible to predict the exact location of a cloud. These possibilities are shown in figure 1. It is desirable to predict acquisition ranges for both situations (target and background in direct light, target and background in cloud shadow) and estimate the probability that the target scene is shadowed by clouds.

The calculation of diffuse and direct illumination is key to the problem of partly cloudy conditions. The revised FASCAT model (Hering and Johnson, 1984), which extended the delta-Eddington solution to include the effects of partly cloudy conditions, has been incorporated. The basic approach is to find the average diffuse illumination from the possible cloud situations and to find "best case" and "worst case" direct illumination. The
average diffuse term is then used with the two extremes of the direct term to give acquisition ranges for the target scene in and out of cloud shadow.

Figure 1. The shadowing problem for partly cloudy conditions.

To define the effects of shadowing by clouds, changes to TARGAC-2 were required in three areas: (1) input of up to three cloud layers; (2) incorporation of the revised FASCAT model; and (3) output of bracketing ranges for the two possible cloud situations. Each is discussed below.
3.1.1. INPUT

TARGAC-2 allowed the user to enter only the cloud base height and amount for a single layer. Cloud thickness and optical parameters were preset for an "average" cloud.

Modeling the effects of clouds on radiation has been improved in TARGAC-3 by allowing the user to enter base height, amount, and cloud type for up to three cloud layers. Cloud thickness and optical parameters are defined for each particular cloud type. The addition of cloud type to the input requirements should not present a hardship to the TARGAC user. Cloud type observations and forecasts are part of many routine civilian and military weather reports that are readily available. This additional input improves the accuracy of the model. Given a cloud type and base altitude, preset values of cloud thickness, asymmetry parameter, optical thickness, and scattering ratio are assigned. These values are described in Section 3.1.2.

In addition to adding cloud type and allowing up to three cloud layers, three other issues affected the selection of inputs for TARGAC-3. Wherever possible, we attempted to use common inputs for the visible and thermal imager sections of the code. Since the thermal imager section already included three cloud layers, we mimicked the TI inputs as closely as possible. For example, we used the same cloud types for each layer and we adopted units of kilometers (km) for cloud base height rather than feet (ft).

We attempted to harmonize inputs between TARGAC-3 and the Air Force EOTDA (Freni et al., 1993). We adopted the EOTDA limits for cloud base heights. We included the Air Force definition of ceiling height: that height at which the summation of cloud cover observed from the surface is more than 4/8 (AWSR 105-24, 1983). Because the Air Force defines cloud coverage in eighths, this means that the coverage must be at least 5/8, or approximately 0.6.

We examined the inputs for the visible and near-IR systems and tried to eliminate redundancies. We removed the significant weather input. This parameter offered choices of different levels of cloud cover (now included as cloud inputs), fog (included as an aerosol selection), and precipitation, which does not directly affect the model for visible and near-IR systems.
Section 4.1.2 describes the specific changes to the TARGAC input selection and the new inputs for interactive and batch modes.

### 3.1.2. FASCAT CALCULATIONS

TARGAC-3 requires optical depths for the standard cloud types. However, there is a problem with defining a typical cloud for a particular cloud type. For example, the vertical extent of a cumulus cloud could range from a few hundred feet for fair weather cumulus to several thousand feet for cumulonimbus. Therefore, the optical depths used in TARGAC-3 represent an "average" cloud for each type.

Table 1 shows the optical depths used by Hering (1983) for various cloud types. Although he presented three representative thicknesses, the table below lists only the average optical depth for each cloud type.

**Table 1. Average optical depths for various clouds.**

<table>
<thead>
<tr>
<th>Cloud Type</th>
<th>Optical Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>St/Sc</td>
<td>22.0</td>
</tr>
<tr>
<td>Cu/Cb</td>
<td>15.0</td>
</tr>
<tr>
<td>As/Ac</td>
<td>15.9</td>
</tr>
<tr>
<td>Thick Ci/Cs</td>
<td>4.4</td>
</tr>
<tr>
<td>Thin Ci/Cs</td>
<td>1.5</td>
</tr>
</tbody>
</table>

According to Joseph et al. (1976), the delta-Eddington optical depth can be expressed as

\[ \tau' = (1 - af) \tau \]

where

- \( a(t) \) = the angle scattering albedo,
- \( f \) = the fractional scattering into the forward peak, and
- \( t \) = the standard optical depth.

Joseph et al. states that according to previous work by van de Hulst (1968) and Hansen (1969), the relationship between the fractional scattering into the forward peak and the asymmetry parameter of the truncated phase function, \( g \), is given by
\[ f = g^2. \]  \hspace{1cm} (2)

Thus, Equation (1) can be rewritten as

\[ \tau' = (1 - ag^2)\tau. \]  \hspace{1cm} (3)

If we assume that the single scattering albedo, \( a \), is 1 in the visible waveband and that the asymmetry parameter for water clouds is 0.85, Equation (3) is reduced to

\[ \tau' = (0.2775)\tau. \]  \hspace{1cm} (4)

If we apply Equation (4) to the values in Table 1, we obtain the delta-Eddington optical depths for clouds used in TARGAC-3. Table 2 shows the results.

Table 2. Delta-Eddington optical depths for average clouds.

<table>
<thead>
<tr>
<th>Cloud Type</th>
<th>Optical Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>St/Sc</td>
<td>6.1</td>
</tr>
<tr>
<td>Cu/Cb</td>
<td>4.2</td>
</tr>
<tr>
<td>As/Ac</td>
<td>4.4</td>
</tr>
<tr>
<td>Thick Ci/Cs</td>
<td>1.2</td>
</tr>
<tr>
<td>Thin Ci/Cs</td>
<td>0.4</td>
</tr>
</tbody>
</table>

In order to determine which atmospheric layers contain clouds, TARGAC-3 uses average geometric thicknesses for the standard cloud types. Table 3 shows these values.

Table 3. Geometric thicknesses for various clouds.

<table>
<thead>
<tr>
<th>Cloud Type</th>
<th>Geometric Thickness (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>St/Sc</td>
<td>0.75</td>
</tr>
<tr>
<td>Cu/Cb</td>
<td>1.5</td>
</tr>
<tr>
<td>As/Ac</td>
<td>0.5</td>
</tr>
<tr>
<td>Thick Ci/Cs</td>
<td>0.5</td>
</tr>
<tr>
<td>Thin Ci/Cs</td>
<td>0.3</td>
</tr>
</tbody>
</table>
The partly cloudy FASCAT computation of average diffuse illumination was designed to handle only one or two cloud layers (Hering and Johnson, 1984). If three cloud layers are entered, then the upper two—those farthest from the ground-based sensor—need to be combined. Cloud fraction and optical depth are averaged over the two layers. Geometric thicknesses are combined. This same procedure is used in the Air Force EOTDA (Higgins et al., 1987).

TARGAC uses 17 layers to model the atmosphere from 20 km down to the surface. Each cloud layer may fill up part or all of one or more atmospheric layers. Figure 2 depicts the 17 atmospheric layers. Two example cloud layers are shown. The upper cloud layer is a combination of a middle altostratus cloud with a thickness of 0.5 km and a high thick cirrus cloud with a thickness of 0.5 km. The combined cloud fills up part of the fifth and sixth atmospheric layers, between 6 and 8 km. The lower cloud layer is a low cumulus cloud with a thickness of 1.5 km. This cloud fills up part or all of atmospheric layers 10 through 12, between 0.5 and 3 km.

We developed a weighted averaging algorithm to assign the appropriate extinction to each atmospheric layer. The cloud extinction is added to the appropriate atmospheric layers. For each atmospheric layer $i$, the extinction is defined as

$$\beta_i = m_{clr} \beta_{clr} + m_{cldl} \beta_{cldl} + m_{cld2} \beta_{cld2}$$

and the asymmetry parameter is

$$g_i = m_{clr} g_{clr} + m_{cldl} g_{cldl} + m_{cld2} g_{cld2}$$

where

$$m_{clr} = \text{fraction of layer } i \text{ that is clear},$$

$$m_{cldl} = \text{fraction that contains part of the upper cloud},$$

$$m_{cld2} = \text{fraction that contains part of the lower cloud},$$

and the extinction coefficient, $\beta$, is related to the optical depth and the layer thickness, $z$, by
\[ \tau = \int_0^z \beta'(z') dz'. \]  

Figure 2. Assignment of clouds to atmospheric layers.
Once the appropriate optical parameters are assigned to each atmospheric layer, TARGAC-3 uses the updated FASCAT model to compute direct and diffuse radiance values. Each cloud layer is assumed to be clear (for clear skies), overcast (for overcast skies), or both (for partly cloudy skies). For partly cloudy skies, it is necessary to calculate the optical parameters and radiance values twice for the atmospheric layers containing clouds: once for a clear path and once for a cloudy path. For clear or overcast skies, these calculations need only be performed once.

The partly cloudy FASCAT model uses a weighted averaging technique of possible cloud situations to obtain the diffuse component of illumination. The following conditions are considered:

1. For clear skies, the diffuse radiance calculation is performed once using clear layers.

2. For one overcast cloud layer (either upper or lower), the radiance is calculated once with clouds in the appropriate atmospheric layer(s).

3. For one partly cloudy layer (either upper or lower), the diffuse radiance calculation is performed twice: once with clouds in the layer(s) and once without. A weighted average of the two is then computed based on the fractional cloud cover.

4. For two overcast layers, the calculation is performed once with clouds in all applicable atmospheric layers.

5. If one of the upper or lower layers is partly cloudy (either upper of lower), the calculation is performed twice and a weighted average is computed.

6. If both layers are partly cloudy, four diffuse radiance calculations are performed: one with two clear layers (CR), one with a lower overcast (L), one with an upper overcast (U), and one with two overcast layers (UL). A weighted average of the four is then computed based on the cloud fractions.
Figure 3 shows this last situation.

Figure 3. Diffuse radiance calculation for two partly cloudy layers.

The direct component of illumination depends on the cloud situation. For clear skies, the direct radiance calculation is performed using clear atmospheric layers. For overcast skies, the calculation is performed using layers with clouds. For partly cloudy skies, the best case direct radiance calculation uses layers without clouds and the worst case calculation uses layers with clouds.

The average diffuse radiance, best and worst case direct radiance, and corresponding optical parameters are used to compute total radiance incident on the target and background, transmission, path radiance, sky-to-ground ratio, and acquisition ranges twice: once for a clear path and once for a cloudy path. The cloud fractions and source zenith angle are then used to estimate the probability that the target scene is in shadow.
Section 4.2 describes the specific changes to the FASCAT calculations and the effects on other parts of the TARGAC model.

### 3.1.3. OUTPUT

Once the diffuse and direct components of illumination are calculated, these values are used to determine the total radiance at the target, the transmitted radiance, the sky-to-ground ratio, the apparent contrast, and the detection and recognition ranges. The average diffuse radiance component is used with the best case or worst case direct component for these calculations. Output is produced for the possible situations (target and background in direct light, target and background in cloud shadow, or both). The result is a single set of output for clear or overcast cases and two sets for partly cloudy cases.

Although under partly cloudy conditions the actual cloud positions cannot be predicted with accuracy, the acquisition ranges and sky-to-ground ratio can be bracketed by providing solutions for the target scene in and out of cloud shadow. The actual values should be between these limits. The probability of being in cloud shadow provides an indicator of the most likely values.

Section 4.1.3. describes the specific changes to the TARGAC output and the values produced.
3.2. TASK 2: SHADOWING BY SMALL-SCALE FEATURES

Shadows caused by small-scale features have an important effect on target acquisition. By small-scale, we mean an object whose size, and to some degree appearance, is on the order of the target's size and/or appearance. Examples include individual trees, clumps of bushes, rock formations, boulders, dunes, or man-made target-like objects. The importance of the small-scale feature is that these objects add clutter to the scene. In most cases, target acquisition performance is inversely proportional to clutter. The higher the clutter, the lower the probability of detection or recognition.

Schmieder et al. (1982) described these clutter effects quite extensively. Although their tests focused on IR imaging systems, they stated that their methodology extended into the visible and near-IR spectrum. Their conclusion was that target scenes could be categorized in terms of the signal-to-clutter ratio (SCR) or the ratio of target contrast to clutter contrast. Higgins et al. (1989) implemented Schmieder's methodology in the Air Force EOTDA in a semi-automated form in which the user selects a scene clutter index in the TV model or scene complexity index in the IR model. Scene complexity in the EOTDA IR model was categorized into four levels (None, Low, Moderate, or High), while in the TV model only three levels are available (Low, Moderate, or High).

The major objective of task 2 is to implement a modified Schmieder methodology for the EO and near-IR parts of the spectrum. During the first year of this effort, we examined the target acquisition relationships, developed an approach to determining the clutter index over a variety of clutter and illumination conditions, and produced a software design document (appendix D).

3.2.1. Target Acquisition Relationships

Many other methods have been proposed for measuring and modeling scene or background clutter or complexity. Table 4 lists some of the better known approaches. The common factor among these methods is that they are computationally intensive.
Table 4. Background clutter/complexity modeling approaches.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Comment</th>
<th>References</th>
</tr>
</thead>
</table>
Table 4. Background clutter/complexity modeling approaches (continued).

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Comment</th>
<th>References</th>
</tr>
</thead>
</table>
In 1993 in a related, classified program for the Department of Defense, we examined how the probability of detecting a target from a frame of imagery changed as a function of background clutter, solar illumination, weather conditions, and sensor collection geometries (Carroll 1993). For that effort, we implemented a simplified probability of detection estimation approach first suggested in a 1970 Rand report (Bailer, 1970). As part of that program, we implemented in software and tested three variations of the clutter/complexity modeling approaches referenced in table 4. The three approaches were auto-regressive, fractal, and moving average.

The result of that effort was a practical implementation of a software solution to the problem of estimating probability of detection from collected imagery. In this program, we have and will continue to make use of the above referenced software in performing static observer testing. The objective is to establish, validate, and quantify the relationship between target acquisition and shadow-caused clutter. Previously, we have used this software to estimate the complexity of classified images. In this program, we will use this software in establishing the complexity of sample images being supplied to us by NVESD, formally the Army Night Vision Laboratory. The following section presents an overview of the Rand approach, which was adapted by Rand for the *RCA Electro-Optics Handbook* and originally published in 1974.

3.2.1.1 Theory of Detection Overview

Imaging sensor performance may be estimated and/or evaluated by application of a target detection/recognition model such as that suggested by Rand, namely

\[ P_{\text{id}} = P_1 \cdot P_2 \cdot P_3 \cdot h \]  

and

\[ P_d = P_1 \cdot P_2 \cdot h \]  

where \( P_{\text{id}} \) is the probability that a target will be identified (recognition in TARGAC terminology) on a display or piece of film and \( P_d \) is the probability of detection; \( P_1 \) is the probability that the observer, searching an area that is known to contain a target, looks with his foveal vision for a specific glimpse time in the direction of the target; \( P_2 \) is the
The probability that if the displayed target image is viewed foveally for one glimpse period, it will, in the absence of noise, have sufficient contrast and size to be detected; \( P_3 \) is the probability that if a target is detected, there will be enough detail shown for it to be recognized (again during a single glimpse and in the absence of noise); and \( h \) is the overall degradation factor arising from noise.

The probability \( P_1 \) is difficult to estimate because it is affected by

1. the solid angle presented to the eye of the search field,
2. the time available to search the image,
3. the number of confusion elements within the scene, and
4. any availability of “cues” or a priori information as to where to look on the image.

The model employs the relation

\[
P_1 = 1 - \exp\left[-\left(\frac{700}{G}\right) \times \left(\frac{A_t}{A_s}\right) \times t\right]
\]  

(10)

where

\[
A_t = \text{area of target}
\]
\[
A_s = \text{area to be searched}
\]
\[
t = \text{search time}
\]
\[
G = \text{congestion factor, usually between 1 and 10, for most real imagery of interest.}
\]

The probability of detection, \( P_2 \), at a contrast ratio (C) of \( C = 1 \), is by definition 50 percent. A useful approximation for \( P_2 \) at other contrasts available at the eye is given by

\[
P_2 = 1/2 \pm 1/2 \times \left\{ 1 - \exp\left[-4.2 \times C^2\right]\right\} 0.5
\]  

(11)

where a minus sign is used when \( C < 0 \). This equation is plotted in figure 4.

The other two factors in the basic equation (\( P_3 \& h \)) are also plotted in figure 4 and can be calculated respectively from

\[
N_r \geq 2 \quad P_3 = 1 - \exp\left[-(N_r/2 - 1)^2\right]
\]  

(12)
\[ Nr < 2 \quad P_3 = 0, \] (13)

where

\[ Nr = \frac{L_{\text{min}}}{\sin \alpha r} \] (14)

is the number of resolution cells contained in the minimum projected target dimension, \( L_{\text{min}} \), \( \alpha \) is the angular resolution of the sensor, \( r \) is the target range, and \( h \) is a function of the displayed signal-to-noise (SNR) ratio.

\[ \text{SNR} \geq 1 \quad h = 1 - \exp \left[ - (\text{SNR} - 1)^2 \right] \] (15)

\[ \text{SNR} < 0 \quad h = 0 \] (16)

Figure 4. Probability factors.
3.2.1.2 The Congestion/Complexity Factor

The Rand approach refers to G as the congestion factor of the image. One can think of the congestion of an image as its complexity or "busyness" with an empty parking lot being very homogeneous and having a very low complexity, and a mixed forest area being very rough and random and having a high complexity. Alternatively (Freni, 1993), one can describe the clutter of a "TV" system as, "The busyness of the target area, including both the number of objects comparable in size and shape to the target and the magnitude of the contrast between scene features". Thus, G maps to what we are calling the scene or background clutter (EO) or complexity (IR). In our implementation of this approach, we found that the use of a fractal dimension estimator provided the best results when compared to experimental studies using real imagery and imagery analysts. The fractal dimension estimator we used was the Hurst dimension (Feders, 1988). The Hurst dimension is a form of fractal dimension that for most real imagery has a value of between 0 and 2. In the above referenced implementation, after calculating the Hurst dimension of an image, we scaled it from the range of (1 to 2) to the desired range (1 to 10). It is our plan in this effort to use a mapping of the Hurst dimension as an approximation of the clutter and complexity. In the later discussion of our approach to 2-d and 3-D clutter, the relationship of this work to the current effort will be evident.

3.2.2. Algorithm Development and Software Design Approach

Our algorithm and software design approach to the problem of predicting the effects of small-scale feature shadows (features about the same size as the target) on the probability of detection and recognition has focused on five goals: algorithm simplicity, software modularity, clear and simple user inputs, flexible fidelity, and speed of execution. In this section, we discuss our overall design approach. Appendix D presents a detailed technical review of the software design.

At the simplest level, one can think of this package as one group of software routines that examines the scene clutter and another group of software routines that examines the target. These two groups pass their output to a third group of routines that calculates the signal-to-clutter ratio. A last group of software calculates the probability of detection and recognition of the target. Let us look at each of these software groups separately.
3.2.2.1 Calculating the Clutter Index

In the PSR approach, the largest group of software is dedicated to calculating the clutter index. Figure 5 presents a graphical overview of the functions performed in this group. Clutter is initially represented as 2-d clutter or 3-d clutter. Clutter is only considered for a small region within the scene. The size of this region must be big enough to provide accurate statistics, yet as small as possible so that it will not clog the computational process. For ease of use, the analyst will be able to select from a set of photographs the 2-d clutter desired: None, Low, Medium, or High. The Air Force EOTDA handles clutter/complexity in a similar manner.
Figure 5. Clutter Index Overview.
However, to maximize fidelity and flexibility when required, the advanced user can input the statistics of the 2-d region. 3-d clutter is handled in a similar manner: either select a clutter type from a set of photos (see table 5) or input a set of statistics for the region. Note: for 3-d clutter we will provide a matrix of photos that combine man-made and natural clutter.

Table 5. Man-made and natural clutter matrix.

<table>
<thead>
<tr>
<th>Man-Made</th>
<th>Natural</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>None-Trees</td>
</tr>
<tr>
<td>Light</td>
<td>Light-Trees</td>
</tr>
<tr>
<td>Average</td>
<td>Average-Trees</td>
</tr>
<tr>
<td>High</td>
<td>High-Trees</td>
</tr>
</tbody>
</table>

Based on the photos selected or the inputs provided by the user, synthetic regions (with the same fractal/Hurst dimension discussed in section 3.2.1.2.) of 2-d and 3-d clutter are generated based on an approximation using circles (2-d) and spheres (3-d).

We have chosen to use spheres as 3-d objects (see plate 1). This choice greatly simplifies the computations, because spheres are rotationally invariant both in azimuth and elevation. We added a SCUD missile resupply vehicle to provide a size reference in the plate. As shown in plate 2, future versions of TARGAC could use combinations of spheres to represent trees, shrubs, and even man-made objects without drastically increasing the complexity of the code.

Based on the location and time-of-day information provided by the user, sun and/or moon 2-d shadows are generated from the 3-d region. These shadows are then added to the 2-d clutter region as shown in figure 5. Both the 2-d and 3-d regions are aspect compressed to the perspective of the observing sensor. Remember that in most cases the observer will be at or near the plane of the ground. The 3-d clutter region will be projected into a 2-d region from the point-of-view of the sensor and will be added to the 2-d plus shadow region.

3-20
Plate 1

Figure 6. Plate 1.
Figure 7. Plate 2.
The clutter index $CI$ as defined below is then computed. Conceptually, a sample region as shown in figure 5 is broken into cells roughly the size of the target, and the background radiance is computed for each cell. The clutter index is the root-mean-square of variations in radiance between adjacent cells as depicted in equation (17).

$$CI = \sqrt{\frac{\sum \sigma_i^2}{N}}$$

(17)

Here the summation of variance is taken over adjacent cells and divided by the total number of adjacent components in the sum.

The CI computation can be performed in one of two ways. The first is to use the statistical information on scene objects and perform a closed form solution assuming independence between the various components. Such a closed form solution although possible is extremely complicated and very inflexible. The second is to perform a fast Monte-Carlo computation on a small, synthetically generated scene. This second approach was adopted for the research effort because of its simplicity and flexibility.

3.2.2.2. Calculating the Signal-to-Clutter Ratio

Once the clutter index has been computed, the signal-to-clutter ratio (SCR) can readily be determined. In the Schmieder approach the pixel with the highest radiance is assigned a count value of 255 and the pixel with the lowest value is assigned a count value of 0. SCR is defined as the maximum difference between the target and background divided by the rms clutter for positive contrast targets

$$SCR = \frac{|\text{Max Target Value (counts)} - \text{Bkgd Mean (counts)}|}{\text{rms clutter (counts)}}$$

(18)

while for negative contrast targets

$$SCR = \frac{|\text{Min Target Value (counts)} - \text{Bkgd Mean (counts)}|}{\text{rms clutter (counts)}}$$

(19)

However, as shown below, the problem becomes more complicated when one has to deal with both shadow and non-shadow backgrounds.
3.2.2.3 Calculating the Probability of Detection & Recognition

TARGAC currently considers targets to have at most length and width (some modules of the code only deal with a minimum target dimension), so in essence targets are cylinders and often only spheres. As long as one limits the use of TARGAC to horizontal and near horizontal line-of-sight (LOS) situations, cylinders provide an excellent form because they are rotationally invariant in azimuth. However, they are not rotationally invariant in elevation. Errors will be introduced as elevation differences increase. With the introduction of small-scale feature shadows, especially the shadow of the target itself, the calculation of the probability of detection becomes complicated. The increased complexity stems from the fact that with the introduction of shadow, the target no longer resides on a single background type. When calculating probability from a statistical basis, even highly complex backgrounds can be represented on average as a single background type. In the general case (with or without shadow), the probability of detecting a target is a function of its size (cycles on target) and the contrast with its surroundings.

![Figure 8. Targets and target shadow.](image)

Figure 8 depicts a cylindrical shaped target and its shadow. For this target, the probability of detection with shadows can be written as:
\[ P(d_{\text{wlw}}) = P(d_{\text{tlw}}) + P(d_{\text{blw}}) (1 - P(d_{\text{tlw}})) \]  \hspace{1cm} (20)

where we have used the rotational invariance of the cylinder to divide its height into two target parts. One target part with only clutter as a background (Ht) and one target part with both clutter and shadowed clutter as a background (Hb).

The above equation states that the probability of detecting the whole target \( P(d_{\text{wlw}}) \) is equal to the probability of detecting the top part \( P(d_{\text{tlw}}) \) of the target (the part of the target bounded by only clutter and target) given the whole target, plus the probability of detecting the bottom part \( P(d_{\text{blw}}) \) of the target (the part of the target bounded by clutter, target, and shadowed clutter) given the whole target, times the quantity one minus the probability of detecting the top of the target given the whole target \( (1 - P(d_{\text{tlw}})) \). This equation accounts for the interaction of the top and bottom parts of the cylinder.

The problem is that accounting for this interaction between the top and bottom parts of the target is difficult. One would like to use \( P(d_{\text{tl}}) \) and \( P(d_{\text{bl}}) \) respectively as the probability of detecting the top part of the target given only the top part of the target and the probability of detecting the bottom part of the target given only the bottom part of the target. Using these two probabilities derives the equation

\[ P(d_{\text{wlw}}) = P(d_{\text{tl}}) + P(d_{\text{bl}}) (1-P(d_{\text{tl}})) + g(b) \]  \hspace{1cm} (21)

Combining the equations (20) and (21) derives

\[ g(b) = \left[ P(d_{\text{tlw}}) + P(d_{\text{blw}}) (1 - P(d_{\text{tlw}})) \right] - \left[ P(d_{\text{tl}}) + P(d_{\text{bl}}) (1 - P(d_{\text{tl}})) \right] \]  \hspace{1cm} (22)

where \( g(b) \) is a non-zero variable dependent on the relationship between the contrast of target with the clutter background and the shadowed clutter background. One would normally expect the value of \( g(b) \) to be quite small. However, during the option year of this effort, we need to either establish a methodology for estimating \( g(b) \) or develop an evidentiary base for ignoring it.
4. IMPACT OF THE REVISIONS

4.1. TASK 1: IMPACTS ON USERS

This section is designed to assist the TARGAC user by describing how to load and compile the new software (the software written during the basic phase of this contract), how to run the software with appropriate inputs, and how to analyze the output. With the exception of some additional cloud inputs and expanded output, the code operates much as it did before. As mentioned above, TARGAC-3 has been implemented for the PC only; the UNIX version will be completed next year. There are no new hardware requirements for the PC version of TARGAC-3.

4.1.1. LOADING AND COMPILING

The TARGAC-3 code was developed using the Microsoft FORTRAN Compiler, Version 5.1. The following instructions anticipate the use of the Microsoft compiler, but the code can be compiled with other compilers. Note that compilation warning messages will appear because of unused and undefined variables that were present in TARGAC-2. The suggested way for loading and compiling the PC version of TARGAC-3 is as follows:

1. Create a working directory with two subdirectories named TARGAC and TCM2. Copy the source code for the TARGAC-3 model into the TARGAC subdirectory. Copy the source code for TCM2 into the TCM2 subdirectory. Copy all batch files and TCM2 target files into the working directory.

2. To create the TARGAC-3 executable file, enter the TARGAC subdirectory. Use the following command to compile with the Microsoft compiler:

   FL /c /Gt0 /AH *.FOR.

   Type the following Microsoft link command with the accompanying switches:

   FL /FeTARGAC3 *.OBJ /link /NOE /E/SE:1024.
Copy the newly created executable file, TARGAC3.EXE, to the working directory.

3. To create the TCM2 executable file, enter the TCM2 subdirectory. Use the following commands to compile with the Microsoft compiler:

   FL /c *.FOR
   FL /c /Od FT3B*.FOR.

Note that the superfile FT3b*.FOR cannot be compiled with the Microsoft FORTRAN optimization option; this causes the program to fail with a runtime error. Type the following Microsoft link command with the accompanying switches:


Copy the newly created executable file, TCM2.EXE, to the working directory.

4. Move to the working directory and type in the command TARGAC-3 to run the model. The batch file TARGAC-3.BAT controls execution.

The Microsoft command line switches are defined as follows:

   /c         compile without linking
   /GT0      allocate all data items to a new data segment
   /Fe       give the executable file the name that follows
   /AH       select the huge library
   /Od       disable optimization
   /link     link the object files listed
   /NOE      prevent link from searching for extended
dictionaries in the libraries
   /E        pack the executable
   /SE       set the maximum number of segments.
Note that Microsoft FORTRAN Version 5.1 by default selects the large library (/AL), uses full optimization (/Ox), generates in-line instructions, and selects coprocessor commands (/FPi87). For more details on the command line switches, see the references provided with the Microsoft FORTRAN Compiler.

4.1.2. INPUT

TARGAC input has undergone several changes. Cloud data are now entered for up to three layers: low, medium, and high. Cloud type is a new input for all cloud layers. Cloud base height limits have changed, along with the unit system for entering cloud base heights. The significant weather input was removed. Inputs have changed for both interactive and batch modes. Note: some of the changes will affect thermal imager runs. Specific changes are described below.

4.1.2.1. Interactive Mode

In interactive mode, cloud information may be specified by the user or read in from a climatological database. If the user chooses to enter cloud information, the cloud fractional amount is required for all three cloud layers. If the cloud fraction is greater than zero for a cloud layer, a cloud base height must be entered for that layer. If it is a low or high cloud, a cloud type must be entered. Since there is only one type of middle cloud supported, a middle cloud type does not have to be entered. Cloud types for the visible and near-IR systems are the same as those for thermal imager systems. The following is a list of cloud types used in TARGAC-3.

**High Cloud Types**
1. Thin Cirrus
2. Thick Cirrus

**Middle Cloud Type**
3. Altostratus/Altocumulus

**Low Cloud Types**
4. Stratus/Stratocumulus
5. Cumulus/Cumulonimbus
The cloud base height limits have changed to match those in the Air Force EOTDA. The new base height limits apply for the visible, near-IR, and thermal imager systems. Cloud base heights are always entered in kilometers, regardless of the sensor system used. The following is a list of the cloud base height limits.

**High Clouds**
6.1 - 13.7 km

**Middle Clouds**
2.0 - 6.1 km

**Low Clouds**
0.1 - 2.0 km

For the visible and near-IR systems, the user is allowed to have a low cloud on the ground. If the visibility is less than 0.55 km, a cloud is assumed to be on the ground. The user is asked to enter a cloud top height, rather than a cloud base height. The cloud fraction is automatically set to 1.0, and the cloud type is set to stratus.

If the user chooses to read in cloud information from a climatological database, only one value of cloud cover and the corresponding base height is available. If the base height is between 6.1 and 13.7 km and the cloud cover is greater than 62.5 percent, the thick cirrus type is used. The thin cirrus type is used for any smaller cloud cover. If the base height is between 2.0 and 6.1 km, the altocumulus type is used. If the base height is between 0.1 and 1.2 km and the cloud cover is greater than 50 percent, the stratus type is used. Otherwise, the cumulus type is used. These types are assigned automatically. The decision process is shown in table 6.

Cloud geometric thickness is automatically assigned based on cloud type, unless there is a low cloud on the ground. The thickness values that are used by TARGAC-3 are given in table 3, section 3.1.2. If there is a low cloud on the ground, the user specifies the cloud thickness by entering a cloud top height. A default value of 0.2 km is used.

The user is not required to specify a ceiling height. Instead, ceiling height is calculated from the cloud inputs for any of the sensor types. The ceiling is defined as that
height at which the cloud cover observed from the surface is more than 60%. This
definition is based on the definition used by the Air Force (AWSR 105-24, 1983).

Table 6. Assignment of cloud type for climatology data.

<table>
<thead>
<tr>
<th>Base Height</th>
<th>Cloud Cover</th>
<th>Cloud Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 - 13.7 km</td>
<td>&gt; 62.5%</td>
<td>Thick Cirrus</td>
</tr>
<tr>
<td>6.1 - 13.7 km</td>
<td>&lt; 62.5%</td>
<td>Thin Cirrus</td>
</tr>
<tr>
<td>2.0 - 6.1 km</td>
<td>any</td>
<td>Altocumulus</td>
</tr>
<tr>
<td>1.2 - 2.0</td>
<td>any</td>
<td>Cumulus</td>
</tr>
<tr>
<td>0.1 - 1.2</td>
<td>&gt;50%</td>
<td>Stratus</td>
</tr>
<tr>
<td>0.1 - 1.2</td>
<td>&lt;50%</td>
<td>Cumulus</td>
</tr>
</tbody>
</table>

The significant weather input was removed, since this information was covered
elsewhere. The input options for significant weather included cloud cover, blowing snow
or sand, fog, drizzle, rain, snow, or thunderstorm. Cloud cover is entered separately. Fog
can be entered as an aerosol type or as a low cloud on the ground. Precipitation is entered
as precipitation type.

The minimum value for the user-supplied background reflectance was changed
from 0.0 to 0.01. As specified in the TARGAC Users Guide (Gillespie, 1993), a value
slightly greater than 0.0 was necessary to prevent a divide-by-zero error.

4.1.2.2. Batch Mode

In batch mode, all inputs are entered in records in the TAC.DAT file. This section
describes the changes made to specific records.

Since the thermal imager section of TARGAC already required three cloud layers,
we decided to use the same batch records to implement three cloud layers in the visible and
near-IR sections: the LCLD, MCLD, and HCLD records. Note that the cloud base height
limits for these records have been changed. In TARGAC-2, the cloud information for
these systems was entered in the METD record. Data must still be entered in all fields of
the METD record, but the cloud data can be any dummy value.
As explained in section 4.1.2.1, the significant weather input was removed. This input was part of the ILUM record. As with the METD record, the data must still be entered in all fields of the record, but the data for the significant weather can be any dummy value.

The background reflectance value was moved from the METD record to the second field of the CONTEXTL record, after the CNTRST field. This value was never saved to the TAC.SAV file after an interactive mode run, causing a problem when TAC.SAV was used for a batch mode run. The minimum value for the field was changed from 0.0 to 0.01 to prevent a divide-by-zero error in the model.

The next few paragraphs describe the records that have been changed either in content or usage: CONT, HCLD, MCLD, LCLD, ILUM, and METD. These descriptions are meant to make updates to the TARGAC Users Guide (Gillespie, 1993) easier.

The **CONT** (Contrast) Record

This record is used with the DVO, II, and SiTV sensor types. The record described here designates that contrast shall be calculated from external input values.

**CNTRST** - Value of inherent contrast.
  Range: 0.1 - 1.0.

**BKREF** - Surface reflectance.
  Range: 0.1 - 1.0

```
  1 2 3 4 5 6 7
1234567890123456789012345678901234567890123456789012345678901234567890
CONTEXTL CNTRST BKREF
```

Sample of CONTEXTL Record:

```
  1 2 3 4 5 6 7
1234567890123456789012345678901234567890123456789012345678901234567890
CONTEXTL 0.5 0.15
```
The HCLD Record

This record is used with any sensor type. The WTME parameter is not used for the DVO, II, and SiTV types.

WTME - Time.
Range: 0000.0 - 2400.0 (HHMM) (GMT)
Default: 1200.0 Hr.

IWX(I,4) - Cloud indicator.
Range: 0.0 - None.
1.0 - Thin
2.0 - Thick
Default: 0.0

WX(I,9) - Cloud fraction.
Range: 0.0 - 1.0
Default: 0.0

WX(I,12) - Cloud base height.
Range: 6.1 - 13.7 km.
Default: 9.0 km.

Sample of HCLD Record:

1 2 3 4 5 6 7
HCLD WTME IWX(I,4) WX(I,9) WX(I,12)
123456789012345678901234567890123456789012345678901234567890
123456789012345678901234567890123456789012345678901234567890
HCLD 1200.0 0.0 0.0 0.0
The MCLD Record

This record is used with any sensor type. The WTME parameter is not used for the DVO, II, and SiTV types.

WTME - Time.
    Range: 0000.0 - 2400.0 (HHMM) (GMT)
    Default: 1200.0 Hr.

IWX(I,5) - Cloud indicator.
    Range: 0.0 - None.
      3.0 - Any middle cloud.
    Default: 0.0

WX(I,10) - Cloud fraction.
    Range: 0.0 - 1.0
    Default: 0.0

WX(I,13) - Cloud base height.
    Range: 2.0 - 6.1 km.
    Default: 4.0 km.

Sample of MCLD Record:

```
1 2 3 4 5 6 7
1234567890123456789012345678901234567890123456789012345678901234567890
MCLD        WTME   IWX(I,5)  WX(I,10)  WX(I,13)
```

Sample of MCLD Record:

```
1 2 3 4 5 6 7
1234567890123456789012345678901234567890123456789012345678901234567890
MCLD        1200.0  3.0    0.5    4.0
```
The LCLD Record

This record is used with any sensor type. The WTME parameter is not used for the DVO, II, and SiTV types.

WTME - Time.
Range: 0000.0 - 2400.0 (HHMM) (GMT)
Default: 1200.0 Hr.

IWX(I,6) - Cloud indicator.
Range: 0.0 - None.
4.0 - Stratus
5.0 - Convective
Default: 0.0

WX(I,11) - Cloud fraction.
Range: 0.0 - 1.0
Default: 0.0

WX(I,14) - Cloud base height.
Range: 0.1 - 2.0 km.
Default: 1.0 km.

Sample of LCLD Record:

LCLD 1200.0 0.0 0.0 0.0
The ILUM Record

This record is used with the DVO, II, and SiTV sensor types. The first record described here designates that the user will input an illumination level in footcandles. The second record indicates that the user will choose a phase of the moon to determine the illumination level. The third record described here designates that TARGAC will have ILUMA calculate the illumination level based on the day and the weather.

**AL** - Value of the illuminance in footcandles (fc).
- Default: 1000.0 fc
- For DVO and SiTV devices the allowed values are 1.0 - 10,000 fc.
- For II devices the allowed values are 0.0001 - 0.01 fc.

**SIGWX** - Significant weather ID. No longer used.

**OBSURF** - Observed state of ground, choices 1-10.
1 - Dry
2 - Moist
3 - Wet
4 - Frozen
5 - Ice
6 - Snow < 0.5 in
7 - 0.5 in < Snow < all
8 - Snow (all)
9 - 0.5 in < Loose, dry snow, dust, sand < all
10 - Loose, dry snow, dust, sand (all)
- Default: 1.0

**PRTYPE** - Precipitation type, choices 1-5.
1 - None
2 - Drizzle
3 - Rain
4 - Snow
5 - Hail
- Default: 1.0

**IMOON** - The phase of the moon.
- Range: -1.0 - Clear sky, full moon conditions.
- -2.0 - Clear sky, quarter moon conditions.
- -3.0 - Clear sky, no moon conditions.
- Default: -1.0

**Sample of ILUMEXTL Record:**
```
1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890
ILUMEXTL  AL  NOT USED  OBSURF  PRTYPE
```

1234567890123456789012345678901234567890123456789012345678901234567890
ILUMEXTL 100.0 0.0 1.0 1.0

4-10
**Sample of ILUMEXTL Record:**

```
    1  2  3  4  5  6  7
123456789012345678901234567890123456789012345678901234567890
ILUMEXTL  IMOON
```

**ILUMINTL -2.0**

```
    1  2  3  4  5  6  7
123456789012345678901234567890123456789012345678901234567890
ILUMINTL            NOT USED OBSURF PRTYPE
```

**Sample of ILUMINTL Record:**

```
    1  2  3  4  5  6  7
123456789012345678901234567890123456789012345678901234567890
ILUMINTL  0.0  1.0  1.0
```

4-11
The METD Record

This record is used with the DVO, II, and SiTV sensor types.

**VIS - Visibility**
- Range: 0.10 - 200.0 (km).
- Default: 7.0 km.

**CF1 - Cloud fraction. No longer used.**

**ZCI - Cloud base height. No longer used.**

**THICK - Cloud thickness.**
- Range: 0.1 - 5.0 (km).
- Default: None.

**TMP - Temperature.**
- Range: -60.0 - 60.0 (degrees Celcius).
- Default: 10.0 degrees.

**TDEW - Dewpoint temperature.**
- Range: -60.0 - TMP above (degrees Celcius).
- Default: 8.0 degrees.

**BKREF - Surface reflectance. No longer used.**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789012345678901234567890123456789012345678901234567890</td>
<td>METD</td>
<td>VIS</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>THICK</td>
<td>TMP</td>
</tr>
</tbody>
</table>

Sample of METD Record:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234567890123456789012345678901234567890123456789012345678901234567890</td>
<td>METD</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>10.0</td>
</tr>
</tbody>
</table>
4.1.3. OUTPUT

The output for TARGAC-3 is in the same basic format as for TARGAC-2. The sky-to-ground ratio has been changed to include values for both in and out of cloud shadow cases when the appropriate conditions exist. The detection and recognition ranges have also been expanded to include clear sky values, in cloud shadow values, or both depending on the cloud conditions. Each condition provides a complete set of detection and recognition ranges for each sensor. No change has been made to the format previously used to report ranges.

The probability of the target scene in cloud shadow has been added. This value is given as a percentage. In addition, information about the cloud situation has been added to the output. Layers are reported as clear, partly cloudy, or overcast. Below is an example of the TARGAC-3 output block.

******************OUTPUTS ******************

SOLAR ELEVATION ANGLE = 25.60 DEGREES.
SOLAR AZIMUTH ANGLE = 82.45 DEGREES.
LUNAR ELEVATION ANGLE = -64.34 DEGREES.
LUNAR AZIMUTH ANGLE = 5.48 DEGREES.
LUNAR PHASE ANGLE = -72.32 DEGREES.

SOLAR ILLUMINANCE = 27198.26 LUMENS/SQ-METER.
LUNAR ILLUMINANCE = .0000 LUMENS/SQ-METER.
NET ILLUMINANCE = 27198.2600 LUMENS/SQ-METER.

INTEGRATED TOTAL FLUX = 292.45 WATTS/SQ-METER.

*** ILLUMINATION VALUE OUT OF RANGE
DEFAULT VALUE WILL BE USED

SKY TO GROUND RATIO (NO CLOUD) IS 1.212847
SKY TO GROUND RATIO (IN CLOUD) IS 7.182998

PROBABILITY OF TARGET IN CLOUD SHADOW: 60.769370
TWO PARTLY CLOUDY LAYERS.
DEVICE TYPE IS I I
CLEAR SKY CASE:

<table>
<thead>
<tr>
<th>DETECTION RANGE (KM)</th>
<th>PROBABILITY LEVEL = .10</th>
<th>.50</th>
<th>.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENSOR ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.0</td>
<td>1.2</td>
<td>.7</td>
</tr>
<tr>
<td>2</td>
<td>1.4</td>
<td>.8</td>
<td>.5</td>
</tr>
<tr>
<td>3</td>
<td>2.2</td>
<td>1.3</td>
<td>.8</td>
</tr>
<tr>
<td>4</td>
<td>.5</td>
<td>.3</td>
<td>.1</td>
</tr>
</tbody>
</table>

DEVICE TYPE IS I I
CLEAR SKY CASE:

<table>
<thead>
<tr>
<th>RECOGNITION RANGE (KM)</th>
<th>PROBABILITY LEVEL = .10</th>
<th>.50</th>
<th>.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENSOR ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>.9</td>
<td>.4</td>
<td>.3</td>
</tr>
<tr>
<td>2</td>
<td>.6</td>
<td>.3</td>
<td>.1</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>.5</td>
<td>.3</td>
</tr>
<tr>
<td>4</td>
<td>.2</td>
<td>.1</td>
<td>.1</td>
</tr>
</tbody>
</table>

DEVICE TYPE IS I I
CLOUD SHADOW CASE:

<table>
<thead>
<tr>
<th>DETECTION RANGE (KM)</th>
<th>PROBABILITY LEVEL = .10</th>
<th>.50</th>
<th>.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENSOR ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.2</td>
<td>.8</td>
<td>.5</td>
</tr>
<tr>
<td>2</td>
<td>.9</td>
<td>.6</td>
<td>.4</td>
</tr>
<tr>
<td>3</td>
<td>1.3</td>
<td>.9</td>
<td>.6</td>
</tr>
<tr>
<td>4</td>
<td>.4</td>
<td>.2</td>
<td>.1</td>
</tr>
</tbody>
</table>

DEVICE TYPE IS I I
CLOUD SHADOW CASE:

<table>
<thead>
<tr>
<th>RECOGNITION RANGE (KM)</th>
<th>PROBABILITY LEVEL = .10</th>
<th>.50</th>
<th>.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENSOR ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>.6</td>
<td>.4</td>
<td>.2</td>
</tr>
<tr>
<td>2</td>
<td>.4</td>
<td>.2</td>
<td>.1</td>
</tr>
<tr>
<td>3</td>
<td>.7</td>
<td>.4</td>
<td>.2</td>
</tr>
<tr>
<td>4</td>
<td>.2</td>
<td>.1</td>
<td>.1</td>
</tr>
</tbody>
</table>
4.2. TASK 1: IMPACTS ON ANALYSTS

This section describes the equations used in TARGAC-3 to predict, direct, and diffuse radiance incident on the target and background, the inherent contrast, and the contrast transmittance. The resulting sky-to-ground ratios and detection/recognition ranges are also described.

4.2.1. DELTA-EDDINGTON RADIATIVE TRANSFER MODEL

The partly cloudy FASCAT model (Hering and Johnson, 1984) has been incorporated into TARGAC-3. This model uses the delta-Eddington approximation to the radiative transfer equation to determine direct and diffuse radiance on a target scene. As described in section 3.1.2, the direct radiance is computed for "best case" and "worst case" cloud conditions. The diffuse radiance is computed as an average over all possible conditions. The total radiance is the sum of the direct and diffuse terms. This section describes the equations used in TARGAC-3 to predict direct, diffuse, and total radiance on the target scene.

TARGAC-3 uses the standard definition of direct radiance:

\[ D(\tau) = F_0 e^{-\mu_0 \tau}, \]  

where

\[ \mu_0 = \cos \theta, \]
\[ \theta = \text{zenith angle}, \]
\[ \tau = \text{optical depth from the top of the atmosphere down to the elevation of interest}, \]

and

\[ \pi F_0 = \text{solar irradiance}. \]
Shettle and Weinman (1970) give the equation of transfer defining the diffuse radiance $I(r, \mu, \phi)$ as

$$
\mu \frac{dI}{dr}(r, \mu, \phi) = -I(r, \mu, \phi) + \frac{a}{4\pi} \int_0^{2\pi} \int_0^{\pi/2} P(\mu, \phi; \mu', \phi') I(r, \mu', \phi') d\mu' d\phi' + \frac{1}{4} a F_0 P(\mu, \phi; \mu_0, \phi_0) e^{-\frac{r}{\mu_0}}
$$

(24)

where

- $\phi$ = azimuth angle,
- $a(\tau)$ = single scattering albedo,

and

$$
P(\mu, \phi; \mu', \phi') = \text{phase function for light incident at angle (} \mu', \phi' \text{) and scattered in the direction (} \mu, \phi \text{)}.
$$

They use Eddington's approximation to express diffuse radiance as a sum of two terms, a directionally independent term, $I_0$, and a directionally dependent term, $I_1$:

$$
I(r, \mu) = I_0(\tau) + I_1(\tau) \mu.
$$

(25)

Shettle and Weinman substitute this expression into the equation of transfer, expand the phase function using a simple cosine dependence, and integrate to obtain a pair of linear differential equations:

$$
\frac{dI_1}{dr} = -3[1 - a(\tau)] I_0 + \frac{3}{4} a(\tau) F_0 e^{-\frac{r}{\mu_0}}
$$

(26)

$$
\frac{dI_0}{dr} = -[1 - a(\tau) g(\tau)] I_1 + \frac{3}{4} a(\tau) g(\tau) \mu_0 F_0 e^{-\frac{r}{\mu_0}}.
$$

(27)

where

- $g(\tau)$ = phase function asymmetry parameter.
For turbid atmospheres, the assumption that there is no absorption of the scattered component,

\[ a(\tau) = 1, \quad (28) \]

is a good approximation in most cases (Wachtmann et al., 1985). This assumption simplifies the solution of equations (26) and (27) and leads to the following conservative solution to the equation of transfer:

\[ I_0(\tau) = B_1 - \frac{3}{4} \mu_0^2 F_0 e^{\frac{-\tau}{\mu_0}} - B_2 T(\tau) \quad (29) \]

\[ I_1(\tau) = B_2 - \frac{3}{4} \mu_0 F_0 e^{\frac{-\tau}{\mu_0}} \quad (30) \]

where

\[ B_2 = \frac{3 \mu_0 F_0 (1 - A) \left[ 2 + 3 \mu_0 + (2 - 3 \mu_0) e^{\frac{-\tau}{\mu_0}} \right]}{4 \left[ 4 + 3(1 - A) T(\tau^*) \right]} \quad (31) \]

\[ B_1 = \left( \frac{3 \mu_0^2}{4} + \frac{\mu_0}{2} \right) F_0 - \frac{2 B_2}{3} \quad (32) \]

the effective optical depth is

\[ T(\tau) = \int_0^\tau [1 - g(\tau')] d\tau' \quad (33) \]

\[ \tau^* = \text{optical thickness of the entire atmosphere,} \]

and

\[ A = \text{ground albedo.} \]
Diffuse irradiance is then computed from $I_0$ and $I_1$:

$$F(\tau) = 2\pi \int_0^{\pm 1} (I_0 + \mu I_1) \mu d\mu = \pi \left[ I_0(\tau) \pm \frac{2}{3} I_1(\tau) \right]$$  \hspace{1cm} (34)

where

$\mu>0$ corresponds to downwelling diffuse irradiance, $F \downarrow(\tau)$,

and

$\mu<0$ corresponds to upwelling diffuse irradiance, $F \uparrow(\tau)$.

Joseph et al. (1976) modified the Eddington approximation to better account for the highly asymmetric phase functions associated with particulate scattering. The delta-Eddington approximation uses a truncated forward scatter peak and a two-term expansion of the phase function. The delta-Eddington approximation is equivalent to the Eddington approximation except the transformed parameters $\tau', a'$, and $g'$ are substituted for $\tau$, $a$, and $g$. The delta-Eddington parameters are defined as follows:

$$\tau' = (1 - af)\tau$$  \hspace{1cm} (35)

$$g' = \frac{g}{1 + g}$$  \hspace{1cm} (36)

$$a' = \frac{(1 - f)a}{1 - af}$$  \hspace{1cm} (37)

where the fractional scattering into the forward peak is defined as

$$f = g^2.$$  \hspace{1cm} (38)

The delta-Eddington parameters result in modified values for the components of diffuse radiance, $I_0(\tau')$ and $I_1(\tau')$, and for direct radiance, $D(\tau')$. 

4-18
Hering used the delta-Eddington approximation in his development of the FASCAT model. This model was developed in three stages. FASCAT computes radiances under clear (1981), overcast (1983), and partly cloudy (1984) meteorological conditions.

Hering's method solves for diffuse radiance for a partly cloudy sky as a weighted average of clear-sky and overcast-sky values. The method allows up to two cloud layers. For two cloud layers, the following situations are possible:

1. Two clear layers. The diffuse radiance components, \( I_0 \) and \( I_1 \), are calculated just once using clear layers.

2. Upper layer overcast, lower layer clear. \( I_0 \) and \( I_1 \) are computed once with the appropriate cloud optical depth added to the optical depth from the top of the atmosphere down to the level of interest. We are mainly interested in targets at or near ground level, so we assume that the level of interest is below any cloud layers.

3. Lower layer overcast, upper layer clear. \( I_0 \) and \( I_1 \) are computed once with the appropriate cloud optical depth added to the optical depth from the top of the atmosphere down to the level of interest.

4. Two overcast layers. \( I_0 \) and \( I_1 \) are computed once with both cloud optical depths added to the atmospheric optical depth.

5. Upper layer partly cloudy, lower layer clear. \( I_0 \) and \( I_1 \) are calculated twice: once with both layers clear and once with the appropriate cloud optical depth included. Weighted averages of the two values for \( I_0 \) and \( I_1 \) are then computed.

6. Upper layer partly cloudy, lower layer overcast. \( I_0 \) and \( I_1 \) are calculated twice: once with both cloud optical depths included and once with just one cloud optical depth included (from the overcast layer). Weighted averages are then computed.

7. Lower layer partly cloudy, upper layer clear. \( I_0 \) and \( I_1 \) are calculated twice: once with both layers clear and once with the appropriate cloud optical depth
included. Weighted averages of the two values for \( I_0 \) and \( I_1 \) are then computed.

8. Lower layer partly cloudy, upper layer overcast. \( I_0 \) and \( I_1 \) are calculated twice: once with both cloud optical depths included and once with just one cloud optical depth included (from the overcast layer). Weighted averages are then computed.

9. Two partly cloudy layers. \( I_0 \) and \( I_1 \) are calculated four times: once using clear layers, once with the lower cloud optical depth included, once with the upper cloud optical depth included, and once with both cloud optical depths included. This scenario is depicted in figure 3 in section 3.1.2. Weighted averages of the four values for \( I_0 \) and \( I_1 \) are then computed.

Hering developed generalized equations to perform the weighted averages:

\[
\overline{I_0} = (1 - F_U)(1 - F_L)I_{0,CR} + F_U(1 - F_L)I_{0,U} + F_L(1 - F_U)I_{0,L} + F_U F_L I_{0,UL}
\]

(39)

\[
\overline{I_1} = (1 - F_U)(1 - F_L)I_{1,CR} + F_U(1 - F_L)I_{1,U} + F_L(1 - F_U)I_{1,L} + F_U F_L I_{1,UL},
\]

(40)

where

\[
F_U(n_U, \mu) = n_U (1.43 - 1.21\mu - 2.00n_U + 1.21n_U\mu + 1.57n_U^2),
\]

(41)

\[
F_L(n_L, \mu) = n_L (1.43 - 1.21\mu - 2.00n_L + 1.21n_L\mu + 1.57n_L^2),
\]

(42)

\( n_U \) = upper cloud fraction,

\( n_L \) = lower cloud fraction,

\( I_{0,CR} \) = \( I_0 \) with no clouds,

\( I_{0,U} \) = \( I_0 \) with upper overcast cloud,

\( I_{0,L} \) = \( I_0 \) with lower overcast cloud,
\( I_{0,UL} = I_0 \) with upper and lower overcast clouds,

\( I_{1,CR} = I_1 \) with no clouds,

\( I_{1,U} = I_1 \) with upper overcast cloud,

\( I_{1,L} = I_1 \) with lower overcast cloud,

and

\( I_{1,UL} = I_1 \) with upper and lower overcast clouds.

Hering's weighting factors \( F_U \) and \( F_L \) are based on work done by Shapiro (1982) to fit SOLMET data. Note that \( F_U \) is zero for a clear upper cloud layer and one for an overcast layer; \( F_L \) is zero for a clear lower cloud layer and one for an overcast layer. Equations (39) and (40), then, can be used to compute \( \bar{I}_0 \) and \( \bar{I}_1 \) values for any of the nine possible cloud situations. Any inapplicable terms cancel out.

According to Hering and Johnson (1984), the downward diffuse radiance below cloud layers has an additional term, which we will denote \( I_{COR} \), to account for forward scattering of direct radiance through the cloud layers:

\[
I_{COR} = \left[ (1 - F_U)(1 - F_L) - G_U G_L \right] D_{CR}(\tau') + \left[ F_U (1 - F_L) - G_L (1 - G_U) \right] D_U(\tau') + \left[ F_L (1 - F_U) - G_U (1 - G_L) \right] D_L(\tau') + \left[ F_U F_L - (1 - G_U)(1 - G_L) \right] D_{UL}(\tau'),
\]

where

\( D_{CR}(\tau') = D(\tau') \) with no clouds,

\( D_U(\tau') = D(\tau') \) with upper overcast cloud,

\( D_L(\tau') = D(\tau') \) with lower overcast cloud,
and

\[ D_{UL}(\tau') = D(\tau) \] with upper and lower overcast clouds.

The factors \( G_U \) and \( G_L \) are the probabilities of a cloud-free solar path through the upper and lower cloud layers. These values are derived by Allen and Malick (1983) as follows:

\[ G(n, \theta) = P_n^{1 + c_n \tan \theta}, \] (44)

where the average height-to-width ratio for a cubical cloud, \( c_n \), is given as

\[ c_n = 0.55 - \frac{n}{2}, \] (45)

the probability of a cloud-free path from the zenith, \( p_n \), is

\[ p_n = 1 - n \frac{1 + 3n}{4}, \] (46)

and \( n \) is the upper or lower cloud fraction. Note that \( G_U \) is one for a clear upper cloud layer and zero for an overcast layer; \( G_L \) is one for a clear lower cloud layer and zero for an overcast layer.

We can now calculate the total radiance incident on the target scene as a sum of direct and diffuse components. For a downward line-of-sight, the average diffuse radiance is

\[ \bar{T} = \bar{T}_0 + \frac{2}{3} \bar{T}_1 + I_{COR}. \] (47)

For an upward line-of-sight, the average diffuse radiance is

\[ \bar{T} = \bar{T}_0 - \frac{2}{3} \bar{T}_1. \] (48)
For a horizontal line-of-sight, we take the average of the upward and downward values:

\[
\bar{I} = \frac{I_0 + I_{\text{cor}}}{2}.
\] (49)

To this average diffuse radiance value, we add the best case direct radiance value to obtain the higher bracketing value of total radiance. We add the worst case direct radiance value to obtain the lower bracketing value. For situation 1, with two clear layers, a single value of total radiance is computed:

\[
R = \bar{I} + D_{\text{cr}}(\tau').
\] (50)

For cloud situations 2 or 3, with one overcast layer, a single value is computed:

\[
R = \bar{I} + D_{u}(\tau')
\] (51)

or

\[
R = \bar{I} + D_{l}(\tau').
\] (52)

For situation 4, with two overcast layers, a single value is computed:

\[
R = \bar{I} + D_{\text{ul}}(\tau').
\] (53)

For cloud situations 5 and 7, with one partly cloudy layer, two bracketing values of total radiance are computed:

\[
R_1 = \bar{I} + D_{\text{cr}}(\tau')
\] (54)

and

\[
R_2 = \bar{I} + D_{l}(\tau')
\] (55)

or

\[
R_2 = \bar{I} + D_{\text{cl}}(\tau').
\] (56)
Here, $R_1$ is the "best case" radiance and $R_2$ is the "worst case" radiance. For situations 6 and 8, with one partly cloudy layer and one overcast layer, two bracketing values of total radiance are again computed:

$$ R_1 = \bar{I} + D_L(\tau') $$

or

$$ R_1 = \bar{I} + D_U(\tau') $$

and

$$ R_2 = \bar{I} + D_{UL}(\tau'). $$

Finally, for cloud situation 9, with two partly cloudy layers, two bracketing values of total radiance are computed:

$$ R_1 = \bar{I} + D_{C\bar{R}}(\tau') $$

and

$$ R_2 = \bar{I} + D_{UL}(\tau'). $$

The other two possibilities (using $D_U$ and $D_L$) are ignored.

Finally, the FASCAT model allows computation of the probability that the target scene is in cloud shadow. This value depends on the probability of a cloud-free path through each cloud layer:

$$ P = 1 - G_U G_L, $$

where $G_U$ and $G_L$ are defined by Equation (29).
4.2.2. INHERENT CONTRAST

The inherent contrast between the target and the background is computed once for clear or overcast conditions and twice for partly cloudy conditions. The bracketing values of total radiance are used for these computations:

\[ C_{0,1} = \frac{R_{0,1} - bR_{0,1}}{bR_{0,1}} \tag{63} \]

and

\[ C_{0,2} = \frac{R_{0,2} - bR_{0,2}}{bR_{0,2}} \tag{64} \]

where the presubscripts designate the target or background, and the postsubscript 0 indicates that the scene is observed at zero distance. Note that the radiance values \( R_0 \) and \( bR_0 \) include a multiplicative factor for the target or background viewing angle and reflectivity. For example, \( tR_{0,1} \) refers to the inherent (zero distance), "best case" radiance reflected off the target.

The radiance reflected off the target is multiplied by the transmittance and added to the path radiance between the target and sensor to obtain the apparent radiance:

\[ tR_{r,1} = tR_{0,1} T(r) + N' \tag{65} \]

and

\[ tR_{r,2} = tR_{0,2} T(r) + N' \tag{66} \]

where \( r \) is the distance between the target and sensor,

\[ R_{r,1} = \text{the best case apparent radiance}, \]

\[ R_{r,2} = \text{the worst case apparent radiance}, \]

\[ T(r) = \text{transmission along the distance } r, \]
and

\[ N^* = \text{path radiance between the target and sensor.} \]

Similar equations are used to compute \( bR_r,1 \) and \( bR_r,2 \).

### 4.2.3. CONTRAST TRANSMITTANCE

The apparent radiance values can then be used to compute apparent contrast. The bracketing values of apparent radiance are used for these computations:

\[
C_{r,1} = \frac{iR_r,1 - bR_r,1}{bR_r,1} \quad (67)
\]

and

\[
C_{r,2} = \frac{iR_r,2 - bR_r,2}{bR_r,2} \quad (68)
\]

The bracketing values of inherent and apparent contrast can then be used to compute contrast transmittance:

\[
\frac{C_{r,1}}{C_{0,1}} = \frac{1}{1 + \frac{N^*}{bR_{0,1}T(r)}} \quad (69)
\]

and

\[
\frac{C_{r,2}}{C_{0,2}} = \frac{1}{1 + \frac{N^*}{bR_{0,2}T(r)}} \quad (70)
\]

The contrast transmittance can also be written as a function of sky-to-ground ratio, \( S_{gr} \):

\[
\text{4-26}
\]
\[ C_{r,1} = C_{0,1} \frac{1}{1 + S_{gr,1} e^{\theta r-1}} \]  

(71)

and

\[ C_{r,2} = C_{0,2} \frac{1}{1 + S_{gr,2} e^{\theta r-1}} \]  

(72)

where the postscripts 1 and 2 again refer to best and worst case values.

4.2.4. SENSOR PERFORMANCE MODEL

The sensor performance model was updated only to include detection and acquisition range prediction for up to two bracketing conditions. No other changes to this model were made.

4.3. TASK 2: ANTICIPATED IMPACTS

This section covers the planned new inputs and modified outputs to TARGAC that will support the addition of small-scale feature shadowing.

4.3.1. INPUTS

Two new types of operator inputs will be needed to support the small-scale scene shadow modifications to TARGAC. The first group consists of inputs related to 2-d and 3-d clutter and the second includes those related to the viewing geometry. Figure 9 may help in the visualization of these geometric inputs. The new inputs are as follows:
2-d clutter type or statistics - as stated previously, the operator will be required to either select the desired 2-d clutter type from a set of supplied images or to input a set of 2-d clutter statistics.

3-d clutter type or statistics - as with 2-d clutter, the operator will be required to either select the desired 3-d clutter type from a set of supplied images, or to input a set of 3-d clutter statistics.

Range of SUN-TARGET-SENSOR angles - Unless the sensor is headed directly at the target, this angle will change as a function of sensor-to-target distance. Thus it is important to calculate detection ranges for various sun-target-sensor angles. Here the operator may input the starting angle, the stopping angle, and the value of each step, or accept the default values.

Sensor-target elevation difference - Unlike the EOTDA, TARGAC was designed as an LOS code, which means that in most cases the target and the observer are at approximately the same elevation. However, since the clutter index will change with large
changes in the observer-target-ground angle, we use the difference between sensor and target elevations as an input in determining how many clutter indexes we must calculate for a specific run.

- Slope angles for the target - these inputs are the elevation angle and rotation angle of the target about the observer-to-target line of sight as projected into the ground plane. Although TARGAC currently considers the earth as flat, the large-scale feature shadowing modifications being performed as part of this effort will be incorporating DTED and ground slope into the code. By adding these two parameters to our calculations, we are facilitating the future incorporation of DTED into the small scale feature shadowing module.

4.3.2. OUTPUTS

The output for TARGAC with the small-scale scene shadowing modifications will be basically in the same format as TARGAC-3. The output will be changed to include the values of the SUN-TARGET-SENSOR AZIMUTH angle. For each angle computed, an array of user defined probability levels (with the defaults set equals to 25, 50, and 75 percent) versus sensor ID number will be calculated for both the detection and recognition cases.
**Sampler Outputs**

- **Solar Elevation Angle**: 25.60 Degrees.
- **Solar Azimuth Angle**: 82.45 Degrees.
- **Lunar Elevation Angle**: -64.34 Degrees.
- **Lunar Azimuth Angle**: 5.48 Degrees.
- **Lunar Phase Angle**: -72.32 Degrees.
- **Solar Illuminance**: 27198.26 Lumens/Sq-Meter.
- **Lunar Illuminance**: 0.0000 Lumens/Sq-Meter.
- **Net Illuminance**: 27198.2600 Lumens/Sq-Meter.
- **Integrated Total Flux**: 292.45 Watts/Sq-Meter.

*** Illumination Value Out of Range  
Default Value Will Be Used

Sky to Ground Ratio is 7.182998

**Device Type is I I**  
SUN-TARGET-SENSOR AZIMUTH = 10 DEGREES:

<table>
<thead>
<tr>
<th>Detection Range (KM)</th>
<th>Probability Level = .10</th>
<th>.50</th>
<th>.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor ID</td>
<td>1.0</td>
<td>1.2</td>
<td>.7</td>
</tr>
<tr>
<td>2</td>
<td>1.4</td>
<td>.8</td>
<td>.5</td>
</tr>
<tr>
<td>3</td>
<td>2.2</td>
<td>1.3</td>
<td>.8</td>
</tr>
<tr>
<td>4</td>
<td>.5</td>
<td>.3</td>
<td>.1</td>
</tr>
</tbody>
</table>

**Device Type is I I**  
SUN-TARGET-SENSOR AZIMUTH = 10 DEGREES:

<table>
<thead>
<tr>
<th>Recognition Range (KM)</th>
<th>Probability Level = .10</th>
<th>.50</th>
<th>.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor ID</td>
<td>1.0</td>
<td>.4</td>
<td>.3</td>
</tr>
<tr>
<td>2.0</td>
<td>.6</td>
<td>.3</td>
<td>.1</td>
</tr>
<tr>
<td>3.0</td>
<td>1.0</td>
<td>.5</td>
<td>.3</td>
</tr>
<tr>
<td>4.0</td>
<td>.2</td>
<td>.1</td>
<td>.1</td>
</tr>
</tbody>
</table>

**Device Type is I I**  
SUN-TARGET-SENSOR AZIMUTH = 20 DEGREES:

<table>
<thead>
<tr>
<th>Detection Range (KM)</th>
<th>Probability Level = .10</th>
<th>.50</th>
<th>.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor ID</td>
<td>1.0</td>
<td>.8</td>
<td>.5</td>
</tr>
<tr>
<td>2.0</td>
<td>.9</td>
<td>.6</td>
<td>.4</td>
</tr>
<tr>
<td>3.0</td>
<td>1.3</td>
<td>.9</td>
<td>.6</td>
</tr>
<tr>
<td>4.0</td>
<td>.4</td>
<td>.2</td>
<td>.1</td>
</tr>
</tbody>
</table>
DEVICE TYPE IS II
SUN-TARGET-SENSOR AZIMUTH = 20 DEGREES:

RECOGNITION RANGE (KM)

PROBABILITY LEVEL = .10

<table>
<thead>
<tr>
<th>SENSOR ID</th>
<th>.10</th>
<th>.50</th>
<th>.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.6</td>
<td>.4</td>
<td>.2</td>
</tr>
<tr>
<td>2</td>
<td>.4</td>
<td>.2</td>
<td>.1</td>
</tr>
<tr>
<td>3</td>
<td>.7</td>
<td>.4</td>
<td>.2</td>
</tr>
<tr>
<td>4</td>
<td>.2</td>
<td>.1</td>
<td>.1</td>
</tr>
</tbody>
</table>

4-31
5. TESTING

Although part of the Electro-Optical Systems Atmospheric Effects Library (EOSAEL), TARGAC is a self-contained module. TARGAC has been under development for a number of years by a number of contractors as a multi-year, incremental research and development program. Each cycle of TARGAC development builds on the previous effort and provides an increasing degree of capability and sophistication. However, as the sophistication increases, the testing requirements for a new module also increase. This fact is exacerbated by the lack of comprehensive documentation and configuration control. PSR and Hughes STX are taking a small step towards the resolution of this problem through the use of formalized Engineering Change Reports (ECRs) as part of our development process and through the use of highly modular code which will facilitate future software modifications.

In this section, we will first examine the testing of the PC version of the shadowing by clouds module and then examine the test planning that has been accomplished for the shadowing by small-scale features module. In addition to the testing and test planning reported here, during the second year of this phase II SBIR, the shadowing by large-scale features module will be tested, as will the integration of all three modules within both the PC and UNIX versions of TARGAC.

5.1. TASK 1: SHADOWING BY CLOUDS MODULE TESTING

Software tests were conducted to analyze the performance of scene shadowing modifications to TARGAC. Whenever feasible, individual routines were tested during the course of modification. For formal testing, three distinct series of tests were conducted to evaluate the model differently: functionality tests, tests against TARGAC-2 sample runs, and comparison tests against the EOTDA. The tests ensured complete program execution, validated system default values, and looked to see that radiance values (both direct and diffuse), the probability that the target scene is in cloud shadow, sky-to-ground ratios, and bracketing range values were within acceptable ranges. Tests were carried out under a pass/fail system. For each failure, appropriate modifications were made to TARGAC-3 to correct the problem.
5.1.1. FUNCTIONALITY TESTS

5.1.1.1. Overview

The first group of tests focused on evaluating the serviceability of the scene shadowing modifications to the TARGAC code. This required that every possible combination of low, middle, and high clouds be reviewed.

5.1.1.2. Cloud Type/Amount/Base Height

This section involved the most extensive testing. Cloud type and base height were varied randomly. Base height extremes were tested. In addition, each cloud level was tested in a clear, partly cloudy, and overcast state. The partly cloudy amounts were varied to encompass a broad spectrum of configurations.

5.1.1.3. Sensor Type

All of the sensor types available in TARGAC were utilized. Direct View Optics, Image Intensifiers, and Silicon Television Systems were extensively tested since they are now sensitive to variable cloud conditions. Thermal Imagers were looked at only to ensure that the modifications did not alter the outputs.

5.1.1.4. Interactive/Batch Mode

Two modes of input were used. All the interactive options were exercised. The batch format was employed to ensure correctness. Default and extreme values affecting modifications to TARGAC inputs were tested.

5.1.1.5. Climatology/User-Specified Data

The climatology information utilized by TARGAC-3 can be specified by the user or obtained from an existing database. Both methods were tested to ensure that reasonable data was entered into the system. Two climatological databases were used: North Atlantic States (73) and Southwestern States (53). A random sampling of user-specified inputs was examined.
5.1.1.6. Results

The total number of possible cloud combination tests that exist are 432. This number was reduced to 66 tests for three reasons:

1. Cloud shadows do not influence thermal imaging systems; therefore, only a limited number of thermal imaging tests were conducted to ensure functionality.

2. The climatology database provides cloud information for only one layer, thereby eliminating an entire series of cloud combination tests.

3. Each cloud combination was tested just once to validate the output. As a result, the many combinations were equally distributed among the possible input selections.

After appropriate code modification, all tests ran naturally to completion. No functionality errors were detected.

5.1.2. TESTS AGAINST TARGAC SAMPLE RUNS

5.1.2.1. Overview

This series of tests simply repeated the tests that were conducted on TARGAC-2. These tests, in addition to the outputs they produced, are listed in the TARGAC Users Manual (Gillespie, 1993). The sample tests were run to verify that the output from TARGAC-3 correlates favorably with the output from TARGAC-2 for clear and overcast sky conditions and improves the output for partly cloudy cases.

We created new batch input files for the TARGAC-3 sample runs. These files contain the same information as those used for the TARGAC-2 sample runs, but some of the records have been changed to correspond to the changes described in section 4.1.2.2. The old and new batch file names for each sample test are shown in table 7.
Table 7. Input files for TARGAC sample runs.

<table>
<thead>
<tr>
<th>Sample #</th>
<th>Old Input File</th>
<th>New Input File</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>TACDVO1.DAT</td>
<td>TACDVO1N.DAT</td>
</tr>
<tr>
<td>1B</td>
<td>TACDVO2.DAT</td>
<td>TACDVO2N.DAT</td>
</tr>
<tr>
<td>2</td>
<td>TACDVO3.DAT</td>
<td>TACDVO3N.DAT</td>
</tr>
<tr>
<td>3</td>
<td>TACDVO4.DAT</td>
<td>TACDVO4N.DAT</td>
</tr>
<tr>
<td>4</td>
<td>TACDVO5.DAT</td>
<td>TACDVO5N.DAT</td>
</tr>
<tr>
<td>5A</td>
<td>TACII3.DAT</td>
<td>TACII3N.DAT</td>
</tr>
<tr>
<td>5B</td>
<td>TACII2.DAT</td>
<td>TACII2N.DAT</td>
</tr>
<tr>
<td>6</td>
<td>TACII1.DAT</td>
<td>TACII1N.DAT</td>
</tr>
<tr>
<td>7A</td>
<td>TACSTV1.DAT</td>
<td>TACSTV1N.DAT</td>
</tr>
<tr>
<td>7B</td>
<td>TACSTV2.DAT</td>
<td>TACSTV2N.DAT</td>
</tr>
<tr>
<td>8</td>
<td>TACTI1.DAT</td>
<td>TACTI1N.DAT</td>
</tr>
<tr>
<td>9</td>
<td>TACTI2.DAT</td>
<td>TACTI2N.DAT</td>
</tr>
<tr>
<td>10</td>
<td>TACUSR1.DAT</td>
<td>TACUSR1N.DAT</td>
</tr>
<tr>
<td>11</td>
<td>TACUSR2.DAT</td>
<td>TACUSR2N.DAT</td>
</tr>
<tr>
<td>12</td>
<td>TACUSR3.DAT</td>
<td>TACUSR3N.DAT</td>
</tr>
</tbody>
</table>

5.1.2.2. Results

After appropriate code modification, all the batch mode tests ran to completion. Comparisons of the output files for TARGAC-2 and TARGAC-3 indicated some differences, as expected. The TARGAC-3 sky-to-ground ratio values for in and out of cloud shadow typically bracketed the TARGAC-2 values, although in several cases both of the TARGAC-3 values were lower than the TARGAC-2 values. Detection and recognition ranges showed smaller variation than the sky-to-ground ratios. Changes in the net illuminance value for sample 1A and the sky-to-ground ratio values for sample 4 require further investigation.
5.1.3. COMPARISON TESTS BETWEEN TARGAC AND THE AIR FORCE EOTDA

5.1.3.1. Overview

In a similar manner to the functionality tests, each sensor type was scrutinized under a number of different cloud conditions using the TARGAC-3 model and Version 3.0 of the Air Force EOTDA. An attempt was made to closely duplicate the input data used in both the TARGAC-3 and EOTDA runs in order to generate similar output ranges. Debug versions of both models were employed so that intermediate radiance values could be monitored.

These tests were deemed useful because the scene shadow modules that were written into the TARGAC-3 program were patterned directly after the code developed for the EOTDA model. They are believed to provide the best program verification available.

One limitation in using the EOTDA model for comparison is the restricted number of sensors that both models have in common. No direct view optical device is currently available on both systems.

5.1.3.2. Results

Limitations were discovered in relating the outputs of the two models. Different units for illumination and radiance values made it difficult to make exact comparisons. The EOTDA assumes a downward line-of-sight, while TARGAC-3 uses a horizontal line-of-sight. In general, illuminance, the probability of the target scene being in cloud shadow, and detection range appeared to vary in a similar manner as cloud conditions were changed. Further investigation is required.

5.2. Task 2: SHADOWING BY SMALL-SCALE FEATURES MODULE TESTING.

5.2.1. Overview

This section describes the plans for testing the small-scale feature shadowing enhancement to the TARGAC code. TARGAC is a complex system designed to operate on IBM-compatible PCs and UNIX-based workstations. Our TARGAC testing goal is to ensure that the initial implementation of small-scale feature shadowing performs as designed, presents predictable and understandable results, and is able to respond rapidly in
both the PC and UNIX environments. Data collected during testing will be used in the
evaluation of the software design and to identify high payoff research and development
areas for future TARGAC improvements.

5.2.2. Scope

This plan covers software verification and validation testing for the small-scale
feature shadowing enhancement to the TARGAC code at PSR, but does not include a
discussion of integration testing of the various modules (enhancements) under independent
development at PSR and Hughes STX. This and related TARGAC improvements are part
of a multi-phase SBIR effort that will incorporate many new concepts and capabilities over
the SBIR program life cycle. This test plan provides a general framework and guide for
TARGAC module testing in the future, but is specifically tailored to address the testing and
validation of the small-scale feature shadowing enhancement to the TARGAC code.

This section is organized in two parts. The first part discusses PSR's testing
approach as it pertains to the testing of these modifications to the TARGAC code. This
includes the approach, constraints, test team, test system configuration, test procedures,
test data, and evaluation recording. The second part describes the schedule that will be
followed during this testing.

5.2.3. Approach

TARGAC must be able to calculate detection and recognition ranges for a sensor-
target combination as a function of user selected probabilities of detection. Thus, the
criterion that must apply to the testing and evaluation of this new TARGAC module should
be fidelity to reproduce selected experimental and theoretical results.

5.2.3.1. Software Verification and Validation Tests

These software verification and validation tests are designed to answer two
important questions: (1) do individual small-scale feature shadowing software component
functions compute correct results and (2) do these software components (function
aggregates) work together as designed? Both quantitative and qualitative results will be
examined during testing.
5.2.3.2. System Integrity Tests

PSR will perform system integrity tests to determine if the small-scale feature shadowing module is producing results that are predictable and explainable. These tests will qualitatively reproduce results expected by analysts that will verify operational correctness given the various constraints.

5.2.3.3. System Fidelity and Performance Tests

PSR will perform system fidelity and performance tests. These tests are designed to measure the utility of TARGAC's small-scale feature shadows model under various scenarios. PSR will test TARGAC results for various sensors at various contrast levels in the following types of scenarios:

- High, medium, and low 2-d clutter with no 3-d clutter. This is the best scenario for comparison of TARGAC against the Air Force EOTDA.

- High, medium, and low 3-d clutter with no 2-d clutter. This scenario will permit comparison of the effects of 3-d object masking/shadows from scattered objects to the effects of 3-d object masking/shadows from tree lines and terrain being developed in the large-scale feature shadowing task.

- High, medium, and low 2-d clutter versus high, medium, and low 3-d clutter. This scenario will produce nine cells per contrast level per sensor type for comparison to real and synthetic imagery.

5.2.4. Test Team

PSR will test TARGAC using the experience of both software engineers and experienced analysts. The objective of the test team will be to evaluate the new TARGAC module's accuracy, functionality, and to a limited extent its utility.

5.2.5. System Configurations

The following system configurations will be used during testing:

- Compaq PC 486/33 w/4 M-Byte of RAM & MS-DOS 6.0
- Sun SPARC-2 Workstation w/32 M-Byte of RAM & SunOS Release 4.1.3
- SGI Workstation w/64 M-Byte of RAM & IRIX 4.0.5
- HP 9000 Workstation (via Internet).

It should be noted that these configurations may be updated before the end of testing. New versions of MS-DOS, SunOS, and SGI OS occur on approximately a six month cycle. For example, MS-DOS 6.2 has just been released by Microsoft.

5.2.6. Test Procedures

TARGAC will be tested against a specific set of test procedures. Each test procedure will include the following items as applicable:

- Background - information to indicate how the function/model is used and any other pertinent information not contained in the body of the test procedure.

- Purpose of test - description of the capabilities to be tested and the criteria the function/model must meet.

- Data requirements - description of data necessary for testing TARGAC capabilities.

- Software requirements - the name and identifier of the software component to be tested.

- Preparation - any setup or preparation required to conduct the test.

- Test procedure - the actual steps required to perform the test.

- Expected results - the outcome required for the test to be considered successful.
5.2.7. Test Data

PSR will work with three distinct types of data in the testing of the small-scale feature shadows module of TARGAC:

- Synthetic data generated by PSR using BRL-CAD, ACAD, and/or EUCLID. (These programs are target modeling codes that run on various work stations to include Sun and SGI. Plates 1 and 2 at the back of this report were generated by PSR using BRL-CAD.)

- Real imagery which is being provided to us (at no cost) by NVESD, formally the Army Night Vision Laboratory.

- Calculated data and the results of other codes such as the Air Force EOTDA and PSR's Probability of Detection Code. All of these codes have problems such as lack of common sensors, different spectral bands of interest, etc.

5.2.8. Evaluation

TARGAC testing will be evaluated quantitatively and qualitatively based on the level of testing as discussed in section 5.2.3. Specifically, software verification and validation tests will be evaluated through quantitative methods that rely on synthesized, idealized data to support test procedures. Results from software verification and validation tests will be documented in test procedures developed for individual software components. Errors encountered during these tests will be captured and documented in discrepancy reports (see figure 10). The Discrepancy Report Form that will be used during testing is a standard PSR form and all items contained within it may not be pertinent to this program.
<table>
<thead>
<tr>
<th>To be completed by originator</th>
<th>Key to Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: __________</td>
<td>0 Enhancement</td>
</tr>
<tr>
<td>Originator: __________</td>
<td>1 Aesthetics only</td>
</tr>
<tr>
<td>Problem Impact: 0 1 2 3 4 5</td>
<td>2 Inconvenient: Impacts ease of use</td>
</tr>
<tr>
<td></td>
<td>3 Important: feature does not work</td>
</tr>
<tr>
<td></td>
<td>4 Very Important: Multiple impacts</td>
</tr>
<tr>
<td></td>
<td>5 Critical: immediate solution required</td>
</tr>
</tbody>
</table>

**To be completed by originator**

Brief Description of Problem:

Detailed Problem Description (include activity on progress):

**To be completed by originator, program manager, or developer**

Module of files affected:

**To be completed by developer**

Description of Changes Made/Solution: Date Completed At PSR: __________

Date Completed On Site: __________

**To be completed by developers**

Assigned at PSR: __________ Date: __________

Assigned on Site: __________ Date: __________

**To be completed by CCB and/or Project Leader**

Date: __________ Project Leader Signature: __________

Date: __________ CCB Signature: __________

*Figure 10. The PSR Test Discrepancy Report Form.*
System integrity tests will be evaluated using qualitative methods that rely on analyst experts to evaluate output results generated by the TARGAC system. Discrepancy reports that describe inconsistencies in expected processing results will be filed if required.

System fidelity and performance tests will be evaluated using both quantitative and qualitative methods that exploit system performance statistics and domain model fidelity characteristics. Discrepancy reports that describe inconsistencies in expected processing results will be filed if required. In addition, model fidelity and performance measurements will be made to help identify system requirements and track performance statistics. These measurements could be used in future development to ensure optimal system performance. During system fidelity and performance testing, comparisons will be made between the PC and UNIX versions of the software.

5.2.9. Schedule

The draft test schedule shown in figure 11 addresses only the testing of the PC and UNIX versions of the small-scale feature shadowing components of TARGAC. This schedule will be integrated with the schedules for the various other activities to be performed by PSR and Hughes STX during the option year of this contract. We must remember that during this same time frame integration testing of the cloud shadowing, small-scale feature shadowing, and large-scale feature shadowing will be taking place. This parallel development approach necessitates close coordination and configuration control between PSR and Hughes STX.

TARGAC small-scale feature shadows testing will take place over a four month period starting May 1, 1994. Software verification and validation tests will be performed May 1 through July 1. System integrity tests will be performed June 1 through August 1, and fidelity tests will be performed July 1 through September 1.
Figure 11. TARGAC small-scale feature shadowing testing schedule.
6. SUMMARY

The goals of this first year of the Phase II SBIR Scene Shadows project were to develop the partly cloudy shadowing model, deliver updated TARGAC software for the PC, determine the relationship between small-scale feature shadows and clutter, and develop an approach to incorporating the temporal and spatial characteristics of small-scale feature shadows into TARGAC. This report describes the Hughes STX revised delta-Eddington model in TARGAC-3. The report also describes software changes that have been implemented and tested. The report describes and discusses the foundations underlying the PSR approach to modeling small-scale scene shadows and provides an overview of the implementation and testing plan PSR has developed.

During the first year of the Phase II SBIR, PSR's accomplishments included:

1. Examination of target acquisition relationships, solving many of the problems facing year two of this effort.

2. Development of an approach to the determining of the clutter index over a variety of clutter and illumination conditions.

3. Production of a software design document for the inclusion of small-scale feature shadows into TARGAC.

During the first year of the Phase II SBIR, Hughes STX's accomplishments included:

1. Development of the partly cloudy shadowing model.

2. Incorporation of the partly cloudy shadowing model into the TARGAC code for the PC and delivery of the updated software.

While PSR was designing the software for the inclusion of small-scale feature shadows into TARGAC, it achieved several corollary objectives in support of the large-scale feature shadowing task:
1. Implemented β software to access DMA DTED data stored on CD ROMs and 9-track tapes.

2. Implemented β software to access DMA DFAD data on 9-track tapes (DFAD is not yet available on CD ROMs).

While Hughes STX was implementing the partly cloudy shadowing model, it achieved several corollary objectives:

1. Cloud inputs were made the same between the visible and thermal imager sections of TARGAC-3.

2. TARGAC-3 cloud inputs were modified to conform to inputs used by the Air Force EOTDA.

3. Several pre-existing errors in TARGAC-2 were fixed. These included faulty COMMON block names in the ELIMIN and STG subroutines, an incorrect lower limit for the background reflectivity input, and incorrect passing of constants to the XSCALE subroutine.

In the second year of the Phase II SBIR, PSR will implement the small-scale feature shadowing changes to both the PC and UNIX versions of TARGAC. PSR will also validate through testing the relationships between small-scale feature shadowing and clutter that were developed during the first year of this effort, and we will perform the final integration and test of the Hughes STX- and PSR-developed modules in the TARGAC code. Hughes STX will implement the cloud shadowing changes to the UNIX version of TARGAC. A study will be conducted to investigate the differences between and applicability of the conservative and non-conservative solutions to the delta-Eddington approximation. PSR and Hughes STX will also implement large-scale feature shadows in TARGAC.
7. BIBLIOGRAPHY

Air Weather Service Regulation 105-24, Volume I, Headquarters Air Weather Service (MAC), Scott AFB, IL, 1 March 1983.


This appendix provides a tree diagram of the TARGAC program. The following notation is used before routine names in the tree:

P Program
S Subroutine
F Function
B Block Data.

This tree diagram is accurate for DVO, II, and SiTV systems. However, for thermal imager systems the PC version breaks after the call to THERML to execute TCM2, a separate program. When TCM2 is done executing, the flow returns to the call to THERMB. The tree diagram for the TCM2 program is unchanged; the diagram is provided by Touart and DeBenedictis (1991).
Program: TARGAC

--- P. TARGAC 
--- D. CLYMBD
--- D. CONSBD
--- D. FCONBD
--- D. FRMTBD
--- D. GEOMBD
--- D. IOUNBD
--- D. SAVEBD
--- D. SPOTBD
--- D. TAR1BD
--- D. TAR2BD
--- D. RHDATA
--- S. REDCOM
--- S. RDTCM2
--- S. CARDRD
--- S. SORT
--- S. HOURLY
--- S. FINDR
--- S. INTCHK
--- S. REALCK
--- S. CNTRAS
--- S. REALCK
--- S. INTCHK
--- S. THERML
--- S. INTCHK
--- S. REALCK
--- F. VAPOR
--- S. TCONVR
--- S. THERMB
--- S. TCONVR
--- S. SGR
--- S. GETDAT
--- S. INTCHK
--- S. REALCK
--- S. CVRTJD
--- S. CLIMAT
--- S. IOOPEN
--- S. XSCALE
--- S. TRPCTL
--- S. INTERP
--- S. SLANT
--- S. TRPCTL
--- S. INTERP
--- S. CASE
--- F. EXPCHK
--- S. SNOSLN
--- S. ICEFOG
--- S. NDXRFR
--- F. RESFN
--- S. ILMDAT
--- S. ILUMA
APPENDIX B: DESCRIPTIONS OF MODIFIED AND NEW ROUTINES

This appendix contains lists of modified TARGAC routines, new routines, and modified COMMON blocks. Descriptions of the modified routines are also included. Note that no new COMMON blocks have been added.

B.1. LIST OF MODIFIED ROUTINES

The following table lists the TARGAC subroutines and functions that have been modified for the new TARGAC-3 code.

<table>
<thead>
<tr>
<th>Subroutines</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTRAS</td>
<td>STG</td>
</tr>
<tr>
<td>CONTST</td>
<td></td>
</tr>
<tr>
<td>DELTED</td>
<td></td>
</tr>
<tr>
<td>ELIMIN</td>
<td></td>
</tr>
<tr>
<td>FINDR</td>
<td></td>
</tr>
<tr>
<td>GETDAT</td>
<td></td>
</tr>
<tr>
<td>ILMDAT</td>
<td></td>
</tr>
<tr>
<td>ILUMA</td>
<td></td>
</tr>
<tr>
<td>INITI</td>
<td></td>
</tr>
<tr>
<td>SGR</td>
<td></td>
</tr>
<tr>
<td>THERMB</td>
<td></td>
</tr>
<tr>
<td>THERML</td>
<td></td>
</tr>
</tbody>
</table>

B.2. LIST OF NEW ROUTINES

The following table lists the subroutines that have been added for the new TARGAC-3 code.

<table>
<thead>
<tr>
<th>Subroutines</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMBIN</td>
</tr>
<tr>
<td>DIFUSE</td>
</tr>
<tr>
<td>GETCLD</td>
</tr>
<tr>
<td>LAYERS</td>
</tr>
<tr>
<td>PCDIF</td>
</tr>
</tbody>
</table>
B.3. LIST OF MODIFIED COMMON BLOCKS

The following table lists the TARGAC COMMON blocks that have been modified for the new TARGAC-3 code.

**COMMON Blocks**

/CLOUD/
/GAUSS/
/RADIA/

B.4. FUNCTIONAL DESCRIPTIONS OF MODIFIED AND NEW ROUTINES

This section provides a brief description of each of the added or modified routines (subroutines or functions). The listing is arranged in alphabetical order. The name of the file containing each routine is shown in parentheses beside the name of the routine.

**SUBROUTINE CNTRAS (TARGAC.FOR)**

Calculates the target-background contrast for direct view optics, image intensifiers, silicon TVs, or thermal imagers.

**SUBROUTINE COMBIN (COMBIN.FOR)**

Combines the cloud properties of two cloud layers into a single "representative" layer.

**SUBROUTINE CONTST (CONTRAST.FOR)**

Calculates the path radiance, the transmission, the direct and diffuse radiance components, the inherent target and background radiances, and the apparent spectral contrast along the given line-of-sight.

**SUBROUTINE DELTED (CONTRAST.FOR)**

Calculates the direct and diffuse radiance components at each level using the delta-Eddington approximation.
SUBROUTINE DIFUSE (DIFUSE.FOR)

Computes the components of diffuse radiance at the target level.

SUBROUTINE ELIMIN (CONTRAST.FOR)

Solves a system of equations using Gaussian elimination.

SUBROUTINE FINDR (FINDR.FOR)

Calculates acquisition ranges for image intensifiers, direct view optics, silicon TVs, and thermal imagers for specified probability levels.

SUBROUTINE GETCLD (GETCLD.FOR)

Sets atmospheric data for layers containing clouds.

SUBROUTINE GETDAT (FINDR.FOR)

Initializes atmospheric data.

SUBROUTINE ILMDAT (ILMDAT.FOR)

Reads illumination or calls ILUMA to compute it as a function of geographical location, date, time, and meteorological conditions.

SUBROUTINE ILUMA (ILUMA.FOR)

Computes total solar/lunar illumination received at the ground as a function of geographical location, date, time, and meteorological conditions.

SUBROUTINE INITI (ILMDAT.FOR)

Initializes radiometric data.
SUBROUTINE LAYERS (LAYERS.FOR)

Computes the extinction coefficients and asymmetry parameters for the atmospheric layers that contain clouds.

SUBROUTINE PCDIF (PCDIF.FOR)

Computes average partly cloudy diffuse radiance component according to the sky condition.

SUBROUTINE SGR (SGR.FOR)

Determines sky-to-ground ratio for visible wavelengths based on illumination and meteorological conditions.

FUNCTION STG (SLNCOF.FOR)

Calculates the sky-to-ground ratio using a delta-Eddington radiative transfer algorithm.

SUBROUTINE THERMB (THERMB.FOR)

Assigns input data and displays inputs for thermal imager systems.

SUBROUTINE THERML (THERML.FOR)

Reads input data for thermal imager systems
This appendix contains Engineering Change Reports (ECRs) for each change made to TARGAC. ECRs are used by Hughes STX to maintain an audit trail of code evolution. An ECR is generated each time an existing routine is modified or a new routine is added. ECRs are numbered sequentially, first by routine number and second by change number. In-line documentation referencing the appropriate ECR number is also provided for each change.

Appendix B gives a functional description for each of the 18 routines that were changed or added. Here, we include the ECRs with code listings. For each changed or added routine, the complete code listing is provided. Our intent is to make it as easy as possible for the TARGAC maintenance programmer to locate each change.

The table below provides a list of all ECRs for the implementation of cloud shadows. The following pages provide all ECRs for each routine, followed by the code listing for that routine.

### TABLE C-1
Engineering Change Reports

<table>
<thead>
<tr>
<th>ECR Numbers</th>
<th>Routine Name</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1 through 1-9</td>
<td>SGR</td>
<td>SGR.FOR</td>
</tr>
<tr>
<td>2-1 through 2-6</td>
<td>STG</td>
<td>SLNCOF.FOR</td>
</tr>
<tr>
<td>3-1</td>
<td>ELIMIN</td>
<td>CONTRAST.FOR</td>
</tr>
<tr>
<td>4-1 and 4-2</td>
<td>ILUMA</td>
<td>ILUMA.FOR</td>
</tr>
<tr>
<td>5-1</td>
<td>COMBIN</td>
<td>COMBIN.FOR</td>
</tr>
<tr>
<td>6-1</td>
<td>INITI</td>
<td>ILMDAT.FOR</td>
</tr>
<tr>
<td>7-1 and 7-2</td>
<td>DELTED</td>
<td>CONTRAST.FOR</td>
</tr>
<tr>
<td>8-1 through 8-5</td>
<td>CONTST</td>
<td>CONTRAST.FOR</td>
</tr>
<tr>
<td>9-1 through 9-6</td>
<td>GETCLD</td>
<td>GETCLD.FOR</td>
</tr>
<tr>
<td>10-1 and 10-2</td>
<td>PCDIF</td>
<td>PCDIF.FOR</td>
</tr>
<tr>
<td>11-1</td>
<td>GETDAT</td>
<td>FINDR.FOR</td>
</tr>
<tr>
<td>12-1 through 12-3</td>
<td>FINDR</td>
<td>FINDR.FOR</td>
</tr>
<tr>
<td>13-1</td>
<td>THERML</td>
<td>THERML.FOR</td>
</tr>
<tr>
<td>14-1 and 14-2</td>
<td>THERMB</td>
<td>THERMB.FOR</td>
</tr>
<tr>
<td>15-1</td>
<td>LAYERS</td>
<td>LAYERS.FOR</td>
</tr>
<tr>
<td>16-1</td>
<td>CNTRAS</td>
<td>TARGAC.FOR</td>
</tr>
<tr>
<td>17-1</td>
<td>DIFUSE</td>
<td>DIFUSE.FOR</td>
</tr>
<tr>
<td>18-1</td>
<td>ILMDAT</td>
<td>ILMDAT.FOR</td>
</tr>
</tbody>
</table>
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: SGR  Old Date: 5/18/93
File Name: SGR.FOR  New Date: 8/11/93

Implemented By: Michael Oberlatz

Reason for Revision: Added code necessary to implement the input of the three cloud layers.

Description of Revision: Added variables, CF2, CF3, ZC2, and ZC3, to contain fractional cloud cover and base height for two additional cloud layers. Added these variables to the CLOUD COMMON block. Added necessary code to input the middle and high cloud data.

Notes:

As appropriate, attach the following:
1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name:   SGR   Old Date:  8/11/93
File Name:     SGR.FOR   New Date:  8/23/93

Implemented By:  Don Hamann

Reason for Revision:  Expanded CLOUD COMMON block to maintain information on multiple cloud layers. Added loop to count the number of cloud layers in order to determine if cloud layers need to be combined.

Description of Revision:  Added variables NUMCLD, MLOOP, CLDG(2), CLDBTA(2), LVRCLD(2), THK1, THK2, THK3, ITY1, ITY2, and ITY3 to CLOUD COMMON block. NUMCLD incremented for each layer containing some cloud fraction.

Notes:

As appropriate, attach the following:
1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: SGR  Old Date: 8/23/93

File Name: SGR.FOR  New Date: 9/14/93

Implemented By: Michael Oberlatz

Reason for Revision: Updated the batch mode for the three cloud layers; added cloud type input; changed the rules for ceiling height, inversion height and cloud thickness.

Description of Revision: Changed the records that the SGR routine uses for the cloud inputs from METD to the records LCLD, MCLD, and HCLD. Added new inputs for the cloud type. The ceiling height rule was changed to be the height of the lowest layer with cloud fraction greater than 0.7. The inversion height rule was changed so that all three cloud layers are checked before it is asked for. Added cloud thickness for each cloud type. Left total THICK the same.

Notes: ____________________________

_____________________________

_____________________________

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-4
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: SGR Old Date: 9/14/93
File Name: SGR.FOR New Date: 9/14/93

Implemented By: Don Hamann

Reason for Revision: To maintain information on the probability of a cloud-free path through each cloud layer and on the probability of the target scene in direct light.

Description of Revision: Added variable PSCLD and array PCF(2) to CLOUD COMMON block to maintain information for each cloud layer.

Notes: 

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-5
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: SGR  Old Date: 9/14/93
File Name: SGR.FOR  New Date: 9/14/93

Implemented By: Don Hamann

Reason for Revision: To save information on the sky to ground (SOG) ratio for both clear and in cloud shadow bracketing conditions.

Description of Revision: Converted the variable SOG into an array containing the clear and cloud shadow sky to ground ratio.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: SGR Old Date: 9/14/93
File Name: SGR.FOR New Date: 9/17/93

Implemented By: Melanie J. Gouveia

Reason for Revision: Fix the error that cloud base heights were out of range for batch mode inputs. Also, add cloud situation flag.

Description of Revision: For batch mode, check the cloud base height inputs only if the appropriate cloud fractions are greater than zero. Add the variable ICLDF to the /CLOUD/ COMMON block.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: SGR  Old Date: 9/17/93
File Name: SGR.FOR  New Date: 10/7/93

Implemented By: Michael Oberlatz

Reason for Revision: Several constant values sent to the XSCALE routine were being changed in XSCALE. This caused a run-time error for the second cycle of a multiple cycle run.

Description of Revision: Variables set to the constant values were substituted for the constant values in the calls to the XSCALE routine.

Notes: Using temporary variables to store the constant values eliminated the run-time error.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: SGR          Old Date: 10/7/93
File Name: SGR.FOR          New Date: 10/7/93

Implemented By: Dan DeBenedictis

Reason for Revision: The background reflectance (BKREF) was not being saved to TAC.SAV when climate data was being used. BKREF is now being saved with record CONTEXTL. Saving RECVAL(8,7) (BKREF) on the METD record is no longer necessary.

Description of Revision: Omitted RECVAL(8,7) from the WRITE statement for writing the record METD to TAC.SAV.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: SGR      Old Date: 10/7/93
File Name: SGR.FOR      New Date: 10/20/93

Implemented By: Dan DeBenedictis

Reason for Revision: All of the conditions included in the significant weather input are now included as part of other inputs. Also, the ceiling height assignment was changed to match the Air Force definition.

Description of Revision: Eliminate the SIGWX input for interactive mode and from the ILUM record for batch mode. The ceiling height is now assigned to that height at which clouds at and below the height cover more than 4/8 of the sky. Since the Air Force defines cloud coverage in eighths, this means that the coverage must be at least 5/8, or approximately 0.6.

Notes: This definition of ceiling height was taken from AWSR 105-24, Vol. 1, 1 March 1983.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
**C**

**C***REV 1/91

**COMMON /ILDATA/ FMONTH, DAY, YEAR, GTIME, SLAT, SLON, ILR1, ILR2, + IR3, RG, FR1, FR2, FR3, SIGWX, OBSURF, CEILHT, + PRTPY, FRC, ITARG
**COMMON /ILUMCM/ ATLTS, AZIS, ALTIN, AZIM, DPHASE, ELUMI, SUNLIT, + MOOLIT, TCLSUN, TCLUDN, RCLSUN, RCLUDN, + RCLUDN
**COMMON /ILLUMI/ AL, ILLUM, L22, AACK, IL1, IL2
**COMMON /IOFILE/IOFILE

**C**

**C-11**
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-3-----------------------------

C GEOMETRIC THICKNESSES FOR THE FIVE CLOUD TYPES

DIMENSION CTHICK(5)
DATA CTHICK /0.3, 0.5, 0.5, 0.75, 1.5/

C-HSTX---SCENE SHADOWS-----------------------------

C***REV 1/91

DATA RAD / 57.29578 /
CRF XSCALE92 DEFAULT VALUES START 17 APR 92
DO 701 IX = 1,3
    DECRPER(IX) = 0.0.
    XMEAN(IX) = 0.0
    XMODE(IX) = 0.0
701 CONTINUE
    IWATER = 0
    NBR = 0
    DO 702 IX = 1,20
        WAVRFN(IX) = 0.0
        RESPFN(IX) = 0.0
702 CONTINUE
ALT = 0.0
CP ALT IS NEVER USED, BUT IT COULD BE IN CALL TO XSSCALE.
CP NUMBER 0.0 IS USED TO CALL XSSCALE INSTEAD. NOV 92 PSG.
CRF XSCALE92 DEFAULT VALUES STOP 17 APR 92

C

C *** THIS SUBROUTINE CALCULATES SKY-TO-GROUND RATIO FOR VISIBLE
C WAVELENGTHS BASED ON VISIBILITY, SURFACE ALBEDO, SUN ANGLE, ETC.
C
CUV DIMENSION ALBEDO(4)
C
C SET DEFAULT AEROSOL STRUCTURE THIS SETS THE VALUE OF
C BETAA AT 18 LEVELS UP TO 20 KM.
C
CALL GETDAT
C
C INPUT DATA FOR SKY TO GROUND RATIO CALCULATIONS
C PROMPT FOR OPTION TO CALL EOSAEL ROUTINE CLIMAT
C
RD = 0.002
RDTEMP = RD
IF (INTER) THEN
  50 CONTINUE
WRITE(IOOUT,*)‘YOU WILL BE ASKED NOW TO PROVIDE INPUT RELATING TO’
WRITE(IOOUT,*)‘THE TIME AND PLACE OF INTEREST AND VARIOUS WEATHER’
WRITE(IOOUT,*)‘RELATED PARAMETERS. SOME OF THIS DATA MAY BE’
WRITE(IOOUT,*)‘SUPPLIED BY THE EOSAEL ROUTINE CLIMAT. IN’
WRITE(IOOUT,*)‘ORDER TO USE CLIMAT YOU WILL NEED TO BE AWARE’
WRITE(IOOUT,*)‘OF THE VARIOUS INPUT PARAMETERS, WHICH ARE’
WRITE(IOOUT,*)‘DEScribed IN VOLUME 8 OF THE EOSAEL REPORTS.’
WRITE(IOOUT,*)‘ADDITIONALLY, INPUT DATA FILES FROM THE CLIMAT’
WRITE(IOOUT,*)‘DIRECTORY ARE NECESSARY. IF YOU WISH TO’
WRITE(IOOUT,*)‘USE CLIMATOLOGICAL DATA WHenever POSSIBLE’
WRITE(IOOUT,*)‘IN THIS ROUTINE, ENTER 1.’
WRITE(IOOUT,*)‘IF YOU WISH TO INPUT YOUR OWN INFORMATION,’
WRITE(IOOUT,*)‘ENTER 0.’
READ(IOIN,*) ICLIM
WRITE(IOOUT,*)ICLIM
WRITE(IOOUT,*)ICLIM
WRITE(IOOUT,*)
IF (ICLIM .NE. 1 .AND. ICLIM .NE. 0) THEN
  WRITE(IOOUT,*)‘YOU MUST ENTER 0 OR 1; TRY AGAIN’
GOTO 50

C-12
ENDIF
ELSE
IF (RECUSE(2)) THEN
  ICLIM = 1
ELSE
  ICLIM = 0
ENDIF
ENDIF
ICLMAT = (ICLIM .EQ. 1)
CP BEGINT WRITTEN INFORMATION ABOUT THE CLIMAT OPTION (C.BACA JULY 92)
IF (ICLMAT) THEN
IF (INTER) THEN
51 CONTINUE
WRITE(IOOUT,*),'CHOOSE A GENERAL LOCATION FROM THE LIST BELOW'
WRITE(IOOUT,*),'
WRITE(IOOUT,*), 1 - FOR CENTRAL EUROPE   (1-4)
WRITE(IOOUT,*), 2 - FOR MID-EAST      (5-10)
WRITE(IOOUT,*), 3 - FOR KOREA        (11-13)
WRITE(IOOUT,*), 4 - FOR ALASKA       (14-16)
WRITE(IOOUT,*), 5 - FOR SCANDINAVIA  (17-18)
WRITE(IOOUT,*), 6 - FOR CENTRAL AMERICA (19-21)
WRITE(IOOUT,*), 7 - FOR MEXICO       (22-25)
WRITE(IOOUT,*), 8 - FOR SOUTH AMERICA (26-31)
WRITE(IOOUT,*), 9 - FOR INDIA        (32-34)
WRITE(IOOUT,*), 10 - FOR SOUTHEAST ASIA (35)
WRITE(IOOUT,*), 11 - FOR SOUTHERN EUROPE (36-47)
WRITE(IOOUT,*), 12 - FOR CANADA AND USA (48-74)
READ(IOIN,*), IGENRL
IF(IGENRL .LT. 1 .OR. IGENRL .GT. 12) THEN
  WRITE(IOOUT,*),'YOU MUST CHOOSE A NUMBER FROM 1 TO 12'
  GOTO 51
ENDIF
IF(IGENRL .EQ. 1) THEN
21 WRITE(IOOUT,*),'INPUT THE REGION CODE NUMBER FOR ' + 'CENTRAL EUROPE'
WRITE(IOOUT,*),'
WRITE(IOOUT,*), REGION 1 - EUROPEAN LOWLANDS'
WRITE(IOOUT,*), REGION 2 - EUROPEAN RHINE VALLEY'
WRITE(IOOUT,*), REGION 3 - EUROPEAN HIGHLANDS'
WRITE(IOOUT,*), REGION 4 - EUROPEAN ALPINE'
READ(IOIN,*), LOCAT
IF(LOCAT .LT. 1 .OR. LOCAT .GT. 4) THEN
  WRITE(IOOUT,*),'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
  GOTO 21
ENDIF
ELSE IF(IGENRL .EQ. 2) THEN
22 WRITE(IOOUT,*),'INPUT THE REGION CODE NUMBER FOR MID-EAST'
WRITE(IOOUT,*),'
WRITE(IOOUT,*), REGION 5 - MIDEAST DESERTS'
WRITE(IOOUT,*), REGION 6 - MIDEAST COASTAL'
WRITE(IOOUT,*), REGION 7 - MIDEAST PERSIAN GULF'
WRITE(IOOUT,*), REGION 8 - MIDEAST RED SEA'
WRITE(IOOUT,*), REGION 9 - MIDEAST EASTERN MOUNTAINS'
WRITE(IOOUT,*), REGION 10 - MIDEAST INDUS VALLEY'
READ(IOIN,*), LOCAT
IF(LOCAT .LT. 5 .OR. LOCAT .GT. 10) THEN
  WRITE(IOOUT,*),'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
  GOTO 22
ENDIF
ELSE IF(IGENRL .EQ. 3) THEN
23 WRITE(IOOUT,*),'INPUT THE REGION CODE NUMBER FOR KOREA'
WRITE(IOOUT,*),'
WRITE(IOOUT,*), REGION 11 - KOREAN EAST COAST'
WRITE(IOOUT,*), REGION 12 - SOUTH KOREA'
WRITE(IOOUT,*), REGION 13 - WEST KOREA'
C-13
READ (IOIN,*) LOCAT
IF (LOCAT .LT. 11 .OR. LOCAT .GT. 13) THEN
  WRITE (IOOUT,*) 'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
  GOTO 23
ENDIF
ELSE IF (IGENRL .EQ. 4) THEN
  WRITE (IOOUT,*) 'INPUT THE REGION CODE NUMBER FOR ALASKA'
  WRITE (IOOUT,*) '  REGION 14 - Alaskan Tundra'
  WRITE (IOOUT,*) '  REGION 15 - Alaskan Subarctic Continental'
  WRITE (IOOUT,*) '  REGION 16 - Alaskan Southern Coast'
  READ (IOIN,*) LOCAT
  IF (LOCAT .LT. 14 .OR. LOCAT .GT. 16) THEN
    WRITE (IOOUT,*) 'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
    GOTO 24
  ENDIF
ELSE IF (IGENRL .EQ. 5) THEN
  WRITE (IOOUT,*) 'INPUT THE REGION CODE NUMBER FOR SCANDINAVIA'
  WRITE (IOOUT,*) '  REGION 17 - Western Scandinavia'
  WRITE (IOOUT,*) '  REGION 18 - Eastern Scandinavia'
  READ (IOIN,*) LOCAT
  IF (LOCAT .LT. 17 .OR. LOCAT .GT. 18) THEN
    WRITE (IOOUT,*) 'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
    GOTO 25
  ENDIF
ELSE IF (IGENRL .EQ. 6) THEN
  WRITE (IOOUT,*) 'INPUT THE REGION CODE NUMBER FOR CENTRAL AMERICA'
  WRITE (IOOUT,*) '  REGION 19 - Central America Pacific Side'
  WRITE (IOOUT,*) '  REGION 20 - Central America Interior'
  WRITE (IOOUT,*) '  REGION 21 - Central America Atlantic Side'
  READ (IOIN,*) LOCAT
  IF (LOCAT .LT. 19 .OR. LOCAT .GT. 21) THEN
    WRITE (IOOUT,*) 'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
    GOTO 26
  ENDIF
ELSE IF (IGENRL .EQ. 7) THEN
  WRITE (IOOUT,*) 'INPUT THE REGION CODE NUMBER FOR MEXICO'
  WRITE (IOOUT,*) '  REGION 22 - Mexico Subtropical'
  WRITE (IOOUT,*) '  REGION 23 - Mexico Pacific'
  WRITE (IOOUT,*) '  REGION 24 - Mexico Highlands'
  WRITE (IOOUT,*) '  REGION 25 - Mexico Tropical'
  READ (IOIN,*) LOCAT
  IF (LOCAT .LT. 22 .OR. LOCAT .GT. 25) THEN
    WRITE (IOOUT,*) 'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
    GOTO 27
  ENDIF
ELSE IF (IGENRL .EQ. 8) THEN
  WRITE (IOOUT,*) 'INPUT THE REGION CODE NUMBER FOR SOUTH AMERICA'
  WRITE (IOOUT,*) '  REGION 26 - South America Tropics'
  WRITE (IOOUT,*) '  REGION 27 - South America Desert West'
  WRITE (IOOUT,*) '  REGION 28 - South America Desert Central'
  WRITE (IOOUT,*) '  REGION 29 - South America Subtropics'
  WRITE (IOOUT,*) '  REGION 30 - South America Subpolar'
  WRITE (IOOUT,*) '  REGION 31 - South America Highlands'
  READ (IOIN,*) LOCAT
  IF (LOCAT .LT. 26 .OR. LOCAT .GT. 31) THEN
    WRITE (IOOUT,*) 'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
    GOTO 28
  ENDIF
ENDIF
ELSE IF(IGENRL .EQ. 9) THEN
  29 WRITE(IOOUT,*)'INPUT THE REGION CODE NUMBER FOR INDIA'
  WRITE(IOOUT,*)'
  WRITE(IOOUT,*)' REGION 32 - INDIA WEST/CENTRAL REGION'
  WRITE(IOOUT,*)' REGION 33 - INDIA NORTHERN VALLEYS'
  WRITE(IOOUT,*)' REGION 34 - INDIA TROPICAL AREA'
  READ(IOIN,*) LOCAT
  IF(LOCAT .LT. 32 .OR. LOCAT .GT. 34) THEN
    WRITE(IOOUT,*)'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
    GOTO 29
  ENDIF
ELSE IF(IGENRL .EQ. 10) THEN
  30 WRITE(IOOUT,*)'INPUT THE REGION CODE NUMBER FOR '+
    'SOUTHEAST ASIA'
  WRITE(IOOUT,*)'
  WRITE(IOOUT,*)' REGION 35 - SOUTHEAST ASIA'
  READ(IOIN,*) LOCAT
  IF(LOCAT .LT. 35 .OR. LOCAT .GT. 35) THEN
    WRITE(IOOUT,*)'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
    GOTO 30
  ENDIF
ELSE IF(IGENRL .EQ. 11) THEN
  31 WRITE(IOOUT,*)'INPUT THE REGION CODE NUMBER FOR '+
    'SOUTHERN EUROPE'
  WRITE(IOOUT,*)'
  WRITE(IOOUT,*)' REGION 36 - EUROPEAN ADRIATIC'
  WRITE(IOOUT,*)' REGION 37 - EUROPEAN AGEAN'
  WRITE(IOOUT,*)' REGION 38 - EUROPEAN BALKAN HIGHLANDS'
  WRITE(IOOUT,*)' REGION 39 - EUROPEAN BALKAN PLAINS'
  WRITE(IOOUT,*)' REGION 40 - EUROPEAN DINARIC ALPS'
  WRITE(IOOUT,*)' REGION 41 - EUROPEAN PO VALLEY'
  WRITE(IOOUT,*)' REGION 42 - EUROPEAN CENT. MEDITERRANEAN'
  WRITE(IOOUT,*)' REGION 43 - EUROPEAN RHONE VALLEY'
  WRITE(IOOUT,*)' REGION 44 - EUROPEAN FRENCH PLATEAU'
  WRITE(IOOUT,*)' REGION 45 - EUROPEAN NW. MEDITERRANEAN'
  WRITE(IOOUT,*)' REGION 46 - EUROPEAN SPANISH PLATEAU'
  WRITE(IOOUT,*)' REGION 47 - EUROPEAN ATLANTIC COAST'
  READ(IOIN,*) LOCAT
  IF(LOCAT .LT. 36 .OR. LOCAT .GT. 47) THEN
    WRITE(IOOUT,*)'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
    GOTO 31
  ENDIF
ELSE
  32 WRITE(IOOUT,*)'INPUT THE REGION CODE NUMBER FOR '+
    'CANADA AND USA'
  WRITE(IOOUT,*)'
  WRITE(IOOUT,*)' 48 - WESTERN CANADA'
  WRITE(IOOUT,*)' 49 - SACRAMENTO VALLEY'
  WRITE(IOOUT,*)' 50 - NORTHERN ROCKY MOUNTAINS'
  WRITE(IOOUT,*)' 51 - CENTRAL ROCKY MOUNTAINS'
  WRITE(IOOUT,*)' 52 - SOUTHERN ROCKY MOUNTAINS'
  WRITE(IOOUT,*)' 53 - SOUTHWESTERN DESERT'
  WRITE(IOOUT,*)' 54 - NORTHERN INTER-MOUNTAIN'
  WRITE(IOOUT,*)' 55 - SOUTHERN INTER-MOUNTAIN'
  WRITE(IOOUT,*)' 56 - CANADIAN PRAIRIE'
  WRITE(IOOUT,*)' 57 - NORTHERN GREAT PLAINS'

C-15
WRITE(IOOUT,*)'YOU MUST CHOOSE A NUMBER FROM THIS LIST'
GOTO 32
ENDIF
ENDIF
52 CONTINUE
WRITE(IOOUT,*)'INPUT THE MONTH OF INTEREST FROM 1 TO 12'
READ(IOIN,*) MONTH
IF (MONTH .LT. 1 .OR. MONTH .GT. 12) THEN
WRITE(IOOUT,*)'THE MONTH MUST BE BETWEEN 1 AND 12'
GOTO 52
ENDIF
53 WRITE(IOOUT,*)
+' INPUT THE CLIMATOLOGY CLASS NUMBER FROM 1 TO 22'
WRITE(IOOUT,*)' 1 - FOG, HAZE AND MIST WITH VIS LT 1 KM'
WRITE(IOOUT,*)' 2 - FOG, HAZE AND MIST WITH VIS GE 1, LT 3 KM'
WRITE(IOOUT,*)' 3 - FOG, HAZE AND MIST WITH VIS GE 3, LT 7 KM'
WRITE(IOOUT,*)' 4 - FOG, HAZE AND MIST WITH VIS GE 7 KM'
WRITE(IOOUT,*)' 5 - DUST WITH VIS LT 3 KM'
WRITE(IOOUT,*)' 6 - DUST WITH VIS GE 3 KM'
WRITE(IOOUT,*)' 7 - DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM'
WRITE(IOOUT,*)' 8 - DRIZZLE, RAIN AND TSTMS WITH VIS GE 1, LT 3KM'
WRITE(IOOUT,*)' 9 - DRIZZLE, RAIN AND TSTMS WITH VIS GE 3, LT 7KM'
WRITE(IOOUT,*)' 10 - DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM'
WRITE(IOOUT,*)' 11 - SNOW WITH VIS LT 1 KM'
WRITE(IOOUT,*)' 12 - SNOW WITH VIS GE 1, LT 3 KM'
WRITE(IOOUT,*)' 13 - SNOW WITH VIS GE 3, LT 7 KM'
WRITE(IOOUT,*)' 14 - SNOW WITH VIS GE 7 KM'
WRITE(IOOUT,*)' 15 - NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M'
WRITE(IOOUT,*)' 16 - NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M'
WRITE(IOOUT,*)' 17 - VIS LT 1 KM AND CEILING HEIGHT LT 300 M'
WRITE(IOOUT,*)' 18 - VIS LT 3 KM AND CEILING HEIGHT LT 1000 M'
WRITE(IOOUT,*)' 19 - CEILING HEIGHT LT 300 M'
WRITE(IOOUT,*)' 20 - CEILING HEIGHT LT 1000 M'
WRITE(IOOUT,*)' 21 - NO CEILING'
WRITE(IOOUT,*)' 22 - ALL CONDITIONS COMBINED'
READ(IOIN,*) ICLASS
WRITE(IOOUT,*)
IF(ICLASS .LT. 1 .OR. ICLASS .GT. 22) THEN
WRITE(IOOUT,*)'THE CLASS NUMBER MUST BE BETWEEN 1 AND 22'
WRITE(IOOUT,*)'HIT RETURN TO CONTINUE'
READ(IOIN,300) IDUM
ENDIF
CP IDUM WILL SHOW UP AS SET BUT NEVER USED IN CODE CHECKERS.
CP IT IS BEING USED TO STOP SCREEN IN THIS FORTRAN CONFIGURATION.
300 FORMAT(I1)
GOTO 53
ENDIF
CP END INFORMATION ABOUT CLIMAT OPTION (C.BACA JULY 92)
CD REV 8/18/89
WRITE(21,54)'CLIM',FLOAT(LOCAT),FLOAT(MONTH),FLOAT(ICLASS)
54 FORMAT(A4,6X,3E10.4)
CP REV 8/18/89
ELSE
   LOCAT = NINT(RECVAL(2,1))
   MONTH = NINT(RECVAL(2,2))
   ICLASS = NINT(RECVAL(2,3))
   CALL INTCHK(47,1,LOCAT,'CLIM','1ST',1)
   CALL INTCHK(12,1,MONTH,'CLIM','2ND',1)
   CALL INTCHK(22,1,ICLASS,'CLIM','3RD',1)
ENDIF
ENDIF
ENDIF
IF (INTER) THEN
   100 WRITE(IOOUT,*)'INPUT THE JULIAN DAY NUMBER 1 TO 366'
      READ(IOIN,*) DATE
      WRITE(IOOUT,*)DATE
      WRITE(IOOUT,*)
      IF (DATE .LT. 1 .OR. DATE .GT. 366) THEN
         WRITE(IOOUT,*)'YOU MUST ENTER A DATE FROM 1 TO 366'
         WRITE(IOOUT,*)
         GOTO 100
      ENDIF
   C***REV 1/91 NEW INTERACTIVE INPUT--REQUESTING YEAR OF INTEREST
   C AS REQUIRED BY ILUMA MODULE
   104 WRITE(IOOUT,*)'INPUT THE YEAR, FROM 1977 TO 1999'
      READ(IOIN,*) YEAR
      WRITE(IOOUT,*)YEAR
      WRITE(IOOUT,*)
      IF (YEAR .LT. 1977.0 .OR. YEAR .GT. 1999.0) THEN
         WRITE(IOOUT,*)'YOU MUST ENTER A YEAR FROM 1977 TO 1999'
         WRITE(IOOUT,*)
         GOTO 104
      ENDIF
   110 WRITE(IOOUT,*)'INPUT THE TIME IN HH.MM ZULU'
      WRITE(IOOUT,*)TIME
      WRITE(IOOUT,*)
      IF (TIME .LT. 0.00 .OR. TIME .GT. 24.00) THEN
         WRITE(IOOUT,*)'YOU MUST ENTER A TIME FROM 00.00 TO 24.00'
         WRITE(IOOUT,*)
         GOTO 110
      ENDIF
   120 WRITE(IOOUT,*)'ENTER THE LATITUDE IN DEGREES + FOR NORTH,'
      WRITE(IOOUT,*)'- FOR SOUTH.'
      WRITE(IOOUT,*)'THE LATITUDE MUST BE BETWEEN -90 AND +90 DEG.'
      READ(IOIN,*) ALAT
      WRITE(IOOUT,*)ALAT
      WRITE(IOOUT,*)
      IF (ALAT .LT. -90.0 .OR. ALAT .GT. 90.0) THEN
         WRITE(IOOUT,*)'YOU HAVE ENTERED AN ILLEGAL VALUE OF ' +
            'LATITUDE -- TRY AGAIN'
         WRITE(IOOUT,*)
         GOTO 120
      ENDIF
   130 WRITE(IOOUT,*)'ENTER THE LONGITUDE IN DEGREES + FOR WEST'
      WRITE(IOOUT,*)'- FOR EAST.'
      WRITE(IOOUT,*)'THE LONGITUDE MUST BE BETWEEN -180 AND +180'
      READ(IOIN,*) ALONG
      WRITE(IOOUT,*)ALONG
      WRITE(IOOUT,*)
      IF (ALONG .LT. -180.0 .OR. ALONG .GT. 180.0) THEN
         WRITE(IOOUT,*)'YOU HAVE ENTERED AN ILLEGAL VALUE OF ' +
            'LONGITUDE -- TRY AGAIN'
         C-17
WRITE(IOOUT,*)
GOTO 130
ENDIF
ELSE
DATE = NINT(RECVAL(5,1))
TIME = RECVAL(5,2)
ALAT = RECVAL(5,3)
ALONG = RECVAL(5,4)
C***REV 1/91 NEW BATCH INPUT--YEAR OF INTEREST REQUIRED BY ILUMA MODULE
YEAR = RECVAL(5,6)
CALL INTCHK (366.1,DATE,'GEOM','1ST',188)
CALL REALCK (24.0,0.0,TIME,'GEOM','2ND',12.0)
CALL REALCK (90.0,-90.0,ALAT,'GEOM','3RD',55.0)
CALL REALCK (180.0,-180.0,ALONG,'GEOM','4TH',9.0)
C***REV 1/91
CALL REALCK (1999.0,1977.0,YEAR,'GEOM','6TH',1991.0)
ENDIF
C***REV 1/91 CALLS CVRTJD TO CONVERT JULIAN DATE TO MONTH AND YEAR,
C CONVERTS TIME TO HUNDREDS HOURS (HHMM) FORMAT, AND
C STORES LATITUDE AND LONGITUDE AS REQUIRED INPUT
C TO THE ILUMA MODULE
CALL CVRTJD (DATE, YEAR, FMONTH, DAY
GTIME TIME
* 100.0
SLAT = ALAT
SLON = -ALONG
C CALL EOSAEL ROUTINE CLIMAT TO OBTAIN CLIMATOLOGICAL DATA
C
IF (ICLMAT) THEN
TIMLOC = (TIME - ALONG / 15.0)
IF (TIMLOC .GT. 23.0) TIMLOC = 24. - TIMLOC
NHOUR = NINT(TIMLOC)
CALL CLIMAT ( LOCAT, MONTH, NHOUR, ICLASS, 0, TEMP, PRESS, RH,
+ AH, DP, VIS, WNDVEL, WINDIR, IPASCT, CLDHT, CLDCVR)
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-3----------------------------------
C ASSIGN APPROPRIATE CLOUD LAYER: CLOUD BASE HEIGHT, FRACTION, AND TYPE.
C THE UNIT CONVERSIONS WERE REMOVED SINCE ALL CLOUD BASE HEIGHTS ARE NOW
C ENTERED IN KILOMETERS INSTEAD OF FEET.
C CF1 = CLDCVR / 100.0
C ZC1 = CLDHT * 3281.0
IF (CLDHT .LE. 2.0) THEN
CF3 = CLDCVR / 100.0
ZC3 = CLDHT
IF (CLDHT .LE. 1.2 .OR. CLDCVR .GT. 50.0) THEN
ITY3 = 4
ELSE
ITY3 = 5
ENDIF
ELSE IF (CLDHT .LE. 6.1) THEN
CF2 = CLDCVR / 100.0
ZC2 = CLDHT
ITY2 = 3
ELSE IF (CLDHT .LE. 20.0) THEN
CF1 = CLDCVR / 100.0
ZC1 = CLDHT
IF (CLDCVR .GT. 67.5) THEN
ITY1 = 2
ELSE
ITY1 = 1
ENDIF
ENDIF
C-HSTX---SCENE SHADOWS-----------------------------------------------
C TMP = TEMP + 273.2
C-18
ENDIF
IF (INTER) THEN
  140 WRITE(IOOUT,*) 'ENTER THE TARGET HEADING BETWEEN 0 AND 360 DEG'
  READ(IOIN,*) TARGAZ
  WRITE(IOOUT,*) TARGAZ
  WRITE(IOOUT,*)
  IF (TARGAZ .LT. 0.0 .OR. TARGAZ .GT. 360.0) THEN
    WRITE(IOOUT,*) 'THE TARGET HEADING MUST BE BETWEEN'
    WRITE(IOOUT,*) ' 0 AND 360 DEG --- TRY AGAIN'
    GOTO 140
  ENDIF
ENDIF
CCB THESE LINES ARE MOVED INTO THE INTERACTIVE BLOCK FROM BELOW ENDIF
CD REV 8/18/89
  152 FORMAT(A4,6X,6E10.4)
  CP152 FORMAT(A4,6X,1110,4E10.4)
  WRITE(21,152) 'GEOM', FLOAT(DATE), TIME, ALAT, ALONG, TARGAZ, YEAR
CD REV 8/18/89
CCB (11/26/91) END MOVED BLOCK
ELSE
  TARGAZ = RECVAL(5,5)
  CALL REALCK(360.0,0.0,TARGAZ,'GEOM','5TH',90.0)
ENDIF
IF (.NOT. ICLMAT) THEN
  IF (INTER) THEN
    700 WRITE(IOOUT,*) 'INPUT THE VISIBILITY IN KILOMETERS'
    READ(IOIN,*) VIS
    WRITE(IOOUT,*) VIS
    WRITE(IOOUT,*)
    IF (VIS .EQ. 0.0) VIS = 0.01
    IF (VIS .LT. 0.0) THEN
      WRITE(IOOUT,*) 'NEGATIVE VISIBILITY IS NOT POSSIBLE.'
      WRITE(IOOUT,*) 'PLEASE CHECK.'
      GOTO 700
   ENDIF
    DVIS = 3.912/VIS
    CONTINUE
C  C-HSTX---SCENE SHADOWS---ECR # HSTX-1-1---------------------------------
C CHANGED THE WORDING OF THE LOW CLOUD COVER INPUT TO READ "LOW
C CLOUDS" INSTEAD OF JUST CLOUDS; CHANGED THE VARIABLE FROM CF1 TO
C CF3; ALLOW BASE HEIGHT ENTRY FOR ANY FRACTION GREATER THAN ZERO;
C CHANGED THE BASE HEIGHT LIMIT TO BE LOW CLOUDS BETWEEN 0.1 AND
C 2.0 KM; THIS LIMIT IS THE SAME AS THE EOTDA VERSION 2.0
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-3---------------------------------
C ADD INPUTS FOR A LOW CLOUD TYPE; ASSIGN CLOUD THICKNESS BASED ON TYPE
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-9---------------------------------
C FOR A CLOUD ON THE GROUND, USE TYPE STRATUS. REARRANGE SOME LINES TO
C MAKE MORE SENSE WITH DVIS IF STATEMENTS.
  IF (DVIS .LT. 7.0) THEN
    149 WRITE(IOOUT,*) 'ENTER THE FRACTIONAL LOW CLOUD COVER'
    WRITE(IOOUT,*) 'BETWEEN 0 AND 1'
    READ(IOIN,*) CF3
    WRITE(IOOUT,*) CF3
    WRITE(IOOUT,*)
    IF (CF3 .LT. 0.0 .OR. CF3 .GT. 1.0) THEN
      WRITE(IOOUT,*) 'THE LOW CLOUD COVER MUST BE BETWEEN'
      WRITE(IOOUT,*) ' 0 AND 1 TRY AGAIN'
      GOTO 149
    ENDIF
    ELSE
      CF3 = 1.0
    ENDIF
    IF (CF3 .GT. 0.0) THEN
C-19
IF (DVIS .LT. 7.0) THEN
  WRITE(IOOUT,*)'ENTER THE LOW CLOUD BASE HEIGHT'
  WRITE(IOOUT,*)'BETWEEN 0.1 AND 2.0 KM'
  READ(IOIN,*) ZC3
  WRITE(IOOUT,*)ZC3
  WRITE(IOOUT,*)
  IF (ZC3 .LT. 0.1 .OR. ZC3 .GT. 2.0) THEN
    WRITE(IOOUT,*)'THE LOW CLOUD BASE HEIGHT MUST BE'
    WRITE(IOOUT,*)'BETWEEN 0.1 AND 2.0 KM'
    WRITE(IOOUT,*)'TRY AGAIN'
    GOTO 150
  ENDIF
  WRITE(IOOUT,*)'ENTER THE LOW CLOUD TYPE'
  WRITE(IOOUT,*)'4 STRATUS/STRATOCUMULUS'
  WRITE(IOOUT,*)'5 CUMULUS/CUMULONIMBUS'
  READ(IOIN,*) ITY3
  WRITE(IOOUT,*) ITY3
  WRITE(IOOUT,*)
  IF (ITY3 .LT. 4 .OR. ITY3 .GT. 5) THEN
    WRITE(IOOUT,*)'THE LOW CLOUD TYPE MUST BE'
    WRITE(IOOUT,*)'4 OR 5 TRY AGAIN'
    GOTO 164
  ENDIF
  THK3 = CTHICK(ITY3)
ELSE
  WRITE(IOOUT,*)'VISIBILITY IS LOW, A CLOUD IS ON THE GROUND'
  WRITE(IOOUT,*)'ENTER THE HEIGHT OF THE LOW CLOUD TOP'
  WRITE(IOOUT,*)'OR -1 IF IT IS UNKNOWN'
  READ(IOIN,*) ZH3
  WRITE(IOOUT,*)ZH3
  WRITE(IOOUT,*)
  IF (ZH3 .EQ. -1.0) ZH3 = 0.2
  IF (ZH3 .LT. 0.1 .OR. ZH3 .GT. 2.0) THEN
    WRITE(IOOUT,*)'THE LOW CLOUD TOP HEIGHT MUST BE'
    WRITE(IOOUT,*)'BETWEEN 0.1 AND 2.0 KM'
    WRITE(IOOUT,*)'TRY AGAIN'
    GOTO 167
  ENDIF
  ZC3 = 0.0
  THK3 = ZH3
  ITY3 = 4
ENDIF
ENDIF
C
C-20
WRITE(IOOUT,*)
IF (CF2 .LT. 0.0 .OR. CF2 .GT. 1.0) GOTO 159
ENDIF
IF (CF2 .GT. 0.0) THEN
ITY2 = 3
THK2 = CTHICK(ITY2)
CONTINUE
WRITE(IOOUT,*)'ENTER THE MIDDLE CLOUD BASE HEIGHT'
WRITE(IOOUT,*)'BETWEEN 2.0 AND 6.1 KM'
READ(IOIN,*) ZC2
WRITE(IOOUT,*)ZC2
WRITE(IOOUT,*)
IF (ZC2 .LT. 2.0 .OR. ZC2 .GT. 6.1) THEN
WRITE(IOOUT,*)'THE MIDDLE CLOUD BASE HEIGHT MUST BE'
WRITE(IOOUT,*)'BETWEEN 2.0 AND 6.1 KM'
WRITE(IOOUT,*)'TRY AGAIN'
END IF
ENDIF
WRITE(IOOUT,*)'ENTER THE FRACTIONAL HIGH CLOUD COVER'
WRITE(IOOUT,*)'BETWEEN 0 AND 1'
READ(IOIN,*) CF1
WRITE(IOOUT,*)CF1
WRITE(IOOUT,*)
IF (CF1 .LT. 0.0 .OR. CF1 .GT. 1.0) THEN
WRITE(IOOUT,*)'THE HIGH CLOUD COVER MUST BE BETWEEN'
WRITE(IOOUT,*)'0 AND 1 TRY AGAIN'
WRITE(IOOUT,*)'ENTER THE FRACTIONAL HIGH CLOUD COVER'
WRITE(IOOUT,*)'BETWEEN 0 AND 1'
READ(IOIN,*) CF1
WRITE(IOOUT,*)CF1
WRITE(IOOUT,*)
IF (CF1 .LT. 0.0 .OR. CF1 .GT. 1.0) GOTO 169
ENDIF
IF (CF1 .GT. 0.0) THEN
CONTINUE
WRITE(IOOUT,*)'ENTER THE HIGH CLOUD BASE HEIGHT'
WRITE(IOOUT,*)'BETWEEN 6.1 AND 13.7 KM'
READ(IOIN,*) ZC1
WRITE(IOOUT,*)ZC1
WRITE(IOOUT,*)
IF (ZC1 .LT. 6.1 .OR. ZC1 .GT. 13.7) THEN
WRITE(IOOUT,*)'THE HIGH CLOUD BASE HEIGHT MUST BE'
WRITE(IOOUT,*)'BETWEEN 6.1 AND 13.7 KM'
WRITE(IOOUT,*)'TRY AGAIN'
GOTO 156
ENDIF
WRITE(IOOUT,*)'ENTER THE HIGH CLOUD TYPE'
WRITE(IOOUT,*)'1 THIN CIRRUS'
WRITE(IOOUT,*)'2 THICK CIRRUS'
READ(IOIN,*) ITY1
WRITE(IOOUT,*)ITY1
WRITE(IOOUT,*)
IF (ITY1 .LT. 1 .OR. ITY1 .GT. 2) THEN
WRITE(IOOUT,*)'THE HIGH CLOUD TYPE MUST BE'
WRITE(IOOUT,*)'1 OR 2 TRY AGAIN'
WRITE(IOOUT,*)'ENTER THE HIGH CLOUD TYPE'
WRITE(IOOUT,*)'1 THIN CIRRUS'
WRITE(IOOUT,*)'2 THICK CIRRUS'
READ(IOIN,*) ITY1
WRITE(IOOUT,*)ITY1
WRITE(IOOUT,*)
IF (ITY1 .LT. 1 .OR. ITY1 .GT. 2) GOTO 174
ENDIF
THK1 = CTHICK(ITY1)

C-21
ENDIF

DUM = 0.0
WRITE(21,399)'HCLD',DUM,FLOAT(ITY1),CF1,ZC1
WRITE(21,399)'MCLD',DUM,FLOAT(ITY2),CF2,ZC2
WRITE(21,399)'LCLD',DUM,FLOAT(ITY3),CF3,ZC3

399 FORMAT(A4,6X,4E10.4)

C-HSTX-SCENE SHADOWS----------------------------------------------
C
ELSE

VIS = RECVAL(8,1)
CALL REALCK(200.0,0.1,VIS,'METD','1ST',7.0)

C-HSTX-SCENE SHADOWS-ECR # HSTX-1-3-------------------------------
C-UPDATED THE BATCH INPUTS TO USE THE LCLD, MCLD, AND HCLD RECORDS
C-FOR CLOUD INPUTS, RATHER THAN METD. THE UNIT CONVERSIONS WERE
C-REMOVED SINCE THE CLOUD BASE HEIGHTS ARE ENTERED IN KILOMETERS
C-STEAD OF FEET.
C-HSTX-SCENE SHADOWS-ECR # HSTX-1-6-------------------------------
C-CHECK CLOUD BASE HEIGHT INPUTS ONLY IF CLOUD FRACTION IS GREATER THAN 0.
C-CF1 = RECVAL(8,2)
C-ZC1 = RECVAL(8,3)
ITY1 = NINT(RECVAL(15,2))
CF1 = RECVAL(15,3)
ZC1 = RECVAL(15,4)
ITY2 = NINT(RECVAL(49,2))
CF2 = RECVAL(49,3)
ZC2 = RECVAL(49,4)
ITY3 = RECVAL(32,2)
CF3 = RECVAL(32,3)
ZC3 = RECVAL(32,4)
CALL INTCHK(2,0,ITY1,'HCLD','2ND',0)
CALL INTCHK(3,3,ITY2,'MCLD','2ND',0)
CALL INTCHK(5,4,ITY3,'LCLD','2ND',0)
CALL REALCK(1.0,0.0,CF1,'HCLD','3RD',0.0)
CALL REALCK(1.0,0.0,CF2,'MCLD','3RD',0.0)
CALL REALCK(1.0,0.0,CF3,'LCLD','3RD',0.0)
IF (CF1.GT.0.) CALL REALCK(13.7,6.1,ZC1,'HCLD','4TH',8.0)
IF (CF2.GT.0.) CALL REALCK(6.1,2.0,ZC2,'MCLD','4TH',4.0)
IF (CF3.GT.0.) CALL REALCK(2.0,0.1,ZC3,'LCLD','4TH',1.0)
IF (ITY3 .GT. 0) THK3 = CTHICK(ITY3)
IF (ITY2 .GT. 0) THK2 = CTHICK(ITY2)
IF (ITY1 .GT. 0) THK1 = CTHICK(ITY1)

DVIS = 3.912/VIS

C-HSTX-SCENE SHADOWS-----------------------------------------------
C-C-HSTX-SCENE SHADOWS-ECR # HSTX-1-1-------------------------------
C-USE THE VARIABLES ZH3, CF3, AND ZC3 INSTEAD OF ZH1, CF1, AND ZC1
C-HSTX-SCENE SHADOWS-ECR # HSTX-1-3-------------------------------
C-THE CEILING HEIGHT IS CHANGED TO BE THE HEIGHT OF THE LOWEST CLOUD
C-LAYER WITH A CLOUD FRACTION OF GREATER THAN 0.7. REMOVE UNIT
C-CONVERSION BECAUSE CLOUD HEIGHTS ARE NOW IN KM, RATHER THAN FEET.
C-CLOUD THICKNESS IS SET TO 0.2 KM IF ANY CLOUD LAYER IS PRESENT.
C-HSTX-SCENE SHADOWS-ECR # HSTX-1-9-------------------------------
C-CHANGE THE CEILING HEIGHT TO MATCH THE AIR FORCE DEFINITION: THE
C-HEIGHT AT WHICH CLOUDS AT AND BELOW THAT HEIGHT COVER MORE THAN 4/8
C-OF THE SKY (FROM AWSR 105-24). 5/8 (0.625) IS THE NEXT INCREMENT.
ZH3 = 300.0

ENDIF

ENDIF

C

IF (CF3 .GT. 0.7) THEN
C
IF (DVIS .LE. 7.0) THEN
C
CEILHT=ZC3*0.0003048
C
THICK = 0.20
C
ELSE
C
CEILHT = 0.0
C
C-22
C THICK = ZH3*0.0003048
C ENDIF
C ELSE
CEILHT = -1.0
THICK = 0.0
C ENDIF
CEILHT = -1.0
FRAC = CF3
IF (FRAC.GT.0.6) THEN
CEILHT = ZC3
ELSE
FRAC = FRAC + CF2
IF (FRAC.GT.0.6) THEN
CEILHT = ZC2
ELSE
FRAC = FRAC + CF1
IF (FRAC.GT.0.6) CEILHT = ZC1
ENDIF
ENDIF
IF ((CF3.GT.0.0) .OR. (CF2.GT.0.0) .OR. (CF1.GT.0.0)) THEN
THICK = 0.20
ELSE
THICK = 0.0
ENDIF
C-HSTX---SCENE SHADOWS-------------------------------
C
C***REV 1/91 STORES FRACTIONAL CLOUD COVER FOR USE BY ILUMA MODULE
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-1-----------------------
C CHANGED THE CLOUD FRACTION INPUTS FOR THE ILUMA VARIABLES -- FR1, C FR2, AND FR3 TO HANDLE THE NEW 3 CLOUD LAYER INPUTS INSTEAD OF THE C PREVIOUS 1 CLOUD LAYER INPUT
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-3-----------------------
C SET ILUMA VARIABLES ILR1, ILR2, AND ILR3 TO CLOUD TYPE INPUTS
C
FRC = CF1
FR1 = CF1
FR2 = CF2
FR3 = CF3
ILR1 = I1Y1
ILR2 = I1Y2
ILR3 = I1Y3
C-HSTX---SCENE SHADOWS-------------------------------
C
IF (INTER) THEN
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-3-----------------------
C THE RULE FOR ASKING FOR THE INVERSION HEIGHT WAS CHANGED TO ACCOUNT C FOR THE THREE CLOUD LAYERS
C IF (CF1 .LT. 0.001) THEN
C IF (CF1 .LT. 0.001 .AND. CF2 .LT. 0.001 .AND. CF3 .LT. 0.001) THEN
WRITE(IOOUT,*)'INPUT RADIATION FOG OR INVERSION HEIGHT IN KM'
WRITE(IOOUT,*)'IF ONE IS PRESENT (OTHERWISE INPUT -1.0)'
READ(IOIN,*) AINVHT
C-23
WRITE(IOOUT,*) AINVHT
WRITE(IOOUT,*)
END IF
ELSE
AINVHT = RECVAL(14,2)
CALL REALCK(10.0,-1.0,AINVHT,'XSCL','2ND',-1.0)
ENDIF
115 CONTINUE
IF (.NOT. ICLMAT) THEN
IF (INTER) THEN
WRITE(IOOUT,*)
WRITE(IOOUT,*) 'ENTER THE TEMPERATURE IN DEG C'
READ(IOIN,*) TMP
WRITE(IOOUT,*) TMP
WRITE(IOOUT,*)
IF (ABS(TMP) .GT. 60.0) THEN
WRITE(IOOUT,*)
+ 'YOU HAVE ENTERED AN EXTREME TEMPERATURE READING'
WRITE(IOOUT,*) 'PLEASE CHECK YOUR READINGS AND REENTER'
WRITE(IOOUT,*) 'THE TEMPERATURE INFORMATION.'
GOTO 115
ENDIF
CONTINUE
WRITE(IOOUT,*) 'ENTER THE DEWPOINT TEMPERATURE IN DEG C'
READ(IOIN,*) TDEW
WRITE(IOOUT,*) TDEW
WRITE(IOOUT,*)
IF (TDEW .GT. TMP) THEN
WRITE(IOOUT,*)
+ 'YOUR DEWPOINT TEMPERATURE INDICATES MORE THAN'
WRITE(IOOUT,*)/+ 100% HUMIDITY. PLEASE CHECK YOUR READINGS AND'
WRITE(IOOUT,*)/+ REENTER THE DEWPOINT TEMPERATURE INFORMATION.'
GOTO 111
ENDIF
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-3-------------------------------
C OMIT CLOUD FRACTION AND BASE HEIGHT FROM METD RECORD; THIS
C INFORMATION IS NOW STORED IN LCLD, MCLD, AND HCLD RECORDS FOR
C THREE CLOUD LAYERS
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-8-------------------------------
C OMIT RECVAL(8,7) (BACKGROUND REFLECTANCE, BKREF) FROM METD RECORD.
C THIS INFORMATION IS NOW STORED IN THE CONTEXTL RECORD.
CD REV 8/18/89
C  WRITE(21,112) 'METD',VIS,CF1,((ZC1/1000),THICK,TMP,TDEW,
C + RECVAL(8,7)
C  WRITE(21,112) 'METD',VIS,DUM,DUM,THICK,TMP,TDEW,
C + RECVAL(8,7)
C  DUM = 0.0
C  WRITE(21,112) 'METD',VIS,DUM,DUM,THICK,TMP,TDEW,DUM
C-HSTX---SCENE SHADOWS-----------------------------------------------
C 112 FORMAT(A4,6X,7E10.4)
CD REV 8/18/89
ELSE
TMP = RECVAL(8,5)
TDEW = RECVAL(8,6)
CALL REALCK(60.0,-60.0,TMP,'METD','5TH',10.0)
CALL REALCK(TMP,-60.0,TDEW,'METD','6TH',8.0)
ENDIF
TMP = TMP + 273.2
RH = 100.0*EXP(6885.06*(1.0/TMP-1.0/(TDEW+273.2))+5.31*
+ ALOG(TMP/(TDEW+273.2)))
ENDIF
CP SIGVIS = 3.912 / VIS USED IN CALL TO OLD XSCALE. PSG NOV 92.

IF (INTER) THEN
  WRITE(IOUT,*)
  WRITE(IOUT,*)'SELECT AEROSOL TYPE'
  WRITE(IOUT,*)' 1-MARITIME AIR MASS'
  WRITE(IOUT,*)' 2-URBAN'
  WRITE(IOUT,*)' 3-RURAL (CONTINENTAL POLAR)' 
  WRITE(IOUT,*)' 4-FOG (HEAVY ADVECTION)'
  WRITE(IOUT,*)' 5-FOG (MODERATE RADIATION)'
  WRITE(IOUT,*)' 6-RAIN (DRIZZLE)'
  WRITE(IOUT,*)' 7-RAIN (WIDESPREAD)'
  WRITE(IOUT,*)' 8-RAIN (THUNDERSTORM)'
  WRITE(IOUT,*)' 9-SNOW'
  WRITE(IOUT,*)' 10-SNOW'
  WRITE(IOUT,*)' 11-ICEFOG'

READ(IOIN,*)IAERO
WRITE(IOUT,*)IAERO
WRITE(IOUT,*)
ELSE
  IAERO = NINT(RECVAL(14,1))
  CALL INTCHK(9,1,IAERO,'XSCL','1ST',3)
ENDIF

TEMP = TMP - 273.2
ISLT = 1
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-9----------------------------------
C SIGWX IS NO LONGER USED; IF FOG AEROSOL SET ILR3 FOR ILUMA HERE.
IF ((IAERO.EQ.4 .OR. IAERO.EQ.5 .OR. IAERO.EQ.11) .AND.
    (CF3 .LE. 0.1)) ILR3 = 2
C-HSTX---SCENE SHADOWS------------------------------------------------------
C
CRF IF (IAERO .GE. 6 .AND. IAERO .LE. 8) THEN 17 APR 92
CRF REPLACE THE PREVIOUS LINE WITH THE FOLLOWING FOR XSCALE92
CALL INTCHK(11,1,IAERO,'XSCL','1ST',3)
ENDIF
ISLT = 1
C
C IOTEMP = IOIN
C IOIN = IPHFUN
CALL IOOPEN(KSTOR,'SCTH.UNT','SCRATCH',0,'FORMATTED',
            'TARGAC','SGR',LUNERR,*105)
RANGE = 5
IF (RANGE .GT. CEILHT .AND. CEILHT .GT. 0.0 ) RANGE = CEILHT
CRF CALL XSCAIL(0.45,0.7,-20.,SIGVIS,XSTRN,IERR,ISLT,3,
               RANGE,90.,CEILHT,0,RNRT,THICK,RH,WNDVEL)
CRF REMOVE THE PREVIOUS 2 LINES AND REPLACE WITH . . .
DUM1 = 90.0
DUM2 = 0.0
CP DUM1 AND DUM2 WERE DEFINED ABOVE TO SEND TO XSCALE BECAUSE
CP THEY ARE MODIFIED IN XSCALE.
CRF THE FOLLOWING 6 LINES FOR XSCALE92. 17 APR 92.

C-25
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-7-----------------------------
C CHANGED THE CONSTANTS SENT TO XSCALE TO VARIABLES. THESE VALUES
C MAY BE CHANGED IN XSCALE. THIS CAUSED PROBLEMS WHEN XSCALE ENDED,
C SINCE THE CONSTANTS WERE NO LONGER THE VALUES THAT THEY WERE
C SUPPOSED TO BE.
C
CALL XSCALE(0.45, 0.7, VIS, 3, 2, ISLT, NBR,
1 RD, DECPER, XMEAN, XMODE, IWATER,
2 WAVRFN, RESPFN,
3 RANGE, 90.0, 0.0, CEILHT, THICK, AINVHT,
4 RNRT, TEMP, WNDVEL, RH, XSTRN, IERR,
5 Q, UM, EXT551, QAVE)
DUM45 = 0.45
DUM7 = 0.7
IDUM3 = 3
IDUM2 = 2
DUM90 = 90.0
DUM0 = 0.0
CALL XSCALE(DUM45, DUM7, VIS, IDUM3, IDUM2, ISLT, NBR,
1 RD, DECPER, XMEAN, XMODE, IWATER,
2 WAVRFN, RESPFN,
3 RANGE, DUM90, DUM0, CEILHT, THICK, AINVHT,
4 RNRT, TEMP, WNDVEL, RH, XSTRN, IERR,
5 Q, UM, EXT551, QAVE)
C-HSTX---SCENE SHADOWS-----------------------------------------------
C
C REWIND IPHFUN
IOTEMP = IOIN
GOTO 106
105 IERR = 1
106 CONTINUE
CP106 CLOSE(KSTOR)
IF (IERR .EQ. 1) THEN
EFLAG = .TRUE.
RETURN
ENDIF
ISLT = 0
CRF ELSE IF (IAERO .EQ. 6 .OR. IAERO .EQ. 10) THEN 17 APR 92
CRF REPLACE THE PREVIOUS LINE WITH THE FOLLOWING FOR XSCALE92
ELSE IF (IAERO .EQ. 9) THEN
CUV TEMP2=TMP-273.2
RD=.002
IF (INTER) THEN
999 WRITE(IOOUT,*) 'INPUT WINDSPEED IN KNOTS'
READ(IOIN,*) WIND
WRITE(IOOUT,*) WIND
WRITE(IOOUT,*)
IF (WIND .LT. 0.0) THEN
WRITE(IOOUT,*)'NEGATIVE WINDSPEED IS NOT POSSIBLE.'
WRITE(IOOUT,*)'REENTER WINDSPEED.'
GOTO 999
ENDIF
IF (WIND .GT. 75.0) THEN
WRITE(IOOUT,*)'YOU HAVE ENTERED AN EXTREMELY HIGH'
WRITE(IOOUT,*)'WINDSPEED.'
WRITE(IOOUT,*)'REENTER WINDSPEED.'
GOTO 999
ENDIF
ELSE
WIND = RECVAL(14, 3)
CALL REALCK(75.0, 0.0, WIND, 'XSCL', '3RD', 0.0)
ENDIF
WNDVEL=WIND*0.5144
C IOTEMP = IOIN
C IOIN = IPHFUN

C-26
CALL IOOPEN(NPLOTU,'SCTH.UNT','SCRATCH',0,'FORMATTED',
+ 'TARGAC','SGR',LUNERR,*1000)

RANGE = 5
IF (RANGE GT CEILHT .AND. CEILHT GT 0.0 ) RANGE = CEILHT

CALL XSCALE(0.45,0.7,-20.,SIGVIS,XSTRN,IERR,ISLT,3,
+ RANGE,90.,CEILHT,0,RNRT,THICK,RH,WNDVEL)

REMOVE THE PREVIOUS 2 LINES AND REPLACE WITH . . .
DUM1 = 90.0
DUM2 = 0.0

DUM1 AND DUM2 WERE DEFINED ABOVE TO SEND TO XSCALE BECAUSE
THEY ARE MODIFIED IN XSCALE.

THE FOLLOWING 6 LINES FOR XSCALE92. 17 APR 92.

CALL XSCALE(0.45,0.7,VIS,3,2,ISLT,NBR,
+ RD,DECPER,XMEAN,XMODE,IWATER,
+ WAVRFN,RESPFN,
+ RANGE,90.0,0.0,CEILHT,THICK,AINVHT,
+ RNRT,TEMP,WNDVEL,RH,XSTRN,IERR,
+ Q,UM,EXT55I,QAVE)

DUM45 = 0.45
DUM7 = 0.7
IDUM3 = 3
IDUM2 = 2
DUM90 = 90.0
DUM0 = 0.0

CALL XSCALE(DUM45,DUM7,VIS,IDUM3,IDUM2,ISLT,NBR,
+ RD,DECPER,XMEAN,XMODE,IWATER,
+ WAVRFN,RESPFN,
+ RANGE,DUM90,DUM0,CEILHT,THICK,AINVHT,
+ RNRT,TEMP,WNDVEL,RH,XSTRN,IERR,
+ Q,UM,EXT55I,QAVE)

REWIND IPHFUN
GOTO 1002

IF (IERR .EQ. 1) THEN 
EFLAG = .TRUE.
RETURN 
ENDIF 
ISLT = 0 
ENDIF 

IF (IAERO .EQ. 6 .OR. IAERO .EQ. 10) THEN 
WRITE(21,1001)'XSCL',FLOAT(IAERO),AINVHT,WIND 
ELSE IF (IAERO .LE. 9 .AND. IAERO .GE. 7) THEN 
WRITE(21,1003)'XSCL',FLOAT(IAERO),AINVHT 
ELSE IF (IAERO .LE. 5 .AND. IAERO .GE. 1) THEN 
WRITE(21,1003)'XSCL',FLOAT(IAERO),AINVHT 
ENDIF 

ADD THE FOLLOWING LINE FOR XSCALE92

C-27
ELSE IF (IAERO .EQ. 11) THEN
WRITE(21,1003)'XSCL',FLOAT(IAERO),AINVHT
ENDIF

1003 FORMAT(A4,6X,2E10.4)
CP1003 FORMAT(A4,6X,1I10,1E10.4)
CD REV 8/18/89
END IF
C IOTEMP = IOIN
C IOIN = IPHFUN
CP CALL IOOPEN(NPLOTU,'SCTH.UNT','SCRATCH',0,'FORMATTED',
CP + 'TARGAC','SGR',LUNERR,*1010)
RANGE=5
IF (RANGE .GT. CEILHT .AND. CEILHT .GT. 0.0 ) RANGE = CEILHT
MAERO = IAERO
CRF IF (IAERO .GE. 4 .AND. IAERO .LE. 9) MAERO=3 17 APR 92
CRF REPLACE THE PREVIOUS LINE WITH THE FOLLOWING LINE FOR XSCALE92
IF (IAERO .GE. 4 .AND. IAERO .LE. 11) MAERO = 3
CRF CALL XSFAIL(0.45,0.7,-20.,SIGVIS,XSTRN,IERR,ISLT,MAERO,
CRF + RANGE,90.,CEILHT,0,RNRT,THICK,RH,WNDVEL)
CRF REMOVE THE PREVIOUS 2 LINES AND REPLACE WITH . . .
DUM1 = 90.0
DUM2 = 0.0
CP DUM1 AND DUM2 WERE DEFINED ABOVE TO SEND TO XSACLE BECAUSE
CP THEY ARE MODIFIED IN XSACLE.
CRF THE FOLLOWING 6 LINES FOR XSACLE92. 17 APR 92.
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-7------------------------
C CHANGED THE CONSTANTS SENT TO XSACLE TO VARIABLES. THESE VALUES
C MAY BE CHANGED IN XSACLE. THIS CAUSED PROBLEMS WHEN XSACLE ENDED,
C SINCE THE CONSTANTS WERE NO LONGER THE VALUES THAT THEY WERE
C SUPPOSED TO BE.
C CALL XSACLE(0.45,0.7,VIS,MAERO,2,ISLT,NBR,
C 1 RD,DECOPER,XMEAN,XMODE,IWATER,
C 2 WAVRFN,RESPFN,
C 3 RANGE,90.0,0.0,CEILHT,THICK,AINVHT,
C 4 0.0,TEMP,0.0,RH,XSTRN,IERR,
C 5 Q,UM,EXT55I,QAVE)
DUM45 = 0.45
DUM7 = 0.7
IDUM2 = 2
DUM90 = 90.0
DUM0 = 0.0
CALL XSACLE(DUM45,DUM7,VIS,MAERO,IDUM2,ISLT,NBR,
1 RD,DECOPER,XMEAN,XMODE,IWATER,
2 WAVRFN,RESPFN,
3 RANGE,DUM90,DUM0,CEILHT,THICK,AINVHT,
4 DUM0,TEMP,DUM0,RH,XSTRN,IERR,
5 Q,UM,EXT55I,QAVE)
C-HSTX---SCENE SHADOWS---------------------------------------------
C
C REWIND IPHFUN
C IOIN = IPHFUN
GOTO 1012
CUV 1010 IERR = 1
1012 CONTINUE
CP1012 CLOSE(NPLOTU)
IF (IERR .EQ. 1) THEN
EFLAG = .TRUE.
RETURN
ENDIF
C
C IF (INTER) THEN
C
C***REV 1/91 ADDED SIGNIFICANT WEATHER ID TO INTERACTIVE PART AS
C NEEDED BY ILUMA MODULE

C-28
ELIMINATE SIGWX; CONDITIONS ARE COVERED BY OTHER INPUTS.

WRITE(IOOUT,*)'SELECT SIGNIFICANT WEATHER ID:

1 - SKY COVER < 50%
2 - SKY COVER = 50%
3 - SKY COVER > 50%
4 - BLOWING SNOW OR SAND
5 - FOG/HAZE (NON-OBSCURING)
6 - DRIZZLE
7 - RAIN
8 - SNOW/RAIN (NO SHOWER)
9 - RAIN/SNOW/HAIL SHOWER
10 - THUNDERSTORM

READ(IOIN,*)SIGWX
WRITE(IOOUT,*)SIGWX

IF (NINT(SIGWX) .LT. 0 .OR. NINT(SIGWX) .GT. 10) THEN
  WRITE(IOOUT,*)'THE SIGNIFICANT WEATHER ID MUST BE BETWEEN', +
  ' 1 AND 10'
  WRITE(IOOUT,*)'TRY AGAIN'
  WRITE(IOOUT,*)
GOTO 990
ENDIF

***REV 1/91 ADDED OBSERVED STATE OF GROUND TO INTERACTIVE PART AS
NEEDED BY ILUMA MODULE

WRITE(IOOUT,*)'SELECT OBSERVED STATE OF GROUND:

1 - DRY
2 - MOIST
3 - WET
4 - FROZEN
5 - ICE
6 - SNOW -- < 0.5
7 - SNOW -- > 0.5 & < ALL
8 - SNOW -- ALL
9 - LOOSE DRY SNOW/DUST/SAND -- >0.5 & <ALL
10 - LOOSE DRY SNOW/DUST/SAND -- ALL

READ(IOIN,*)OBSURF
WRITE(IOOUT,*)OBSURF
WRITE(IOOUT,*)

IF (NINT(OBSURF) .LT. 0 .OR. NINT(OBSURF) .GT. 10) THEN
  WRITE(IOOUT,*)'THE OBSERVED STATE OF GROUND MUST BE', +
  ' BETWEEN 1 AND 10'
  WRITE(IOOUT,*)'TRY AGAIN'
  WRITE(IOOUT,*)
GOTO 991
ENDIF

***REV 1/91 ADDED PRECIPITATION TYPE TO INTERACTIVE PART
AS NEEDED BY ILUMA MODULE

WRITE(IOOUT,*)'SELECT PRECIPITATION TYPE:

1 - NONE
2 - DRIZZLE
3 - RAIN
4 - SNOW
5 - HAIL

READ(IOIN,*)PRTYP
WRITE(IOOUT,*)PRTYP
WRITE(IOOUT,*)

IF (NINT(PRTYP) .LT. 0 .OR. NINT(PRTYP) .GT. 5) THEN
  WRITE(IOOUT,*)'THE PRECIPITATION TYPE BETWEEN', +
  ' BETWEEN 1 AND 5'
  WRITE(IOOUT,*)'TRY AGAIN'
  WRITE(IOOUT,*)

GOTO 992
ENDIF
ELSE
C***REV 1/91 ADDED SIGNIFICANT WEATHER ID, OBSERVED STATE OF GROUND, AND PRECIPITATION TYPE TO BATCH PORTION AS NEEDED BY ILUMA MODULE
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-9-----------------------------
C ELIMINATE SIGWX; CONDITIONS ARE COVERED BY OTHER INPUTS.
C SIGWX = RECVAL(7,2)
C CALL REALCK(10.0,1.0,SIGWX,'ILUM','2ND',1.0)
C-HSTX---SCENE SHADOWS----------------------------------------------
C
OBSURF = RECVAL(7,3)
PRTYP = RECVAL(7,4)
CALL REALCK(10.0,0.0,OBSURF,'ILUM','3RD',1.0)
CALL REALCK( 5.0,0.0,PRTYP,'ILUM','4TH',1.0)
ENDIF
C
C RD = RDTMP
TARGAZ = TARGAZ + 180.0
IHOUR = NINT(TIME)
ZTIME = TIME * 100.0
IMIN = NINT(ZTIME) - IHOUR * 100
XMIN = FLOAT(IMIN) / 60.0
TIME = FLOAT(IHOUR) + XMIN
C
C SUBROUTINE ZEN TO OBTAIN THE SOLAR ZENITH AND AZIMUTH ANGLES
C
CALL ZEN(DATE,TIME,ALAT,ALONG,COSZ,SUNAZ)
C***REV 1/91 CALL TO SOLUN COMMENTED OUT--REPLACED BY CALL TO ILUMA
C CALL SOLUN(DATE,TIME,ALAT,ALONG,COSZ,SUNAZ,COSM,LUNAZ)
C***REV 1/91 CALLS ILUMA FOR SOLAR AND LUNAR ELEVATIONS AND AZIMUTHS.
C ELEVATIONS CONVERTED TO ZENITHS
CALL ILMDAT ( IDEV )
ZENSUN = (90.0 - ALTS)/RAD
COSZ = COS(ZENSUN)
SUNAZ = AZIS + 180.0
C IF (SUNAZ .GT. 360.0) SUNAZ = SUNAZ - 360.0
C
ZENLUN = (90.0 - ALTMN)/RAD
COSM = COS(ZENLUN)
LUNAZ = AZIM + 180.0
C IF (LUNAZ .GT. 360.0) LUNAZ = LUNAZ - 360.0
C
THE PARAMETER FLIP IS USED TO TEST AGAINST BECAUSE THERE ARE PROBLEMS WITH MICROSOFT FORTRAN 5.0 IN TESTING AGAINST THIS PARAMETER. PSG 26 MARCH 1993. MICROSOFT FORTRAN THINKS THAT COSZ IS LESS THAN 0.0, GREATER THAN .972, LESS THAN 9.72, AND LESS THAN 1.0. CONFUSING N'EST PAS?????
FLIP = 0.0
IF (COSZ .LE. FLIP) THEN
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,*) ' CAUTION, SUN IS BELOW THE HORIZON.'
WRITE(IOOUT,*)' VALUES FOR THE SKY TO GROUND RATIO
MAY BE ERRONEOUS. LUNAR AZIMUTH
AND ZENITH ANGLE USED INSTEAD OF SOLAR AZIMUTH AND ZENITH ANGLE.'
C-30
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
  WRITE(NDIRTU,*)' CAUTION, SUN IS BELOW THE HORIZON."
  WRITE(NDIRTU,*)' VALUES FOR THE SKY TO GROUND RATIO'
  WRITE(NDIRTU,*)' MAY BE ERRONEOUS. LUNAR AZIMUTH'
  WRITE(NDIRTU,*)' AND ZENITH ANGLE USED INSTEAD OF'
  WRITE(NDIRTU,*)' SOLAR AZIMUTH AND ZENITH ANGLE."
ENDIF
COSZ = COSM
SUNAZ = LUNAZ
ENDIF
RHO = BKGREF
C
C USE STG TO FIND THE SKY TO GROUND RATIO
C
CPUSB STG2 WILL LOOK LIKE IT IS USED BEFORE SET TO CODE CHECKERS.
CPUSB STG2 IS A VALUE RETURNED BY FUNCTION STG. 18 NOV 92.
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-1-5------------------------
C CHANGE SKY-TO-GROUND RATIO VARIABLE TO AN ARRAY IN ORDER TO SAVE
C INFORMATION FOR PARTLY CLOUDY SITUATIONS. ADD LOOP AND OUTPUT
C STATEMENTS TO REPORT BOTH SKY-TO-GROUND RATIOS.
C
SOG=STG(XLAMB,COSZ,SUNAZ,VIS,TARGAZ,RHO,STG2)
SOG(1)=STG(XLAMB,COSZ,SUNAZ,VIS,TARGAZ,RHO,STG2)
SOG(2) = STG2
C
IF ( MLOOP .EQ. 1 ) THEN
  JSTOP = 1
ELSE
  JSTOP = 2
END IF
DO 500 JCLOUD=1,JSTOP
IF (SOG(JCLOUD) .LT. 0.5) THEN
  IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
    IF ((JCLOUD.EQ.2).OR.((MLOOP.EQ.1).AND.(NUMCLD.GT.0))) THEN
      WRITE(IOOUT,*)
      1 'SKY TO GROUND RATIO (IN CLOUD) IS LESS THAN 0.5'
    ELSE
      WRITE(IOOUT,*)
      1 'SKY TO GROUND RATIO (NO CLOUD) IS LESS THAN 0.5'
    ENDIF
  ELSE
    WRITE(IOOUT,*)
    1 'SKY TO GROUND RATIO (IN CLOUD) IS',SOG(JCLOUD)
  ENDIF
  WRITE(IOOUT,*)'THE VALUE HAS BEEN RESET TO 0.5'
END IF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
  IF ((JCLOUD.EQ.2).OR.((MLOOP.EQ.1).AND.(NUMCLD.GT.0))) THEN
    WRITE(NDIRTU,*)'SKY TO GROUND RATIO (IN CLOUD) IS LESS THAN 0.5'
  ELSE
    WRITE(NDIRTU,*)'SKY TO GROUND RATIO (NO CLOUD) IS LESS THAN 0.5'
  ENDIF
  WRITE(NDIRTU,*)'THE VALUE HAS BEEN RESET TO 0.5'
END IF
END IF
C-31
WRITE(IOOUT,*)'  
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
  WRITE(NDIRTU,*)'  
  IF ((JCLD.EQ.2).OR.((MLOOP.EQ.1).AND.(NUMCLD.GT.0))) THEN
    WRITE(NDIRTU,*)
    'SKY TO GROUND RATIO (IN CLOUD) IS ',SOG(JCLD)
  ELSE
    WRITE(NDIRTU,*)
    'SKY TO GROUND RATIO (NO CLOUD) IS ',SOG(JCLD)
  ENDIF
  WRITE(NDIRTU,*)'  
ENDIF
500 CONTINUE
C-HSTX---SCENE SHADOWS------------------------------------------
C
RETURN
END
C
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name:  STG                   Old Date:  1/14/93

File Name:  SLNCOF.FOR                New Date:  8/11/93

Implemented By:  Michael Oberlatz

Reason for Revision:  Corrects an error in naming of the CLOUD COMMON block and adds three cloud layers to the model.

Description of Revision:  Changed the name of the CLOUDS COMMON block to CLOUD to match the SGR function. Added the variables ZC2, ZC3, CF2, and CF3 to the CLOUD COMMON block to implement the three cloud layers.

Notes:  

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: STG Old Date: 8/11/93
File Name: SLNCOF.FOR New Date: 8/23/93

Implemented By: Dan DeBenedictis

Reason for Revision: Set atmospheric data for layers with and/or without clouds. Add loop structure for possible cloud situations.

Description of Revision: 1) Re-dimension the variables I0, I1, SURF, SURFO, F, PTHRD, and ISTAR to support partly cloudy situations. 2) Add the variables NUMCLD, MLOOP, CLDBTA, CLDG, LYRCLD, THK, and ITY to the COMMON block CLOUD. 3) Call GETCLD. 4) Loop through the number of cloud situations. (See code for more details.) Pass the loop counter to the subroutine DELTED; pass the counter and SZA to CONTST. Assign STG and STG2 as bracketing values, depending on MLOOP.

Notes: 

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: STG  Old Date: 8/23/93
File Name: SLNCOF.FOR  New Date: 8/27/93

Implemented By: Don Hamann

Reason for Revision: To maintain information on the probability of a cloud-free path through each cloud layer and on the probability of the target scene in direct light.

Description of Revision: Added variable PSCLD and array PCF(2) to CLOUD COMMON block to maintain probability information.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: STG          Old Date: 8/27/93

File Name: SLNCOF.FOR      New Date: 9/15/93

Implemented By: Melanie J. Gouveia

Reason for Revision: Fix the error that, for clear sky cases, gave 100.0 for the probability that the target is in cloud shadow.

Description of Revision: Initialize the probability that the target is in direct light to 100.0, before any clouds are accounted for.

Notes:

As appropriate, attach the following:
  1. Code listing with changes highlighted
  2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: STG  Old Date: 9/15/93
File Name: SLNCOF.FOR  New Date: 9/17/93

Implemented By: Don Hamann

Reason for Revision: Probability of cloud free path through layer requires initialization for clear sky situation. Added flag to keep track of the cloud situation.

Description of Revision: Added variable ICLDF to CLOUD COMMON block to maintain information on cloud situation. ICLDF is initialized to 1, PCF is initialized to 1, and PSCLD is initialized to 100.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: STG
File Name: SLNCOF.FOR
Old Date: 9/17/93
New Date: 10/20/93
Implemented By: Dan DeBenedictis

Reason for Revision: To find a weighted average of the extinction coefficient (BETAA) and the asymmetry parameter (G) for all atmospheric layers that contain some or all of a cloud layer.

Description of Revision: Call the routine LAYERS inside the DO loop controlled by MLOOP.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
FUNCTION STG(XLAMB,SZA,SUNAZ,VS,TARGAZ,ALBE)
COMMON/ATMOS/PRES(21),Z(21),RHO(21),NLEV,NLAY

C-HSTX---SCENE SHADOWS---ECR # HSTX-2-2-----------------------------
C RE-DIMENSIONED THE FOLLOWING VARIABLES TO SUPPORT PARTLY CLOUDY
C COMPUTATION:
C IO(20), I1(20)    ===> IO(20,4), I1(20,4)
C SURF(30), SURFO(30) ===> SURF(30,4), SURFO(30,2)
C F(20), PTHRD(30) ===> F(20,4), PTHRD(30,2)
C ISTAR(20,30) ===> ISTAR(20,30,2)
C
COMMON/RADIA/A(20),G(20),GP(20),BETAA(20),BETAR(20),
1    BETA(20),K(20),P(20),ALPHA(20),BET(20),
2    TAU(0:20),IO(20,4),I1(20,4),F(20,4),TF(20),FNOT, 
3    NANG,PF(180,20),ANG(180,20),I2(20),TAUP(0:20), 
4    ISTAR(20,30,2),PTHRD(30,2),SING(30),AP(20),PHE(30), 
5    ALBEDO,SURF(30,4),SURFO(30,2),TAUSTR,VIS,CAPTP(20), 
6    B1,B2

C-39
C-HSTX---SCENE SHADOWS-----------------------------------------------
C
COMMON/GEOM/THETO,PHIO,UNOT,XLAM,NOBS,PHI(30),BTA(30),NLOS,
1 XOB(30),YOB(30),ZOB(30),THETOB(30),PHIOB(30)
COMMON/TARGS/NTARG,XTARG(29),YTARG(29),ZTARG(29),TTARG(29)
1 ,XTARG(29),RTARG(29),CTARG(29)

C-HSTX---SCENE SHADOWS---ECR # HSTX-2-1-----------------------------------
C CHANGED THE CLOUDS COMMON BLOCK NAME TO CLOUD TO MATCH UP WITH
C THE SGR FUNCTION. THIS CORRECTS AN ERROR IN TARGAC AND ADDS THE
C THREE CLOUD LAYERS TO THE MODEL.
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-2-----------------------------------
C ADDED VARIABLES NUMCLD, MLOOP, CLDBTA(2), CLDG(2), LYRCLD(2), THK, ITY
C TO CLOUD COMMON BLOCK TO MAINTAIN INFORMATION ON MULTIPLE CLOUD LAYERS.
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-3-----------------------------------
C ADDED VARIABLE PSCLD AND ARRAY PCF(2) TO CLOUD COMMON BLOCK TO MAINTAIN
C INFORMATION ON PROBABILITY OF CLOUD-FREE PATH THROUGH EACH CLOUD LAYER
C AND THE PROBABILITY OF THE TARGET IN DIRECT LIGHT.
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-5-----------------------------------
C ADDED VARIABLE ICLDF TO CLOUD COMMON BLOCK TO MAINTAIN INFORMATION ABOUT
C CLOUD SITUATION.
CUV COMMON /CLOUDS/ZC1,CF1
C COMMON /CLOUDS/ZC1,ZC2,ZC3,CF1,CF2,CF3
C COMMON /CLOUD/ZC1,ZC2,ZC3,CF1,CF2,CF3
C COMMON /CLOUD/NUMCLD, MLOOP, CLDBTA(2),CLDG(2), LYRCLD(2),
C + THK1, THK2, THK3, ITY1, ITY2, ITY3,
C + ZC1,ZC2,ZC3,CF1,CF2,CF3
C COMMON /CLOUD/ICLDF, NUMCLD, MLOOP, CLDBTA(2),CLDG(2), LYRCLD(2),
C + THK1, THK2, THK3, ITY1, ITY2, ITY3,
C + ZC1,ZC2,ZC3,CF1,CF2,CF3,PCF(2),PSCLD
C-HSTX---SCENE SHADOWS-----------------------------------------------
C
COMMON/BCONST/DGTRD,LUOUT,LUIN
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
CRF COMMON/XSCL/ZZZ(999),BBETA(999),RELH(999),NNZPTS,SLNFLG
CRF REMOVE SLNFLG FROM XSCL COMMON BLOCK FOR UPGRADE TO XSCALE92
CRF 17 APR 92
COMMON/XSCL/ZZZ(999),BBETA(999),RELH(999),NNZPTS
REAL I0,I1,I2,ISTAR,K
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-6-----------------------------------
C ADD TEMPORARY VARIABLE TO STORE ATMOSPHERIC LAYER EXTINCTION, WITHOUT
C CLOUDS.
REAL BETAIAI(20)
C-HSTX---SCENE SHADOWS-----------------------------------------------------
C
CRF LOGICAL SLNFLG
CUV CHARACTER*1 ANSW
C
******************************************************************************
C VARIABLES
******************************************************************************
C
C PRES REAL VECTOR OF ATMOSPHERIC PRESSURES IN MB.
C Z REAL VECTOR OF ATMOSPHERIC LEVELS IN KM.
C A REAL VECTOR OF SINGLE SCATTERING ALBEDOS
C RHO REAL VECTOR OF ATMOSPHERIC DENSITY IN GM**M-3
C G REAL VECTOR OF ASYMMETRY PARAMETERS
C GP REAL VECTOR OF MODIFIED ASYMMETRY PARAMETERS
C BETAA REAL VECTOR OF AEROSOL EXTINCTION COEFS IN KM**-1
C BETAR REAL VECTOR OF RAYLEIGH SCATTERING COEFS IN KM**-1
C BETA REAL VECTOR OF TOTAL EXTINCTION COEFS IN KM**-1
C K REAL VECTOR OF PARAMETER USED IN DELTAED SOLUTION

C-40
REAL VECTOR OF PARAMETER USED IN DELTAED SOLUTION
C ALPH L REAL VECTOR OF PARAMETER USED IN DELTAED SOLUTION
C BET REAL VECTOR OF PARAMETER USED IN DELTAED SOLUTION
C TAU REAL VECTOR OF LAYER OPTICAL THICKNESSES
C NLEV INTEGER OF THE NUMBER OF ATMOSPHERIC LAYERS
C XLAM REAL OF THE WAVELENGTH USED IN MICRO METERS
C THETO REAL OF THE SOLAR ZENITH ANGLE IN DEGREES
C PHI0 REAL OF THE SOLAR AZIMUTH IN DEGREES WEST OF SOUTH
C UNOT REAL VALUE OF THE COSINE OF THE SOLAR ZENITH ANGLE
C PHNOT REAL VALUE OF THE SOLAR AZIMUTH ANGLE IN DEGREES
C THETOBREAL VALUE OF THE OBSERVATION ZENITH ANGLE IN DEGREES
C PHI0B REAL VALUE OF THE OBSERVATION AZIMUTH ANGLE IN DEGREES
C ZOB REAL VALUE OF THE OBSERVER HEIGHT IN KM.
C ALBEDO REAL VALUE OF THE SURFACE ALBEDO
C NANG INTEGER VALUE OF THE NUMBER OF ANGLES FOR EXACT PHASE FUNCTION
C PF MATRIX(NANG,NLAY) OF EXACT PHASE FUNCTION INFORMATION
C NOBS NUMBER OF OBSERVATION COORDINATES EQUAL TO UNITY IN THIS CASE
C XOB X COORDINATE OF THE OBSERVER
C YOB Y COORDINATE OF THE OBSERVER
C ZOB Z COORDINATE OF THE OBSERVER
C THETOB ZENITH ANGLE OF THE LINE OF SIGHT
C PHI0B AZIMUTH ANGLE OF THE LINE OF SIGHT

C***************************************************************

C NANG = 17
NLEV = 18
ALBEDO = ALBE
XLAM = XLAMB
THETO = ACOS(SZA) / 3.14159 * 180.0
PHI0 = SUNAZ / 3.14159 * 180.0
VIS = VS
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-6---------------------------------
C STORE INITIAL VALUES OF LAYER EXTINCTION, WITHOUT CLOUDS.
DO 51 I = 1,17
   BETAI(I) = BETAA(I)
51 CONTINUE
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-7---------------------------------
C *** SET EXTINCTION COEFFICIENTS ACCORDING TO THE FOLLOWING SCHEME
C 1ST -- DEFAULT COEFFICIENTS HAVE ALREADY BEEN ASSIGNED BY
C CALLING GETDATA IN THE SUBROUTINE SGR
C 2ND -- USE A SCALING HEIGHT ACCORDING TO ELTERMAN FOR THE
C LAYER BETWEEN 5 KM AND 2 KM
C 3RD -- USE XSCALE TO SET THE EXTINCTION COEFFICIENTS IN THE
C LOWEST TWO KILOMETERS -- VALUES OF EXTINCTION COEFFICIENT
C HAVE ALREADY BEEN ASSIGNED IN THE BBETA ARRAY FROM THE
C CALL TO XSCALE IN THE SUBROUTINE SGR
C 4TH -- PRESENCE OF A CLOUD SUPERCEDES ALL OTHER RULES IN
C THE LAYER ICLoud
C
BETAV = 3.912 / VIS
SCHT = - 5.0 / LOG( 0.005 / BETAV )
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-8---------------------------------
C DEFINE THE NUMBER OF CLOUD SITUATIONS AND LOOP THROUGH. IF THERE ARE
C CLOUDS, GET CLOUD OPTICAL PARAMETERS.
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-9---------------------------------
C INITIALIZE PROBABILITY THAT TARGET IS IN DIRECT LIGHT (NO CLOUDS).
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-10---------------------------------
C INITIALIZE PROBABILITY OF CLOUD-FREE PATH THROUGH LAYER. INITIALIZE
C THE CLOUD SITUATION FLAG (NO CLOUDS).
C MLOOP = 1
C One iteration of the loop for the conditions:

C-41
both layers clear
one layer overcast, one layer clear
both layers overcast

M_LOOP = 2

Two iterations of the loop for the conditions:
one partly cloudy layer, one clear layer
one partly cloudy layer, one overcast layer

First iteration for:
two clear layers
one overcast layer, one clear layer

Second iteration for:
one clear layer, one overcast layer
two overcast layers

M_LOOP = 4

Four iterations of the loop for the condition:
two partly cloudy layers

First iteration for:
two clear layers

Second iteration for:
high cloud layer only

Third iteration for:
low cloud layer only

Fourth iteration for:
both cloud layers

PSCLD = 100.
PCF(1) = 1.0
PCF(2) = 1.0

IF (NUMCLD.GT.0) THEN
CALL GETCLD
ELSE
M_LOOP = 1
ICLF = 1
ENDIF

DO 200 I = 1, M_LOOP

C-HSTX---SCENE SHADOWS------------------------------------------------------

C

SET THE LAYER VALUES FOR THE SINGLE PARTICLE SCATTERING ALBEDO AND THE ASYMMETRY PARAMETER. THE ASYMMETRY PARAMETER WILL BE ADJUSTED SO THAT LAYERS WITHOUT AEROSOL WILL HAVE PARAMETERS 0

DO 30 N =1,NLEV-1
A(N) = 0.99999
G(N) = 0.8045

C-HSTX---SCENE SHADOWS---ECR # HSTX-2-6------------------------------------

C RESET EXTINCTION VALUES FOR THE ATMOSPHERIC LAYERS, NOT INCLUDING CLOUDS.
BETAA(N) = BETAAI(N)

C-HSTX---SCENE SHADOWS------------------------------------------------------

C

30 CONTINUE

C

** SCALE THE EXTINCTION COEFFICIENT BETWEEN 2 AND 5 KM

C

DO 50 N = 1, NLAY
IF(Z(N+1) .LT. 5.0 .AND. Z(N+1) .GT. 2.0) THEN
BETAA(N) = BETAV * EXP(-Z(N) / 2.0 / SCHT)
ELSE IF(Z(N+1) .LE. 2.0) THEN
ZHOLD = (Z(N) + Z(N+1)) / 2.0
CJ CHANGE INTERPOLATION FROM BESSEL TO CUBIC SPLINE
CJ AUGUST 1992 J.FITZGERREL, P.GILLESPIE
BETAA(N) = BSLIF(ZHOLD,ZZZ,BBETA,NNZPTS)
BETAA(N) = CUBINT(ZHOLD,ZZZ,BBETA,NNZPTS)
ENDIF
C WRITE(\*,*) 'BETAA(\*,\*) = ',BETAA(N)
50 CONTINUE
*** SET THE ASYMMETRY FACTOR AND EXT. COEFF. FOR CLOUD IN THIS LAYER

GETCLD FINDS THE LEVEL OF CLOUD BASE AND THE CLOUD LAYERS; CODE HAS BEEN REMOVED FROM STG. ASYMMETRY FACTOR AND EXT. COEFF. FOR CLOUD LAYERS DEPEND ON ITERATION.

CALL LAYERS TO FIND WHICH ATMOSPHERIC LAYERS CONTAIN CLOUDS AND COMPUTE A WEIGHTED AVERAGE FOR THE ASYMMETRY FACTOR AND EXTINCTION COEFFICIENT. THIS ECR OVERRIDES ECR # HSTX 2-2 AT THIS JUNCTION OF THE CODE.

IF (NUMCLD.NE.0) THEN
  IF (MLOOP.EQ.1) THEN
    CALL LAYERS(ZC1,THK1,ZC2,THK2,CLDBTA,CLDG)
  ENDIF
  IF (MLOOP.EQ.2.AND.NUMCLD.EQ.1.AND.I.EQ.2) THEN
    CALL LAYERS(ZC1,THK1,0.0,0.0,CLDBTA,CLDG)
  ELSEIF (MLOOP.EQ.2.AND.NUMCLD.EQ.2.AND.I.EQ.1) THEN
    CALL LAYERS(ZC1,THK1,0.0,0.0,CLDBTA,CLDG)
  ELSEIF (MLOOP.EQ.2.AND.NUMCLD.EQ.2.AND.I.EQ.2) THEN
    CALL LAYERS(ZC1,THK1,0.0,0.0,CLDBTA,CLDG)
  ELSEIF (MLOOP.EQ.4) THEN
    IF (I.EQ.2) THEN
      CALL LAYERS(ZC1,THK1,0.0,0.0,CLDBTA,CLDG)
    ELSEIF (I.EQ.3) THEN
      CALL LAYERS(ZC1,THK1,0.0,0.0,CLDBTA,CLDG)
    ELSEIF (I.EQ.4) THEN
      CALL LAYERS(ZC1,THK1,0.0,0.0,CLDBTA,CLDG)
  ENDIF
ENDIF
ENDIF
C-HSTX---SCENE SHADOWS-----------------------------------------------

C CP WRITE(*,*)'G AND BETAA ARE ',G(1),BETAA(1)
C C *** SET THE VIEWING GEOMETRY
C
NOSB = 1
NTARG = 1
NTARG = 2
RTARG(1) = 0.10
XOB(1) = 0.0
YOB(1) = 0.0
CP ZOB(1) = 0.002
ZOB(1) = 0.0
XTARG(1) = 0.0
YTARG(1) = 100.0
CP ZTARG(1) = 10.0
ZTARG(1) = 0.0
CP THTARG(1) = 180.00000
THTARG(1) = 90.00000
AZTARG(1) = TARGAZ
CP THETOB(1) = 84.289407
THETOB(1) = 90.0
PHIOB(1) = TARGAZ
XOB(2) = 0.0
YOB(2) = 0.0
CP ZOB(2) = 0.002
ZOB(2) = 0.0
XTARG(2) = 0.0

C-43
YTARG(2) = 100.0  
ZTARG(2) = 0.0  
CP THITARG(2) = 180.0000  
THITARG(2) = 0.0000  
AZTARG(2) = TARGAZ  
CP THETOB(2) = 90.000000  
THETOB(2) = 90.000000  
PHIOB(2) = TARGAZ  
CP RTARG(2) = .33  
CP WRITE(*,*),'WHAT IS THE VALUE OF THE GROUND RTARG?'  
CP READ(*,*)RTARG(2)  
RTARG(2) = ALBEDO  
CP WRITE(*,*),'PHIOB IS ',PHIOB(1),PHIOB(2)  
C  
*** CALL MAIN ROUTINES  
C  
CALL INITI  
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-2----------------------------------  
C ADDED ARGUMENT I TO DELTED AND CONTST; ALSO SZA TO CONTST  
CALL DELTED (I)  
CP WRITE(*,*),'BEFORE CALL TO CONTRAST '  
CP CALL CONTRAST  
C  
C-HSTX---SCENE SHADOWS-----------------------------------------------  
C  
CP WRITE(*,*),'AFTER CALL TO CONTRAST '  
CP WRITE(*,*),'USE THE ACTUAL SURFACE ALBEDO?'  
CP WRITE(*,*)'FIRST THE UPGRADED SURFACE REFLECTANCE,...'  
CP READ(*,100)ANSW  
100 FORMAT(A)  
CP IF((ANSW.EQ.'N').OR.(ANSW.EQ.'N'))THEN  
CP C-HSTX---SCENE SHADOWS---ECR # HSTX-2-2-------------------------  
C COMPUTE SKY-TO-GROUND RATIO FOR TARGET IN AND OUT OF CLOUD SHADOW  
IF (MLOOP.EQ.1) THEN  
STG = PTHRD(I,1) / SURFO(2,1)  
ELSE  
STG = PTHRD(1,1) / SURFO(2,1)  
STG2 = PTHRD(1,2) / SURFO(2,2)  
ENDIF  
C-HSTX---SCENE SHADOWS-----------------------------------------------  
C  
CP WRITE(*,*),'THE SKY TO GROUND RATIO IS ',STG  
CP WRITE(*,*)'PTHRD AND SURFO IN STG ',PTHRD(1,1),SURFO(1,1)  
CP STG = PTHRD(1) / SURFO(1)  
CP WRITE(*,*),'SURF AFTER THE SGR CALC IS ',SURFO(1,1)  
CP ELSE  
CP WRITE(*,*),'OLD SGR CALC, NLAY, TF ',NLAY,TF(NLAY)  
CP WRITE(*,*),'ALBEDO & PTHRD ',ALBEDO,PTHRD(1,1)  
C  
C-HSTX---SCENE SHADOWS---ECR # HSTX-2-2-------------------------  
C COMMENT OUT PREVIOUS VALUE OF STG2  
C STG2 = PTHRD(1) / (TF(NLAY) * ALBEDO / 3.1416)  
C-HSTX---SCENE SHADOWS-----------------------------------------------  
C  
CP WRITE(*,*),'TF AND ALBEDO AT THE SGR CALC ARE ',TF(NLAY),ALBEDO  
CP END IF  
RETURN  
END  
C  
<END OF UNIT: STG>
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: ELIMIN          Old Date: 2/25/93
File Name: CONTRAST.FOR      New Date: 8/11/93

Implemented By: Dan DeBenedictis

Reason for Revision: An error in TARGAC was discovered: the COMMON block AGAUSS should be named GAUSS. GAUSS is used in the routine DELTED, which fills in the arrays stored in GAUSS.

Description of Revision: The COMMON block AGAUSS is renamed GAUSS.

Notes: ________________________________

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-45
SUBROUTINE ELIMIN(N,ERR)

C-HSTX---SCENE SHADOWS---ECR # HSTX-3-1-----------------------------
C CHANGED THE AGAUS COMMON BLOCK NAME TO GAUSS TO MATCH UP WITH
C THE DELTED ROUTINE. THIS CORRECTS AN ERROR IN TARGAC.
C
COMMON/AGAUS/ AA(40,40),BB(40),X(40)
COMMON/GAUSS/ AA(40,40),BB(40),X(40)

C-HSTX---SCENE SHADOWS-----------------------------------------------
C
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,IRELH,
1 KSTOR,NPLOTU,STDERR
COMMON /CONST/PI,P12,PIRAD,TWOP1,TORRMB,CDEGK
COMMON /IOFILE/IOFILE
INTEGER IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT, IRELH,
   KSTOR,NPLOTU,STDERR,IOFILE
DIMENSION A(40,41),Y(40),ID(40)

NN = N + 1
DO 200 I = 1, N
   A(I,NN) = BB(I)
DO 200 J = 1, N
   A(I,J) = AA(I,J)

K = 1
1 CONTINUE
DO 21 I = 1, N
   ID(I) = I
2 CONTINUE
   KK = K + 1
   IS = K
   IT = K
   B = ABS(A(K,K))
   DO 3 I = K, N
      DO 3 J = K, N
         IF(ABS(A(I,J)) - B) 3,3,31
      31 IS = I
         IT = J
         B = ABS(A(I,J))
   3 CONTINUE
   IF(IS - K) 4,4,41
41 DO 42 J = K, NN
      C = A(IS,J)
      A(IS,J) = A(K,J)
42 A(K,J) = C
4 CONTINUE
   IF(IT - K) 5,5,51
51 IC = ID(K)
      ID(K) = ID(IT)
      ID(IT) = IC
   DO 52 I = 1, N
      C = A(I,IT)
      A(I,IT) = A(I,K)
52 A(I,K) = C
5 CONTINUE
   IF(A(K,K))6,102,6
6 CONTINUE
   DO 7 J = KK,NN
      A(K,J) = A(K,J)/A(K,K)
   DO 7 I = KK, N
      W = A(I,K)*A(K,J)
      A(I,J) = A(I,J) - W
   IF(ABS(A(I,J)) - .0001 * ABS(W))71,7,7
71 A(I,J) = 0.
7 CONTINUE
   K = KK
   IF(K - N)2,81,102
81 IF(A(N,N))\$8,102,8
8 CONTINUE
Y(N) = A(N,NN)/A(N,N)
NM = N - 1
DO 9 I = 1,NM
K = N - I
KK = K + 1
9 CONTINUE
Y(K) = A(K,NN)
Y(K) = Y(K) -A(K,J) * Y(J)
DO 9 J = KK,N
Y(K) = Y(K) -A(K,J) * Y(J)
CONTINUE
DO 10 I = 1, N
DO 10 J = 1,N
101 X(I) = Y(J)
CONTINUE
RETURN
102 IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,1000)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,1000)
ENDIF
RETURN
1000 FORMAT("")
C THE COMMENT BELOW HAS TO DO WITH THE DELTA EDDINGTON SOLN
1001 FORMAT('NO UNIQUE SOLUTION')
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: ILUMA               Old Date: 3/23/93
File Name: ILUMA.FOR              New Date: 8/11/93

Implemented By: Dan DeBenedictis

Reason for Revision: Cloud fraction (FR1, FR2, & FR3) and cloud type (ILR1, ILR2, & ILR3) for each layer are now user inputs.

Description of Revision: Removed all initialization and setting of the variables FR1, FR2, FR3, ILR1, ILR2, and ILR3. Also ID2 is set to 1 for clear or 2 for cloudy. This eliminates the algorithm to set ID2 to a value from 1 to 8 based on cloud fraction and ceiling height. ILR1, ILR2, and ILR3 are input with the Thermal model cloud index. ILUMA converts these numbers to the ILUMA cloud index.

Notes: 

As appropriate, attach the following:
1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: ILUMA   Old Date: 8/11/93

File Name: ILUMA.FOR   New Date: 10/18/93

Implemented By: Dan DeBenedictis

Reason for Revision: Remove references to and use of the variable SIGWX. All cloud information (type, fraction, and height) and fog are input elsewhere.

Description of Revision: Commented out any code that referenced SIGWX.

Notes:

As appropriate, attach the following:
1. Code listing with changes highlighted
2. Test records
SUBROUTINE ILUMA ( IPRFLG, IERR )

This subroutine reads all input data except flag IPRFLG from
the standard EOSAEL input unit IOIN. The calling program should
set IPRFLG = 0 to suppress normal outputs to LUN IOOUT, or to
non-zero value to activate normal output listings.

This program computes total solar/lunar illumination (watt/m**2)
received at the ground as a function of geographical location,
date, Greenwich mean time GMT), and meteorological conditions.
The albedo option allows one to account for realistic albedo
conditions of the ground surface. It utilizes a three layer model
atmosphere characterized by the state of cloudiness in each layer.

This code is a combination of several earlier codes:

Ralph Shapiro: "Solar radiative flux calculations from standard
surface meteorological observations," Air Force Geophysics
Laboratory report AFGL-TR-82-0039, March 1, 1982.

L.D. Duncan/D.P. Sauter "Natural illumination under realistic
weather conditions", Atmospheric Sciences Laboratory Report

Report PHL 1982-13 (the Netherlands) by Ir. A.C. Van Bochove,
contains the algorithms used for finding the positions of the
sun and moon, and the overall lunar/solar illumination constants.

This version adapted by A. Miller (NMSU) from L.D. Duncan's (ASL)
program 'illum.pro' to conform (more or less) to D.P. Sauter's
(ASL) Turbo-Pascal codes. (April 1987) updated by D. Sauter
in February, 1990 to incorporate several program revisions.

This driver accepts inputs in the EOSAEL card-order, although
not all the cards which are now allowed are actually used.

Conversion from old form to above will be via:

IH = ILR1 (high clouds, IH = 1, 2, 3)
IM = ILR2 + 3 (middle clouds, IM = 4, 5)
IL = ILR3 + 5 (low clouds, IL = 6, 7, 8, 9)

Input:
Cards may be inserted in any order with the exception of
sentinel cards ('GO' or 'DONE' - which signify that execution is to
begin). All input data are entered under format (A4,6X,7E10.4).

***

Note that all data must be entered with properly placed decimal
points even if they seem as though they should be integers.
(even the time on 24-HR type clock)

***

Mnemonic      Input variable description
DATE  FMONTH, DAY, YEAR, GTIME
FMONTH IS MONTH (1. - 12.)
DAY IS DAY OF MONTH (1. - 31.)
YEAR IS A.D. YEAR (1977.-1999. FOR MOON CASES)
GTIME IS GREENWICH MEAN TIME IN FORM HHMM. - WHERE THE
DECIMAL POINTS ARE MANDATORY

************************************************************
** THERE ARE NO DEFAULTS FOR THESE EXCEPT 1200. FOR TIME **
** PROGRAM WILL ABORT ON ILLEGAL MONTH,DAY OR YEAR **
************************************************************

GEOS  PARAMETERS DESCRIBING THE LOCATION AND TIME OF
THE OBSERVATION POINT
SLAT - LOCAL LATITUDE
SLON - LOCAL LONGITUDE (+EAST, -WEST)
(DEFAULT VALUES ARE 0.0, AND 0.0)

CLDS  STATE OF CLOUDINESS IN EACH OF 3 LAYERS OF THE ATMOSPHERE
ILR1=1./2./3. CLEAR/THIN CI-CS/THICK CI-CS
ILR2=1./2. CLEAR/AS-AC
ILR3=1./2./3./4. CLEAR/FOG-SMOKE/SC-ST/CU-CB
** REMEMBER THAT DECIMAL POINTS ARE NECESSARY **
(DEFAULT VALUES FOR THIS CARD ARE: 1., 1., 1.)

ALBD  RG - SURFACE ALBEDO
(DEFAULT VALUE OF RG IS 0.2)

CLFR  FRI, FR2, FR2 CLOUD FRACTIONS (0.0 - 1.0) IN HIGH, MIDDLE
AND LOW LEVELS RESPECTIVELY.
(DEFAULT VALUES ARE 0.0 FOR ALL THREE.)

WEAX  SIGWX, OBSURF, CEILHT, PRTYP, FRC

STATE OF WEATHER AND SURFACE CONDITIONS:
(USED IF INCOMING DATA IS FOR SURFACE BASED MET.DATA.
IF THIS CARD IS PRESENT, DATA ON THE 'CLDS', 'CLFR' AND
'ALBD' CARDS WILL BE IGNORED.)

SIGWX - SIGNIFICANT WEATHER ID, FROM LIST:
1 - SKY COVER < 50 %  2 - SKY COVER = 50 %
3 - SKY COVER > 50 %  4 - BLOWING SNOW OR SAN/
5 - FOG/HAZE (NON-OBSCURING)  6 - DRIZZLE
7 - RAIN  8 - SNOW/RAIN (NO SHOWER)
9 - RAIN/SNOW/HAIL SHOWER  10 - THUNDERSTORM

OBSURF IS THE OBSERVED STATE OF THE GROUND FROM THE FOLLOWING
CHOICES:
1 - DRY  2 - MOIST  3 - WET  4 - FROZEN
5 - ICE  6 - SNOW < 0.5  7 - 0.5 < SNOW < ALL'
8 - SNOW (ALL)  9 - 0.5 < (LOOSE DRY SNOW/DUST/SAND) < ALL
10 - ALL = (LOOSE DRY SNOW/DUST/SAND)

CEILHT IS THE OBSERVED CEILING HEIGHT IN KM

PRTYP IS PRECIPITATION TYPE, FROM THE LIST:
1 - NONE  2 - DRIZZLE  3 - RAIN
4 - SNOW  5 - HAIL

FRC IS THE FRACTION OF SKY COVERED BY CLOUDS
*GO* is instruction to begin execution (which loops back for additional data and 'GO' cards until the 'DONE' card (see below) is encountered. The 'GO' card separates discrete sets of parameters — such as different times of day, etc.

*DONE* card informs the program that no more sets of data are to be expected. When the 'DONE' card is detected, a flag is set to bypass the instructions to loop back to the beginning for additional data upon completion of the computations which use the data which were immediately ahead of the 'DONE' card.

***> If only one 'PASS' is to be done, use only the 'DONE' card and do not use a 'GO' card.

***> Sample card setup for two sets of data (or two passes) might contain the following cards (from which numerical data have been omitted):

- DATE
- GEOS
- ALBD
- WEAX
- GO

DATE (different from the first one) — see next note also done

***> Any parameter which is not redefined in data sets which are subsequent to the first one (the one before the first 'GO' card) normally retain their most recent value. This may lead to incorrect behavior if 'SINGLE-LAYER' and 'THREE-LAYER' forms of input data are used within the same data file (deck) since the user may not know what values have been assigned by the Sauter-Duncan algorithm to some parameters.

***> Note also that the Sauter-Duncan procedures include a term for galactic light. The normal 'THREE-LAYER' procedure (Shapiro) does not include that term, so ostensibly identical 'SINGLE-LAYER' and 'THREE-LAYER' data sets may not in fact yield the same final results for the illuminances.

**Outputs:**

- **ELUMII** = solar + lunar + galactic illumination in W/m² using 93 lumen per watt conversion factor
- **SUNLIT** = solar horizontal illuminance in lumen/sq-meter
- **MOOLIT** = lunar horizontal illuminance in lumen/sq-meter
- **ALT** = solar elevation in degrees (above local horizontal)
- **AZI** = solar azimuth in degrees clockwise from north
- **ALT** = lunar elevation angle in degrees
- **AZIM** = lunar azimuth angle in degrees
- **DPHASE** = phase of moon (-180. = new moon, +180. = dark)

------ Additional outputs which are not printed -------

- **TCLSUN** = fractional transmittance of sunlight for clear sky
- **TCLLUN** = fractional sky transmittance of moon light

The following four quantities are very approximate and are obtained using a "seat of the pants" one layer atmosphere delta-EDDINGTON
C METHOD WITH SAME TRANSMITTANCE AS THE THREE LAYER COMPOSITE DERIVED

C USING THE SHAPIRO EQUATIONS TOGETHER WITH:

C SS ALBEDO (CLEAR) = 0.98, ASSYM G = 0.0
C SS ALBEDO (CLDY) = 0.98, ASSYM G = .85
C (SEE SUBROUTINE DIRDIF)

C RCLSUN = DIRECT/DIFFUSE FLUX RATIO FOR CLEAR SKY (SUN ANGLES)
C RCDSUN = DIRECT/DIFFUSE FLUX RATIO FOR CLOUDY SKY (SUN)
C RCLLUN = CLEAR RATIO FOR CASE OF MOON POSITION ANGLES
C RCDLUN = CLDY RATIO FOR MOON CASE

C EOSAEL STANDARD COMMON BLOCKS AND DEFINITIONS

C COMMON /IODFILE/IOFILE
C COMMON /CONST/ PI, PI2, PIRAD, TWOP, TORRM, CDEGK
C COMMON /CLYMAT/ TEMP, PRESS, RH, AH, DP, VIS, CLDATM, CLDHYT,
+ FOGPRB, WNDVEL, WNDDIR, IPASCT
C
C CHARACTER FMTSO0*234, FMTSO1*120, FMTSO2*122, FMTSO3*73,
+ FMTSO4*122, FMTSO5*111, FMTSO6*122, FMTSO7*230,
+ FMTSO11*15, FMTSO12*15, FMTSO13*152
C
C COMMON /FMTSBD/ FMTSO0, FMTSO1, FMTSO2, FMTSO3, FMTSO4, FMTSO5,
+ FMTSO6, FMTSO7, FMTSO11, FMTSO12, FMTSO13
C
C INTEGER IOIN, IOOUT, IPHFUN, LONUT, NDIRTU, NCLIMT, IRELH,
+ KSTOR, NPLOTU, STDERR
C
C COMMON /IOUNIT/ IOIN, IOOUT, IPHFUN, LONUT, NDIRTU, NCLIMT,
+ IRELH, KSTOR, NPLOTU, STDERR
C
C COMMON /GEOMET/ PTS(15), IGEOSS
C COMMON /ILUMCM/ ALTS, AZIS, ALTM, AZIM, DPHASE, ELUMI, SUNLIT,
+ MOOLIT, TCLSUN, TCLLUN, RCLSUN, RCDSUN, RCLLUN,
+ RCDLUN

C***REV 2/91
C COMMON /ILDATA/ FMONTH, DAY, YEAR, ST, SLAT, SLON, ILR1, ILR2,
+ ILR3, RG, FR1, FR2, FR3, SIGWX, OBSURF, CEILHT,
+ PRITY, FRC, ITARG
C
C INTEGER DATE, TIME, IODEFILE
C REAL MOOLIT
C
C DIMENSION DAT(7)
C CHARACTER CARD*80, ALPHA1*4, ALPHA2*4, RNAME(8)*4
C CHARACTER*5 IH1(3), IH2(2), IH3(4)
C CHARACTER REVNO*16, REVDAT*9
C CHARACTER*32 SCCS
C LOGICAL WARNING
C SAVE WARNING
C EXTERNAL ILUMBD

C DATA WARNING / .TRUE. /
C DATA RNAME / 'DATE', 'GEOS', 'CLDS', 'ALBD', 'CLFR', 'WEAX',
+ 'GO', 'DONE' /
C DATA IH1 / 'CLEAR', 'CI-CS', 'CI-CS' /
C DATA IH2 / 'AS-AC', 'AS-AC' /
C DATA IH3 / 'CLEAR', 'F-K', 'SC-ST', 'CU-CB' /
C
C DATA RAD / 57.29578 /
C DATA SCCS / 'O(#) iluma.f 2.2 02/27/90' /
C DATA REVNO / 'EOSAEL87 REV 2.2' /
C DATA REVDAT / '02/27/90' /
C
C 30 FORMAT(1HO,///' *** UNRECOGNIZABLE INPUT DATA CARD DETECTED IN ROU
1TINE ILUMA *** '1
C10X,'FIRST CONTENTS WERE : ',1X,2A4//)
40 FORMAT(1H1)
50 FORMAT(5X,25(1H*),' INPUTS ',25(1H*)/
2 /5X,'FOR THE DATE OF ',12,1H/,I2,1H/,I2)
60 FORMAT (5X,'DAYTIME CONDITIONS'//74(1H')//)
70 FORMAT (5X,'NIGHTTIME CONDITIONS'//74(1H')//)
80 FORMAT (5X,'GREENWICH TIME '//74(IH')//)
1 /SX,'LOCAL LATITUDE = ','F7.2,' DEG ','
2 /SX,'LOCAL LONGITUDE = ','F7.2,' DEG')
90 FORMAT (5X,'PHASE ANGLE = ','F7.2,' DEG',
1 / 5X, 'MOON' S ZENITH ANGLE = ','F7.2,' DEG')
100 FORMAT (5X,'FIRST LAYER: FRACTIONAL COVER: ','F5.2,' TYPE: ','A5)
110 FORMAT (57X,'- THIN CLOUD')
120 FORMAT (57X,'- THICK CLOUD')
130 FORMAT (5X,'SECOND LAYER: FRACTIONAL COVER: ','F5.2,' TYPE: ','A5)
140 FORMAT (5X,'THIRD LAYER: FRACTIONAL COVER: ','F5.2,' TYPE: ','A5)
150 FORMAT (SX,'SURFACE ALBEDO ',F6.3)
160 FORMAT (////5X,25(1H*),' OUTPUTS ',25(1H*)/
C 5X,'SOLAR ELEVATION ANGLE = ','F7.2,' DEGREES.'/
C 5X,'SOLAR AZIMUTH ANGLE = ','F7.2,' DEGREES.'/
C 5X,'LUNAR ELEVATION ANGLE = ','F7.2,' DEGREES.'/
C 5X,'LUNAR AZIMUTH ANGLE = ','F7.2,' DEGREES.'/
C 5X,'LUNAR PHASE ANGLE = ','F7.2,' DEGREES.'/
C 5X,'SOLAR ILLUMINANCE = ','F11.2,' LUMENS/SQ-METER.'/
C 5X,'LUNAR ILLUMINANCE = ','F11.4,' LUMENS/SQ-METER.'/
C 5X,'NET ILLUMINANCE = ','F11.4,' LUMENS/SQ-METER.'/
C 5X,'INTEGRATED TOTAL FLUX = ','F9.2,' WATTS/SQ-METER.')
C 170 FORMAT (' ILUMA: MONTH, MM =',I12,
1 /' MM SHOULD BE GREATER THAN 0 AND SMALLER THAN 13')
180 FORMAT (' ILUMA: DAY, ID =',I12,
1 /' ID SHOULD BE GREATER THAN 0 AND SMALLER THAN 32')
190 FORMAT(' THIS IS A SPARE FORMAT FOR FUTURE USE IF NEEDED')
C 200 FORMAT (' ILUMA: LUNAR ZENITH ANGLE, ZN =','E12.4,
C 1 /' ZN SHOULD BE SMALLER THAN 85 DEG')
210 FORMAT(//// ILUMA DIAGNOSTIC: FATAL ERROR NO. ',I3,' OCCURRED IN SUBR
IOUTINE ILLUM.')/
220 FORMAT(//// ILUMA: PREMATURE END OF INPUT DATA FILE ENCOUNTERED ON
I1UN ',I3//' PROBABLE CAUSE: MISSING 'DONE' CARD//)
C IF (WARNING) THEN
C PRINT MAIN HEADER
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,FMTS10) ' I L U M A '
1 , ' I L U M A '
2 , ' I L U M A '
3 , ' I L U M A '
4 , ' I L U M A '

WARNING = .FALSE.
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTUFMTS10) ' I L U M A '
1 , ' I L U M A '
2 , ' I L U M A '
3 , ' I L U M A '
4 , ' I L U M A '

WARNING = .FALSE.
ENDIF
ENDIF
C DEFAU = 1.E-30
MAXNAM = 8
IPASS = 0
IEND = 0

230 CONTINUE
IPASS = IPASS + 1
IDWX = 0
ID2 = 0
IERR = 0

C***REV 2/91
IF (ITARG.EQ.1 .OR. ITARG.EQ.2) GOTO 325
C********
240 CONTINUE
C
READ IN THE DATA
C
READ(IOIN, '(A)', END=440) CARD
READ(CARD, FMTS11, ERR=242) ALPHA1, ALPHA2, (DAT(L), L = 1, 7)
GOTO 243

242 IF(IOFILE.EQ.0 .OR. IOFILE.EQ.2) THEN
WRITE(IOOUT, '(A)') 'A NON EOSAEL FORMAT INPUT CARD ENCOUNTERED'
WRITE(IOOUT, '(A)') CARD
ENDIF
IF(IOFILE.EQ.1 .OR. IOFILE.EQ.2) THEN
WRITE(NDIRTU, '(A)') 'A NON EOSAEL FORMAT INPUT CARD ENCOUNTERED'
WRITE(NDIRTU, '(A)') CARD
ENDIF
STOP
243 CONTINUE

C
MAKE SURE THAT THE IDENTIFIER CHARACTERS ARE ALL UPPERCASE
C
CALL UCA (ALPHA1)
C
CHECK FOR CARD TYPES
C
DO 250 KK = 1, MAXNAM
IF (ALPHA1 .NE. R NAMES(KK)) GOTO 250
GOTO (270, 280, 290, 300, 310, 320, 340, 330), KK
250 CONTINUE

C
UNKNOWN CARD TYPE
C
IF(IOFILE.EQ.0 .OR. IOFILE.EQ.2) THEN
WRITE(IOOUT, 30) ALPHA1, ALPHA2
ENDIF
IF(IOFILE.EQ.1 .OR. IOFILE.EQ.2) THEN
WRITE(NDIRTU, 30) ALPHA1, ALPHA2
ENDIF
IERR = 1
GOTO 450

C
DATE CARD
C
270 CONTINUE
FMONTH = DAT(1)
DAY = DAT(2)
YEAR = DAT(3)
MM = NINT(FMONTH)
ID = NINT(DAY)
IY = NINT(YEAR-1900.0)
DATE = 10000*IY+100*MM+ID
ST = DAT(4)

C CONVERT TO INTERNAL FORM = HH.MM

NHRS = AINT(ST/100.0)
NMIN = AINT(ST-100*NHRS)
TIME = NHRS*100+NMIN
ST = NHRS+FLOAT(NMIN)/100.0
GOTO 240

C.....OBSERVATION POINT PARAMETERS CARD.....

280 SLAT = DAT(1)
SLON = DAT(2)
GOTO 240

C.....STATE OF CLOUDINESS CARD.....

290 ILR1 = IFIX(DAT(1))
ILR2 = IFIX(DAT(2))
ILR3 = IFIX(DAT(3))
GOTO 240

C.....SURFACE ALBEDO CARD.....

300 RG = DAT(1)
GOTO 240

C.....CLOUD FRACTIONS CARD

310 CONTINUE
FR1 = DAT(1)
FR2 = DAT(2)
FR3 = DAT(3)
GOTO 240

C SURFACE MET DATA CARD

320 CONTINUE
C-HSTX---SCENE SHADOWS---ECR # HSTX-4-2---------------------------
C REMOVE SIGWX; THIS VARIABLE IS NO LONGER USED.
C SIGWX = DAT(1)
C-HSTX---SCENE SHADOWS---------------------------------------------
C OBSURF = DAT(2)
CEILHT = DAT(3)
PRTP = DAT(4)
FRC = DAT(5)
IDWX = 1
C RESET ANY DATA LEFT OVER FROM 3-LAYER INPUT MODE
C-HSTX---SCENE SHADOWS---ECR # HSTX-4-1--------------------------------
C REMOVE INITIALIZATION OF FR1, FR2, AND FR3.
C-HSTX---SCENE SHADOWS---------------------------------------------
C GOTO 240
C
C***REV 2/91

325 MM = NINT(FMONTH)
ID = NINT(DAY)
IY = NINT(YEAR-1900.)
DATE = 10000*IY+100*MM+ID

C-56
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-4-1---------------------------
C REMOVE INITIALIZATION OF FR1, FR2, AND FR3.
C-HSTX---SCENE SHADOWS---------------------------------------------
C
ELSE
IDWX = 0
ENDIF
C****
C
....START EXECUTION.....
C
330 CONTINUE
C
DONE CARD COMES HERE
C
IEND = 1
340 CONTINUE
C
SET DEFAULT VALUES AND CHECK INPUT CONSISTENCY
C
SOME OF THESE MAY NEED TO BE CHANGED TO REFLECT USE OF DIFFERENT
SUN / MOON LOCATING Routines
C
IF (MM .LT. 1 .OR. MM .GT. 12) GOTO 390
IF (ID .LT. 1 .OR. ID .GT. 31) GOTO 400
IF (SLAT .LT. DEFALT) SLAT = 0.0
IF (SLON .LT. -180.0 .OR. SLON .GT. 180.0) SLON = 0.0
IF (ST .LT. 0.0 .OR. ST .GT. 24.0) ST = 12.0
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-4-1---------------------------
C REMOVE CONSISTENCY CHECK OF ILR1, ILR2, AND ILR3.
C-HSTX---SCENE SHADOWS---------------------------------------------
C
IF (RG .LT. 0.0 .OR. RG .GT. 1.0) RG = 0.0
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-4-1---------------------------
C REMOVE CONSISTENCY CHECK OF FR1, FR2, AND FR3.
C-HSTX---SCENE SHADOWS---------------------------------------------
C
IF ANY WEAx DATA ARE OUT OF ALLOWED RANGE:
C
IF (IDWX .NE. 0) THEN
C
IF (OBSURF .LT. 1.0 .OR. OBSURF .GT. 10.0) OBSURF = 1.0
IF (CEILHT .LT. 0.0) CEILHT = 9999.9
IF (PRTYP .LT. 1.0 .OR. PRTYP .GT. 5.0) PRTYP = 1.0
IF (FRC .LT. 0.0 .OR. FRC .GT. 1.0) FRC = 0.0
ENDIF
C
OTHER INITIALIZATIONS
C
GLOWRA IS USED ONLY FOR SAUTER-DUNCAN NIGHT-SKY ILLUMINATION

C-57
ADJUSTMENT

SKIP THE NEXT STEPS IS 1-LAYER INPUT DATA FLAG IS NOT SET BY CALLE

IF (IDWX .EQ. 0) GOTO 350

GLOWRA = 1.0

C-HSTX---SCENE SHADOWS---ECR # HSTX-4-2--------------------------------
C IDWX AND SIGWX NO LONGER USED; ILR3 FOR FOG IS SET IN SGR.
C IDWX = SIGWX
C-HSTX---SCENE SHADOWS---------------------------------------------

IDSURF = OBSURF
IDPR = PRTyp

SET GROUND ALBEDO BASED ON SURFACE CONDITIONS

IF (IDSURF .GE. 1 .AND. IDSURF .LE. 4) RG = 0.25
IF (IDSURF .EQ. 5) RG = 0.6
IF (IDSURF .EQ. 6) RG = 0.43
IF (IDSURF .GE. 7 .AND. IDSURF .LE. 10) RG = 0.6

C-HSTX---SCENE SHADOWS---ECR # HSTX-4-1--------------------------------
C REMOVE LOGIC TO DEFINE FR1, FR2, FR3, ILR1, ILR2, AND ILR3. THEY
C ARE NOW USER INPUTS. ILR1, ILR2, AND ILR3 ARE ENTERED WITH THE
C CLOUD INDICES USED BY THE THERMAL MODEL; THEY MUST BE CHANGED TO
C MATCH WHAT ILUMA IS ACCUSTOMED TO. ALSO, CHANGE THE LOGIC TO DEFINE
C ID2 FROM VALUES OF 1 TO 8 TO VALUES OF 1 FOR CLEAR AND 2 FOR CLOUDY.

ID2 = 1
IF (FR1.GE.0.1.OR.FR2.GE.0.1.OR.FR3.GE.0.1) ID2 = 2
IF (IDPR .GT. 1 .AND. IDPR .LT. 6) ID2 = 2

IF (ILR1.EQ.0) THEN
ILR1 = 1
ELSEIF (ILR1.EQ.1) THEN
ILR1 = 2
ELSEIF (ILR1.EQ.2) THEN
ILR1 = 3
ENDIF

IF (ILR2.EQ.0) THEN
ILR2 = 1
ELSEIF (ILR2.EQ.3) THEN
ILR2 = 2
ENDIF

IF (ILR3.EQ.0) THEN
ILR3 = 1
ELSEIF (ILR3.EQ.4) THEN
ILR3 = 3
ELSEIF (ILR3.EQ.5) THEN
ILR3 = 4
ENDIF

C-HSTX---SCENE SHADOWS---------------------------------------------
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-4-2--------------------------------
C IDWX NO LONGER USED; ILR3 FOR FOG IS SET IN SGR.
C IF (FR3 .EQ. 0.1 .AND. IDWX .EQ. 5) ILR3 = 2
C-HSTX---SCENE SHADOWS---------------------------------------------
C
350 CONTINUE

C
IF (IDWX .EQ. 0) GLOWRA = 0.0

TRANSLATIONS
IH = ILR1
IM = ILR2+3
IL = ILR3+5
MM = NINT(FMONTH)
NHRG = NRHS

IF NHRG (GMT HOUR) IS NEGATIVE, THEN THE GREENWICH DATE IS REALLY THE PREVIOUS CALENDAR DAY

IF (NHRG .LT. 0) THEN
NHRG = NHRG+24
DATE = DATE-1
ENDIF

IF (NHRG .GT. 24) THEN
NHRG = NHRG-24
DATE = DATE+1
ENDIF

SET THE TIME TO GMT CLOCK
TIME = NHRG*100+NMIN

PRINT THE INPUT DATA

IF (IPRFLG .NE. 0) THEN
IF (IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
IF (IPASS .NE. 1) WRITE(IOOUT, 40)
WRITE(IOOUT, 50)MM,ID,IY
WRITE(IOOUT, 80)NHRG,NMIN,SLAT,SLON
WRITE(IOOUT, 100)FR1,1H1(ILR1)
IF (ILR1 .EQ. 2) WRITE(IOOUT, 110)
IF (ILR1 .EQ. 3) WRITE(IOOUT, 120)
WRITE(IOOUT, 130)FR2,1H2(ILR2)
WRITE(IOOUT, 140)FR3,1H3(ILR3)
WRITE(IOOUT, 150)RG
ENDIF
IF (IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
IF (IPASS .NE. 1) WRITE(NDIRTU, 40)
WRITE(NDIRTU, 50)MM,ID,IY
WRITE(NDIRTU, 80)NHRG,NMIN,SLAT,SLON
WRITE(NDIRTU, 100)FR1,1H1(ILR1)
IF (ILR1 .EQ. 2) WRITE(NDIRTU, 110)
IF (ILR1 .EQ. 3) WRITE(NDIRTU, 120)
WRITE(NDIRTU, 130)FR2,1H2(ILR2)
WRITE(NDIRTU, 140)FR3,1H3(ILR3)
WRITE(NDIRTU, 150)RG
ENDIF
ENDIF

COMPUTE ILLUMINATION

CALL ILLUM ROUTINE TO GET SOLAR/LUNAR POSITIONS AND RAW ILLUMIN.

CALL ILLUM ( DATE, TIME, SLON, SLAT, AZISUN, ALTSUN, AZIMOO, +
        ALMOO, SUNLIT, MOOLIT, DPHASE, IERR )

TEST FOR ERROR IN ILLUM

IF (IERR .NE. 0) GOTO 430

GET SOLAR EFFECTS
ANGLE = PI2-ALTSUN
COZEN = COS(ANGLE)

DUNCAN - SAUER ADJUSTMENT SETS ZENITH ANGLE TO 5 DEGREES IF IT IS LESS THAN THAT.

IF (COZEN .LT. 0.09) COZEN = 0.09
CALL FRATRN ( IH, IM, IL, FR1, FR2, FR3, RG, COZEN, CLDCLR,
+ ID2, TCLR )
TCLSUN = TCLR

GET CRUDE IDEA OF DIRECT TO DIFFUSE FLUX RATIO

SSALB = 0.98
HGFAC = 0.0

RCLSUN IS DIRDIF RATIO FOR SUN ANGLE

IF (SUNLIT .LT. 1.0E-15 .OR. COZEN .LE. 0.0) THEN

RCLSUN = 0.0
RCDSUN = 0.0
GOTO 360
ENDIF

CALL DIRDIF ( SSALB, HGFAC, COZEN, SUNLIT, TCLR, RG, RCLSUN )

DO IT AGAIN FOR CLOUDY CASE

TCLD = TCLR*CLDCLR
HGFAC = 0.85

RCDSUN IS DIRDIF RATIO FOR CLDY SUN POS.

CALL DIRDIF ( SSALB, HGFAC, COZEN, SUNLIT, TCLD, RG, RCDSUN )

ADJUST SOLAR FLUX TO CLOUDY CONDITIONS

360 CONTINUE
SUNLIT = SUNLIT*CLDCLR

AND REPEAT FOR MOON

ANGLE = PI2-ALTMOO
COZEN = COS(ANGLE)
IF (COZEN .LT. 0.09) COZEN = 0.09
CALL FRATRN ( IH, IM, IL, FR1, FR2, FR3, RG, COZEN, CLDCLR,
+ ID2, TCLR )
TCLLUN = TCLR
HGFAC = 0.0

IF (MOOLIT .LE. 1.0E-5 .OR. COZEN .LT. 0.0) THEN

RCLLUN = 0.0
RCDLUN = 0.0
GOTO 370
ENDIF

CALL DIRDIF ( SSALB, HGFAC, COZEN, MOOLIT, TCLR, RG, RCLLUN )
TCLD = TCLR*CLDCLR
HGFAC = 0.85
CALL DIRDIF ( SSALB, HGFAC, COZEN, MOOLIT, TCLD, RG, RCDLUN )

370 CONTINUE
MOOLIT = MOOLIT*CLDCLR
WEIRD THINGS SOMETIMES HAPPEN

IF (SUNLIT .LT. 0.0) SUNLIT = 0.0

IF (MOOLIT .LT. 0.0) MOOLIT = 0.0

BRIGHT = SUNLIT+MOOLIT

CALL FRATRN TO COMPUTE GALACTIC LIGHT ATTENUATION (ASSUMES ZENITH ANGLE OF 45 DEGREES)

CALL FRATRN (IH, IM, IL, FR1, FR2, FR3, RG, 0.707, CLDCLR, +
ID2, TCLR)

GLOWRA = CLDCLR

GET FINAL OVERALL ILLUMINANCE ELUMI

ELUMI = BRIGHT+GLOWRA*.0008

CONVERT FLUX FROM LUX TO W/M^2 USING 93 LUMEN/WATT FACTOR FROM F.W. SEARS 'OPTICS'

SUNGLO = SUNLIT/93.0

AMOON = MOOLIT/93.0

ELUMII = ELUMI/93.0

CONVERT ANGLES TO DEGREES

AZIS = AZISUN*RAD

AZIM = AZIMOO*RAD

ALTS = ALTSSUN*RAD

ALTM = ALTSMOO*RAD

PRINT RESULTS

IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
IF(IPRFLG.NE.0) THEN
WRITE(IOOUT,*)'HIT ENTER TO CONTINUE'
READ(IOOUT,*)
WRITE(IOOUT, 160)ALTS,AZIS,ALTM,AZIM,DPHASE,$
SUNLIT,MOOLIT,ELUMI,ELUMII
ENDIF
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
IF(IPRFLG.NE.0) WRITE(NDIRTU, 160)ALTS,AZIS,ALTM,AZIM,DPHASE,$
SUNLIT,MOOLIT,ELUMI,ELUMII
ENDIF

380 CONTINUE

LOOP TO beginning IF NO DONE CARD HAS BEEN FOUND

IF (IEND.NE.1) GOTO 230
GOTO 450

....PRINT ERROR MESSAGES.....

390 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT, 170)MM
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU, 170)MM
ENDIF
IERR = 2
GOTO 450
400 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT, 180)ID
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU, 180)ID
ENDIF
IERR = 3
GOTO 450
C
IERR = 4 IS NOW A SPARE
410 CONTINUE
GOTO 450
C420 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT, 200)ZN
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU, 200)ZN
ENDIF
IERR = 5
C
GOTO 450
430 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT, 210)IERR
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU, 210)IERR
ENDIF
GOTO 450
440 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT, 220)IOIN
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU, 220)IOIN
ENDIF
IERR = 7
450 CONTINUE
C
RETURN
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: COMBIN          Old Date: ___  ___
File Name: COMBIN.FOR         New Date: 8/17/93

Implemented By: Don Hamann

Reason for Revision: Required a routine to combine the cloud properties of two cloud layers into a single "representative" layer. The cloud shadowing algorithm that is employed allows for only two cloud layers, whereas input accepts up to three cloud layers.

Description of Revision: The routine utilizes the logic employed to combine layers in the Air Force EOTDA. The combined cloud fraction is from the perspective of a ground observer. The extinction coefficient and thickness of the combined layer are weighted averages of the individual values. The asymmetry parameter is a straight average. The base height is that of the lower layer.

Notes: ________________________

_____________________________

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-63
C-HSTX---SCENE SHADOWS---ECR # HSTX-5-1

C NEW ROUTINE TO COMBINE TWO CLOUD LAYERS.

C-HSTX---SCENE SHADOWS----------------------------------------------------------

C*-----------------------------------------------------------------------------*

C<BEGIN>
C<IDENTIFICATION> NAME: COMBIN
C TYPE: FORTRAN SUBROUTINE
C FILENAME: COMBIN.FOR

C<DESCRIPTION>
C This routine combines the cloud properties of two cloud layers into a single "representative" layer. It assumes that all clouds are randomly distributed. This routine assumes that clouds are actually present in the two cloud layers being combined.

C<INPUT>
C COMMON BLOCK VARIABLES:
C NONE

C<OUTPUT>
C COMMON BLOCK VARIABLES:
C NONE

C<CALLED ROUTINES>
C NONE

C<PARAMETER>
C CALLING SEQUENCE:
C COMBIN (CFL,THL,BAL,CGL,HTL,CFU,THU,BAU,CGU,CFNEW,THNEW,BANEW,
C CGNEW,HTNEW,NCLD)

C INPUT:
C HTL Lower cloud base height. (km)
C CFL Cloud fraction of lower cloud layer. (fraction 0.-1.)
C CGL Cloud asymmetry parameter of lower cloud layer.
C CFU Cloud fraction of upper cloud layer. (fraction 0.-1.)
C THL Cloud thickness of lower layer. (km)
C THU Cloud thickness of upper layer. (km)
C CGU Cloud asymmetry parameter of upper cloud layer.
C BAL Ext. coeff. of lower cloud layer. (km-1)
C BAU Ext. coeff. of upper cloud layer. (km-1)

C OUTPUT:
C HTNEW Cloud base of combined layer. (km)
C CFNEW Cloud fraction of combined layer. (fraction 0.-1.)
C THNEW Cloud thickness of combined layer. (km)
C BANEW Extinction coefficient of combined layer. (km-1)
C CGNEW Cloud asymmetry parameter of combined layer.
C NCLD Number of cloud layers.

C<HISTORY>
C CREATED AUGUST, 1993. HUGHES STX CORPORATION.

C<END>

C List of variables:
C BAL Extinction coefficient of lower cloud layer. (km-1)
C BANEW Extinction coefficient of combined layer. (km-1)
C BAU Extinction coefficient of upper cloud layer. (km-1)
C CFL Cloud fraction of lower cloud layer. (fraction 0.-1.)
C CFNEW Cloud fraction of combined layer. (fraction 0.-1.)
C CFU Cloud fraction of upper cloud layer. (fraction 0.-1.)
C CGL Cloud asymmetry parameter of lower cloud layer.

C-64
SUBROUTINE COMBIN (CFL, THL, BAL, CGL, HTL, CFU, THU, BAU, CGU, CFNEW, THNEW, BANEW, CGNEW, HTNEW, NCLD)

THIS ROUTINE ASSUMES THAT CLOUDS ARE ACTUALLY PRESENT IN THE TWO CLOUD LAYERS BEING COMBINED.

IF ( CFL .EQ. 0.0 .OR. CFU .EQ. 0.0 ) THEN
  WRITE(*,*)
  WRITE(*,*)'ERROR IN SUBROUTINE COMBIN: CLOUD FRACTION IS ZERO IN AT LEAST ONE OF THE TWO CLOUD LAYERS BEING PROCESSED BY THIS ROUTINE. THE OUTPUT PARAMETERS OF COMCLD ARE BEING SET TO -999.'
  CFNEW = -999.
  THNEW = -999.
  BANEW = -999.
  CGNEW = -999.
  HTNEW = -999.
ELSE
  COMPUTE THE CLOUD EXTINCTION COEFFICIENT OF COMBINED CLOUD LAYER
  BANEW = (CFL*BAL*THL + CFU*BAU*THU)/(CFL*THL + CFU*THU)
  COMPUTE THE CLOUD FRACTION OF THE COMBINED CLOUD LAYER
  CFNEW = CFL + CFU * ( 1. - CFL )
  COMPUTE THE CLOUD THICKNESS OF THE COMBINED CLOUD LAYER
  THNEW = (CFL*THL + CFU*THU)/CFNEW
  COMPUTE THE ASYMMETRY PARAMETER OF COMBINED CLOUD LAYER
  *** NOTE: THIS VALUE IS THE AVERAGE OF TWO VALUES
  CGNEW = ( CGU + CGL ) * 0.5
  THE CLOUD BASE HEIGHT OF THE COMBINED LAYER IS ASSUMED TO BE THE BASE HEIGHT OF THE LOWER LAYER.
  HTNEW = HTL
  SET CLOUD FRACTION OF HIGH CLOUD TO ZERO INDICATING THAT THREE CLOUD LEVELS HAVE BEEN REDUCED INTO TWO CLOUD LEVELS.
  CFU = 0.0
  REDUCE NUMBER OF CLOUD LAYERS FROM THREE TO TWO.
  NCLD = NCLD - 1
ENDIF
C
RETURN
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: INITI          Old Date: 3/2/93
File Name: ILMDAT.FOR        New Date: 8/17/93

Implemented By: Don Hamann

Reason for Revision: In order to support partly cloudy radiance computation, the directionally dependent diffuse, directionally independent diffuse and direct radiance variables have to be collected and stored for all possible cloud situations or for bracketing conditions (target scene in and out of cloud shadow).

Description of Revision: Re-dimensioned the following variables in RADIA COMMON block:

- I0(20), I1(20) => I0(20,4), I1(20,4)
- SURF(30), SURFO(30) => SURF(30,4), SURFO(30,2)
- F(20), PTHRD(30) => F(20,4), PTHRD(30,2)
- ISTAR(20,30) => ISTAR(20,30,2)

Notes: 

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
SUBROUTINE INITI
COMM/ATMOS/PRES(21), Z(21), RHO(21), NLEV, NLAY
C
C-HSTX --- SCENE SHADOWS --- ECR # HSTX-6-1----------------------------------------
C RE-DIMENSIONED THE FOLLOWING VARIABLES TO SUPPORT PARTLY CLOUDY
C COMPUTATION:
C I0(20), I1(20) ===> I0(20,4), I1(20,4)
C SURF(30), SURFO(30) ===> SURF(30,4), SURFO(30,2)
C F(20), PTHR(30) ===> F(20,4), PTHR(30,2)
C ISTAR(20,30) ===> ISTAR(20,30,2)
C
C COMMON/RADIA/A(20), G(20), GP(20), BETA(20), BETAR(20),
C 1 BETA(20), K(20), P(20), ALPHA(20), BET(20),
C 2 TAU(0:20), I0(20), I1(20), F(20), TF(20), FNOT,
C 3 NANG, PF(180,20), ANG(180,20), I2(20), TAUP(0:20),
C 4 ISTAR(20,30), PTHR(30), SING(30), AP(20), PHF(30)
C 5 , ALBEDO, SURF(30), SURFO(30), TAUSTR, VIS, CAPTP(20),
C 6 B1, B2
C
C COMMON/RADIA/A(20), G(20), GP(20), BETA(20), BETAR(20),
C 1 BETA(20), K(20), P(20), ALPHA(20), BET(20),
C 2 TAU(0:20), I0(20,4), I1(20,4), F(20,4), TF(20), FNOT,
C 3 NANG, PF(180,20), ANG(180,20), I2(20), TAUP(0:20),
C 4 ISTAR(20,30,2), PTHR(30,2), SING(30), AP(20), PHF(30)
C 5 , ALBEDO, SURF(30,4), SURFO(30,2), TAUSTR, VIS, CAPTP(20),
C 6 B1, B2
COMMON/GEOM/THETO,PHIO,UNOT,XLAM,NOBS,PHI(30),BTA(30),NLOS,
1 XOB(30),YOB(30),ZOB(30),THETOB(30),PHIOB(30)
COMMON/BCONST/DGTRD,LUOUT,LUIN
COMMON/IOUNIT/ IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT,
+ IRELH, KSTOR, NPLOTU, STDERR
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON/TARGS/NTARG,XTARG(29),YTARG(29),ZTARG(29),THTARG(29)
1 AZTARG(29),RTARG(29),CTARG(29)
COMMON/IOFILE/IOFILE
CUV REAL IO,II,I2,ISTAR,K,ISTR
REAL IO,II,I2,ISTAR,K
INTEGER IOFILE
CP LOGICAL DIAG
C----------- -------------------------------------------------------------
DATA BIGEXP /88./
C**************************************************************************
C
C PRES REAL VECTOR OF ATMOSPHERIC PressURES IN MB.
C Z REAL VECTOR OF ATMOSPHERIC LEVELS IN KM.
C A REAL VECTOR OF SINGLE SCATTERING ALBEDOS
C RHO REAL VECTOR OF ATMOSPHERIC DENSITY IN GM**M-3
C G REAL VECTOR OF ASYMMETrY PARAMETERS
C GP REAL VECTOR OF MODIFIED ASYMMETrY PARAMETERS
C BETAA REAL VECTOR OF AEROSOL EXTINCTION COEFS IN KM**-1
C BETAR REAL VECTOR OF RAYLEIGH SCATTERING COEFS IN KM**-1
C BET REAL VECTOR OF TOTAL EXTINCTION COEFS IN KM**-1
C K REAL VECTOR OF PARAMETER USED IN DELTAED SOLUTION
C P REAL VECTOR OF PARAMETER USED IN DELTAED SOLUTION
C ALPHA REAL VECTOR OF PARAMETER USED IN DELTAED SOLUTION
C TAU REAL VECTOR OF LAYER OPTICAL THICKNESSES
C NLEV INTEGER OF THE NUMBER OF ATMOSPHERIC LAYERS
C XLAM REAL OF THE WAVELENGTH USED IN MICRO METERS
C THETO REAL OF THE SOLAR ZENITH ANGLE IN DEGREES
C PHIO REAL OF THE SOLAR AZIMUTH IN DEGREES WEST OF SOUTH
C UNOT REAL VALUE OF THE COSINE OF THE SOLAR ZENITH ANGLE
C PHNOT REAL VALUE OF THE SOLAR AZIMUTH ANGLE IN DEGREES
C THETOBREAL VALUE OF THE OBSERVATION ZENITH ANGLE IN DEGREES
C PHIOB REAL VALUE OF THE OBSERVATION AZIMUTH ANGLE IN DEGREES
C ZOB REAL VALUE OF THE OBSERVER HEIGHT IN KM.
C ALBEDO REAL VALUE OF THE SURFACE ALBEDO
C I0 REAL VECTOR OF EDDINGTON INTENSITY
C I1 REAL VECTOR OF MODIFIED INTENSITY
C I2 REAL VECTOR OF MODIFIED INTENSITY
C ISTAR REAL ARRAY OF PATH FUNCTION

C**************************************************************************
C
C *** THIS ROUTINE INITIALIZES THE RADIOMETRIC DATA
C
C *** CALCULATE THE RAYLEIGH SCATTERING COEFFICIENT
C
NANG = 179
DO 6 N = 1,NLEV - 1
DO 5 L = 1,NANG
ANG(L+1,N) = 180.0 / FLOAT(NANG + 1) * FLOAT(L)
P(L,N) = -9.99
5 CONTINUE
ANG(I,N) = 0.0

C-69
\[ \text{ANG(NANG,N)} = 180.0 \]

\[ \text{CONTINUE} \]

\[ \text{CP DIAG = .FALSE.} \]

\[ \text{RHOSTP = RHO(1)} \]

\[ \text{SIGTOP = SIGRAY(XLAM,RHOSTP)} \]

\[ \text{DO } 10 \text{ N = 1,NLEV-1} \]

\[ \text{BETAR(N)} = (RHO(N) + RHO(N+1))/2.0 / RHOSTP * SIGTOP \]

\[ \text{CONTINUE} \]

\[ \text{PI = ASIN(1.0) * 2.0} \]

\[ \text{DGTRD = PI / 180.0} \]

\[ \text{CUV POVR2 = PI / 2.0} \]

\[ \text{FNOT = 1.0} \]

\[ \text{NLAY = NLEV - 1} \]

\[ \text{*** CALCULATE THE COSINE OF THE SOLAR ZENITH ANGLE} \]

\[ \text{UNOT = COS(THETO * DGTRD)} \]

\[ \text{IF(UNOT.LT.0.0)THEN} \]

\[ \text{IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN} \]

\[ \text{WRITE(IOOUT,*)}' ' \]

\[ \text{WRITE(IOOUT,*)}' ' CAUTION, YOUR INPUTS HAVE SPECIFIED A ' \]

\[ \text{WRITE(IOOUT,*)}' ' TIME AND LOCATION SUCH THAT THE SUN ' \]

\[ \text{WRITE(IOOUT,*)}' ' AND THE MOON ARE BELOW THE HORIZON,' \]

\[ \text{WRITE(IOOUT,*)}' ' SKY TO GROUND RATIO ' \]

\[ \text{WRITE(IOOUT,*)}' ' CALCULATIONS MAY BE ERRONEOUS' \]

\[ \text{WRITE(IOOUT,*)}' ' \]

\[ \text{ENDIF} \]

\[ \text{IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN} \]

\[ \text{WRITE(NDIRTU,*)}' ' \]

\[ \text{WRITE(NDIRTU,*)}' ' CAUTION, YOUR INPUTS HAVE SPECIFIED A ' \]

\[ \text{WRITE(NDIRTU,*)}' ' TIME AND LOCATION SUCH THAT THE SUN ' \]

\[ \text{WRITE(NDIRTU,*)}' ' AND THE MOON ARE BELOW THE HORIZON,' \]

\[ \text{WRITE(NDIRTU,*)}' ' SKY TO GROUND RATIO ' \]

\[ \text{WRITE(NDIRTU,*)}' ' CALCULATIONS MAY BE ERRONEOUS' \]

\[ \text{WRITE(NDIRTU,*)}' ' \]

\[ \text{ENDIF} \]

\[ \text{END IF} \]

\[ \text{*** FIND SCALE HEIGHT (SCHT) AS PROPOSED BY ELTERMAN} \]

\[ \text{BET55 = 3.912 / VIS} \]

\[ \text{BET55 = BET55 - BETAR(NLAY)} \]

\[ \text{SCHT = -5.0/(ALOG(5.0E-3/BET55))} \]

\[ \text{*** CALCULATE THE ORDINARY AND MODIFIED OPTICAL THICKNESS OF EACH} \]

\[ \text{CUV SCAPT} \]

\[ \text{SCAPTP = 0.0} \]

\[ \text{TAUSTR = 0.0} \]

\[ \text{TAUSRPF = 0.0} \]

\[ \text{DO 20 N = 1,NLAY} \]

\[ \text{IF(2(N+1) .LE. 5.0 .AND. BETAA(N) .EQ. 0.0) THEN} \]

\[ \text{BETAA(N) = BET55 * EXP(-2(N+1) / SCHT)} \]

\[ \text{ENDIF} \]

\[ \text{BETA(N) = BETAA(N) + BETAR(N)} \]

\[ \text{*** ADJUST THE HENYEY GREESTEIN ASYMMETRY PARAMETER TO ACCOUNT FOR RAYLEIGH SCATTERING} \]

\[ \text{G(N) = G(N) * (BETA(N) - BETAR(N)) / BETA(N)} \]

\[ \text{TAU(N) = (Z(N) - Z(N+1)) * BETA(N)} \]

\[ \text{TAUP(N) = (1.0 - A(N) * G(N) * G(N)) * TAU(N)} \]

\[ \text{TAUSTR = TAUSTR + TAU(N)} \]

\[ \text{TAUSRPF = TAUSRPF + TAUP(N)} \]

\[ \text{TAU(N) = TAUSTR} \]

\[ \text{C-70} \]
TAUP(N) = TAUSTR
IF(DIAG) WRITE(LUOUT,1000)N,BETAR(N),BETAA(N),BETA(N),TAU(N),
TAUSRP
CONTINUE
C1000 FORMAT(' DIAGNOSTIC',I3,7E10.3)
DO 20 N = 1, NLAY
*** CALCULATE DELTA-EDDINGTON PARAMETERS
*** THE NEXT THREE STATEMENTS IMPLEMENT THE DELTA EDDINGTON METHOD
FP = G(N) * G(N)
AP(N) = A(N) * (1.0 - FP) / (1.0 - A(N) * FP)
GP(N) = (G(N) - FP) / (1.0 - FP)
CAPTP(N) = (1.0 - GP(N)) * (Z(N) - Z(N+1)) * BETA(N) *
1
(1.0 - A(N) * G(N) * G(N))
SCAPTP = SCAPTP + CAPTP(N)
CAPTP(N) = SCAPTP
OMINA = (1.0 - AP(N))
OMINAG = (1.0 - AP(N) * GP(N))
K(N) = (3.0 * OMINA * OMINAG) ** 0.5
OMINKU = (1.0 - K(N) * K(N) * UNOT * UNOT)
P(N) = (3.0 * OMINA / OMINAG) ** 0.5
ALPHA(N) = 3.0 * AP(N) * FNOT * UNOT * UNOT *
1
(1.0 + GP(N) * OMINA) / (4.0 * OMINKU)
BET(N) = 3.0 * AP(N) * FNOT * UNOT *
1
(1.0 + 3.0 * GP(N) * OMINA * UNOT * UNOT)
2 / (4.0 * OMINKU)
IF(DIAG) WRITE(LUOUT,1010)N,OMINA,OMINAG,OMINKU,K(N),P(N)
DO 30 N = 1, NLAY
*** INITIALIZE COEF MATRIX AND COLUMN VECTORS
DO 40 N = 1, 2 * NLAY
BB(N) = 0.0
X(N) = 0.0
DO 50 M = 1, 2 * NLAY
AA(N,M) = 0.0
50 CONTINUE
40 CONTINUE
*** LOAD 2N BY 2N MATRIX FOR SOLUTION BY GAUSSIAN ELIMINATION
DO 65 N = 2, 2 * (NLAY -1), 2
I = N / 2
IP1 = I + 1
NP1 = N + 1
NP2 = N + 2
NM1 = N - 1
EMKT = EXP(-K(I) * TAUP(I))
EPKT = EXP(K(I) * TAUP(I))
EMKPT = EXP(-K(IP1) * TAUP(I))
EPKPT = EXP(K(IP1) * TAUP(I))
DEXPO = TAUP(I) / UNOT
IF (ABS(DEXPO) .GT. BIGEXP) THEN
EMTUO = 0.0
ELSE
EMTUO = EXP(-DEXPO)
C-71
C============================================================================
AA(N,N1) = EMK
AA(N,N) = EPK
AA(N,NP1) = -EMK
BB(N) = EMTUO * (ALPHA(I) - ALPHA(IP1))
AA(NP1,N1) = AA(N,N1) * P(I)
AA(NP1,N) = -AA(N,N) * P(I)
AA(NP1,NP1) = -AA(N,NP1) * P(IP1)
BB(NP1) = EMTUO * (BET(I) - BET(IP1))

65 CONTINUE

TAUST = TAUSRP
B2 = (3.0 * UNOT * FNOT * (1.0 - ALBEDO) * (2.0 + 3.0 * UNOT) +
    2.0 - 3.0 * UNOT) * EXP(-TAUST/UNOT)) / (4.0 * (4.0 +
    3.0 * UNOT +

1) (1.0 - ALBEDO) * SCAPTP))
AA(NP2,NP1) = (1.0 - ALBEDO - 2.0 * (1.0 + ALBEDO) * P(IP1) / 3.0) * EXP(-K(IP1) * TAUST)
AA(NP2,NP2) = (1.0 - ALBEDO + 2.0 * (1.0 + ALBEDO) * P(IP1) / 3.0) * EXP(K(IP1) * TAUST)

DEXPO = TAUST / UNOT

C============================================================================
IF (ABS(DEXPO) .GT. BIGEXP) THEN
TEMP = 0.
ELSE
TEMP = EXP(-DEXPO)
ENDIF
BB(NP2) = ((1.0 - ALBEDO) * ALPHA(IP1) - 2.0 *
    1) (1.0 + ALBEDO) * BET(IP1) / 3.0 +
    2) ALBEDO * UNOT * FNOT) * TEMP
C============================================================================
CP IF(DIAG) THEN
CP DO 70 I = 1, 2 * NLAY
CP WRITE(LUOUT,1040) I,(AA(I,J),J=1,2*NLAY),BB(I)
C1040 FORMAT(' DIAGNOSTIC',113,9E10.3,E12.3)
CP70 CONTINUE
CP ENDIF
CP NLOS = 0
DO 77 N = 1, NOBS
    DO 75 M = 1, NTARG
    NLOS = NLOS + 1
    DX = (XTARG(M) - XOB(N))
    DY = (YTARG(M) - YOB(N))
    DZ = (ZTARG(M) - ZOB(N))
    R = DX * DX + DY * DY
    R = R**0.5
    THETOB(NLOS) = 90.0 - ATAN2(DZ,R)/DGTRD
    PHIOB(NLOS) = ATAN2(DX,DY)/DGTRD
CP THETOB(2) = 95.5
CP PHIOB(2) = 180.0
CP PHIOB(2) = TARGAZ
CP WRITE('**') 'PHIOB IS ',PHIOB(NLOS)
C WRITE('**',1200) XOB(N),XTARG(M),YOB(N),YTARG(M),ZOB(N),
C 1 ZTARG(M),THETOB(NLOS),PHIOB(NLOS)
75 CONTINUE
77 CONTINUE
C
C *** CALCULATE THE RELATIVE AZIMUTH ANGLE AND ANGLE BETWEEN THE
C LINE TO SIGHT AND THE INCOMING DIRECT BEAM RADIATION
C
DO 80 N = 1, NLOS
PHI(N) = (PHIOB(N) - PHI0) * DGTRD
C-72
U = COS( THETOB(N) * DGTRD)
BTA(N) = U * UNOT + (1.0 - U * U)**0.5
     * (1.0 - UNOT * UNOT)**0.5 * COS( PHI(N))

CONTINUE

*** CALCULATE THE COSINE OF THE ANGLE BETWEEN THE NORMAL
    TO THE TARGET AND THE SOLAR DIRECTION

DO 90 N = 1, NTARG
   U = COS(THTARG(N) * DGTRD)
   PHIHOL = (AZTARG(N) - PHIO) * DGTRD
   WRITE(*,*)'U AND UNOT ARE ',U,UNOT
   CTARG(N) = U * UNOT + (1.0 - U * U)**0.5
     * (1.0 - UNOT * UNOT)**0.5 * COS(PHIHOL)
   WRITE(*,*)'CTARG IS ',CTARG(N)
90 CONTINUE

*** IF LAYER PHASE FUNCTION IS NOT INPUT THEN USE
    A HENYEW GREENSTEIN APPROXIMATION FOR THE LAYER

DO 100 N = 1, NLAY
   DO 110 L = 1, NANG
      BTT = COS(ANG(L,N) * DGTRD)
      IF(PF(L,N) .EQ. -9.99) PF(L,N) = (1.0 - G(N) * G(N)) / 1
        * (4.0 * PI * (1.0 - 2.0 * G(N) * 2 BTT + G(N) * G(N)) ** 1.5)
      PF(L,N) = PF(L,N) * BETAA(N) + 3.0 / 16.0 * PI
     * (1.0 + BTT * BTT) * BETAR(N)
      PF(L,N) = PF(L,N) / (BETAA(N) + BETAR(N))
110 CONTINUE
100 CONTINUE

*** CONVERT THE PHASE FUNCTION ANGLES TO THEIR RESPECTIVE COSINES

DO 120 N = 1, NLAY
   DO 130 L = 1, NANG
      ANG(L,N) = COS(ANG(L,N) * DGTRD)
130 CONTINUE
120 CONTINUE
RETURN
END

<END OF UNIT: INITI>
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: DELTED          Old Date: 8/11/93
File Name: CONTRAST.FOR        New Date: 8/17/93

Implemented By: Don Hamann

Reason for Revision: Enabled FASCAT calculations to be performed on atmospheric layers for all possible cloud situations.

Description of Revision: Added cloud situation iteration number as an argument. Re-dimensioned the variables I0, I1, SURF, SURFO, P, PTHRD, and ISTAR. Perform delta-Eddington calculations using atmospheric parameters for the particular cloud situation.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-74
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: DELTED                  Old Date: 9/14/93
File Name: CONTRAST.FOR                New Date: 9/14/93

Implemented By: Dan DeBenedictis

Reason for Revision: ISTAR needs to be saved only for bracketing cloud situations.

Description of Revision: Index ISTAR with II for the first and last cloud situations. Overwrite ISTAR values for other cloud situations. Set II index equal to the cloud situation loop counter, or to 2 if the counter is greater than 2.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
C-<BEGIN>
C<IDENTIFICATION> NAME: DELTED
C TYPE: FORTRAN SUBROUTINE
C FILENAME: SACC.F
C<DESCRIPTION>
C DELTA-EDDINGTON ROUTINE CALCULATED RADIATIVE INTENSITIES AND FLUXES
C
C<INPUT>
C COMMON BLOCK VARIABLES:
C / ATMOS / NLAY, NLEV
C / CONST / DGTRD, DIAG, LUOUT, PI
C / GAUSS / X
C / GEOM / NOBS, UNOT, BTA, THETOB
C / RADIA / FNOT, ALPHA, BET, BETA, BETAA, BETAR, F, G, ISTAR,
C IO, II, I2, K, P, PHF, TAU, TAUP, TF
C<OUTPUT>
C COMMON BLOCK VARIABLES:
C / RADIA / F, G, IO, II, I2, ISTAR, PHF, TF
C<CALLED ROUTINES>
C ELIMIN - (SUBR) SOLVES SYSTEM OF EQUATIONS
C<PARAMETER>
C I - CLOUD SITUATION NUMBER
C
C CALLING SEQUENCE:
C DELTED (I)
C<HISTORY>
C UPDATED FALL 1988. MODIFIED BY OPTIMETRICS, INC IN JANUARY 1989 TO I
C PERFORMANCE PREDICTIONS FOR A PARTICULAR WEAPON SYSTEM.
C<END>
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-7-1-----------------------------
C ADD CLOUD SITUATION NUMBER AS ARGUMENT
C SUBROUTINE DELTED
C-HSTX---SCENE SHADOWS-----------------------------
C COMMON/ATMOS/PRES(21),Z(21),RHO(21),NLEV,NLAY
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-7-1-----------------------------
C RE-DIMENSIONED THE FOLLOWING VARIABLES TO SUPPORT PARTLY CLOUDY
C COMPUTATION:
C IO(20), I1(20) ===> IO(20,4), II(20,4)
C SURF(30), SURFO(30) ===> SURF(30,4), SURFO(30,2)
C F(20), PTHRD(30) ===> F(20,4), PTHRD(30,2)
C ISTAR(20,30) ===> ISTAR(20,30,2)
C
C-HSTX---SCENE SHADOWS-----------------------------
C COMMON/RADIA/A(20),G(20),GP(20),BETAA(20),BETAR(20),
C 1 BETA(20),K(20),P(20),ALPHA(20),BET(20),
C 2 TAU(0:20),IO(20),I1(20),F(20),TF(20),FNOT,
C 3 NANG,PF(180,20),ANG(180,20),I2(20),TAUP(0:20),
C 4 ISTAR(20,30),PTHRD(30),SING(30),AP(20),PHF(30)
C 5 ALBEDO,SURF(30),SURFO(30),TAUSTR,VIS,CAPTP(20),
C 6 B1,B2
C
C-HSTX---SCENE SHADOWS-----------------------------
C COMMON/RADIA/A(20),G(20),GP(20),BETAA(20),BETAR(20),
C 1 BETA(20),K(20),P(20),ALPHA(20),BET(20),
C 2 TAU(0:20),IO(20,4),I1(20,4),F(20,4),TF(20),FNOT,
C 3 NANG,PF(180,20),ANG(180,20),I2(20),TAUP(0:20),
ISTAR(20,30,2), PTHRD(30,2), SING(30), AP(20), PHF(30)
, ALBEDO, SURF(30,4), SURFO(30,2), TAUSTR, VIS, CAPTP(20),
B1, B2

C-HSTX---SCENE SHADOWS-----------------------------------------------

COMMON/GEOM/THETO, PHI0, UNOT, XLAM, NOBS, PHI(30), BTA(30), NLOS,
XOB(30), YOB(30), ZOB(30), THETOB(30), PHIOB(30)

COMMON/TARGS/NTARG, XTARG(29), YTARG(29), ZTARG(29), THTARG(29),
AZTARG(29), RTARG(29), CTARG(29)

COMMON/BCONST/DGTRD, LUOUT, LUIN
COMMON/CONST/PI, PI2, PIRAD, TWOPHI, TORMB, CDEGK
COMMON/GAUSS/AA(40,40), BB(40), BB(40), BB(40)
COMMON/IOFILE/IOFILE
COMMON/IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, IRELH,
STERR

INTEGER IOFILE

DIMENSION C1(20), C2(20), AN(180), PFN(180)

REAL IO, I1, I2, ISTAR, K

CP

LOGICAL DIAG

DATA BIGEXP /88.0/

C

*** CALL SUBROUTINE TO SOLVE SYSTEM OF EQUATIONS

C

WRITE(*,*)'IN DELTED NLAY AND ERR ARE ',NLAY, ERR
CALL ELIMIN(2 * NLAY, ERR)

C

*** FORMULATE C1 AND C2 VECTORS

C

NN = 0
DO 10 N = 1, NLAY
    NN = NN + 1
    NN = NN + 1
CP IF (DIAG) WRITE(1UOUT, 1020) N, C1(N), C2(N)
C1020 FORMAT('DIAGNOSTIC', I3, 2E10.3)
10 CONTINUE

C

*** CALCULATE INTENSITIES AND FLUXES

C

DO 20 N = 1, NLEV-1
    ARG = K(N) * TAUP(N)
    IF (ARG .GT. 50.) THEN
        IF (IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
            WRITE(IOOUT,*) 'ARG IS TOO LARGE'
        ENDIF
        IF (IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
            WRITE(NDIRTU,*) 'ARG IS TOO LARGE'
        ENDIF
        STOP
    ENDIF
    DEXPO = TAUP(N) / UNOT
C
    IF (ABS(DEXPO) .GT. BIGEXP) THEN
        ETUO = 0.0
    ELSE
        ETUO = EXP(-DEXPO)
    ENDIF

C---------------------------------------------------------------------

C-HSTX---SCENE SHADOWS---ECR # HSTX-7-1-----------------------------
C RE-DIMENSION ARRAYS IO, II, AND F TO ACCOUNT FOR CLOUD CONTRIBUTION.
C
I = 1

One iteration of the loop for the conditions:
both layers clear

C-77
C one layer overcast, one layer clear
C both layers overcast
I = 2

Two iterations of the loop for the conditions:
one partly cloudy layer, one clear layer
one partly cloudy layer, one overcast layer

I = 4
Four iterations of the loop for the condition:
two partly cloudy layers

I0(N,I) = B1 - 0.75 * UNOT * UNOT * FNOT * ETUO - B2 * CAPTP(N)
I1(N,I) = B2 - 0.75 * UNOT * FNOT * ETUO
F(N,I) = PI * (I0(N,I) + 2.0 / 3.0 * I1(N,I))
TF(N) = F(N,I) + UNOT * PI * FNOT * ETUO
CP WRITE(*,*)'I0 AND I1 IN TF ARE ',I0(N,I),II(N,I)
CP WRITE(*,*)'TF IS ',TF(N)
CP IF(DIAG) WRITE(LUOUT,1030)N,EMK,EPK,ETUO,I0(N,I),I1(N,I),
CP + F(N,I),TF(N)
C1030 FORMAT(' DIAGNOSTIC',13,7E10.3)
C-HSTX---SCENE
SHADOWS---------------------------------------------

C 20 CONTINUE
C
*** CALCULATE THE VALUE OF I2 AND ISTAR
C
SINOT = (1.0 - UNOT * UNOT) ** 0.5
DO 30 N = NLAY, 1, -1
DO 30 L = 1, NLOS
   AN(LL) = ANG(LL,N)
   PFN(LL) = PF(LL,N)
25 CONTINUE
CSTHOB = COS(THETOB(L) * DGTRD)
TAUX = TAUP(N)
SINOB = (1.0 - CSTHOB * CSTHOB) ** 0.50
BT = BTA(L)
IF (BT .GT. COS(25.0 * DGTRD)) TAUX = TAU(N)
CJ CHANGE INTERPOLATION FROM BESSEL TO CUBIC SPLINE
CJ AUGUST 1992 J.FITZGERELLE, P.GILLESPIE
CJ PHF(L) = CUBINT(BT,AN,PFN,NANG)
CJ PHF(L) = BSLIF(BT,AN,PFN,NANG)
C---------------------------------------------------------------------
IF (ABS(TAUX/UNOT) .GT. BIGEXP) THEN
   TERM = 0.0
ELSE
   TERM = EXP(-TAUX/UNOT)
ENDIF
EPS = PI * FNOT * TERM
C=========================================================================
C CPHF(L) = ((1.0 - G(N) * G(N))/(4.0 * PI * ( 1.0 - 2.0 * G(N))
C 1 BT + G(N) * G(N) ** 1.5) * BETAA(N) + 3.0 / (16.0 * PI) *
C 2 (1.0 + COS(BT) ** 2) * BETAR(N)) / BETA(N)
C C-HSTX---SCENE SHADOWS---ECR # HSTX-7-1---------------------------------
C RE-DIMENSIONED ARRAYS I0, I1, AND ISTAR TO ACCOUNT FOR CLOUD
CONTRIBUTION.
C I = 1 - CLEAR SKY.
C I = 2 - TOP CLOUD OR BOTTOM CLOUD ONLY (ONE CLOUD).
C I = 4 - TOP AND BOTTOM CLOUD (TWO CLOUDS).
C-HSTX---SCENE SHADOWS---ECR # HSTX-7-2---------------------------------
C ISTAR NEEDS TO BE CALCULATED FOR ONLY CLEAR OR OVERCAST
C
I2(N) = 4.0 * PI * I0(N,I) * PHF(L) * EPS * SINOB * SINOT / 
1 (4.0 * PI * I0(N,I) + PHF(L) * EPS)
C-78
CP IF(DIAG) WRITE(LUOUT,*) ' TERMS ', TERM1, TERM2, TERM3, TERM4
C
II = I
IF (I.GT.2) II = 2
ISTAR(N,L,II) = BETA(N) * (PHF(L) * EPS + IO(N,I) + G(N) * I1(N,I) * CSTHOB + G(N) * I2(N) * COS(PHI(L)))
C-HSTX---SCENE SHADOWS-----------------------------------------------
C
30 CONTINUE
RETURN
END
C
C <END OF UNIT: DELTED>
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: CONTST Old Date: 8/17/93
File Name: CONTRAST.FOR New Date: 8/20/93

Implemented By: Dan DeBenedictis

Reason for Revision: Compute average value of diffuse radiance over four possible cloud situations. Compute bracketing values of path radiance, direct radiance, and total radiance for the target scene in and out of cloud shadow.

Description of Revision: Re-dimension the variables IO, I1, SURF, SURF0, F, PTHRD, and ISTAR. Add cloud situation number (I) and solar zenith angle (SZA) as arguments. Add the COMMON block CLOUD to access cloud information. Call PCDIF to compute the average diffuse radiance and the correction for forward scattering.

Notes: 

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: CONTST  Old Date: 8/20/93

File Name: CONTRAST.FOR  New Date: 8/27/93

Implemented By: Don Hamann

Reason for Revision: To maintain information on the probability of a cloud-free path through each cloud layer and on the probability of the target scene in direct light.

Description of Revision: Added variable PSCLD and array PCF(2) to CLOUD COMMON block to maintain probability information.

Notes: 

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-81
Routine Name: CONTST          Old Date: 8/27/93
File Name: CONTRAST.FOR      New Date: 9/14/93

Implemented By: Dan DeBenedictis

Reason for Revision: Bracketing values of transmittance (TN) must be computed for clear and/or overcast.

Description of Revision: Re-dimension the variable TN and use SAVE to save the values of TN for each call to CONTST.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: CONTST                   Old Date: 9/14/93

File Name: CONTRAST.FOR                New Date: 9/17/93

Implemented By: Don Hamann

Reason for Revision: Fix an error in the calls to PCDIF. Use temporary variables in the calls. Add a flag to keep track of the cloud situation.

Description of Revision: Added variable ICLDF to CLOUD COMMON block to maintain information on cloud situation. Created local array SURFA(4) to temporarily store direct radiance for call to PCDIF. Store directionally dependent diffuse, directionally independent diffuse, and direct radiance in temporary variables for call to PCDIF. Which values are loaded depends on the cloud situation.

Notes: ICLDF flag = 1 for two clear layers, 2 for one overcast layer, 3 for two overcast layers, 4 for one clear layer and one partly cloudy, 5 for one overcast layer and one partly cloudy, and 6 for two partly cloudy layers.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: CONTST                Old Date: 9/17/93
File Name: CONTRAST.FOR              New Date: 10/20/93

Implemented By: Melanie J. Gouveia

Reason for Revision: Update equations for direct and diffuse radiance to more closely match the literature (Shettle and Weinman, 1970; Joseph, Wiscombe, and Weinman, 1976; Hering and Johnson, 1984). Make the subroutine more modular by moving the setup for and calls to PCDIF to the new DIFUSE routine.

Description of Revision: Use the Delta-Eddington optical depth, rather than regular optical depth, for all direct radiance calculations. Use diffuse = I0 - 2/3 I1 for upward line-of-sight, diffuse = I0 + 2/3 I1 + CORR for downward LOS, and diffuse = I1 + CORR/2 for horizontal LOS.

Notes: The upward and downward LOS equations are found in the literature. The horizontal LOS equation is an average of the upward and downward.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-84
NAME: CONTRAST
TYPE: FORTRAN SUBROUTINE
FILENAME: SACC.F

DESCRIPTION
CALCULATES THE PATH RADIANCES, THE TRANSMITTED BACKGROUND RADIANCES
AND THE APPARENT SPECTRAL CONTRASTS ALONG THE VARIOUS LINES OF SIGH

INPUT
COMMON BLOCK VARIABLES:
/ ATMOS / NLAY, NLEV, Z
/ CONST / DGTAD, DIAG, LOUT
/ GEOM / NOBS, UNOT, THETOB
/ RADI / ALBEDO, FNOT, TAUSTR, BETA, I0, I1, ISTAR
/ TARGS / NTARG, CTARG, THTARG, ZTARG
/ CLOUD / NUMCLD, MLOOP, ZC1, ZC2, CF1, CF2

OUTPUT
COMMON BLOCK VARIABLES:
/ RADI / PTHRBD, SURF

CALLED ROUTINES
CJ BSL1F - (SUBR) ACCESSES BESSEL INTERPOLATION ROUTINE
CJ CUBINT - (SUBR) ACCESSES CUBIC SPLINE ROUTINE
C J SIMPNE - (SUBR) PERFORMS SIMPSON RULE INTEGRATION

PARAMETER
I - CLOUD SITUATION NUMBER
SZA - SOLAR ZENITH ANGLE

CALLING SEQUENCE:
CONTST (I,SZA)

HISTORY
UPDATED FALL 1988

END

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------------
ADD CLOUD SITUATION NUMBER AND SOLAR ZENITH ANGLE AS ARGUMENTS
SUBROUTINE CONTST
SUBROUTINE CONTST (I,SZA)

CP PARAMETER(MAXEXP = 30)
COMMON/ATMOS/PRES(21),Z(21),RHO(21),NLEV,NLAY

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------------
RE-DIMENSIONED THE FOLLOWING VARIABLES TO SUPPORT PARTLY CLOUDY
COMPUTATION:
I0(20), I1(20) ====> I0(20,4), I1(20,4)
SURF(30), SURFO(30) ====> SURF(30,4), SURFO(30,2)
F(20), PTHRBD(30) ====> F(20,4), PTHRBD(30,2)
ISTAR(20,30) ====> ISTAR(20,30,2)

COMMON/RADI/A(20),G(20),GP(20),BETAA(20),BETAR(20),
1 1 BETA(20),K(20),F(20),ALPHA(20),BET(20),
2 TAU(0:20),I0(20),I1(20),F(20),FNOT,
3 NANG,PF(180,20),ANG(180,20),I2(20),TAUP(0:20),
4 ISTAR(20,30),PTHBD(30),SING(30),AP(20),PHF(30)
5 ,ALBEDO,SURF(30),SURFO(30),TAUSTR,VIS,CAPTP(20),
6 ,B1,B2
COMMON/RADI/A(20),G(20),GP(20),BETAA(20),BETAR(20),
1 BETA(20),K(20),F(20),ALPHA(20),BET(20),

C-85
C-HSTX---SCENE SHADOWS---------------------------------------------------

COMMON/GEOM/THETO,PHIO,UNOT,XLAM,NOBS,PHI(30),BTA(30),NLOS,
  XOB(30),YOB(30),ZOB(30),THETOB(30),PHIOB(30)
COMMON/BCONST/DGTRD,LUOUT,LUIN
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON/TARGS/NTARG,XTARG(29),YTARG(29),ZTARG(29),THTARG(29),
  AZTARG(29),RTARG(29),CTARG(29)
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,IRELH,
  NCLIMT,NPLOTU,STDERR
COMMON/IOFILE/IOFILE

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-----------------------------------
C ADD CLOUD COMMON BLOCK TO ACCESS CLOUD INFORMATION.

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-2-----------------------------------
C ADDED VARIABLE PSCLD AND ARRAY PCF(2) TO CLOUD COMMON BLOCK TO MAINTAIN
C INFORMATION ON PROBABILITY OF CLOUD-FREE PATH THROUGH EACH CLOUD LAYER
C AND THE PROBABILITY OF THE TARGET IN DIRECT LIGHT.

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3-----------------------------------
C ADDED VARIABLE ICLDF TO CLOUD COMMON BLOCK TO MAINTAIN INFORMATION ABOUT
C CLOUD SITUATION.
C COMMON /CLOUD/NUMCLD, MLOOP, CLDBTA(2),CLDG(2), LYRCLD(2),
  THK1, THK2, THK3, ITY1, ITY2, ITY3,
  ZC1,ZC2,ZC3,CF1,CF2,CF3
COMMON /CLOUD/ICLDF,NUMCLD, MLOOP, CLDBTA(2),CLDG(2), LYRCLD(2),
  THK1, THK2, THK3, ITY1, ITY2, ITY3,
  ZC1,ZC2,ZC3,CF1,CF2,CF3,PCF(2),PSCLD

C-HSTX---SCENE SHADOWS---------------------------------------------------

INTEGER IOFILE
REAL I0,I1,I2,ISTAR,K
LOGICAL DIAG

*** THIS SUBROUTINE CALCULATES THE PATH RADIANCES,
*** THE TRANSMITTED BACKGROUND RADIANCES AND THE APPARENT
*** SPECTRAL CONTRASTS ALONG THE VARIOUS LINES OF SIGHT.

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3-----------------------------------
C RE-DIMENSION AND SAVE THE TRANSMISSION VALUE (TN) TO COMPUTE THE
C TRANSMITTED TARGET RADIANCE AT OBSERVER WITH AND WITHOUT CLOUDS
C DIMENSION R(20),RAD(21),PD(20),TN(20)
C DIMENSION R(20),RAD(21),PD(20),TN(20,2)
SAVE TN

C-HSTX---SCENE SHADOWS---------------------------------------------------

*** INITIALIZE

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-----------------------------------
C SET IN/OUT OF CLOUD SHADOW FLAG
II = I
  IF (II.GT.1) II = 2

C-HSTX---SCENE SHADOWS---------------------------------------------------

L = 0
IER = 0
WRITE(*,*) 'NOBS AND NTARG ARE ',NOBS,NTARG
DO 222 N = 1,NOBS
  DO 333 M = 1,NTARG
    DIST = 0.0
  222 CONTINUE
  CONTINUE

C-86
SUM = 0.0
ISTOP = 0
CSTHOB = COS(THETOB(L) * DGTRD)

*** FOR UPWARD DIRECTED LINES OF SIGHT
FIND THE LEVEL IN WHICH THE OBSERVATION IS TAKEN
AND THE NUMBER OF LEVELS THRU WHICH THE TRANSMISSION
MUST BE CALCULATED

IF(THETOB(L) .GE. 0.0 .AND. THETOB(L) .LT. 85.0) THEN
CP WRITE(*,*)'UPWARD LINES OF SITE AND THETOB ',THETOB(L)
DO 999 J = 1, NLEV
   IF(Z(J) .GE. ZOB(N)) THEN
      NLOCO = J
   ENDIF
   IF(Z(J) .GE. ZTARG(M)) THEN
      NLOCT = J
   ENDIF
999 CONTINUE
NP = NLOCO - NLOCT
C
R(1) = 0.0
DIST = (Z(NLOCO) - ZOB(N))
R(2) = DIST / CSTHOB
SUM = SUM + DIST / CSTHOB * BETA(NLOCO)
SLP = DIST / CSTHOB
TRV= 1.0 / BETA(NLOCO) * (1.0 - EXP(-R(2) * BETA(NLOCO)))
TRV = TRV / SLP

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------
C RE-DIMENSIONED ARRAY ISTAR TO ACCOUNT FOR CLOUD CONTRIBUTION.
RAD(1) = ISTAR(NLOCO,L,II)
RAD(2) = ISTAR(NLOCO-1,L,II)

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3-------------------------------
C RE-DIMENSIONED ARRAY TN TO ACCOUNT FOR CLOUD CONTRIBUTION.
TN(1,II) = TRV
IF(TN(1,II) .GE. .005) ISTOP = 1
TN(2,II) = EXP(-SUM)
IF(TN(2,II) .GE. .005) ISTOP = 2
PD(1) = TN(1,II) * RAD(1)
PD(2) = TN(2,II) * RAD(2)

C-HSTX---SCENE SHADOWS----------------------------------------------
C
IF (NP .GT. 1) THEN
   DO 888 J = 1, NP - 1
      DIST = Z(NLOCO-J) - Z(NLOCO-J+1)
      R(J+2) = R(J+1) + DIST / CSTHOB
      SLP = DIST / CSTHOB
      TRV= 1.0 / BETA(NLOCO-J) *
      (1.0 - EXP(-DIST * BETA(NLOCO-J) / CSTHOB)) / SLP
      SUM = SUM + DIST / CSTHOB * BETA(NLOCO-J)
   1
   2
888 CONTINUE

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------
C RE-DIMENSIONED ARRAY ISTAR TO ACCOUNT FOR CLOUD CONTRIBUTION.
RAD(J+2) = ISTAR(NLOCO-J-1,L,II)

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3-------------------------------
C RE-DIMENSIONED ARRAY TN TO ACCOUNT FOR CLOUD CONTRIBUTION.
TN(J+2,II) = TN(J+1,II) * TRV
TN(J+2,II) = EXP(-SUM)
PD(J+2) = TN(J+2,II) * RAD(J+2)
IF(TN(J+2,II) .GE. 0.005) ISTOP = J + 2

C-HSTX---SCENE SHADOWS-----------------------------------------------
C
888 CONTINUE
ENDIF
DIST = ZTARG(M) - Z(NLOCT+I)
CP WRITE(*,*)'ZTARG AND ZNLOCT+I ',ZTARG(M),Z(NLOCT+I)
R(NP+2) = R(NP+1) + DIST / CSTHOB +.001
SUM = SUM + DIST / CSTHOB * BETA(NLOCT)
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------
C RE-DIMENSIONED ARRAY ISTAR TO ACCOUNT FOR CLOUD CONTRIBUTION.
RAD(NP+2) = ISTAR(NLOCT,L,II)
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3-------------------------------
C RE-DIMENSIONED ARRAY TN TO ACCOUNT FOR CLOUD CONTRIBUTION.
TN(NP+2,II) = EXP(-SUM)
PD(NP+2) = TN(NP+2,II) * RAD(NP+2)
C-HSTX---SCENE SHADOWS-----------------------------------------------

C *** CALCULATE THE PATH RADIANCE FROM THE OBSERVER TO THE TARGETS ABOVE
C
C *** INTEGRATE THE PATH FUNCTION FROM OBSERVER TO TARGET
C
IF(ISTOP .LE. 1) STOP 'ISTOP'
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------
C RE-DIMENSIONED ARRAY PTHRD TO ACCOUNT FOR CLOUD CONTRIBUTION.
PTHRD(L,II) = SIMPNE(R,PD,ISTOP,IER)
C-HSTX---SCENE SHADOWS-----------------------------------------------

C *** ADD TARGET TRANSMITTED RADIANCES
C
C *** START WITH THE CONTRIBUTION FROM THE TARGET ILLUMINATED BY THE DIRECT BEAM USING SCALED OPTICAL DEPTH SO AS TO INCLUDE THE FORWARD SCATTERED CONTRIBUTION HERE
C
1 TAUTRG= TAUP(NLOCT) - (ZTARG(M) - Z(NLOCT+1))
2 * BETA(NLOCT) * (1.0 - A(NLOCT) * G(NLOCT))
CP WRITE(*,*)'RTARG IS ',RTARG(M)
CP WRITE(*,*)'CTARG IS ',CTARG(M)
CP WRITE(*,*)'FNOT IS ',FNOT
CP WRITE(*,*)'EXP IS ',EXP(-TAUTRG/UNOT)
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------
C RE-DIMENSIONED ARRAY SURF TO ACCOUNT FOR CLOUD CONTRIBUTION.
SURF(L,I) = FNOT * CTARG(M) * RTARG(M) * EXP(-TAUTRG/UNOT)
CP WRITE(*,*)'UPWARD LOS SURF DIR IS ',SURF(L,I)
1 IF (SURF(L,I) .LT. 0.0) SURF(L,I) = 0.0
C-HSTX---SCENE SHADOWS-----------------------------------------------

C *** ADD THE CONTRIBUTION FROM THE TARGET ILLUMINATED BY THE DIFFUSE FLUX WHICH IS ASSUMED TO BE THE UPWARD FLUX UNLESS THE TARGET NORMAL IS HORIZONTAL IN WHICH CASE THE FLUX IS THE AVERAGE OF THE UPWARD AND DOWNWARD VALUES
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------
C COMPUTE AVERAGE COMPONENTS OF DIFFUSE RADIANCE OVER ALL CLOUD SITUATIONS AND TOTAL RADIANCE FOR TARGET IN DIRECT LIGHT AND IN CLOUD SHADOW. RE-DIMENSIONED ARRAYS SURF AND SURFO TO ACCOUNT FOR C CLOUD CONTRIBUTION.

C-88
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3-----------------------------------
C RE-DIMENSIONED ARRAY TN TO ACCOUNT FOR CLOUD CONTRIBUTION.
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-4-----------------------------------
C STORE DIRECTIONALLY DEPENDENT DIFFUSE, DIRECTIONALLY INDEPENDENT
C DIFFUSE, AND DIRECT RADIANCE IN TEMPORARY VARIABLES FOR CORRECT
C COMPUTATION OF AVERAGE DIFFUSE RADIANCE FOR PARTICULAR CLOUD
C SITUATION. USE TEMPORARY VARIABLES IN CALLS TO PCDIFF.
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-5-----------------------------------
C MOVE TEMPORARY VARIABLES AND PCDIFF CALLS TO DIFUSE SUBROUTINE.
C ELIMINATE DEPENDENCE ON THTARG.

IF (I.EQ.MLOOP) THEN
    CALL DIFUSE (L,NLOCT,SZA,AVEIO,AVEII,CORR)
    CALL DIFUSE (L,NLOCT,SZA,AVEIO,AVEII,CORR)
C
    IF (THTARG(M) .NE. 90.0) THEN
        SURF(L,1) = SURF(L,1) + RTARG(M) * (AVEIO - 2.0 / 3.0 * AVEII)
        SURF(L,2) = SURF(L,MLOOP) + RTARG(M) * (AVEIO - 2.0 / 3.0 * AVEII)
    ELSE
        SURF(L,1) = SURF(L,1) + AVEIO * RTARG(M)
        SURF(L,2) = SURF(L,MLOOP) + AVEIO * RTARG(M)
        ENDIF
    ENDIF

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-------------------------------
C
C
C
C-89
C RE-DIMENSIONED ARRAY ISTAR TO ACCOUNT FOR CLOUD CONTRIBUTION.
   RAD(1) = ISTAR(NLOCO,L,II)
   RAD(2) = ISTAR(NLOCO+1,L,II)
C-HSTX---SCENE SHADOWS---------------------------------------------
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3-----------------------------
C RE-DIMENSIONED ARRAY TN TO ACCOUNT FOR CLOUD CONTRIBUTION.
   TN(1,II) = TRV
   TN(2,II) = EXP(-SUM)
   IF(TN(2,II) .GE. 0.005) ISTOP = 2
   PD(1) = RAD(1) * TN(1,II)
   PD(2) = RAD(2) * TN(2,II)
C-HSTX---SCENE SHADOWS---------------------------------------------
C
IF(NP .GT. 1) THEN
   DO 555 J = 1, NP - 1
      DIST = Z(NLOCO+J) - Z(NLOCO+J+1)
      R(J+2) = R(J + 1) + DIST / ABS(CSTHOB)
      SLP = DIST / ABS(CSTHOB)
      TRV= 1.0 / BETA(NLOCO-J) * (1.0 - EXP(-DIST * BETA(NLOCO-J)) / ABS(CSTHOB)) / SLP
      SUM = SUM + DIST / ABS(CSTHOB) * BETA(NLOCO-J)
   1
   2
   IF(TN(J+2,II) .GE. 0.005) ISTOP = J + 2
   PD(J+2) = TN(J+2,II) * RAD(J+2)
C-HSTX---SCENE SHADOWS---------------------------------------------
C
555 CONTINUE
ENDIF
DIST = Z(NLOC1) - ZTARG(M)
CP WRITE(*,*)'R, NP ',R,NP
CP WRITE(*,*)' R(NP+I) ',R(NP+1)
CP WRITE(*,*)'ZLOCT, ZTARG ',Z(NLOC1),ZTARG(M)
CP WRITE(*,*)'SUM = SUM + DIST / ABS(CSTHOB) * BETA(NLOC1)
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-----------------------------
C RE-DIMENSIONED ARRAY ISTAR TO ACCOUNT FOR CLOUD CONTRIBUTION.
   RAD(NP+2) = ISTAR(NLOC1,L,II)
C-HSTX---SCENE SHADOWS---------------------------------------------
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3-----------------------------
C RE-DIMENSIONED ARRAY TN TO ACCOUNT FOR CLOUD CONTRIBUTION.
   TN(NP+2,II) = EXP(-SUM)
   PD(NP+2) = RAD(NP+2) * TN(NP+2,II)
C-HSTX---SCENE SHADOWS---------------------------------------------
C
*** CALCULATE THE PATH RADIANCE TO THE TARGETS BELOW
C
*** INTEGRATE PATH FUNCTION FROM OBSERVER TO TARGET
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1-----------------------------
C RE-DIMENSIONED ARRAY PTHRD TO ACCOUNT FOR CLOUD CONTRIBUTION.
   PTHRD(L,II) = SIMPNE(R,PD,ISTOP,IER)
C
C-90
*** ADD TARGET TRANSMITTED RADIANCES
FIRST CALCULATE THE INHERENT RADIANCES DUE TO THE
DIRECT BEAM AND FORWARD SCATTERED COMPONENT

\[ \text{TAUTG} = \text{TAVP(NLOCT)} - (\text{ZTARG(M)} - \text{Z(NLOCT+1)}) \]
\[ * \text{BETA(NLOCT)} * (1.0 - \text{A(NLOCT)} * \text{G(NLOCT)} * \text{G(NLOCT)}) \]

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1---------------------------------------------
C RE-DIMENSIONED ARRAY SURF TO ACCOUNT FOR CLOUD CONTRIBUTION.

C-HSTX---SCENE SHADOWS---ECR # HSTX-8-5---------------------------------------------
C USE DELTA-EDDINGTON OPTICAL DEPTH, RATHER THAN REGULAR OPTICAL
C OPTICAL DEPTH, IN DIRECT RADIANCE CALCULATION. ELIMINATE REGULAR
C OPTICAL DEPTH AND ADDDIFF VARIABLE FROM CALCULATION. SCALED OPTICAL
C DEPTH INCLUDES FORWARD SCATTERED COMPONENT.

\[ \text{TUTRG} = \text{TAVP(NLOCT)} - (\text{ZTARG(M)} - \text{Z(NLOCT+1)}) \]
\[ * \text{BETA(NLOCT)} \]
\[ \text{ADDIFF} = \text{FNOT} * \text{UNOT} \]
\[ * (\exp(-\text{TAUTG} / \text{UNOT}) - \exp(-\text{TUTRG} / \text{UNOT})) \]
\[ \text{SURF}(L,I) = \text{FNOT} * \text{CTARG}(M) * \text{RTARG}(M) * \exp(-\text{TUTRG} / \text{UNOT}) \]
\[ \text{SURF}(L,I) = \text{FNOT} * \text{CTARG}(M) * \text{RTARG}(M) * \exp(-\text{TAUTG} / \text{UNOT}) \]

IF (SURF(L,I) .LT. 0.0) SURF(L,I) = 0.0

C-HSTX---SCENE SHADOWS--------------------------------------------------------
C
*** ADD THE INHERENT RADIANCE DUE TO THE AVERAGE OF THE U
DOWNWARD DIFFUSE TERMS UNLESS THE UNIT NORMAL IS VERT
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-1---------------------------------------------
C COMPUTE AVERAGE COMPONENTS OF DIFFUSE RADIANCE OVER ALL CLOUD
C SITUATIONS AND TOTAL RADIANCE FOR TARGET IN DIRECT LIGHT AND IN
C CLOUD SHADOW. RE-DIMENSIONED ARRAYS SURF AND SURFO TO ACCOUNT FOR
C CLOUD CONTRIBUTION.
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-3---------------------------------------------
C RE-DIMENSIONED ARRAY TN TO ACCOUNT FOR CLOUD CONTRIBUTION.
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-4---------------------------------------------
C STORE DIRECTIONALLY DEPENDENT DIFFUSE, DIRECTIONALLY INDEPENDENT
C DIFFUSE, AND DIRECT RADIANCE IN TEMPORARY VARIABLES FOR CORRECT
C COMPUTATION OF AVERAGE DIFFUSE RADIANCE FOR PARTICULAR CLOUD
C SITUATION. USE TEMPORARY VARIABLES IN CALLS TO PCDIF.
C-HSTX---SCENE SHADOWS---ECR # HSTX-8-5---------------------------------------------
C MOVE TEMPORARY VARIABLES AND PCDIF CALLS TO DIFUSE SUBROUTINE.
C ELIMINATE DEPENDENCE ON THTARG. MODIFY DOWNWARD DIFFUSE EQUATION
C TO MATCH EOTDA.

IF (1.EQ.MLOOP) THEN
CALL DIFUSE (L,NLOCT,SZA,AVEIO,AVEI1,CORR)
C
IF(THTARG(M) .EQ. 0.0) THEN
SURFO(L,1) = SURF(L,1) + RTARG(M) * 
(F(NLOCT) / PI + ADDIFF)
SURFO(L,2) = SURF(L,MLOOP) + RTARG(M) * 
(F(NLOCT) / PI + ADDIFF)
SURFO(L,1) = SURF(L,1) + RTARG(M) * 
(AVEIO + 2.0 / 3.0 * AVEI1 + CORR)
SURFO(L,2) = SURF(L,MLOOP) + RTARG(M) * 
(AVEIO + 2.0 / 3.0 * AVEI1 + CORR)
ELSE
SURFO(L,1) = SURF(L,1) + RTARG(M) * (AVEIO + 
ADDIFF / 2.0)

C-91
SURF0(L,2) = SURF(L,MLOOP) + RTARG(M) * (AVEIO + ADDIFF / 2.0)

ENDIF

WRITE(*,*)'SURF DIR + DIF IS ',SURF0(L,1),SURF0(L,2)
WRITE(*,*)'IO AND ADDIFF ARE ',AVEIO,ADDIFF

*** MULTIPy BY THE TRANSMITTANCE BETWEEN THE TARGET AND OBSERVER

WRITE(*,*)'PTHRD AND SURFO IN CNTRST ',PTHRD(1),SURFO(1)
SURF(L,1) = SURF0(L,1) * TN(NP+2,II)
SURF(L,2) = SURF0(L,2) * TN(NP+2,II)
ENDIF

ELSE IF THE OBSERVATION ANGLE IS NEARLY HORIZONTAL US MODIFIED PATH INTEGRATION TO FIND PATH RADIANCE

ELSE IF(THETOB(L) .GE. 85.0 .AND. THETOB(L) .LE. 95.0) THEN
WRITE(*,*)'HORIZONTAL LINE OF SIGHT'
WRITE(*,*)'PHIOB IS ',PHIOB(L)
DO 444 J = NLEV, 1, -1
IF(Z(J) .LE. ZOB(N)) THEN
NLOCO = J - 1
ENDIF
IF(Z(J) .LE. ZTARG(M)) THEN
NLOCT = J - 1
ENDIF
444 CONTINUE
NP = NLOCT - NLOCO

DIST = (ZOB(N) - ZTARG(M))**2 + (YOB(N) - YTARG(M))**2 + (XOB(N) - XTARG(M))**2
DIST = SQRT(DIST)
DEFAC = (1.0 - A(NLOCO) * G(NLOCO) * G(NLOCO))
DEFAC = 1.0
SUM = DIST * BETA(NLOCO) * DEFAC

*** ADD TARGET TRANSMITTED RADIANCES
FIRST CALCULATE THE INHERENT RADIANCES DUE TO THE DIRECT BEAM AND FORWARD SCATTERED COMPONENT

TAUTGP = TAUP(NLOCT) - (ZTARG(M) - Z(NLOCT+1))
1 * BETA(NLOCT) * (1.0 - A(NLOCT) * G(NLOCT) * G(NLOCT))

C-92
C-HSTX---SCENE SHADOWS---ECR HSTX-8-1-----------------------------------
C RE-DIMENSIONED ARRAY SURF TO ACCOUNT FOR CLOUD CONTRIBUTION.
C-HSTX---SCENE SHADOWS---ECR HSTX-8-5-----------------------------------
C USE DELTA-EDDINGTON OPTICAL DEPTH, RATHER THAN REGULAR OPTICAL
C OPTICAL DEPTH, IN DIRECT RADIANCE CALCULATION. ELIMINATE REGULAR
C OPTICAL DEPTH AND ADDIFF VARIABLE FROM CALCULATION. SCALED OPTICAL
C DEPTH INCLUDES FORWARD SCATTERING COMPONENT.
C-HSTX---SCENE SHADOWS---ECR HSTX-8-1-------------------------------

\[\text{TAUTRG} = \tau(nloct) - (z_{\text{TARG}} - z(nloct+1))\]
\[\text{ADDIFF} = \text{FNOT} \times \text{UNOT} \times \left(\exp(-\text{TAUTRG} / \text{UNOT}) - \exp(-\text{TAUTGP} / \text{UNOT})\right)\]
\[\text{SURF}(L, I) = \text{FNOT} \times \text{CTARG} \times \text{RTARG} \times \left(\exp(-\text{TAUTGP} / \text{UNOT})\right)\]

CP WRITE(*,*)'HORIZONTAL LOS SURF DIR IS ',SURF(L,I)
IF (SURF(L,I) .LT. 0.0) SURF(L,I) = 0.0
C-HSTX---SCENE SHADOWS----------------------------------------------------------------
C*** ADD THE INHERENT RADIANCE DUE TO THE AVERAGE OF THE UPWARD AND DOWNWARD DIFFUSE TERMS UNLESS THE UNIT NORMAL IS VERTICAL
C-HSTX---SCENE SHADOWS---ECR HSTX-8-1-----------------------------------
C COMPUTE AVERAGE COMPONENTS OF DIFFUSE RADIANCE OVER ALL CLOUD SITUATIONS AND TOTAL RADIANCE FOR TARGET IN DIRECT LIGHT AND IN CLOUD SHADOW. RE-DIMENSIONED ARRAYS SURF AND SURFO TO ACCOUNT FOR CLOUD CONTRIBUTION.
C-HSTX---SCENE SHADOWS---ECR HSTX-8-3-----------------------------------
C RE-DIMENSIONED ARRAY TN TO ACCOUNT FOR CLOUD CONTRIBUTION.
C-HSTX---SCENE SHADOWS---ECR HSTX-8-4-----------------------------------
C STORE DIRECTIONALLY DEPENDENT DIFFUSE, DIRECTIONALLY INDEPENDENT DIFFUSE, AND DIRECT RADIANCE IN TEMPORARY VARIABLES FOR CORRECT COMPUTATION OF AVERAGE DIFFUSE RADIANCE FOR PARTICULAR CLOUD SITUATION. USE TEMPORARY VARIABLES IN CALLS TO PCDIF.
C-HSTX---SCENE SHADOWS---ECR HSTX-8-5-----------------------------------
C MOVE TEMPORARY VARIABLES AND PCDIF CALLS TO DIFUSE SUBROUTINE.
C ELIMINATE DEPENDENCE ON THTARG. MODIFY HORIZONTAL DIFFUSE EQUATION TO BE AVERAGE OF UPWARD AND DOWNWARD. CORRECT TRANSMITTANCE TO BE TN(1,1) AND TN(1,2) FOR HORIZONTAL LINE-OF-SIGHT, RATHER THAN TN(NP+2,1) AND TN(NP+2,2).

IF (I.EQ.MLOOP) THEN
CALL DIFUSE (L,NLOCT,SZA,AVEIO,AVEI1,CORR)
C
IF (THTARG .EQ. 0.0) THEN
\[\text{SURFO}(L, 1) = \text{SURF}(L, 1) + \text{RTARG} \times \left(\text{AVEIO} + \text{ADDIFF} / 2.0\right)\]
\[\text{SURFO}(L, 2) = \text{SURF}(L, MLOOP) + \text{RTARG} \times \left(\text{AVEIO} + \text{ADDIFF} / 2.0\right)\]
ELSE
\[\text{SURFO}(L, 1) = \text{SURF}(L, 1) + \text{RTARG} \times \left(\text{AVEIO} + \text{ADDIFF} / 2.0\right)\]
\[\text{SURFO}(L, 2) = \text{SURF}(L, MLOOP) + \text{RTARG} \times \left(\text{AVEIO} + \text{ADDIFF} / 2.0\right)\]
ENDIF

CP WRITE(*,*)'SURF DIR + DIF IS ',SURFO(L,1),SURFO(L,2)
CP WRITE(*,*)'I0, ADDIFF, AND F ARE ',AVEIO,
CP & ADDIFF, F(NLOCT)
C
C*** MULTIPLY BY THE TRANSMITTANCE BETWEEN THE TARGET AND OBSERVER

C-93
C CP WRITE(*,*)'PTHRD AND SURFO IN CNTRST', PTHRD(1,1), SURFO(1,1)
SURF(L,1) = SURFO(L,1) * TN(1,1)
SURF(L,2) = SURFO(L,2) * TN(1,2)
ENDIF
C-HSTX---SCENE SHADOWS-------------------------------------
C ELSE
IF(IOFILE.EQ.0 .OR. IOFILE.EQ.2) THEN
WRITE(IOOUT,*) ' OBSERVATION ANGLE OUT OF RANGE
ENDIF
IF(IOFILE.EQ.1 .OR. IOFILE.EQ.2) THEN
WRITE(NDIRTU,*) ' OBSERVATION ANGLE OUT OF RANGE
ENDIF
ENDIF
333 CONTINUE
222 CONTINUE
RETURN
END
C C <END OF UNIT: CONTRAST>
C C-94
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: GETCLD          Old Date: New
File Name: GETCLD.FOR          New Date: 8/17/93

Implemented By: Don Hamann

Reason for Revision: New routine to set the atmospheric data for the cloud layers found.

Description of Revision: If there are three cloud layers, call COMBIN to combine the middle and high layers. Define the cloud base height, fraction, extinction coefficient, and asymmetry parameter for the remaining cloud layers. Find the atmospheric layer containing each cloud. Set the number of cloud situations.

Notes: It is assumed that the sensor is not in a cloud layer. It is also assumed that, for this routine to be called, there is at least one cloud layer.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: GETCLD          Old Date: 8/17/93
File Name: GETCLD.FOR          New Date: 8/27/93

Implemented By: Don Hamann

Reason for Revision: To generate values for the probability of a cloud-free path through the cloud layers and the probability that the target scene is in direct light.

Description of Revision: Added variable PSCLD and array PCF(2) to the CLOUD COMMON block. Added the GEOM COMMON block to access the variable THETO and the BCONST COMMON block to access the variable DGTRD, which are both used to compute the probability.

Methodology based on Allen and Malick (1983).

Notes: Assume that the target is below all input cloud layers.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: GETCLD            Old Date: 8/27/93
File Name: GETCLD.FOR            New Date: 9/14/93

Implemented By: Michael Oberlatz

Reason for Revision: The unit conversion for the cloud base heights from feet to kilometers is no longer needed, since the heights are input in kilometers rather than feet.

Description of Revision: The unit conversion is commented out.

Notes: ________________________________

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-97
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: GETCLD               Old Date: 9/14/93
File Name: GETCLD.FOR               New Date: 9/15/93

Implemented By: Melanie J. Gouveia

Reason for Revision: Fix the error that, for clear sky cases, gave 100.0 for the probability that the target is in cloud shadow. Also fix the error in the determination of the number of cloud situations.

Description of Revision: Remove the initialization of the probability that the target is in direct light to 100.0; this is now done in STG. Set MLOOP = 1 for 2 clear layers, 1 overcast and 1 clear layer, and 2 overcast layers. Set MLOOP = 2 for 1 partly cloudy layer and 1 clear, and 1 partly cloudy layer and 1 overcast. Set MLOOP = 4 for 2 partly cloudy layers.

Notes:  

As appropriate, attach the following:
1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: GETCLD              Old Date: 9/15/93
File Name: GETCLD.FOR              New Date: 9/17/93

Implemented By: Don Hamann

Reason for Revision: In order to prevent uninitialized data from corrupting the proper execution of the program, all cloud shadow parameters are reset. Added flag to keep track of the cloud situation.

Description of Revision: Added variable ICLDF to CLOUD COMMON block to maintain information on cloud situation. Set ICLDF, depending on number of clouds and cloud fractions. Reset CLDBTA, CLDG, LYRCLD, PCF, cloud fraction, base height, thickness, and type as necessary.

Notes: ICLDF flag = 1 for two clear layers, 2 for one overcast layer, 3 for two overcast layers, 4 for one clear layer and one partly cloudy, 5 for one overcast layer and one partly cloudy, and 6 for two partly cloudy layers.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: GETCLD  Old Date: 9/17/93
File Name: GETCLD.FOR  New Date: 10/20/93

Implemented By: Don Hamann

Reason for Revision: Cloud information is reorganized in GETCLD to facilitate more efficient processing. The cloud thickness and cloud type variables were not properly assigned during the reorganization process in GETCLD.

Description of Revision: Reset the array values for cloud type, ITY, and cloud thickness, THK, according to an array of cloud indices, INXCLD.

Notes: As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
**NAME:** GETCLD

**TYPE:** FORTRAN SUBROUTINE

**FILENAME:** GETCLD.FOR

**DESCRIPTION:**
GETCLD defines the number of cloud layers, asymmetry parameters, and the extinction coefficients for the cloud layers found. GETCLD determines the probability of a cloud free path. GETCLD also calls COMBIN to combine three cloud layer properties into two cloud layer properties if there are more than two cloud layers. It is assumed that the sensor is not in a cloud layer.

**INPUT**

**COMMON BLOCK VARIABLES:**

- / CLOUD / ITY1, ITY2, ITY3: Cloud types for each cloud layer.
- / CLOUD / CF1, CF2, CF3: Cloud cover fractions for each cloud.
- / CLOUD / ZC1, ZC2, ZC3: Cloud base heights for each cloud.
- / CLOUD / THK1, THK2, THK3: Cloud thickness for each cloud.
- / GEOM / THETO: Zenith angle of illumination source (deg).
- / BCONST/ DGTRD: Degrees to radians conversion factor.

**OUTPUT**

**COMMON BLOCK VARIABLES:**

- / CLOUD / ICLDF: Cloud situation index.
- / CLOUD / NUMCLD: Number of cloud layers.
- / CLOUD / CLDG: Array of cloud asymmetry parameters.
- / CLOUD / LYRCLD: Array of layer position of clouds.
- / CLOUD / CF1, CF2: High and low cloud fractions.
- / CLOUD / ZC1, ZC2: High and low cloud base heights.
- / CLOUD / ITY1, ITY2: High and low cloud types.
- / CLOUD / PCF(2): High and low cloud-free path probability.
- / CLOUD / PSCLD: Probability that target is in direct light.

**CALLED ROUTINES:**

COMBIN - (SUBROUTINE) COMBINES MIDDLE AND HIGH CLOUDS INTO ONE CLOUD LAYER.

**PARAMETER**

**CALLING SEQUENCE:**

GETCLD

**HISTORY:**

CREATED AUGUST, 1993. HUGHES STX CORPORATION.

**List of variables:**

- AMT: Cloud fraction for a particular layer. (0.0 - 1.0)
- CB: Array of cloud extinction coefficients for five cloud types.
- CF: Array of cloud fractions. (fraction 0-1)
- I = 1 for upper cloud layer.
- I = 2 for lower cloud layer.
- CG: Array of cloud asymmetry parameters for five cloud types.
CLDBTA Array of extinction coefficients.
I = 1 for upper cloud layer
I = 2 for lower cloud layer.

CLDG Array of cloud asymmetry parameters.
I = 1 for upper cloud layer.
I = 2 for lower cloud layer.

CN Intermediate variable.

DGTRD Degrees to radians conversion factor.

ICLD Cloud index.

ICLDF Cloud situation index.
  ICLDF = 1 two clear layers
  ICLDF = 2 one overcast layer
  ICLDF = 3 two overcast layers
  ICLDF = 4 one clear layer, one partly cloudy layer
  ICLDF = 5 one overcast layer, one partly cloudy layer
  ICLDF = 6 two partly cloudy layers

ILR Array of cloud types. (1-5)
I = 1 for upper cloud layer.
I = 2 for lower cloud layer.

ILR = 1 Thin Ci (high)
ILR = 2 Thick Ci (high)
ILR = 3 As/Sc (middle)
ILR = 4 St/Sc (low)
ILR = 5 Cu/Cb (low)

INXCLD Array of cloud indices.
I = 1 for upper cloud layer
I = 2 for lower cloud layer.

LYRCLD Array of layer position of clouds.
I = 1 for upper cloud layer
I = 2 for lower cloud layer.

N Cloud layer index.

NUMCLD Number of cloud layers.

PN Intermediate variable.

PCF Array of probability of cloud-free path. (0.0 - 1.0)
I = 1 for upper cloud layer.
I = 2 for lower cloud layer.

PSCLD Probability of target in direct light. (0.0 - 100.0)

THETO Solar zenith angle. (degrees)

Z Array of atmospheric layer heights. (1000 ft.)

ZC Array of cloud base heights. (km)
I = 1 for upper cloud layer
I = 2 for lower cloud layer.

ZCLOUD Cloud base height. (km)

SUBROUTINE GETCLD
C-HSTX---SCENE SHADOWS----------------------------------------------------------
C COMMON/ATMOS/PRES(21),Z(21),RHO(21),NLEV,NLAY
C DIMENSION CG(5),CB(5),INXCLD(2),CF(3),ZC(3),ITY(3),THK(3)
C LOGICAL SET(2) / .FALSE., .FALSE. / 
C SET COMMON VALUES TO ARRAY NOTATION FOR CONVENIENT MANIPULATION
C EQUIVALENCE ( ZC1, ZC(1) ), ( CF1, CF(1) ), 
+ ( ITY1, ITY(1) ), ( THK1, THK(1) )
C EQUIVALENCE ( ZC2, ZC(2) ), ( CF2, CF(2) ),
+ ( ITY2, ITY(2) ), ( THK2, THK(2) )
C EQUIVALENCE ( ZC3, ZC(3) ), ( CF3, CF(3) ),
+ ( ITY3, ITY(3) ), ( THK3, THK(3) )
C
C EXTINCTION COEFFICIENTS FOR THE FIVE CLOUD TYPES
C DATA CB/1.40, 2.40, 8.80, 2.80, 8.13/
C ASYMMETRY PARAMETERS FOR THE FIVE CLOUD TYPES
C DATA CG/0.85, 0.85, 0.85, 0.85, 0.85/
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-9-2--------------------------------------
C INITIALIZE PROBABILITY THAT TARGET IS IN DIRECT LIGHT (NO CLOUDS).
C-HSTX---SCENE SHADOWS---ECR # HSTX-9-4--------------------------------------
C REMOVE INITIALIZATION FOR PROBABILITY THAT TARGET IS IN DIRECT LIGHT.
C THIS IS NOW DONE IN STG.
C PSCLD = 100.
C-HSTX---SCENE SHADOWS----------------------------------------------------------
C INITIALIZE CLOUD OUTPUTS BASED ON NUMBER OF CLOUDS FOUND.
C IF ( NUMCLD .EQ. 3 .OR. NUMCLD .EQ. 2 ) THEN
C IF NUMBER OF CLOUD LAYERS IS THREE COMBINE THREE CLOUD LAYERS INTO TWO CLOUD LAYERS.
C IF ( NUMCLD .EQ. 3 ) THEN
C NOTE: SENSOR IS ASSUMED TO BE BELOW MIDDLE CLOUDS
C COMBINE HIGH AND MIDDLE CLOUDS
C ILO = 2
C HI = 1
C CFLO = CF(ILO)
C TKLO = THK(ILO)
C BALO = CB(ITY(ILO))
C CGLO = CG(ITY(ILO))
C HTLO = ZC(ILO)
C CFHI = CF(IHI)
C TKHI = THK(IHI)
C BAHI = CB(ITY(IHI))
C CGHI = CG(ITY(IHI))
C CALL COMBIN(CFLO, TKLO, BALO, CGLO, HTLO, 
+ CFHI, TKHI, BAHI, CGHI, 
+ CFNEW, TKNEW, BANEW, CGNEW, HTNEW, NUMCLD )
C CLDBTA(ILO) = BANEW
CLDG(ILO) = CGNEW
THK(ILO) = TKNEW
CF(ILO) = CFNEW
CF(IHI) = CFHI

INDICATE CLDBTA AND CLDG VALUES HAVE BEEN COMPUTED.

SET(ILO) = .TRUE.

END IF

SET LAYER INDICES FOR CLOUD PROPERTIES AT THE PROPER CLOUD LAYER

IF (CF3 .GT. 0.0) INXCLD(2) = 3
IF (CF1 .GT. 0.0) INXCLD(1) = 1
IF (CF2 .GT. 0.0 .AND. CF3 .GT. 0.0) INXCLD(1) = 2
IF (CF2 .GT. 0.0 .AND. CF1 .GT. 0.0) INXCLD(2) = 2

ELSE IF ( NUMCLD .EQ. 1 ) THEN

SET LAYER INDICES FOR CLOUD PROPERTIES AT THE PROPER CLOUD LAYER

DO 10 ICLD = 1,3
   IF ( CF(ICLD) .GT. 0.0 ) INXCLD(1) = ICLD
10 CONTINUE

ELSE

ERROR IN NUMBER OF CLOUD LEVELS DETECTED.
RETURN

END IF

LOOP ON NUMBER OF CLOUD LAYERS

DO 30 ICLD = 1, NUMCLD

DEFINE CLOUD BOTTOM.

ZC(ICLD) = ZC(INXCLD(ICLD))

DEFINE THE CLOUD FRACTION.

CF(ICLD) = CF(INXCLD(ICLD))

C-HSTX---SCENE SHADOWS---ECR # HSTX-9-6-----------------------------------
C ASSIGN THE THICKNESS AND CLOUD TYPE ACCORDING TO THE CLOUD INDEX
C VARIABLE.

C DEFINE THE CLOUD THICKNESS

THK(ICLD) = THK(INXCLD(ICLD))

C DEFINE THE CLOUD TYPE

ITY(ICLD) = ITY(INXCLD(ICLD))

C-HSTX---SCENE SHADOWS-----------------------------------------------------
C C-HSTX---SCENE SHADOWS---ECR # HSTX-9-2-----------------------------------
C COMPUTE PROBABILITY OF A CLOUD-FREE SOLAR PATH AS EXPRESSED BY ALLEN
C AND MALICK, 1983. ALSO CALCULATE PROBABILITY OF TARGET IN DIRECT LIGHT.

AMT = CF(ICLD)
PN = 1.0 - AMT * (1.0 + 3.0 * AMT) / 4.0
CN = 0.55 - AMT / 2.0

C-104
PCF(ICLD) = PN**(1.0 + CN * TAN(THETO * DGTRD))
PSCLD = PSCLD * PCF(ICLD)

C-HSTX---SCENE SHADOWS-----------------------------------------------
C DEFINE THE CLOUD EXTINCTION COEFFICIENT UNLESS PREVIOUSLY SET.
C
IF ( .NOT. SET(ICLD) ) CLDBTA(ICLD) = CB(ITY(INXCLD(ICLD))))

C DEFINE THE CLOUD ASYMMETRY PARAMETER UNLESS PREVIOUSLY SET.
C
IF ( .NOT. SET(ICLD) ) CLDG(ICLD) = CG(ITY(INXCLD(ICLD))))

C DETERMINE THE ATMOSPHERIC LAYER IN WHICH THE CLOUD IS FOUND.
C NOTE: LOGIC MIRRORS TARGAC CODE PREVIOUSLY CONTAINED IN ROUTINE STG.
C
IF ( CF(ICLD) .GT. 0.0 ) THEN

C-HSTX---SCENE SHADOWS---ECR # HSTX-9-3-----------------------------------
C REMOVED THE UNIT CONVERSION SINCE THE VALUE IS ALREADY ENTERED IN KM
C
ZCLOUD = ZC(ICLD) * 0.0003048
ZC = ZC(ICLD)

C-HSTX---SCENE SHADOWS-----------------------------------------------
C ELSE
ZCLOUD = 40.0
ENDIF
C
DO 20 N = 1, NLEV
IF ( ZCLOUD .LT. Z(N) ) LYRCLD(ICLD) = N
20 CONTINUE
C
30 CONTINUE
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-9-4-----------------------------------
C CORRECT DETERMINATION OF THE NUMBER OF CLOUD SITUATIONS.
C
MLOOP = 1
One iteration of the loop for the conditions:
both layers clear
one layer overcast, one layer clear
both layers overcast

MLOOP = 2
Two iterations of the loop for the conditions:
one partly cloudy layer, one clear layer
one partly cloudy layer, one overcast layer

First iteration for:
two clear layers
one overcast layer, one clear layer
Second iteration for:
one clear layer, one overcast layer
two overcast layers

MLOOP = 4
Four iterations of the loop for the condition:
two partly cloudy layers

First iteration for:
two clear layers
Second iteration for:
high cloud layer only
Third iteration for:
low cloud layer only
Fourth iteration for:
both cloud layers

C-HSTX---SCENE SHADOWS---ECR # HSTX-9-5-----------------------------------
C DETERMINE AND SAVE THE CLOUD SITUATION FLAG.

C-105
IF ( NUMCLD .EQ. 1 ) THEN
  IF ( CF1 .EQ. 1.0 .OR. CF2 .EQ. 1.0 ) THEN
    MLOOP = 1
    ICLDF = 2
  ELSE
    MLOOP = 2
    ICLDF = 4
  END IF
ELSE IF ( NUMCLD .EQ. 2 ) THEN
  IF ( CF1 .EQ. 1.0 .AND. CF2 .EQ. 1.0 ) THEN
    MLOOP = 1
    ICLDF = 3
  ELSE IF ( CF1 .NE. 1.0 .AND. CF2 .NE. 1.0 ) THEN
    MLOOP = 4
    ICLDF = 6
  ELSE
    MLOOP = 2
    ICLDF = 5
  END IF
END IF

C-HSTX---SCENE SHADOWS-----------------------------------------------
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-9--5-------------------------------
C ZERO OUT VALUES FOR CLEAR LAYERS.
DO 40 ICLD = NUMCLD+1, 3
  CF(ICLD) = 0.0
  ZC(ICLD) = 0.0
  THK(ICLD) = 0.0
  ITY(ICLD) = 0
  IF ( ICLD .EQ. 2 ) THEN
    CLDBTA(ICLD) = 0.0
    CLDG(ICLD) = 0.0
    LYRCLD(ICLD) = 0
    PCF(ICLD) = 1.0
  END IF
40 CONTINUE
C-HSTX---SCENE SHADOWS-----------------------------------------------
C
RETURN
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: PCDIF               Old Date: New
File Name: PCDIF.FOR               New Date: 8/17/93

Implemented By: Dan DeBenedictis

Reason for Revision: New routine to compute average partly cloudy diffuse radiance or correction for forward scattering.

Description of Revision: Computes radiance as a weighted average of up to four possible cloud situations: two clear layers, upper overcast and lower clear, upper clear and lower overcast, and two overcast layers. Methodology based on Hering and Johnson (1984).

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-107
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: PCDIF       Old Date: 8/17/93
File Name: PCDIF.FOR       New Date: 9/17/93

Implemented By: Don Hamann

Reason for Revision: Array of direct radiance values entering subroutine was changed to a one-dimensional array. This was done to correctly align the stored radiance values for the cloud situation currently being examined.

Description of Revision: Re-dimensioned two-dimensional array, HS, to a one-dimensional array containing four values for possible cloud situations.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
C-HSTX---SCENE SHADOWS---ECR  # HSTX-10-1-------------------------------
C NEW ROUTINE TO COMPUTE AVERAGE PARTLY CLOUDY DIFFUSE RADIANCE.
C-HSTX---SCENE SHADOWS-----------------------------------------------
C******************************************************************************
C<BEGIN>
C<IDENTIFICATION>  NAME:  PCDIF
C TYPE:  FORTRAN SUBROUTINE
C FILENAME:  PCDIF.FOR
C******************************************************************************
C<DESCRIPTION>
C Computes partly cloudy radiance according to the sky condition.
C The algorithms allows for up to 2 cloud layers. This routine
C was duplicated from Hering's FASCAT 2, CLDFR. For a description
C of the equations, see Hering's final report, FASCAT, 1984.
C******************************************************************************
C<INPUT>
C COMMON BLOCK VARIABLES:
C NONE
C******************************************************************************
C<OUTPUT>
C COMMON BLOCK VARIABLES:
C NONE
C******************************************************************************
C<CALLED ROUTINES>
C NONE
C******************************************************************************
C<PARAMETER>
C CALLING SEQUENCE:
C PCDIF (CLR, A, B, AB, ANS, IOPT, PCF, ZNILL, NUMCLD, CF1, CF2, HS)
C INPUT:
C CLR (REAL)  Radiance, clear sky.
C A (REAL)  Radiance, top cloud.
C B (REAL)  Radiance, bot cloud.
C AB (REAL)  Radiance, two clouds.
C IOPT (INTEGER)  Flag indicates above or below cloud.
C PCF (REAL)  Probability of cloud free path.
C ZNILL (REAL)  Zenith angle of illumination source.
C NUMCLD (INTEGER)  Number of clouds (1 or 2)
C CF1 (REAL)  Cloud fraction of the top cloud layer.
C CF2 (REAL)  Cloud fraction of the lower cloud layer.
C HS(M) (REAL)  Direct radiance for the M-th cloud situation.
C OUTPUT:
C ANS (REAL)  Partly cloudy radiance.
C******************************************************************************
C<HISTORY>
C CREATED AUGUST, 1993. HUGHES STX CORPORATION.
C******************************************************************************
C<END>
C******************************************************************************
C List of variables:
C A  Radiance, top cloud.
C AB  Radiance, two clouds.
C ABLF  Contribution for top and bot cloud. (w/m**2/SR/um)
C ABLFA  Contribution for top and bot cloud, ref pt below cloud.
C (w/m**2/SR/um)
C ALF  Contribution for top cloud. (w/m**2/SR/um)
C AMT(I)  Cloud amount of the I-th level ( 1 = top).
C ANS  Partly cloudy radiance.
C ATRANS  Transmittance of the top level.
C ATRAN1  Transmittance of the top level minus 1.
C B  Radiance, bottom cloud.
C BLF  Contribution for bot cloud. (w/m**2/SR/um)
C BLFA  Contribution for bot cloud ref pt below cloud.
C (w/m**2/SR/um)

C-109
SUBROUTINE PCDIF(CLR, A, B, AB, ANS, IOPT, PCF, ZNULL, NUMCLD, CF1, CF2, HS)

C-HSTX---SCENE SHADOWS---ECR # HSTX-10-2---------------------------------

CORRECT HS ARRAY TO BE JUST ONE-DIMENSIONAL.

DIMENSION AMT(2), PCF(2), HS(4)

C-HSTX---SCENE SHADOWS----------------------------------------------------

DEGRAD = .01745329
CTHES = COS(ZNULL * DEGRAD)
IF (NUMCLD.LE.1) B = 0.0
AMT(1) = CF1
AMT(2) = CF2
ATRANS = 1.43 * AMT(1) - 1.21 * CTHES * AMT(1) - 2.0 * AMT(1)**2 $ + 1.21 * CTHES * AMT(1)**2 + 1.57 * AMT(1)**3
BTRANS = 1.43 * AMT(2) - 1.21 * CTHES * AMT(2) - 2.0 * AMT(2)**2 $ + 1.21 * CTHES * AMT(2)**2 + 1.57 * AMT(2)**3
ATRAN1 = 1.0 - ATRANS
BTRAN1 = 1.0 - BTRANS
IF (IOPT.EQ.1) THEN
  CL = ATRAN1 * BTRAN1 * CLR
  ALF = ATRANS * BTRAN1 * A
  BLF = BTRANS * ATRAN1 * B
  ABLF = ATRANS * BTRANS * AB
  ANS = CL + ALF + BLF + ABLF
ELSE
  C-HSTX---SCENE SHADOWS---ECR # HSTX-10-2---------------------------------

CORRECT HS ARRAY TO BE JUST ONE-DIMENSIONAL.

CLA = ((ATRAN1 * BTRAN1) - PCF(1) * PCF(2)) * HS(1) + CTHES
ALFA = ((ATRANS * BTRAN1) - PCF(2) * (1.0 - PCF(1))) * HS(2) + CTHES
BLFA = ((BTRANS * ATRAN1) - PCF(1) * (1.0 - PCF(2))) * HS(3) + CTHES
ABLFA = ((ATRANS * BTRANS) - (1.0 - PCF(1)) * (1.0 - PCF(2))) * HS(4) + CTHES
ANS = CLA + ALFA + BLFA + ABLFA

C-HSTX---SCENE SHADOWS----------------------------------------------------

ENDIF

RETURN
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: GETDAT          Old Date: 3/19/93
File Name: FINDR.FOR          New Date: 8/17/93

Implemented By: Don Hamann

Reason for Revision: To support partly cloudy radiance computation by creating additional storage locations for possible cloud situations.

Description of Revision: Re-dimensioned the following variables in RADIA COMMON block: I0(20), I1(20) => I0(20,4), I1(20,4)
SURF(30), SURF0(30) => SURF(30,4), SURF0(30,2)
F(20), PTHR(30) => F(20,4), PTHR(30,2)
ISTAR(20,30) => ISTAR(20,30,2)

Notes: 

As appropriate, attach the following:
1. Code listing with changes highlighted
2. Test records
**SUBROUTINE GETDAT**

**C** 
THIS ROUTINE INITIALIZES DATA

**C** -STX TCM2 --- ECR ASL-2-1-----------------------------------

**C** REMOVED THE COMMON BLOCKS:
- /TARDIM/ /COEFFS/ /CLOUDS/ /DATRAD/

**C** REMOVED THE DATA STATEMENTS:
- XTEMP1, YRAD1, ALFT1-ALFT3, ALFB1-ALFB29, NLYTR1, NLYBK1
- DELTR1, DEBK1

**C** REMOVED THE INITIALIZING OF THE VARIABLES IN THE COMMON BLOCK COEFFS

**C** THEY ARE NO LONGER NEEDED DUE TO THE INCORPORATION OF TCM2.

**C** -STX TCM2---------------------------------------------

**C** COMMON/ATMOS/PRES(21),Z(21),RHO(21),NLEV,NLAY

**C** HSTX --- SCENE SHADOWS --- ECR # HSTX-11-1-------------------

**C** RE-DIMENSIONED THE FOLLOWING VARIABLES TO SUPPORT PARTLY CLOUDY COMPUTATION:
- IO(20), I1(20) ==> IO(20,4), I1(20,4)
- SURF(30), SURFO(30) ==> SURF(30,4), SURFO(30,2)
- F(20), PTHRD(30) ==> F(20,4), PTHRD(30,2)
- ISTAR(20,30) ==> ISTAR(20,30,2)

**C** COMMON/RADIA/A(20),G(20),GP(20),BETAA(20),BETAR(20),
- BETA(20),K(20),P(20),ALPHA(20),BET(20),
- TAU(0:20),IO(20),I1(20),F(20),TF(20),FNOT,
- NANG,PF(180,20),ANG(180,20),I2(20),TAUP(0:20),
- ISTAR(20,30),PTHRD(30),SING(30),AP(20),PHF(30)
- ALBEDO,SURF(30,4),SURFO(30,2),TAUSTR,VIS,CAPTP(20),
- B1,B2

**C** CONVERT DATA ARRAYS

**C** DO 50 I = 1,17
- PRES(I) = PRES1(I)
- Z(I) = Z1(I)
- RHO(I) = RH01(I)
- BETAA(I) = BETAA1(I)

**C** 50 CONTINUE
- PRES(18) = PRES1(18)
- Z(18) = Z1(18)
- RHO(18) = RH01(18)
- NLEV = 18

**RETURN**

**END**

C-112
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: FINDR  Old Date: 8/17/93
File Name: FINDR.FOR  New Date: 9/14/93

Implemented By: Don Hamann

Reason for Revision: To compute and output bracketing ranges for sensors for both clear and cloud shadow conditions.

Description of Revision: Change the variable SOG to an array to store bracketing values. Add the CLOUD COMMON block to access cloud information. Output the probability that the target is in cloud shadow and information about the cloud conditions. Loop over up to 2 sky conditions to compute and output bracketing ranges.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-113
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: FINDR Old Date: 9/14/93
File Name: FINDR.FOR New Date: 9/17/93

Implemented By: Melanie J. Gouveia

Reason for Revision: Fix the error that, for thermal imager devices, printed bracketing ranges and cloud information. Also add cloud situation flag, to obtain more information than just the number of cloud situations.

Description of Revision: Run through the "bracketing ranges" loop only once for thermal imager devices. Print cloud information only for visible devices. Add the variable ICLDF to the /CLOUD/ COMMON block; this flag is 1 for 2 clear layers, 2 for 1 overcast layer and 1 clear, 3 for 2 overcast layers, 4 for 1 partly cloudy layer and 1 clear, 5 for 1 partly cloudy layer and 1 overcast, and 6 for 2 partly cloudy layers.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: FINDR  Old Date: 9/17/93
File Name: FINDR.FOR  New Date: 10/7/93

Implemented By: Michael Oberlatz

Reason for Revision: Several constant values sent to the XSCALE routine were being changed in XSCALE. This caused a run-time error for the second cycle of a multiple cycle run.

Description of Revision: Variables set to the constant values were substituted for the constant values in the calls to the XSCALE routine.

Notes: Using temporary variables to store the constant values eliminated the run-time error.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
SUBROUTINE FINDR ( ITYPE, CFLAG )

*** THIS SUBROUTINE CALCULATES ACQUISITION RANGES FOR
IMAGE INTENSIFIERS (NIGHT VISION DEVICES), FOR
DIRECT VIEW OPTICS INCLUDING THE HUMAN EYE, BINOCULARS, AND
PERISCOPE, FOR SILICON TVS, AND FOR THERMAL IMAGERS
FOR THREE FIXED LEVELS OF PROBABILITY.

A TV SYSTEMS PERFORMANCE PARAMETERS
A1 I**2 SYSTEMS PERFORMANCE PARAMETERS (1ST PARA=SYSTEM,
2ND = VARIABLE, 3RD=CONTRAST LEV., 4TH=LIGHT LEVEL)
AL ILLUMINATION (FT-CDLS)
ACK ILLUMINATION LEVEL INDICATOR
AMAG SENSOR MAGNIFICATION ARRAY
CD CLEAR DAY PARAMETERS
C APPARENT CONTRAST
Clim LOWER USABLE CONTRAST THRESHOLD FOR DEVICE TYPE
Cntrst TARGET/BACKGROUND INHERENT CONTRAST
Cim MINIMUM TARGET DIMENSION (METERS)
DepS ARRAY OF SENSOR DEPRESSION ANGLES (DEG)
Ho HEAVY OVERCAST PARAMETERS
Icrsd CROSS INDEX FOR DEVICE TYPE (INDEXED TO 5)
Ictyp INDEX FOR SPECTRAL REGION IN CALL TO CONTRAST
Icycle ACQUISITION LEVEL--RESOLVABLE CYCLES
Idev INDEX OF DEVICE TYPE: 1 FOR DVO, 2 FOR I**2, 3 FOR TV
4 FOR THERMAL IMAGERS, OR 5 FOR USER DEFINED
Ipf INDEX FOR FIXED PROBABILITY LEVELS
Ittype INDEX OF DEVICE TYPE: 1 = I**2, 2 = DVO, 3 = TV
4 = TI
Lsc INDEX FOR SENSOR NUMBER
Nsnsr NUMBER OF SENSORS OF TYPE IDEV
Oc OVERCAST DAY PARAMETERS
Pf VALUES FOR FIXED LEVELS OF PROB. OF PERFORMANCE
Pr ARRAY OF RANGES FOR FIXED PROB. LEVELS
Ps ARRAY OF PROBABILITIES
Pss PROBABILITY OF ACQUISITION AT RANGE R
R RANGE (KM)
Rc RESOLVABLE CYCLES
RnDelta DELTA-Ta COMPUTED BY TCM2
Sog SKY TO GROUND RATIO
So SUNSET OVERCAST PARAMETERS
Temps2 TARGET TEMPERATURES COMPUTED BY TCM2
Vis VISIBILITY (KM)
Wave WAVELENGTH USED IN SKY TO GROUND CALCULATION

PARAMETER(IDVOP = 1,INTENS = 2,ITELE = 3,ITHER = 4,IUSR = 5)
REAL INDEX

C-HSTX---SCENE SHADOWS---ECR # HSTX-12-1-----------------------------
C CHANGE SKY-TO-GROUND RATIO VARIABLE TO AN ARRAY IN ORDER TO SAVE
C INFORMATION FOR POSSIBLE SKY CONDITIONS. INCLUDE /CLOUD/ COMMON BLOCK
C TO ACCESS CLOUD INFORMATION.
C-HSTX---SCENE SHADOWS---ECR # HSTX-12-2-----------------------------
C ADDED VARIABLE ICLDF TO CLOUD COMMON BLOCK TO MAINTAIN INFORMATION ABOUT
C CLOUD SITUATION.
C REAL SOG
REAL SOG(2)
COMMON /CLOUD/ICLDF,NMCLD, MLOOP, CLDBTA(2),CLDG(2), LYRCLD(2),
+ THK1, THK2, THK3, ITY1, ITY2, ITY3,
+ ZC1, ZC2, ZC3, CF1, CF2, CF3, PCF(2), PSCLD
C-HSTX---SCENE SHADOWS---------------------------------------------
C COMMON/IOFILE/IOFILE

C-116
COMMON/INOUT/INTER, IRPT, EFLAG
COMMON/IOUNIT/IOIN, IOOUT, IPHFUN, IOUNIT, NDIRTU, NCLIMT, IRELH,
  KSTOR, NPLLOT, STDERR
COMMON /IBLOCK/ RECVAL(111, 7), RECUSE(21), I PTR(21), IBEGIN(22)
COMMON /CHBLCK/ RECFLD(21)
COMMON/RFUN/AL(4, 4, 4, 3), A(7), AT(5, 7), ILOOP, IAL, C, RCC3(2),
  RC3(2, 3), ICLRC, IFIRST, ALOGC,
  AA(7), CC(30), YY(30), AAMAG, NPTS
COMMON/SCALE/XTEMPC, RD, AINVHT
CP THE SCALEX COMMON BLOCK IS USED TO GET INFO FROM SGR TO
CP FINDR FOR XSCALE CALLS. PSG DEC 92
CRF 17 APR 92 UPGRADE TO XSCALE92 COMMON/SCALE/XTEMPC, RD, AINVHT
COMMON/ILLUM/AL, ILLUM, L22, ACK, ILL, IL2
C XRANG1 AND XRANG2 HAVE BEEN ADDED TO THE SINANG COMMON BLOCK
C SEE COMMENTS NEAR THE CALLS TO XSCALE FOR REASONING. RF AUG89.
COMMON/SINANG/SINANG, R, XSTRN1, XSTRN2, XRANG1, XRANG2
CRF 17 APR 92 COMMON/XSCL/ZZZ(999), BETA(999), RELH(999), NNZPTS, SLNFLG
CRF REMOVE SLNFLG FROM XSCALE COMMON BLOCK IN UPGRADE TO XSCALE92
COMMON/XSCL/ZZZ(999), BETA(999), RELH(999), NNZPTS
COMMON/ASCENE/NTARID, NEKID, ITARID, IBKID, RLATT, RLONG, ELEV
  RPCT,, GENERIC / GNCRT, GNRICB
C STX TCM2 -- ECR ASL-7-1 ---------------------------------------------
C ADD COMMON BLOCK TEMPS2
COMMON/TEMPS2/TEMPB(10), RNDELT(5)
C INCREASE THE DIMENSIONS OF THE TARGET DIMENSION ARRAYS
COMMON/TARDIM/XSIZE(22), YSIZE(22), ZSIZE(22)
C CHANGE VARIABLE DEP TO ALT
  COMMON/THRM/NRTM, TTARG, TBKG, TEMPC, DEWP, ALT, VIS, IAERO, RNRT, CEIL
  + , AINV, WINDS, ELEVA, ASP, ID, IO
C INCREASE THE DIMENSIONS OF THE TIME ARRAYS
COMMON/TIMES/NTIM, NRUNTM, TRLTOT(7), IDATE, ITIMOT, TOT,
  + , TDATE(11), JT, TIME(11)
C ADD COMMON BLOCK INTZZ
COMMON /INTZZ/ IFUN, IDEV, CLIM5, NPROB, PFF(3)
COMMON/EXTCN/BETA1, BETA2, EXTC1, EXTC2
C ADD DEPS = ARRAY OF SENSOR DEPRESSION ANGLES (DEG)
C FOR WHICH TCM2 COMPUTES TEMPERATURES
REAL DEPS(5)
C STX TCM2---------------------------------------------
CHARACTER PLACE*3, CDEV*4
DIMENSION WAVE(5), ALL(3), A6(7), AT1(7), AT2(7), AT3(7)
DIMENSION SMEX(3)
DIMENSION ICRST(5), CLIM(5)
DIMENSION PS(20, 3), PR(20, 3), PF(3), RNAME(4), NSNSR(5)
DIMENSION BETAV(999), BETA2(999), EXTC1(999), EXTC2(999)
DIMENSION EXTC(2)
INTEGER IOIN, IOOUT, IPHFUN, IOUNIT, NDIRTU, NCLIMT
INTEGER IRELH, KSTOR, STDERR, NPLLOT, IOFILE
CHARACTER CHRDEV*20, NMSND*15, AMOK*1, RNAME*11, RECFLD*4
C CHARACTER*15 NMSND
LOGICAL ILOOP, IAL, IFIRST, IMONO, IUP, IDOWN, INTER, CFLAG, EFLAG
LOGICAL RECUSE, GNRICB
CRF XSCALE92 DEFAULT VALUES START 17 APR 92
DIMENSION Q(3, 2), QAVE(2), DEEPC(3), XMEAN(3), XMODE(3),
  + WAVRFN(20), RESRFN(20)
COMPLEX UM
CRF XSCALE92 DEFAULT VALUES STOP 17 APR 92
DATA DEPS/70., 30., 10.5., 0.1/, DATA NSNSR/14, 4, 1, 5, 1/
DATA ICRST/1, 2, 3, 4, 5/
DATA WAVE/0.55, 0.65, 0.70, 10.0, 0.0/
DATA CLIM/0.020, 0.01, 0.010, 0.062, 0.0/
CUV DATA ALL/0.01, 0.001, 0.0001/
DATA A2/0.017, 0.126, 0.01384,
1.84055,0.126,0.9,0.03687,1.0,0.0,0.72074,0.07119,
2.0,0.0,0.0,0.043,0.136,0.02418,2.87864,0.136,0.454,
3.0.05507,1.50680,0.454,0.910,0.113,0.99335,0.91,1.0,0.78141,
4.0.07255,0.127,0.381,0.07332,2.74653,0.381,0.912,0.15917,1.45473,
5.0.912,1.0,0.81656,0.09212,0.0.0.0.0.0.
DATA A3/0.02,0.09,0.01374,3.37995,
1.0.09,0.25,0.031,1.91679,0.25,0.92,0.05267,1.4302,0.92,1.0,
2.0.69676,0.13897,0.04,0.2,0.02904,3.85956,0.2,0.86,0.06319,
3.2.30429,0.86,1.0,0.70616,0.17396,0.0.0.0.0,
4.0.116,0.88,0.1004,3.44559,0.88,1.0,0.78033,0.19080,
5.0,0.0,0.0,0.0,0.0.
DATA A4/0.02,0.066,0.01270,
1.2.74653,0.066,0.942,0.03878,0.88612,0.942,1.0,0.55019,
2.0.14938,0.0,0.0,0.089,0.165,0.04982,
3.3.4295,0.165,0.871,0.12779,0.73817,0.871,1.0,0.58444,0.15346,
4.0,0.0,0.0,0.288,0.405,0.2048,1.70463,0.405,
5.0.918,0.30831,0.68193,0.0,0.0,0.0,0.0,0.0.
DATA A5/0.02,0.054,0.01232,
1.7.07417,0.054,0.945,0.00946,8.34029,0.945,1.0,0.68714,
2.0.57725,0.0,0.0,0.0,0.02,0.9,0.02059,9.31636,
3.0.9,1.0,0.66963,0.72914,0.0,0.0,0.0,0.0,0.0,
4.0.1,0.88,0.07068,9.90871,0.88,1.0,
5.0.74509,0.65388,0.0,0.0,0.0,0.0,0.0,0.0,0.0.
DATA A6/38.122,3.455,2.894,3.786,1.223,0.012/,
1.3.62122,0.859897,-0.075664,0.007368,-0.004656,
1.0.000614,0.000013/-
DATA AT1/4.797984,0.938367,-0.235704,0.010507,0.012939,
1.-0.000996,-0.000203/-
DATA AT2/0.248778,0.061437,-0.018147,0.003975,-0.000452,
1.0.000321,-0.000111/
DATA PF/0.75,0.50,0.25/
CONST=0.017453293
CRF XSCALE92 DEFAULT VALUES START 17 APR 92
DO 701 IX=1,3
DECPER(IX) = 0.0
XMEAN(IX) = 0.0
XMODE(IX) = 0.0
701 CONTINUE
IWATER = 0
NBR = 0
DO 702 IX = 1,20
WAVRFN(IX) = 0.0
RESPFN(IX) = 0.0
702 CONTINUE
ALT = 0.0
CRF XSCALE92 DEFAULT VALUES STOP 17 APR 92
C
C RESTRUCTURE DATA ARRAYS
C
RGNAME(3) = 'RECOGNITION'
RGNAME(4) = 'RECOGNITION'
RGNAME(1) = 'DETECTION'
DO 800 I = 1,4
DO 800 J = 1,4
DO 800 K = 1,3
A1(2,I,J,K) = A3(I,J,K)
A1(3,I,J,K) = A4(I,J,K)
800 CONTINUE
DO 300 I = 1,7
A(I) = A6(I)
AT(I) = AT1(I)
AT(2,I) = AT1(I)
AT(3,I) = AT2(I)

C-118
AT(4,I) = AT2(I)
AT(5,I) = AT3(I)
300 CONTINUE

C SET DEVICE TYPE

45 CONTINUE
CRF THE FOLLOWING LINE IS COMMENTED OUT TO ALLOW THE USER DEFINED
CRF MODE TO OPERATE PROPERLY
CRF IDEV = ICRST(ITYPE)
C--STX TCM2--ECR ASL-7-1 --------------------------------------------------
C FOR THE PC VERSION
C FOR MAINFRAME COMMENT OUT THESE FOUR LINES UP TO NEXT STX LABEL
C IF CFLAG IS TRUE THIS MEANS TCM2 DONE NOW JUMP TO THE OUTPUT
IF (CFLAG) THEN
  CFLAG = .FALSE.
GOTO 777
ENDIF
C--STX TCM2---------------------------------------------------------------
CRF MOVED THE FOLLOWING LINE FROM 10 LINES ABOVE; THIS ALLOWS USER
CRF DEFINED MODE TO OPERATE. 5 MAR 92
  IDEV = ICRST(ITYPE)
  IF (IDEV .EQ. IUSR) THEN
    IF (INTER) THEN
      WRITE(IOOUT,*)
      'YOU HAVE CHOSEN TO RUN THIS PORTION OF THE TARGET'
      WRITE(IOOUT,*)
      'ACQUISITION MODEL FOR A USER SPECIFIED DEVICE. IN'
      WRITE(IOOUT,*)
      'ORDER TO CALCULATE DETECTION AND RECOGNITION RANGES'
      WRITE(IOOUT,*)
      'FOR YOUR DEVICE YOU MUST SPECIFY THE PERFORMANCE'
      WRITE(IOOUT,*)'PARAMETERS FOR THE DEVICE. AS A MINIMUM YOU WILL'
      WRITE(IOOUT,*)'BE ASKED TO SUPPLY INFORMATION ABOUT THE SENSOR'
      WRITE(IOOUT,*)
      'PERFORMANCE CURVE; I.E. THE CURVE WHICH SPECIFIES'
      WRITE(IOOUT,*)'SPATIAL FREQUENCY AS A FUNCTION OF CONTRAST.'
      WRITE(IOOUT,*)'PRESS C TO CONTINUE '
      READ(IOIN,*)CKEY
      WRITE(IOOUT,*)'THE SPATIAL FREQUENCY MUST HAVE UNITS OF CYCLES'
      WRITE(IOOUT,*)'PER MILLIRADIAN. YOU MUST ALSO KNOW THE'
      WRITE(IOOUT,*)
      'WAVELENGTH REGION IN WHICH THE SENSOR IS RESPONSIVE'
      WRITE(IOOUT,*)
      'IF YOUR SENSOR PERFORMANCE FUNCTION IS OF THE FORM,'
      WRITE(IOOUT,*)'
      WHERE Y IS SPATIAL FREQUENCY IN CYCLES PER'
      WRITE(IOOUT,*)'MILLIRADIAN AND EACH TERM T(I) IS OF THE FORM '
      WRITE(IOOUT,*)'
      T(I) = A(I) * LN(C) ** I,'}
      WRITE(IOOUT,*)
      'WHERE C IS CONTRAST, THEN YOU NEED ONLY'
      WRITE(IOOUT,*)'SPECIFY THE CONSTANTS A(I), I = 0,6 WHICH'
      WRITE(IOOUT,*)' WILL BE REQUESTED LATER.'
      WRITE(IOOUT,*)'PRESS C TO CONTINUE '
      READ(IOIN,*)CKEY
      WRITE(IOOUT,*)'IF YOU DO NOT KNOW THE COEFFICIENTS'
      WRITE(IOOUT,*)'IN THE ABOVE FORM, BUT KNOW AT LEAST'
      WRITE(IOOUT,*)'FIVE POINTS ON THE CURVE, YOU MAY ENTER '
      WRITE(IOOUT,*)'THE INFORMATION IN A POINT BY POINT FORMAT.'
      WRITE(IOOUT,*)'THE FINAL FORM FOR THE INFORMATION WHICH WILL '
      WRITE(IOOUT,*)'BE ACCEPTED BY THIS MODULE, IS A SEPARATE'
      WRITE(IOOUT,*)'USER SUPPLIED FUNCTION WHICH MUST BE NAMED'

C-119
WRITE(IOOUT,*)'"USERRC". THIS FUNCTION MUST BE COMPILED AND'
WRITE(IOOUT,*)'LINKED PRIOR TO THE CONTINUED EXECUTION OF THIS'
WRITE(IOOUT,*)'MODULE.'
WRITE(IOOUT,*)'PRESS C TO CONTINUE '
READ(IOIN,*)CKEY
WRITE(IOOUT,*)'IF YOU DO NOT HAVE THE NECESSARY INFORMATION'
WRITE(IOOUT,*)'ON SENSOR PERFORMANCE IN ANY OF THE THREE FORMATS'
WRITE(IOOUT,*)'GIVEN ABOVE, YOU MAY EXIT AT THIS TIME BY SIMPLY'
WRITE(IOOUT,*)'ENTERING "0" AND "RETURN". IF YOU WISH TO CONTINUE,'
WRITE(IOOUT,*)'SPECIFY THE FORM OF THE SENSOR PERFORMANCE CURVE'
WRITE(IOOUT,*)'BY ENTERING A NUMBER TO THE LEFT OF THE LIST BELOW'
WRITE(IOOUT,*)'0. NO INFORMATION AVAILABLE -- HALT EXECUTION'
WRITE(IOOUT,*)'1. COEFFICIENTS -- A(I)'
WRITE(IOOUT,*)'2. POINT BY POINT ENTRY'
WRITE(IOOUT,*)'3. USER DEFINED ROUTINE'
READ(IOIN,*)IFUN
WRITE(IOOUT,*)IFUN
WRITE(IOOUT,*)
IF(IFUN .EQ. 0) THEN
RETURN
ELSE IF (IFUN .EQ. 1) THEN
WRITE(IOOUT,*)
WRITE(IOOUT,*)'YOU HAVE CHosen to ENTER the SENSOR PERFORMANCE'
WRITE(IOOUT,*)'IN THE FORM OF EXPANSION COEFFICIENTS.'
WRITE(IOOUT,*)
WRITE(IOOUT,*)'YOU will BE ASKED to PROVIDe SEVEN COEFFICIENTS'
WRITE(IOOUT,*)'A(0),A(1),...,A(6)'
WRITE(IOOUT,*)
DO 50 I = 1,7
WRITE(IOOUT,*)'ENTER THE COEFFICIENT FOR SUBSCRIPT ', I-1
READ(IOIN,*)AA(I)
WRITE(IOOUT,*)AA(I)
50 CONTINUE
ELSE IF(IFUN .EQ. 2) THEN
WRITE(IOOUT,*)
WRITE(IOOUT,*)
WRITE(IOOUT,*)'YOU HAVE CHosen to ENTER the SENSOR PERFORMANCE'
WRITE(IOOUT,*)'IN A POINT BY POINT FORMAT. YOU MAY ENTER FROM'
WRITE(IOOUT,*)'FROM 5 TO 30 POINTS ALONG THE CURVE. YOUR POINTS'
WRITE(IOOUT,*)'MUST BE ENTERED IN PAIRS --C(I),Y(I)--, WHERE'
WRITE(IOOUT,*)'Y(I) IS THE SPATIAL FREQUENCY IN CYCLES PER'
WRITE(IOOUT,*)'MILLIRADIAN, AND C(I) IS THE CONTRAST AT THE'
WRITE(IOOUT,*)'IPTH POINT. YOU MUST ENTER THE PAIRS SUCH THAT'
WRITE(IOOUT,*)'THE VALUES OF C(I) ARE EITHER MONOTONICALLY '
WRITE(IOOUT,*)'INCREASING OR DECREASING. THE VALUES OF C(I)'
WRITE(IOOUT,*)'MUST BE BETWEEN 0 AND 1. YOU MUST INCLUDE ONE'
WRITE(IOOUT,*)'PAIR FOR A CONTRAST OF 0.0 AND ONE PAIR FOR A'
WRITE(IOOUT,*)'CONTRAST OF 1.0. BEGIN BY ENTERING THE NUMBER OF'
WRITE(IOOUT,*)'POINTS, FROM 5 TO 30, WHICH YOU INTEND TO ENTER.'
WRITE(IOOUT,*)'YOUR FIRST CONTRAST VALUE MUST BE EITHER'
WRITE(IOOUT,*)'A 0.0 OR 1.0'
55 CONTINUE
WRITE(IOOUT,*)
WRITE(IOOUT,*)'ENTER THE NUMBER OF POINTS ON THE SENSOR'
WRITE(IOOUT,*)'PERFORMANCE CURVE'
WRITE(IOOUT,*)
READ(IOIN,*)NPTS
WRITE(IOOUT,*)NPTS
WRITE(IOOUT,*)
IF (NPTS .LT. 5 .OR. NPTS .GT. 30) THEN
   WRITE(IOOUT,*)'YOU MAY ONLY ENTER FROM 5 TO 30 POINTS'
   GO TO 55
ENDIF
IMONO = .TRUE.
IUP = .FALSE.
IDOWN = .FALSE.
DO 60 I = 1, NPTS
   WRITE(IOOUT,*)'ENTER PAIR C(I), Y(I) FOR POINT ',I
   READ(IOIN,*)CC(I),YY(I)
   WRITE(IOOUT,*)CC(I),YY(I)
   WRITE(IOOUT,*)
   IF(CC(I) .LT. 0. .OR. CC(I) .GT. 1.0) THEN
      WRITE(IOOUT,*)'YOU HAVE ENTERED A VALUE OF CONTRAST OUTSIDE'
      WRITE(IOOUT,*)'OF THE ALLOWABLE RANGE (FROM 0.0 TO 1.0)'
      WRITE(IOOUT,*)'TRY AGAIN'
      GO TO 55
   ENDIF
   IF(I .GE. 2 ) THEN
      CHOLD = CC(I-1)
      IF(CHOLD .GT. CC(I)) IDOWN = .TRUE.
      IF(CHOLD .LT. CC(I)) IUP = .TRUE.
      IMONO = .NOT. ( IUP .AND. IDOWN)
      IF (.NOT. IMONO) THEN
         WRITE(IOOUT,*)'THE CONTRAST VALUES MUST BE MONOTONIC'
         WRITE(IOOUT,*)'TRY AGAIN'
         GO TO 55
      ENDIF
   ENDIF
   ELSE IF(CC(1) .NE. 0.0 .AND. CC(1) .NE. 1.0) THEN
      WRITE(IOOUT,*)'IF THE CONTRAST ARRAY IS TO BE MONOTONIC'
      WRITE(IOOUT,*)'AND INCLUDE 0 AND 1 THEN YOUR LIST'
      WRITE(IOOUT,*)'MUST START WITH 0 OR 1. TRY AGAIN'
      GO TO 55
   ENDIF
60 CONTINUE
CD REV 8/18/89
IFUN = .TRUE.
ELSE IF( IFUN .EQ. 3) THEN
   WRITE(IOOUT,*)'YOU HAVE CHOSEN TO ENTER YOUR OWN SENSOR '
   WRITE(IOOUT,*)'PERFORMANCE CURVE VIA A USER DEFINED FUNCTION.'
   WRITE(IOOUT,*)'FOR INFORMATION ON THE NATURE OF THIS CURVE'
   WRITE(IOOUT,*)'PLEASE CONSULT THE USERS MANUAL ON TARGAC.'
   WRITE(IOOUT,*)
C-121
NOTE THAT AT THIS TIME IN THE EXECUTION YOU MUST
WRITE(IOOUT,*)'HAVE PREVIOUSLY COMPILED AND LINKED A FUNCTION'
WRITE(IOOUT,*)'NAMED "USRFUN" IN ORDER TO CONTINUE EXECUTION'
WRITE(IOOUT,*)'IF THIS IS NOT THE CASE, YOU MAY EXIT THE'
WRITE(IOOUT,*)'PROGRAM BY ENTERING A 0 TO THE PROMPT BELOW.'
WRITE(IOOUT,*)
WRITE(IOOUT,*)'IF YOU WISH TO HALT THE PROGRAM HERE ENTER "0"
WRITE(IOOUT,*)
WRITE(IOOUT,*)'AND "RETURN", TO CONTINUE ENTER "1" AND "RETURN"'
READ(IOIN,*)IHALT
WRITE(IOOUT,*)IHALT
WRITE(IOOUT,*)
IF(IHALT .EQ. 0) RETURN
ELSE
WRITE(IOOUT,*)'YOU MUST ENTER A CHOICE FROM 0 TO 3; TRY AGAIN'
GO TO 45
ENDIF
WRITE(IOOUT,*)
WRITE(IOOUT,*)'YOU MAY ALSO WANT TO ENTER A MAGNIFICATION FACTOR'
WRITE(IOOUT,*)'FOR YOUR DEVICE. IF YOU DO NOT KNOW THE'
WRITE(IOOUT,*)'MAGNIFICATION FACTOR ENTER 1.0'
READ(IOIN,*)AAMAG
WRITE(IOOUT,*)AAMAG
WRITE(IOOUT,*)
WRITE(IOOUT,*)'YOU ARE NOW ASSURED OF HAVING SENSOR PERFORMANCE'
WRITE(IOOUT,*)'INFORMATION FOR YOUR SENSOR. YOU MAY FACILITATE'
WRITE(IOOUT,*)
WRITE(IOOUT,*)'THE REMAINDER OF THE EXECUTION OF THIS PROGRAM IF'
WRITE(IOOUT,*)
WRITE(IOOUT,*)'YOU CAN CLASSIFY IT AS ONE OF THE MAIN SENSOR TYPES'
WRITE(IOOUT,*)'IE A DVO, I**2, TV, OR TI. CHOOSING ONE OF'
WRITE(IOOUT,*)
WRITE(IOOUT,*)'THE MAIN DEVICE TYPES FOR YOUR SENSOR AT THIS POINT'
WRITE(IOOUT,*)'WILL ONLY AFFECT THE SPECTRAL INFORMATION PASSED'
WRITE(IOOUT,*)'TO THE CONTRAST AND/OR THE SKY/GROUND ROUTINES'
WRITE(IOOUT,*)
CONTINUE
WRITE(IOOUT,*)'CHOOSE AN OPTION'
WRITE(IOOUT,*)
WRITE(IOOUT,*)'1. DVO'
WRITE(IOOUT,*)'2. IMAGE INTENSIFIER'
WRITE(IOOUT,*)'3. SILICON TV'
WRITE(IOOUT,*)'4. THERMAL IMAGER'
READ(IOIN,*)IDEV
WRITE(IOOUT,*)IDEV
WRITE(IOOUT,*)
IF (IDEV .LT. 1 .OR. IDEV .GT. 4) THEN
WRITE(IOOUT,*)'YOU MUST MAKE A SELECTION FROM THE LIST BELOW'
GOTO 65
ENDIF
CONTINUE
WRITE(IOOUT,*)'ENTER THE LOWER LIMIT OF CONTRAST WHICH IS'
WRITE(IOOUT,*)'CHARACTERISTIC OF YOUR SENSOR.'
READ(IOIN,*)CLIM(5)
WRITE(IOOUT,*)CLIM(5)
WRITE(IOOUT,*)
IF(CLIM(5) .LT. 0.0) THEN
WRITE(IOOUT,*)'YOUR CONTRAST LIMIT MUST BE POSITIVE'
GOTO 66
ENDIF
CONTINUE
CD
REV 8/18/89
IF (IFUN .EQ. 1) THEN
WRITE(21,61)'IFUN',FLOAT(IFUN),FLOAT(IDEV),AAMAG,CLIM(5)
WRITE(21,62)'COEF',AA(1),AA(2),AA(3),AA(4),AA(5),AA(6),AA(7)
ELSE IF (IFUN .EQ. 2) THEN
WRITE(21,61)'IFUN',FLOAT(IFUN),FLOAT(IDEV),AAMAG,CLIM(5)
WRITE(21,62)'POIN',3.0,CC(1),YY(1),CC(2),YY(2),CC(3),YY(3)

C
IF (NPTS .LT. 6.0) THEN
  INDEX = 5.0
ELSE
  INDEX = 6.0
ENDIF

C
WRITE(21,62)'POIN',INDEX,CC(4),YY(4),CC(5),YY(5),
  & CC(6),YY(6)
IF (CC(7) .GT. 0.0) THEN
  IF (NPTS .LT. 8.0) THEN
    INDEX = INDEX + 1
  ELSE IF (NPTS .GT. 8.0) THEN
    INDEX = INDEX + 3
  ELSE
    INDEX = INDEX + 2
  ENDIF
ENDIF
WRITE(21,62)'POIN',INDEX,CC(7),YY(7),CC(8),YY(8),
  & CC(9),YY(9)
ENDIF
IF (CC(10) .GT. 0.0) THEN
  IF (NPTS .LT. 11.0) THEN
    INDEX = INDEX + 1
  ELSE IF (NPTS .GT. 11.0) THEN
    INDEX = INDEX + 3
  ELSE
    INDEX = INDEX + 2
  ENDIF
ENDIF
WRITE(21,62)'POIN',INDEX,CC(10),YY(10),CC(11),YY(11),
  & CC(12),YY(12)
ENDIF
IF (CC(13) .GT. 0.0) THEN
  IF (NPTS .LT. 14.0) THEN
    INDEX = INDEX + 1
  ELSE IF (NPTS .GT. 14.0) THEN
    INDEX = INDEX + 3
  ELSE
    INDEX = INDEX + 2
  ENDIF
ENDIF
WRITE(21,62)'POIN',INDEX,CC(13),YY(13),CC(14),YY(14),
  & CC(15),YY(15)
ENDIF
IF (CC(16) .GT. 0.0) THEN
  IF (NPTS .LT. 17.0) THEN
    INDEX = INDEX + 1
  ELSE IF (NPTS .GT. 17.0) THEN
    INDEX = INDEX + 3
  ELSE
    INDEX = INDEX + 2
  ENDIF
ENDIF
WRITE(21,62)'POIN',INDEX,CC(16),YY(16),CC(17),YY(17),
  & CC(18),YY(18)
ENDIF
IF (CC(19) .GT. 0.0) THEN
  IF (NPTS .LT. 20.0) THEN
    INDEX = INDEX + 1
  ELSE IF (NPTS .GT. 20.0) THEN
    INDEX = INDEX + 3
  ELSE
    INDEX = INDEX + 2
  ENDIF
ENDIF
WRITE(21,62)'POIN',INDEX,CC(19),YY(19),CC(20),YY(20),
  & CC(21),YY(21)
ENDIF IF (CC(22) .GT. 0.0) THEN
  IF (NPTS .LT. 23.0) THEN
    INDEX = INDEX + 1
  ELSE IF (NPTS .GT. 23.0) THEN
    INDEX = INDEX + 3
  ELSE
    INDEX = INDEX + 2
  ENDF
  WRITE(21,62) 'POIN', INDEX, CC(22), YY(22), CC(23), YY(23), &
      CC(24), YY(24)
ENDIF IF (CC(25) .GT. 0.0) THEN
  IF (NPTS .LT. 26.0) THEN
    INDEX = INDEX + 1
  ELSE IF (NPTS .GT. 26.0) THEN
    INDEX = INDEX + 3
  ELSE
    INDEX = INDEX + 2
  ENDF
  WRITE(21,62) 'POIN', INDEX, CC(25), YY(25), CC(26), YY(26), &
      CC(27), YY(27)
ENDIF IF (CC(28) .GT. 0.0) THEN
  IF (NPTS .LT. 29.0) THEN
    INDEX = INDEX + 1
  ELSE IF (NPTS .GT. 29.0) THEN
    INDEX = INDEX + 3
  ELSE
    INDEX = INDEX + 2
  ENDF
  WRITE(21,62) 'POIN', INDEX, CC(28), YY(28), CC(29), YY(29), &
      CC(30), YY(30)
ENDIF ELSE IF (IFUN .EQ. 3) THEN
  WRITE(21,61) 'IFUN', FLOAT(IFUN), FLOAT(IDEV), AAMAG, CLIM(5)
ENDIF

ELSE
  IF (.NOT. RECUSE(6)) THEN
    IF (IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
      WRITE(IOOUT,*)'**** DEVICE TYPE MISSING OR OUT OF RANGE ****'
      WRITE(IOOUT,*)
    ENDIF
    IF (IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
      WRITE(NDIRTU,*)'**** DEVICE TYPE MISSING OR OUT OF RANGE ****'
      WRITE(NDIRTU,*)
    ENDIF
    EFLAG = .TRUE.
  RETURN
ENDIF

IFUN = NINT(RECVAL(6,1))
IDEV = NINT(RECVAL(6,2))
AAMAG = RECVAL(6,3)
CLIM(5) = RECVAL(6,4)
IF (IDEV .LT. 1 .OR. IDEV .GT. 4) THEN
  IF (IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
    WRITE(IOOUT,*)'**** DEVICE TYPE MISSING OR OUT OF RANGE ****'
    WRITE(IOOUT,*)
  ENDIF
  IF (IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
    WRITE(NDIRTU,*)'**** DEVICE TYPE MISSING OR OUT OF RANGE ****'
    WRITE(NDIRTU,*)
  ENDIF
  EFLAG = .TRUE.
ENDIF

C-124
RETURN
ENDIF
IF(IFUN .EQ. 1) THEN
  IF (.NOT. RECUSE(3)) THEN
    IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
      WRITE(IOOUT,*)'***** COEFFICIENT RECORD EXPECTED *****'
    ENDIF
    IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
      WRITE(NDIRTU,*)'***** COEFFICIENT RECORD EXPECTED *****'
    ENDIF
    EFLAG = .TRUE.
    RETURN
  ENDIF
ENDIF
ICTR = 1
DO 69 J = IBEGIN(21),IPTR(21)
  IEOREC = NINT(RECVAL(J,1))
  IF (IEOREC .GT. ICTR + 3) IEOREC = ICTR + 3
  DO 68 K = ICTR,IEOREC
    IVAL = 2*(MOD(K+2,3)+I)
    CC(K) = RECVAL(J,IVAL)
    YY(K) = RECVAL(J,IVAL+1)
  CONTINUE
  ICTR = K
CONTINUE
69 CONTINUE
NPTS = ICTR - 1
IDOWN = .FALSE.
IUP = .FALSE.
IMONO = .TRUE.
DO 70 I = 2,NPTS
  CHOLD = CC(I-1)
  IF (CHOLD .GT. CC(I)) IDOWN = .TRUE.
  IF (CHOLD .LT. CC(I)) IUP = .TRUE.
  IMONO = .NOT. (IUP .AND. IDOWN)
  IF (.NOT. IMONO) THEN
    WRITE(IOOUT,*)
    + '***THE CONTRAST VALUES ARE NOT MONOTONIC.'
    WRITE(IOOUT,*)'CHANGE THE POINT RECORDS AND TRY AGAIN.'
    EFLAG = .TRUE.
    RETURN
  ENDIF
CONTINUE
70 CONTINUE
ENDIF
ENDIF
CRF REV 1/27/92
CRF ADDED THE FOLLOWING LINE, CLIM(IDEV) IS USED BELOW FOR THE CONTRAST
CRF LIMIT, THE USERS LIMIT HAS BEEN ENTERED INTO CLIM(5). SO CLIM(5)

C-125
CRF MUST BE ASSIGNED TO CLIM(IDEV)
CLIM(IDEV) = CLIM(5)
CRF REV 1/27/92
ENDIF
CD REV 8/18/89
72 FORMAT(A4,6X,5E10.4)
CP72 FORMAT(A4,6X,11E10,4E10.4)
CD REV 8/18/89
IF(INTER) THEN
75 CONTINUE
WRITE(IOOUT,*) 'YOU MAY ENTER FROM 1 TO 3 LEVELS OF PROBABILITY'
WRITE(IOOUT,*) 'FOR WHICH THE DETECTION AND RECOGNITION'
WRITE(IOOUT,*) 'RANGES WILL BE CALCULATED.'
WRITE(IOOUT,*) 'ENTER THE NUMBER OF PROBABILITY LEVELS DESIRED'
READ(IOIN,*) NPROB
WRITE(IOOUT,*) NPROB
WRITE(IOOUT,*)
IF (NPROB .LT. 1 .OR. NPROB .GT. 3) THEN
76 CONTINUE
WRITE(IOOUT,*) 'THE NUMBER OF LEVELS MUST BE 1, 2, OR 3'
GO TO 75
ENDIF
CD REV 8/18/89
ELSE
NPROB = NINT(RECVAL(I,I))
CALL INTCHK(3,1,NPROB,'AQUI','1ST',3)
DO 78 I = 1, NPROB
PF(I) = RECVAL(1,I+I)
PROBY = .25*I
IF (I .EQ. 1) THEN
PLACE = '2ND'
ELSE IF (I.EQ. 2) THEN
PLACE = '3RD'
ELSE
PLACE = '4TH'
ENDIF
CALL REALCK(0.9,0.1,PF(I),'AQUI',PLACE,PROBY)
78 CONTINUE
ENDIF
C--STX TCM2 ECR ASL-7-1 -------------------------------------------
C CALL CNTRAS FOR ALL SENSORS EXCEPT THERMALS
IF(IDEV .NE. IETHER) THEN
CALL CNTRAS(IDEV,CNTRST,BKGREF)
CNTRST = ABS(CNTRST)
CNTRST = AMIN1(CNTRST,1.0)
ENDIF
C CALL THERML FOR INPUTS
IF (IDEV.EQ.IETHER) THEN
CALL THERML

C******************************************************************************
C FOR THE PC VERSION
C FOR MAINFRAME COMMENT OUT THE FOLLOWING LINES UP TO THE NEXT STX LABEL
C SAVE INPUTS FOR TCM2 AND FOR SECOND PASS OF TARGAC

WRITE(41,*)ITYPE,IDEV,NPROB,PF(1),PF(2),PF(3)
WRITE(41,*)RECVAL
WRITE(41,*)RECUSE
WRITE(41,*)IPTR
WRITE(41,*)IBEGIN
WRITE(41,28)RECFLD(11)

28 FORMAT(A4)
C IF USER DEFINED AND THERMAL SAVE INPUTS
IF (ITYPE.EQ.5) THEN
  WRITE(41,*)IFUN,AAMAG,CLIM(5),NPTS
  WRITE(41,*)AA
  WRITE(41,*)CC
  WRITE(41,*)YY
ENDIF

C STX
C FOR THE PC VERSION
C FOR MAINFRAME COMMENT OUT THE FOLLOWING LINE
RETURN
C FOR THE PC VERSION COMMENT OUT THE FOLLOWING LINE
C FOR MAINFRAME ACTIVATE THE FOLLOWING LINE
C CALL TCM2
ENDIF

C AFTER TCM2 IS DONE JUMP TO HERE AND CALL THERMB TO DISPLAY INPUTS AND
C TEMPERATURES FROM TCM2
777 IF (IDEV.EQ.ITHER) THEN
  CLIM(5)=CLIM5
  DO 2 I=1,3
    PF(I)=PFF(I)
  CONTINUE
  CALL THERMB
C STX TCM2 ASL-7-1 **********************
C CNTRST DEFINED AS THE VALUE FOR DEP=0
CNTRST = RNDELT(5)
C
IF (GNRICT) THEN
  WRITE(*,*)' ENTER THE TARGET EFFECTIVE WIDTH (M) '
  READ(*,*)YEFF
  WRITE(*,*)
  WRITE(*,*)' ENTER THE TARGET EFFECTIVE HEIGHT (M) '
  READ(*,*)XEFF
ENDIF
ENDIF
C
C STX TCM2 -----------------------------------------------
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
  WRITE(NDIRTU,105)CNTRST
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
  WRITE(IOOUT,105)CNTRST
ENDIF

105 FORMAT(' CONTRAST = ',F6.2)
CONLIM = CLIM(IDEV)
IF(ABS(CNTRST) .LT. CONLIM) THEN
  CFLAG = .TRUE.
  RETURN
ENDIF
IF(IDEV .NE. ITHEN) THEN
  IF(INTER) THEN

C-127
CONTINUE
WRITE(IOOUT,*)'INPUT TARGET MINIMUM DIMENSION IN METERS'
WRITE(IOOUT,*)'(FOR EXAMPLE 2.4 FOR HEIGHT OF TANK OR)
WRITE(IOOUT,*)' 0.5 FOR WIDTH OF MAN')
READ(IOIN,*)DIM
WRITE(IOOUT,*) DIM
WRITE(IOOUT,*)
IF(DIM .LT. 0.0) THEN
  WRITE(IOOUT,*)'THE TARGET DIMENSION MUST BE POSITIVE'
  GOTO 106
ENDIF

CD REV 8/18/89
WRITE(21,72)'AQUI',FLOAT(NPROB),PF(1),PF(2),PF(3),DIM
CD REV 8/18/89
107 CONTINUE
ELSE
  DIM = RECVAL(1,5)
  CALL REALCK(50.,0.2,DIM,'AQUI','5TH',2.4)
ENDIF
ENDIF

CALL SKY TO GROUND ROUTINE TO CALCULATE THE SKY TO
GROUND RATIO FOR VISIBLE OR NEAR VISIBLE DEVICES

WVLNG = WAVE(IDEV)
IF (IDEV .NE. IOTHER) THEN
C***REV 1/91
  CALL SGR (BKGREF, WVLNG, VIS, SOG,
  TTEMPC, DDEWP, ICLIM, +
  IDEV )
  IF (EFLAG) RETURN
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-12-1-----------------------------------
C THE FOLLOWING LINE WAS MOVED TO WITHIN SKY CONDITION LOOP.
C
SG=SOG
C-HSTX---SCENE SHADOWS------------------------------------------------------
C
TEMPC = TTEMPC
DEWP = DDEWP
ENDIF
ATTN=3.912/VIS
TEMPA = TEMPC + 273.16

CALL SUBROUTINE SMINFO TO QUERY THE USER ABOUT THE INCLUSION
OF SMOKE SCREEN EFFECTS. *** J. N. CRAIN, 8/12/86 ***

SMEL = 0.
SMANGE = 0.

CALL SMINFO(TEMP,DEWP,SMEL,SMEX,SMANGE,ISMYE,AMOK)

MC SCHEMATIC IMAGER/SCREEN DIAGRAM, IF SMOKE SCREEN PRESENT.
C
IF (ISMYE.GT.1 .AND. AMOK .NE. 'N') THEN
  IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
    WRITE(NDIRTU,*)'** IMAGER/SCREEN SCHEMATIC (NOT TO SCALE) **'
    WRITE(NDIRTU,*)''
  ENDIF
  IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
    WRITE(IOOUT,*)'** IMAGER/SCREEN SCHEMATIC (NOT TO SCALE) **'
    WRITE(IOOUT,*)''
  ENDIF
ENDIF
IF (SMANGE.EQ.0.) THEN
  IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
    WRITE(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
      WRITE(IOOUT,*)'** IMAGER/SCREEN SCHEMATIC (NOT TO SCALE) **'
      WRITE(IOOUT,*)''
  ENDIF
ENDIF
C-128
WRITE(NDIRTU,*)'[IMAGER](((SCREEN)))-----------------*TARGET*'
WRITE(NDIRTU,912) SMEL*1000.
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,*)'[IMAGER](((SCREEN)))-------------*TARGET*'
WRITE(IOOUT,912) SMEL*1000.
ENDIF
912 FORMAT(10X,'<-',F7.2,' M ->')
ELSE
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,*)'[IMAGER]----------------- ((((SCREEN)))',
&
'[IMAGER]-----------------*TARGET*'
WRITE(NDIRTU,915) SMANGE*1000,SMEL*1000
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,*)'[IMAGER]----------------- ((((SCREEN)))',
&
'[IMAGER]-----------------*TARGET*'
WRITE(IOOUT,915) SMANGE*1000,SMEL*1000
ENDIF
915 FORMAT(10X,'<-',F7.2,' M ->','<-',F7.2,' M ->')
ENDIF
ENDIF
C
C OBTAIN ILLUMINATION DATA
C
C CALL ILMDAT(IDEV)
C
C CHECK FOR SLANT PATH CONDITIONS
C
C CRF NO DEPRESSION ANGLE FOR NON THERMAL DEVICES 27 FEB 92
DEP = 0.0
C STX TCM2 ECR ASL-7-1 -------------------------------------
C CALCULATE DEPRESSION ANGLE AT 30000 FT
CSTX FIX 1 10-31-91 DEPRESSION ANGLE ONLY FOR THERMAL AS YET C. BACA
IF(IDEV .EQ. ITHER) THEN
CRF CHANGE AL TO ALl IN FOLLOWING TWO LINES, AL IS ILLUMINATION
ALl = ALT/3280.8
DEP = ASIN(AL1/9.144111192)
DEP = 180.0 * DEP/3.14159
C BECAUSE OF ERROR IN COMPUTING TRANSMISSION ALONG A SLANT PATH
C THE FOLLOWING LINE PREVENTS TRANSMISSION CALCULATIONS ALONG A SLANT PATH
C WHEN FIXED DELETE FOLLOWING LINE
DEP = 0.0
ENDIF
C STX TCM2 ---------------------------------------------------
C
IF(ABS(DEP) .LE. 2.0) THEN
ISLT = 0
ANGLE = 0.0
ELSE
ISLT = 1
ANGLE = DEP
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,*)'CAUTION - SMOKE EFFECTS MAY NOT BE'
WRITE(NDIRTU,*)'FOR SLANT PATHS'
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,*)'CAUTION - SMOKE EFFECTS MAY NOT BE'
WRITE(IOOUT,*)'FOR SLANT PATHS'
ENDIF
ENDIF
C
C CALCULATE HUMIDITIES
C
C-129
AH = 1322.8314 * EXP(25.22 * (1.0 - 273.16 / (DEWP + 273.16)) + 5.31 * ALOG(273.16 / (DEWP + 273.16)) / TEMPA

RH = 100.0 * EXP(6885.06 * (1.0 / TEMPA - 1.0 / (DEWP + 273.16)) + 5.31 * ALOG(TEMPA / (DEWP + 273.16))

CONVERT RAINRATE

RNRT = RNRT * 25.4

FOR SNOW SET PARTICLE SIZE

CRF 17 APR 92 IF (IAERO .EQ. 9) THEN

CRF CHANGE THE PREVIOUS LINE FOR XSCALE92 AEROSOLS 17 APR 92

IF (IAERO .EQ. 10) THEN

CUV RD = 0.002

ENDIF

SET CLOUD THICKNESS FOR THERMAL DEVICES

IF (IDEV .EQ. IHER .AND. CEIL .NE. -1.) THEN

IF (INTER) THEN

CONTINUE

WRITE(IOOUT,*) 'ENTER THE CLOUD THICKNESS (IF KNOWN)'
WRITE(IOOUT,*) 'LESS THAN 10.0 KM, OTHERWISE ENTER 0.0'
READ(IOIN,*) THICK
WRITE(IOOUT,*) THICK
WRITE(IOOUT,*)

IF (THICK .LT. 0.0 .OR. THICK .GT. 10.) THEN

WRITE(IOOUT,*) 'YOU HAVE ENTERED AN ILLEGAL VALUE'
WRITE(IOOUT,*) 'FOR CLOUD THICKNESS'
GOTO 118

ENDIF

ELSE

THICK = RECVAL(8,4)
CALL REALCK(10.0,0.0,THICK,'METD','4TH',0.0)

ENDIF

CEILHT = CEIL
CUV AINVHT = AINV
CUV WIND = WINDS * 0.5144

OBTAIN SOUNDING INFORMATION

IF (IDEV .EQ. IHER .AND. ISLT .EQ. 1) THEN

IF (INTER) THEN

WRITE(IOOUT,*) 'IN ORDER TO CALCULATE GASEOUS ABSORPTION'
WRITE(IOOUT,*) 'FOR SLANT PATH CASES, A VERTICAL SOUNDING'
WRITE(IOOUT,*) 'IS NEEDED. YOU MAY USE YOUR OWN SOUNDING'
WRITE(IOOUT,*) 'FILE (SEE DOCUMENTATION FOR FILE REQUIREMENTS)'
WRITE(IOOUT,*) 'BY ENTERING THE NAME OF THE FILE'
WRITE(IOOUT,*) 'ENCLOSED IN QUOTATION MARKS AT THIS TIME.'
WRITE(IOOUT,*) 'OTHERWISE ENTER NONE IN SINGLE QUOTES'
WRITE(IOOUT,*) 'AND A SOUNDING WILL BE SELECTED FROM THE'
WRITE(IOOUT,*) 'STANDARD ATMOSPHERES.'
READ(IOIN,*) NMSND
WRITE(IOOUT,*)
ELSE

NMSND = RECFLD(11)

ENDIF

CALL FOR SOUNDING FILE

CALL GETSND(RLATT,IDATE,NMSND)

ENDIF
IF(IDEV .EQ. IHER) THEN
  IF(INTER) THEN
    WRITE(IOOUT,*)'ENTER THE SURFACE PRESSURE IN MILLIBARS'
    READ(IOIN,*)PRES
    WRITE(IOOUT,*)PRES
    WRITE(IOOUT,*)
    IF(PRES .LT. 0.0 .OR. PRES .GT. 1030.0) THEN
      WRITE(IOOUT,*)'YOU HAVE ENTERED AN INCORRECT VALUE'
      WRITE(IOOUT,*)'FOR SURFACE PRESSURE'
      GOTO 121
    ENDIF
    CD REV 8/18/89
    WRITE(21,122)'SONDNONE',PRES
    122 FORMAT(A8,2X,E10.4)
  ELSE
    PRES = RECVAL(11,1)
    CALL REALCK(1030.0,500.0,PRES,'SOND','1ST',1030.0)
  ENDIF
  IF(WVLNG .GT. 3.0 .AND. WVLNG .LT. 5.0) THEN
    IOTEMP = IOIN
    IOIN = IPHFUN
    CALL IOOPEN(KSTOR,'SCTH.UNT','SCRATCH',0,'FORMATTED',
                 'TARGAC','FINDR',LUNERR,*130)
    XRANG2 = 20.0
    HERE AND THE IN THE FOLLOWING XSCALE CALL FOR THE 8-11 BAND,
    THE PATH LENGTH (IN XRANG2) IS REDUCED UNTIL THE TRANSMITTANCE
    (IN XSTRN) IS MEANINGFULLY GREATER THAN ZERO.
    THE SUBROUTINE EXTIC TAKES THE LOG OF XSTRN DIVIDED BY XRANG
    TO DETERMINE THE AVERAGE EXTINCTION. SINCE LOG(0) IS UNDEFINED,
    THIS IS A BAD THING TO ALLOW. THIS TEST AVOIDS THE PROBLEM.
    RF AUG89
    CRF233 CALL XSCALE(3.0,5.0,VIS,IAERO,IDUM2,ISLT,NBR,
                   RD,DECPER,XMEAN,XMODE,IWATER,
                   XRANG2,ANGLE,CEILHT,ICLIM,RNRT,THICK,RH,WINDS)
    CRF REMOVE THE PREVIOUS TWO LINES, REPLACE WITH THE FOLLOWING 6 LINES
    CRF FOR XSCALE92. 17 APR 92
    C-HSTX---SCENE SHADOWS---ECR # HSTX-12-3---------------------------
    C CHANGED THE CONSTANTS SENT TO XSCALE TO VARIABLES. THESE VALUES
    MAY BE CHANGED IN XSCALE. THIS CAUSED PROBLEMS WHEN XSCALE ENDED,
    SINCE THE CONSTANTS WERE NO LONGER THE VALUES THAT THEY WERE
    SUPPOSED TO BE.
    C 233 CALL XSCALE(3.0,5.0,VIS,IAERO,2,ISLT,NBR,
                  RD,DECPER,XMEAN,XMODE,IWATER,
                  XRANG2,ANGLE,CEILHT,ICLIM,RNRT,THICK,RH,WINDS)
    233 CALL XSCALE(DUM3,DUM5,VIS,IAERO,IDUM2,ISLT,NBR,
                  RD,DECPER,XMEAN,XMODE,IWATER,
2 WAVRFN,RESPFN,
3 XRANG2,ANGLE,ALT,CEILHT,THICK,AINVHT,
4 RNRT,TEMP,WINDS,RH,XSTRN1,IERR,
5 Q,UM,EXT551,QAVE)

C-HSTX---SCENE SHADOWS-------------------------------

C IF( XSTRN2 .LT. 1.0E-15 ) THEN
XSTRN2 = XSTRN2 / 2.0
IF( XSTRN2 .GT. 0.05 ) GOTO 233
XSTRN2 = XSTRN2 * 2.0
ENDIF
C REWIND IPHFUN
C IOIN = IOTEMP
GOTO 135
130 IERR = 1
135 CLOSE(KSTOR)
IF(IERR .EQ. 1) THEN
EFLAG = .TRUE.
RETURN
ENDIF
JWAVE = 2
IF (ISLT .EQ. 1) THEN
BETA2(1) = 0.0
DO 150 J = 1, NNZPTS
    BETA2(J+I) = BETA(J)
150 CONTINUE
CALL SLNCOF(EXTC1,EXTC2,SINANG,ELEVA)
CP ALT = 1.0 * SINANG
CP CHANGED THE ABOVE LINE TO THE ONE BELOW 16-06-92 TO
CP ADHERE TO ANSI STANDARDS
R = 1.0
ELSE
ALT = 1.0
ENDIF
CALL EXTIC(JWAVE,ALT,ISLT,EXTC,AH,PRES,VIS,TEMPA,RH,IAERO)
ELSE IF(WVLNG .GT. 8.0 .AND. WVLNG .LT. 11.0) THEN
C IOTEMP = IOIN
C IOIN = IPHFUN
CALL IOOPEN(KSTOR,'SCTH.UNT','SCRATCH',0,'FORMATTED',
+ 'TARGAC','FINDR',LUNERR,*155)
XRANG1 = 20.0
CRF 235 CALL XSCAIL(8.0,11.0,-20.,ATTN,XSTRN1,IERR,ISLT,IAERO,
CRF + XRANG1,ANGLE,CEILHT,ICLIM,RNRT,THICK,RH,WINDS)
CRF REMOVE THE PREVIOUS 2 LINES, REPLACE WITH FOLLOWING 6 LINES
CRF FOR XSCALE92. 17 APR 92.
C CRF C-HSTX---SCENE SHADOWS---ECR # HSTX-12-3----------------------
C CHANGED THE CONSTANTS SENT TO XSCALE TO VARIABLES. THESE VALUES
C MAY BE CHANGED IN XSCALE. THIS CAUSED PROBLEMS WHEN XSCALE ENDED,
C SINCE THE CONSTANTS WERE NO LONGER THE VALUES THAT THEY WERE
C SUPPOSED TO BE.
C 235 CALL XSCALE(8.0,11.0,VIS,IAERO,2,ISLT,NBR,
C 1 RD,DECPER,XMEAN,XMODE,IWATER,
C 2 WAVRFN,RESPFN,
C 3 XRANG1,ANGLE,CEILHT,THICK,AINVHT,
C 4 RNRT,TEMP,WINDS,RH,XSTRN1,IERR,
C 5 Q,UM,EXT551,QAVE)
DUMB = 8.0
DUM11 = 11.0
IDUM2 = 2
235 CALL XSCALE(DUMB,DUM11,VIS,IAERO,IDUM2,ISLT,NBR,
 1 RD,DECPER,XMEAN,XMODE,IWATER,
 2 WAVRFN,RESPFN,
 3 XRANG1,ANGLE,CEILHT,THICK,AINVHT,
IF( XSTRN1 .LT. 1.OE-15 ) THEN
   XRANG1 = XRANG1 / 2.0
   IF( XRANG1 .GT. 0.05 ) GOTO 235
   XRANG1 = XRANG1 * 2.0
ENDIF

C REWIND IPHFUN
C IOIN = IOTEMP
GOTO 157

155 IERR = 1
157 CLOSE(KSTOR)
IF(IERR .EQ. 1) THEN
   EFLAG = .TRUE.
   RETURN
ENDIF

JWAVE = 1
IF(ISLT .EQ. 1) THEN
   BETA1(1) = 0.0
   DO 160 J = 1,NNZPTS
      BETA1(J+1) = BETA(J)
   160 CONTINUE
   CALL SLNCOF(EXTC1,EXTC2,SINANG,ELEVA)
   CP ALT = 1.0 * SINANG
   CP CHANGED THE ABOVE LINE TO THE ONE BELOW 16-06-92 TO
   CP ADHERE TO ANSI STANDARDS
   ALT = SINANG
   R = 1.0
   ELSE
      ALT = 1.0
   ENDIF
   CALL EXTIC(JWAVE,ALT,ISLT,EXTC,AH,PRES,VIS,TEMPA,RH,IAERO)
ELSE
   IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
      WRITE(IOOUT,*) 'PROGRAM HALTED BECAUSE WAVELENGTH OUT OF RANGE'
   ENDIF
   IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
      WRITE(NDIRY,*) 'PROGRAM HALTED BECAUSE WAVELENGTH OUT OF RANGE'
   ENDIF
   STOP
ENDIF
ENDIF

C *** DO LOOP FOR 2 ACQUISITION LEVELS--DETECTION AND RECOGNITION
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-12-1-----------------------------------
C CHANGE SKY-TO-GROUND RATIO VARIABLE TO AN ARRAY AND ADD DO LOOP IN ORDER
C TO COMPUTE ACQUISITION LEVELS FOR UP TO TWO POSSIBLE SKY CONDITIONS.
C WRITE PROBABILITY THAT TARGET SCENE IS IN CLOUD SHADOW. WRITE SKY
C CONDITION.
C-HSTX---SCENE SHADOWS---ECR # HSTX-12-2-----------------------------------
C REMOVE EXTRA ITERATION FOR THERMAL IMAGER DEVICES.
C
IF (IDEV .EQ. ITHER) THEN
   JSTOP = 1
ELSE
   IF ( MLOOP .EQ. 1 ) THEN
      JSTOP = 1
   ELSE
      JSTOP = 2
   END IF
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2)
   WRITE(NDIRY,*) 'PROBABILITY OF TARGET IN CLOUD SHADOW:',
2 100.0 - PSCLD
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
2 WRITE(IOOUT,*) 'PROBABILITY OF TARGET IN CLOUD SHADOW:',
2 100.0 - PSCLD
C
IF ( NUMCLD .EQ. 2 ) THEN
1 IF ( CF1 .NE. 1.0 .AND. CF2 .NE. 1.0 ) THEN
1 WRITE(NDIRTU,*) 'TWO PARTLY CLOUDY LAYERS.'
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(IOOUT,*) 'TWO PARTLY CLOUDY LAYERS.'
1 ELSE IF ( CF1 .NE. 1.0 .OR. CF2 .NE. 1.0 ) THEN
1 WRITE(NDIRTU,*) '1 PTLY CLDY LRY, 1 OVRCST LRY.'
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(IOOUT,*) '1 PTLY CLDY LRY, 1 OVRCST LRY.'
1 ELSE
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(NDIRTU,*) 'TWO OVERCAST LAYERS.'
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(IOOUT,*) 'TWO OVERCAST LAYERS.'
END IF
ELSE IF ( NUMCLD .EQ. 1 ) THEN
1 IF ( CF1 .NE. 1.0 ) THEN
1 WRITE(NDIRTU,*) 'ONE PARTLY CLOUDY LAYER.'
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(IOOUT,*) 'ONE PARTLY CLOUDY LAYER.'
1 ELSE
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(NDIRTU,*) 'ONE OVERCAST LAYER.'
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(IOOUT,*) 'ONE OVERCAST LAYER.'
END IF
ELSE
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(NDIRTU,*) 'BOTH LAYERS CLEAR.'
1 IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
1 WRITE(IOOUT,*) 'BOTH LAYERS CLEAR.'
END IF
ENDIF
C
DO 500 JCLoud=1,JSTOP
C-HSTX---SCENE SHADOWS-----------------------------------------------
C
DO 500 JCYCLE=1,2
C
C *** INCREMENT RANGE
C
R = 0.0
R0 = 0.50
C RF REV 1/27/92
C RF REMOVED FOLLOWING LINE, THIS WRITES OVER IDEV WHICH IS STILL
C RF NEEDED
C RF IDEV = ICRST(ITYPE)
C RF REV 1/27/92
NSN = NSNSR(ITYPE)
DO 5 IPF=1,3
DO 5 LSC = 1,NSN
PR(LSC,IPF)=0.0
PS(LSC,IPF)=99.
5 CONTINUE
6 CONTINUE
IF(IDEV .EQ. IOTHER) THEN
DUM1 = 0.5
CALL EXTIC(JWAVE,DUM1,ISLT,EXTC,AH,PRES,VIS,TEMPA,RH,IAERO)
ICYCLE = 3 * JCYLE - 2

C FOR THERMAL IMAGER, START RANGE MARCHING DIRECTLY OVER TARGET.
CRF DISAGREE NOW: THIS IS NOT THE AIR FORCE 5 MAR 92
CRF
ELSE
ICYCLE = 2 * JCYLE - 1
ENDIF
CFLAG = .FALSE.
R=R+0.1
SMATH = 0.
IF (R.GT.SMANGE) SMATH = AMIN1(R-SMANGE,SMEL)
KEY = 0

C FOLLOWING CALCULATIONS WERE RELOCATED FROM UPSTREAM IN FINDR AND
C COMMENTS ADDED.
C CALCULATE EFFECTIVE TARGET SIZE FOR MENU TARGETS
C
C AS SUGGESTED BY C. BACA ON HP VERSION, CHANGE IF BELOW TO
C LINE BELOW IT, THAT IS, IF NOT GENERIC AND YES THERMAL
C IF (.NOT. GNRICT) THEN
CIF (.NOT. GNRICT .AND. IDEV .EQ. ITHER) THEN
C DEP=ASIN(ALT/R)
C REPLACE THE LINE COMMENTED OUT ABOVE WITH THE IF BLOCK
C DIRECTLY BELOW
C IF (ALT .GT. R) THEN
DEP = ASIN(1.0)
ELSE
DEP = ASIN(ALT/R)
ENDIF
C
C COMPUTE YEFF
C
C DIM=AMIN1(XEFF,YEFF)
CCB CHANGED THE ABOVE COMMENTED LINE TO THE FOLLOWING
IF (IDEV .EQ. ITHER) DIM=AMIN1(XEFF,YEFF)
C
C *** CALCULATE APPARENT CONTRAST
C
C IF(IDEV .NE. ITHER) THEN
SMOKE = EXP(SMEX(3)*SMATH)
SG = SOG(JCLOUD)
C=ABS(CNTRST/(1.0+SG*(SMOKE*EXP(ATTN*R)-1.0)))
C = AMIN1(C,1.0)

C-135
ALOGC = ALOG(C)

IF (C .LT. CONLIM .AND. R .LT. .11) THEN
  CFLAG = .TRUE.
  RETURN
ENDIF
ELSE
  IF (R-RO) .GE. 0.25) THEN
    RO = RO + 0.50
    CALL EXTIC(JWAVE,RO,ISLT,EXTC,AH,PRES,VIS,TEMPA,RH,IAERO)
  ENDIF
  ATTN = EXTC(JWAVE)
ENDIF

C STX TCM2 ECR ASL-7-1 ++++++++++++++++
C EVALUATE INHERENT CONTRAST BY INTERPOLATING ON TCM2 ARRAY.
  CNTRST=UNILIN(DEPS,RNDELT,5,DEP,IEX)
CPUBS IEX ABOVE APPEARS TO BE USED BEFORE SET, BUT IT IS RETURNED
CPUBS BY FUNCTION UNILIN 18 NOV 92 PSG
C STX TCM2

C = ABS(CNTRST*EXP(-ATTN*R)*EXP(-SMEX(JWAVE)*SMATH))
ALOGC = ALOG(C)

IF (C .LT. CONLIM .AND. R .LT. .21) THEN
  CFLAG = .TRUE.
  RETURN
ENDIF
ENDIF

IFIRST = .TRUE.
DO 220 LSC=1,NSN
  ILOOP = .FALSE.
  RC = RCF(ITYPE,LSC,IFUN,CFLAG)
  IF(CFLAG) THEN
    IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
      WRITE(NDIRTU,*)'*** CONTRAST WAS TOO LOW FOR DEVICE',LSC
      WRITE(NDIRTU,*)' RETRY WITH DIFFERENT COMBINATION OF'
      WRITE(NDIRTU,*)' TARGET AND BACKGROUND OR DIFFERENT
    ENDIF
    IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
      WRITE(IOOUT,*)'*** CONTRAST WAS TOO LOW FOR DEVICE',LSC
      WRITE(IOOUT,*)' RETRY WITH DIFFERENT COMBINATION OF'
      WRITE(IOOUT,*)' TARGET AND BACKGROUND OR DIFFERENT
    ENDIF
    ILOOP = .TRUE.
  ENDIF
  IF( .NOT. ILOOP) THEN
    CALL ACQUIR(PSS,R,RC,ICYCLE,DIM)
  ENDIF
330 CONTINUE
ENDIF
IF(Key.EQ.1) GO TO 1
DO 225 IPF = 1,NPROB
  DO 225 LSC = 1,NSN
    IF(Abs(PS(LSC,IPF)-PF(IPF)) .GE. Abs(PSS - PF(IPF))) THEN
      PS(LSC,IPF)=PSS
      PR(LSC,IPF)=R
      IF(PSS.GT. .00000001) KEY=1
    ENDIF
    CONTINUE
  ENDIF
220 CONTINUE
1 CONTINUE
DO 225 IPF = 1,NPROB
DO 225 LSC = 1,NSN
IF(PR(LSC,IPF).EQ.0.) PR(LSC,IPF) = 9999999.

CONTINUE
IF(ICYCLE.EQ.1) THEN
  IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
    WRITE(IOOUT,*)'HIT ENTER TO CONTINUE'
    READ(IOOUT,*)
  ENDIF
ENDIF
IF(ICYCLE.EQ.3.OR.ICYCLE.EQ.4) THEN
  IF(ICRST(ITYPE).EQ.1) THEN
    IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
      WRITE(IOOUT,*)'HIT ENTER TO CONTINUE'
      READ(IOOUT,*)
    ENDIF
  ENDIF
ENDIF

C
C *** OUTPUT RANGES FOR SET LEVELS OF PROBABILITY
C
CRF REV 1/27/92
CRF REPLACED IDEV WITH ICRST(ITYPE) IN NEXT 5 STATEMENTS
IF(ICRST(ITYPE).EQ.1)CDEV(1:4) = 'DVO'
IF(ICRST(ITYPE).EQ.2)CDEV(1:4) = 'I I'
IF(ICRST(ITYPE).EQ.3)CDEV(1:4) = 'SITV'
IF(ICRST(ITYPE).EQ.4)CDEV(1:4) = 'TI'
IF(ICRST(ITYPE).EQ.5)CDEV(1:4) = 'USR'

CRF REV 1/27/92
CHRDEV(1:16) = ' DEVICE TYPE IS '
CHRDEV(17:20) = CDEV
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
  WRITE(IOOUT,100)'CHRDEV'
  WRITE(IOOUT,100)CHRDEV
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
  WRITE(NDIRTU,100)'CHRDEV'
  WRITE(NDIRTU,100)CHRDEV
ENDIF

C-HSTX---SCENE SHADOWS---ECR # HSTX-12-1-------------------------------------
C ADDITIONAL OUTPUT INFORMATION PROVIDED TO INDICATE CLEAR SKY OR CLOUD
C SHADOW AcQuisITION RANGES.
IF(IDEV.NE.IETHER) THEN
  IF (((JCLOUD.EQ.2).OR.((MLOOP.EQ.1).AND.(NUMCLD.GT.0))) THEN
    IF(IOFILE.EQ.1.OR.IOFILE.EQ.2)
      WRITE(NDIRTU,*) 'CLOUD SHADOW CASE:'
    IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
      WRITE(IOOUT,*) 'CLOUD SHADOW CASE:'
  ELSE
    IF(IOFILE.EQ.1.OR.IOFILE.EQ.2)
      WRITE(NDIRTU,*) 'CLEAR SKY CASE:'
    IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
      WRITE(IOOUT,*) 'CLEAR SKY CASE:'
  ENDIF
ENDIF

C-HSTX---SCENE SHADOWS----------------------------------------------------------
C
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2)
  WRITE(NDIRTU,90) (RGNAME(ICYCLE)),(PF(J),J=1,NPROB)
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2)
  WRITE(IOOUT,90) (RGNAME(ICYCLE)),(PF(J),J=1,NPROB)
90  FORMAT(' ',/,A11,1X,'RANGE (KM)'/ PROBABILITY LEVEL = ', +
        3(F3.2,14X))
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
  WRITE(NDIRTU,92) 'SENSOR ID'
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
   WRITE(IOOUT,92) 'SENSOR ID'
ENDIF
92  FORMAT(A10)
DO 420 LSC=1,NSN
   IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
      WRITE(NDIRTU,91)LSC,(PR(LSC,IPF),IPF=1,NPROB)
   END IF
   IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
      WRITE(IOOUT,91)LSC,(PR(LSC,IPF),IPF=1,NPROB)
   END IF
420  CONTINUE
500 CONTINUE
RETURN
100 FORMAT(A20)
1000 FORMAT(10I5)
1100 FORMAT(8F10.7)
1200 FORMAT(A15)
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: THERML Old Date: 3/23/93
File Name: THERML.FOR New Date: 9/14/93

Implemented By: Michael Oberlatz

Reason for Revision: The cloud base height limits were changed to match the base height limits in the visible section.

Description of Revision: The high cloud base height limits were changed from 9.0 - 20.0 km to 6.1 - 13.7 km. The middle cloud base height limits were changed from 4.0 - 8.0 km to 2.0 - 6.1 km. The low cloud base height limits were changed from 1.0 - 4.0 km to 0.1 - 2.0 km.

Notes: 

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-139
SUBROUTINE THERML

C THE SUBROUTINE THERML HAS BEEN SPLIT INTO TWO ROUTINES: THERML & THERMB.
C THERML CONTAINS ONLY BLOCK ONE FROM THE ORIGINAL MODEL WITHOUT ANY
C WRITING OF INPUTS TO OUTPUT FILES.
C THERML NOW HAS THE TARGETS AND BACKGROUNDS OF TCM2.
C THE ARRAYS FOR TARGETS AND BACKGROUND HAVE BEEN REDIMENSIONED FROM 8 TO
C 23 AND FROM 30 TO 39.
C WEATHER & TIME ARRAYS THAT WERE DIMENSIONED BY 17 ARE NOW DIMENSIONED BY
C 11.
C C
C COMMON BLOCKS AND DIMENSION STATEMENTS NO LONGER NEEDED HAVE BEEN DELETED.

COMMON/WEATHER/WX(11,19),IXW(11,8),ALB,TCORE,TBAR
COMMON/TIMES/NTIM,NRUNTM,TRTOT(7),IDATE,ITIMOT,TOT,
+JDATE(11),JTIME(11)
COMMON/ASCENE/NTARID,NBKID,ITARID,IBKID,RLATT,RLONG,ELEV,VSPEED,
+THEAD
COMMON/OFILE/TITLE
COMMON/THRM/NRTM,TTRG,TBKG,TMPC,DEWP,ALT,VS,IAERO,RNRT,CEIL
+AINV,WINDS,ELEVA,ASP,IDTG
COMMON/INOUT/INTER,IRPT,EFLAG
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,IRELH,
+KSTOR,NPLLOTU,STDERR
COMMON/IBLOCK/RECVAL(111,7),RECUSE(21),IPTR(21),IBEGIN(22)

C COMMON / GNERIC / GNRICT,GNRICB

C***REV 1/91

COMMON / ILCNT/ FMONTH, DAY, YEAR, GTIME, SLAT, SLON, ILR1, ILR2,
+ILR3, RG, FR1, FR2, FR3, SIGWX, OBSURF, CEILHT,
+FRTP, FRC, ITARG
COMMON / ILCNT/ ALTS, AZIS, ALTIN, AZIM, DPHASE, ELUMI, SUNLIT,
+MOOLIT, TCLSUN, TCLLUN, RCLSUN, RCDLUN,
+RCDLUN

INTEGER IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT
INTEGER IRELH,KSTOR,NPLLOTU,STDERR
REAL IH,MOOLIT
CHARACTER*15 TITLE
CHARACTER*1 ICDLAY(3)
DIMENSION IBKIDC(39),ITGDSC(23)
LOGICAL INTER,RECUSE,GNRICT,GNRICB,EFLAG
CHARACTER*36 IBKIDC,ITGDSC
DATA ICDLAY/'H','M','L'/

CP DATA CONST/.017453293/
CP CONST IS COMMENTED OUT BECAUSE IT IS NOT
CP USED. CONST IS CONVERSION FROM DEGREES TO RADIANS.
CP ICDLAY APPEARS TO BE FOR USE WITH THE LOW MEDIUM HIGH
CP CLUTTER MODEL. FUTURE USE. PSG NOV 92

C C C
C********************************************************************************* C C BEGIN BLOCK 1 C C READ INPUT PARAMETERS FROM UNIT 5 AND ENSURE C THEY ARE REALISTIC AND CONSISTENT. SUB- C ROUTINES REALCK AND INTCHK ARE USED TO FLAG C QUESTIONABLE DATA.

C-140
C

C SET INPUT RECORD COUNTER INDEX
C
C INITIALIZE BACKGROUND AND TARGET DESCRIPTORS
C

C++++++ STX ECR ASL-9-1 ++++++++++++++++++++++++++++++++++
C***** TCM2 BACKGROUNDS *********
IBKDSC(1) = ' TALL GRASS GROWING'
IBKDSC(2) = ' TALL GRASS DORMANT'
IBKDSC(3) = ' MOWED GRASS GROWING'
IBKDSC(4) = ' MOWED GRASS DORMANT'
IBKDSC(5) = ' DECIDUOUS TREES GROWING'
IBKDSC(6) = ' DECIDUOUS TREES DORMANT'
IBKDSC(7) = ' CONIFEROUS TREES GROWING'
IBKDSC(8) = ' CONIFEROUS TREES DORMANT'
IBKDSC(9) = ' DIRT ROAD DRY'
IBKDSC(10) = ' ROCK'
IBKDSC(11) = ' PLOWED FIELD WET'
IBKDSC(12) = ' SNOW ON ROAD'
IBKDSC(13) = ' SALT DRY'
IBKDSC(14) = ' SALT DAMP'
IBKDSC(15) = ' ROAD'
IBKDSC(16) = ' CONCRETE SLAB'
IBKDSC(17) = ' GRASS FIELD'
IBKDSC(18) = ' TREES'
IBKDSC(19) = ' DIRT ROAD WET'
IBKDSC(20) = ' PLOWED FIELD DRY'
IBKDSC(21) = ' CLAY ROAD'
IBKDSC(22) = ' UNDISTURBED SNOW'
IBKDSC(23) = ' ASPHALT'
IBKDSC(24) = ' THIN CONCRETE SLAB'
IBKDSC(25) = ' THICK CONCRETE'
IBKDSC(26) = ' THICK ASPHALT'
IBKDSC(27) = ' SOIL MOIST'
IBKDSC(28) = ' STANDARD SAND'
IBKDSC(29) = ' STANDARD WATER'
IBKDSC(30) = ' FOLIAGE DORMANT AND SPARSE'
IBKDSC(31) = ' FOLIAGE DORMANT AND MEDIUM'
IBKDSC(32) = ' FOLIAGE DORMANT AND DENSE'
IBKDSC(33) = ' FOLIAGE GROWING AND SPARSE'
IBKDSC(34) = ' FOLIAGE GROWING AND MEDIUM'
IBKDSC(35) = ' FOLIAGE GROWING AND DENSE'
IBKDSC(36) = ' FOLIAGE VIGOROUS AND SPARSE'
IBKDSC(37) = ' FOLIAGE VIGOROUS AND MEDIUM'
IBKDSC(38) = ' FOLIAGE VIGOROUS AND DENSE'
IBKDSC(39) = ' GENERIC BACKGROUND'

C***** TCM2 TARGETS ************
ITGDSC(1) = ' T62 - OFF'
ITGDSC(2) = ' T62 - IDLE'
ITGDSC(3) = ' T62 - EXERCISED'
ITGDSC(4) = ' ZIL - OFF'
ITGDSC(5) = ' ZIL - IDLE'
ITGDSC(6) = ' ZIL - EXERCISED.'
ITGDSC(7) = ' GENERATOR - OFF'
ITGDSC(8) = ' GENERATOR - IDLE'
ITGDSC(9) = ' BUNKER'
ITGDSC(10) = ' DAM'
ITGDSC(11) = ' POL'
ITGDSC(12) = ' BRIDGE'
ITGDSC(13) = ' HYDROELECTRIC POWER PLANT'
ITGDSC(14) = ' FT72 - OFF'
ITGDSC(15) = ' FT72 - IDLE'
ITGDSC(16) = ' FT72 - EXERCISED'
ITGDSC(17) = ' FCHOP(APACHE) - OFF'
ITGDSC(18) = ' FCHOP(APACHE) - HOVER'

C-141
ITGDSC(19) = 'ANTI-TANK VEHICLE BRDM-2 OFF'
ITGDSC(20) = 'ANTI-TANK VEHICLE BRDM-2 IDLE'
ITGDSC(21) = 'ANTI-TANK VEHICLE BRDM-2 EXERCISED'
ITGDSC(22) = 'POINT SUR SHIP'
ITGDSC(23) = 'GENERIC TARGET'

C--STX-TCM2-----------------------------------------------
C
CD
SET GENERIC FLAGS
GNRICT = .FALSE.
GNRICB = .FALSE.
CD
C***REV 1/91
ITARG = 1
C
C INPUT TARGET TYPE
C
IF(INTER) THEN
20  CONTINUE
C STX TCM2 -- DISPLAY 23 TARGETS NOW (TCM1 HAD 8)
WRITE(IOOUT,*) 'ENTER THE TARGET ID NUMBER FROM 0 TO 23'
WRITE(IOOUT,*) 'BY SELECTING THE NUMBER FROM THE LIST BELOW'
DO 25 N = 1,11
WRITE(IOOUT,1000) N,ITGDSC(N),N+12,ITGDSC(N+12)
25 CONTINUE
WRITE(IOOUT,1000) 12,ITGDSC(12)
READ(IOIN,*) NTARID
WRITE(IOOUT,*) NTARID
WRITE(IOOUT,*)
CD
SET GENERIC FLAG IS GENERIC TARGET IS CHOSEN
C STX TCM2 ECR ASL-8-1 ++++++++++++++++++++++++++++++++++++++
C--GENERIC TARGET IS NOW 23 (TCM1 GENERIC TARGET = 8)
C
IF (NTARID .EQ. 23) THEN
  GNRICT = .TRUE.
ELSE
  GNRICT = .FALSE.
ENDIF
CD
IF (NTARID .LT. 1 .OR. NTARID .GT. 23) THEN
  WRITE(IOOUT,*) 'THE TARGET ID NUMBER MUST BE BETWEEN 1 AND 23'
  GOTO 20
ENDIF
C
C--STX--TCM2 ASL ECR-8-1 +++++++++++++++++++++++++++++++++++
C TCM2 REQUIRES TARGET SPEED OF ONLY EXERCISED VEHICLES (NEW INPUT)
IF (INTER) THEN
  IF (NTARID .EQ.3 .OR. NTARID .EQ.6 .OR. NTARID .EQ.17 .OR. NTARID .EQ.21)
1    THEN
    WRITE(IOOUT,*)'ENTER THE VEHICLE SPEED IN METERS PER SECONDS'
    READ(IOIN,*)VSPEED
  ENDIF
ENDIF
IF (VSPEED.LT.0.OR.VSPEED.GT.25) THEN
  WRITE(IOOUT,*) 'THE RANGE FOR VEHICLE SPEED IS 0 TO 25 M/S'
  GOTO 17
ENDIF
ENDIF
ELSE
  VSPEED = NINT(RECVAL(12,3))
  CALL REALCK(25.,1.,VSPEED,'TARG','3RD',5.0)
ENDIF

C--STX-TCM2-----------------------------------------------

C
C READ NUMBER OF BACKGROUNDS
C
IF(INTER) THEN
  30 CONTINUE
C STX TCM2 ECR ASL-8-1 ++++++++++++++++++
C--DISPLAY 39 BACKGROUNDS NOW (TCM1 HAD 30)
C
WRITE(IOOUT,*) 'ENTER THE BACKGROUND ID NUMBER FROM 1 TO 39'
WRITE(IOOUT,*)
'BY SELECTING THE NUMBER FROM THE LIST BELOW'
DO 33 N = 1,19
  WRITE(IOOUT,1000) N,IBKDSC(N),N+20,IBKDSC(N+20)
33 CONTINUE
WRITE(IOOUT,1000) 20,IBKDSC(20)
READ(IOIN,*) NBKID
WRITE(IOOUT,*) NBKID
WRITE(IOOUT,*)
CD IF THE GENERIC BACKGROUND IS CHOSEN, THE FLAG MUST BE SET
C STX TCM2 ASL-8-1 +++++++++++++++++++++
C--GENERIC BACKGROUND IS NOW 39 (TCM1 IT WAS 30)
IF (NBKID.EQ.39) THEN
  GNRICB = .TRUE.
ELSE
  GNRICB = .FALSE.
ENDIF
IF (NBKID .LT. 1 .OR. NBKID .GT. 39) THEN
  WRITE(IOOUT,*) 'THE BACKGROUND ID NUMBER MUST BE ' 
  WRITE(IOOUT,*) ' BETWEEN 1 AND 39'
  GOTO 30
ENDIF
C*******************************************************************************
C IF THE GENERIC BACKGROUND IS CHOSEN, THE FLAG MUST BE SET'
C STX TCM2 ASL-8-1 +++++++++++++++++++
C--GENERIC BACKGROUND IS NOW 39 (TCM1 IT WAS 30)
IF (NBKID.EQ.39) THEN
  GNRICB = .TRUE.
ELSE
  GNRICB = .FALSE.
ENDIF
IF (NBKID .LT. 1 .OR. NBKID .GT. 39) THEN
  WRITE(IOOUT,*) 'THE BACKGROUND ID NUMBER MUST BE ' 
  WRITE(IOOUT,*) ' BETWEEN 1 AND 39'
  GOTO 30
ENDIF
C*******************************************************************************
ELSE
  NBKID = NINT(RECVAL(12,2))
C STX TCM2 ECR ASL-8-1 RESET THE UPPER BOUND
  CALL INTCHK(39,1,NBKID,'TARG','2ND',3)
ENDIF
C
CD IF THE GENERIC BACKGROUND IS CHOSEN ASK THE USER THE
CD EFFECTIVE TEMPERATURE OF THE BACKGROUND
CD IF (GNRICB) THEN
WRITE(IOOUT,*)' INPUT THE EFFECTIVE BACKGROUND TEMPERATURE'
READ(IOIN,*)TBKG
WRITE(IOOUT,*)TBKG
ENDIF
CD TG = NTARID
C STX TCM2 ECR ASL-8-1 +++++++++++++++++++
C REMOVED PROMPT FOR OPERATING CONDITION AND CALL TO DCODET
C
C READ IN LATITUDE,LONGITUDE,DAY-OF-THE-YEAR,
C TIME-OVER-TARGET,AND TARGET AREA ELEVATION.
C
IF(INTER) THEN
  40 CONTINUE
WRITE(IOOUT,*) 'ENTER THE LATITUDE IN DEG; POS. FOR N'
READ(IOIN,*) RLATT
C-143
WRITE(IOOUT,*) RLATT
WRITE(IOOUT,*)
IF(RLATT .GT. 90. .OR. RLATT .LT. -90.) THEN
  WRITE(IOOUT,*) 'THE LATITUDE MUST BE BETWEEN +/- 90 DEG'
  GOTO 40
ENDIF

50 CONTINUE
WRITE(IOOUT,*) 'ENTER THE LONGITUDE IN DEG; POS FOR W'
READ(IOIN,*) RLONG
WRITE(IOOUT,*) RLONG
WRITE(IOOUT,*)
IF(RLONG .GT. 180. .OR. RLONG .LT. -180.) THEN
  WRITE(IOOUT,*) 'THE LONGITUDE MUST BE BETWEEN +/- 180 DEG'
  GOTO 50
ENDIF

60 CONTINUE
WRITE(IOOUT,*) 'ENTER THE JULIAN DATE BETWEEN 1 AND 366'
WRITE(IOOUT,*) 'COUNTING FROM JANUARY 1987'
READ(IOIN,*) IDATE
WRITE(IOOUT,*) IDATE
WRITE(IOOUT,*)
IF(IDATE .LT. 1 .OR. IDATE .GT. 366) THEN
  WRITE(IOOUT,*) 'THE DAY NUMBER MUST BE BETWEEN 1 AND 366'
  GOTO 60
ENDIF

C***REV 1/91 NEW INTERACTIVE INPUT--REQUESTING YEAR OF INTEREST
C AS REQUIRED BY ILUMA MODULE

65 CONTINUE
WRITE(IOOUT,*) 'ENTER THE YEAR BETWEEN 1977 AND 1999'
READ(IOIN,*) YEAR
WRITE(IOOUT,*) YEAR
WRITE(IOOUT,*)
IF(YEAR .LT. 1977 .OR. YEAR .GT. 1999) THEN
  WRITE(IOOUT,*) 'THE YEAR MUST BE BETWEEN 1977 AND 1999'
  GOTO 65
ENDIF

C

70 CONTINUE
WRITE(IOOUT,*) 'ENTER THE TIME OF INTEREST IN HHMM Z'
READ(IOIN,*) ITIMOT
WRITE(IOOUT,*) ITIMOT
WRITE(IOOUT,*)
IF(ITIMOT .LT. 0 .OR. ITIMOT .GT. 2359) THEN
  WRITE(IOOUT,*) 'THE TIME OF INTEREST MUST BE BETWEEN 0 AND 2359 HOURS ZULU.'
  GOTO 70
ENDIF

80 CONTINUE
WRITE(IOOUT,*) 'ENTER THE TARGET ELEVATION IN FT'
READ(IOIN,*) ELEV
WRITE(IOOUT,*) ELEV
WRITE(IOOUT,*)
IF(ELEV .LT. 0. .OR. ELEV .GT. 20000.) THEN
  WRITE(IOOUT,*) 'THE ELEV MUST BE BETWEEN 0 AN 20000 FT'
  GOTO 80
ENDIF
ELSE
  RLATT = RECVAL(9,1)
  RLONG = RECVAL(9,2)
  IDATE = NINT(RECVAL(9,3))
  ITIMOT = NINT(RECVAL(9,4))
  ELEV = RECVAL(9,5)
C***REV 1/91 NEW BATCH INPUT--YEAR OF INTEREST REQUIRED BY ILUMA MODULE
YEAR = RECVAL(12,6)
C

C-144
CALL REALCK(90.0,-90.0,RLATT,'SITE','1ST',55.0)
CALL REALCK(180.0,-180.0,RLONG,'SITE','2ND',-9.0)
CALL INTCHK(366,1,DATE,'SITE','3RD',188)
CALL INTCHK(2359,0,ITIMOT,'SITE','4TH',1200)
CALL REALCK(20000.0,0.0,ELEV,'SITE','5TH',1000.0)
C***REV 1/91
CALL REALCK(1999.0,1977.0,YEAR,'TARG','6TH',1991.0)
C
ENDIF
C
C***REV 1/91 STORES LATITUDE AND LONGITUDE AS NEEDED BY ILUMA MODULE
SLAT = RLATT
SLON = -RLONG
C
C CONVERT ITIMOT TO TIME IN HOURS (TOT)
ITOT=ITIMOT/100
IMIN=ITIMOT-(ITOT*100)
XMIN=REAL(IMIN)/60.
TOT=REAL(ITOT)+XMIN
IF(INTER) THEN
IF (.NOT. GNRICT) THEN
86 CONTINUE
C STX TCM2-ECR ASL-8-1 ++++++++++++++++++++++++++++++++ ++++++++++++++++++++++++++++++
C TCM2 REQUIRES TARGET HEADING (NEW INPUT)
WRITE(IOOUT,*) 'INPUT THE TARGET HEADING, 0 - 360 DEG'
READ(IOIN,*) THEAD
WRITE(IOOUT,*) THEAD
WRITE(IOOUT,*)
IF(THEAD .LT. 0.0 .OR. THEAD .GT. 360.) THEN
WRITE(IOOUT,*)'THE TARGET HEADING MUST BE BETWEEN 0 AND 360 DEG'
GOTO 86
ENDIF
ELSE
C STX TCM2-ECR ASL-8-1****************************
C READ IN TARGET ASPECT ANGLE AND TARGET HEADING.
C
ASP = RECVAL(12,4)
CALL REALCK(360.,1.,ASP,'TARG','4TH',360.0)
THEAD = RECVAL(12,7)
CALL REALCK(360.,1.,THEAD,'TARG','7TH',90.0)
ENDIF
C
C STX TCM2-ECR ASL-8-1***********************************
C INPUT SENSOR ALTITUDE INSTEAD OF SENSOR DEPRESSION ANGLE
C
IF(INTER) THEN
90 CONTINUE
WRITE(IOOUT,*) 'INPUT THE ALTITUDE OF THE SENSOR IN FT'
WRITE(IOOUT,*)
READ(IOIN,*) ALT
WRITE(IOOUT,*) ALT
WRITE(IOOUT,*)
IF(ALT .LT. 0.0 .OR. ALT .GT. 30000.) THEN

C-145
WRITE(IOOUT,*) 'THE SENSORS ALTITUDE MUST BE BETWEEN'
WRITE(IOOUT,*) '0 AND 30000 FT'
GOTO 90
ENDIF
ELSE
C
READ SENSOR ALTITUDE STX
C
ALT = RECVAL(12,5)
CALL REALCK(30000.0,0.0,ALT,'TARG','5TH',100.0)
ENDIF
C
STX-TCM2---------------------------------------------
IF(INTER) THEN
  100 CONTINUE
  WRITE(IOOUT,*) 'INPUT THE SURFACE ALBEDO FROM 0.0 TO 1.0'
  READ(IOIN,*) ALB
  WRITE(IOOUT,*) ALB
  WRITE(IOOUT,*)
  C STX CHANGED IF STATEMENT SO ALBEDO CANNOT BE 0 OR 1 (REAL WORLD)
  IF(ALB .LE. 0.0 .OR. ALB .GE. 1.0) THEN
    WRITE(IOOUT,*) 'THE SURFACE ALBEDO MUST BE '
    WRITE(IOOUT,*) ' BETWEEN 0.0 AND 1.0'
    GOTO 100
  ENDIF
  110 CONTINUE
  WRITE(IOOUT,*) 'INPUT THE AVERAGE AIR TEMPERATURE IN DEG C FOR'
  WRITE(IOOUT,*) 'THE 24 H PERIOD PRIOR TO THE TIME OF INTEREST'
  READ(IOIN,*) TBAR
  WRITE(IOOUT,*) TBAR
  WRITE(IOOUT,*)
  IF (TBAR .LT. -60.0 .OR. TBAR .GT. 60.0) THEN
    WRITE(IOOUT,*) 'YOU HAVE ENTERED AN EXTREME TEMPERATURE'
    WRITE(IOOUT,*) 'THIS PROGRAM ACCEPTS TEMPERATURES BETWEEN'
    WRITE(IOOUT,*) '-60.0 AND +60.0 DEGREES C'
    GOTO 110
  ENDIF
  150 CONTINUE
ENDIF
CD REV 8/18/89 STX
C STX TCM2 ECR ASL-8-1+++++++++++++++++++++++++++++++++++++++++++C
C CHANGE 'TARG' RECORD: CHANGE ITARID TO VSPEED, CHANGE DEP TO ALT,
C ADD THEAD
WRITE(21,116)'TARG',FLOAT(NTARID),FLOAT(NBKID),VSPEED,
& ASP,ALT,YEAR,THEAD
WRITE(21,117)'SITE',RLATT,RLONG,FLOAT(IDATE),FLOAT(ITIMOT),
& ELEV,TBAR,ALB
116 FORMAT(A4,6X,7E10.4)
117 FORMAT(A4,6X,7E10.4)
C STX-TCM2---------------------------------------------
CD REV 8/18/89
WRITE(IOOUT,*)' SELECT AEROSOL TYPE'
WRITE(IOOUT,*)' 1-MARITIME AIR MASS'
WRITE(IOOUT,*)' 2-URBAN'
WRITE(IOOUT,*)' 3-RURAL (CONTINENTAL POLAR)'
WRITE(IOOUT,*)' 4-FOG (HEAVY ADVECTION)'
WRITE(IOOUT,*)' 5-FOG (MODERATE RADIATION)'
WRITE(IOOUT,*)' 6-RAIN (DRIZZLE)'
WRITE(IOOUT,*)' 7-RAIN (WIDESPREAD)'
WRITE(IOOUT,*)' 8-RAIN (THUNDERSTORM)'
C STX TCM2 CHANGED IAEROL TO IAERO THE VARIABLE IN THE COMMON BLOCK THRM
READ(IOIN,*) IAERO
WRITE(IOOUT,*) IAERO
WRITE(IOOUT,*)
IF(IAERO .LT. 1 .OR. IAERO .GT. 8) THEN
  WRITE(IOOUT,*) 'THE AEROSOL MODEL MUST BE BETWEEN 1 AND 8'
C-146
C READ IN GROUND ALBEDO AND AVERAGE AIR TEMPERATURE
C FOR THE 24 HOUR PERIOD PRIOR TO TOT.
C
ALB = RECVAL(9,7)
TBAR = RECVAL(9,6)
IAERO = NINT(RECVAL(14,1))
C
C STX RESET LIMITS OF ALBEDO
CALL REALCK(0.99999,0.00001,ALB,'SITE','7th',0.15)
CALL REALCK(60.0,-60.0,TBAR,'SITE','6th',10.0)
C STX CHANGE IAER0L TO IAERO
CALL INTCOK(8,1,IAERO,'XSCL','1st',3)
ENDIF
C
C***REV 1/91 STORES SURFACE ALBEDO FOR USE BY ILUMA MODULE
RG = ALB
C
IF(INTER) THEN
120 CONTINUE
C
120 TCM2 ECR ASL-8-1 **********************************************************
C TCM2 CAN ONLY HANDLE 30 HOURS OF MET DATA,
C SO RANGE IS 6 TO 30 FORMERLY 6 TO 48
C AND MAXIMUM OF 11 NOT 17 MET INPUT TIMES

WRITE(IOOUT,*) 'INPUT THE NUMBER OF TIMES AT WHICH'
WRITE(IOOUT,*) 'METEOROLOGICAL DATA IS AVAILABLE.'
WRITE(IOOUT,*) 'MET DATA IS REQUIRED FOR FROM 6 TO'
WRITE(IOOUT,*) '30 HRS PRIOR TO THE TIME OF INTEREST.'
WRITE(IOOUT,*) 'REPORTED IN 3 HR INTERVALS AND IS ALSO'
WRITE(IOOUT,*) 'REQUIRED AT THE TIME OF INTEREST. Thus,'
WRITE(IOOUT,*) 'THE NUMBER OF TIMES AT WHICH MET DATA IS'
WRITE(IOOUT,*) 'AVAILABLE MUST BE AT LEAST 3 AND AT MOST 11'
READ(IOIN,*) NTIM
WRITE(IOOUT,*) NTIM
WRITE(IOOUT,*) IF(NTIM .LT. 3 .OR. NTIM .GT. 11) THEN
WRITE(IOOUT,*) 'YOU HAVE ENTERED AN IMPROPER VALUE'
WRITE(IOOUT,*) 'FOR THE NUMBER OF TIMES AT WHICH MET'
WRITE(IOOUT,*) 'DATA IS AVAILABLE'
GOTO 120
ENDIF
ELSE
C
C READ IN THE NUMBER OF TIMES FOR WHICH METEOROLOGICAL
C DATA WILL BE INPUT. AT LEAST SIX HOURS OF ANTICEDENT
C METEOROLOGICAL DATA ARE REQUIRED TO COMPUTE TARGET/
C BACKGROUND TEMPERATURES AT A GIVEN TIME.
C
NTIM = NINT(RECVAL(13,1))
CALL INTCOK(11,3,NTIM,'TIME','1ST',3)
ENDIF
C
C STX TCM2-----------------------------------------------
C
IF(INTER) THEN
WRITE(IOOUT,*) 'YOU WILL NOW BE ASKED TO INPUT THE'
WRITE(IOOUT,*) 'METEOROLOGICAL DATA VALID FOR TIMES'
WRITE(IOOUT,*) 'IN THREE HOUR INCREMENTS STARTING'
WRITE(IOOUT,*) 'AT THE EARLIEST TIME AND PROCEEDING TO THE'
WRITE(IOOUT,*) 'TIME OF INTEREST. AT EACH PROMPT THE RANGE'
WRITE(IOOUT,*) 'OF PERMISSIBLE VALUES WILL BE DISPLAYED.'
CD REV 9/22/89
DO 450 I=1,NTIM
WX(I,14) = -1.0
GO TO 450
END
CONTINUE
DO 400 I = 1, NTIM
WX(I,1) = -3.* (NTIM - I)
WTME = ABS(WX(I,1))
WRITE(IOOUT,*) 'YOU WILL NOW BEGIN ENTERING GROUP', I, 'DATA'
WRITE(IOOUT,*) 'MEASURED AT OR FORECAST FOR', WTME, ' HOURS'
WRITE(IOOUT,*) 'BEFORE THE TIME OF INTEREST'
CONTINUE
WRITE(IOOUT,*) 'ENTER THE WEATHER INDEX AS FOLLOWS:'
WRITE(IOOUT,*) '1 - FOR NO PRECIPITATION'
WRITE(IOOUT,*) '2 - FOR RAIN'
WRITE(IOOUT,*) '3 - FOR SNOW'
READ(IOIN,*) IWX(I,1)
WRITE(IOOUT,*) IWX(I,1)
IF(IWX(I,1) .LT. 1 .OR. IWX(I,1) .GT. 3) THEN
  WRITE(IOOUT,*) 'THE WEATHER INDEX MUST BE BETWEEN 1 AND 3'
  GOTO 130
ENDIF
IF(IWX(I,1) .EQ. 2) THEN
  CONTINUE
 WRITE(IOOUT,*) 'ENTER THE PRECIPITATION RATE'
 WRITE(IOOUT,*) 'FROM 0.0 TO 5.0 INCHES/HOUR'
 READ(IOIN,*) WX(I,2)
 WRITE(IOOUT,*) WX(I,2)
 WRITE(IOOUT,*) WX(I,2)
 WRITE(IOOUT,*)
 IF(WX(I,2) .LT. 0.0 .OR. WX(I,2) .GT. 5.0) THEN
  WRITE(IOOUT,*) 'YOU HAVE ENTERED AN EXTREME VALUE FOR'
  WRITE(IOOUT,*) 'PRECIPITATION RATE. PLEASE ENTER A NUMBER'
  WRITE(IOOUT,*) 'BETWEEN 0.0 AND 5.0'
  GOTO 140
 ENDIF
 ELSE
  WX(I,2) = 0.0
 ENDIF
 CONTINUE
 WRITE(IOOUT,*) 'ENTER THE SURFACE AIR TEMPERATURE'
 WRITE(IOOUT,*) 'FROM -60.0 TO 60.0 DEG C'
 READ(IOIN,*) WX(I,3)
 WRITE(IOOUT,*) WX(I,3)
 WRITE(IOOUT,*)
 IF(WX(I,3) .GT. WX(I,2)) THEN
  WRITE(IOOUT,*) 'YOU HAVE ENTERED A DEWPOINT TEMPERATURE'
  WRITE(IOOUT,*) 'WHICH RESULTS IN A RELATIVE HUMIDITY OF'
  WRITE(IOOUT,*) 'OVER 100 PERCENT. PLEASE CHECK AND REENTER'
  GOTO 170
 ENDIF
 TMPVP1 = VAPOR(WX(I,4))
 IF(TMVPVP1 .LT. 0.0) THEN
  WRITE(IOOUT,*) 'YOU HAVE ENTERED AN UNREALISTICALLY LOW'
  WRITE(IOOUT,*) 'VALUE FOR THE DEWPOINT TEMPERATURE. '
  WRITE(IOOUT,*) 'PLEASE CHECK AND REENTER'

GOTO 170
ENDIF

180 CONTINUE
WRITE(IOOUT,*) 'ENTER THE SURFACE WIND SPEED IN KNOTS'
WRITE(IOOUT,*) 'FROM 0.0 TO 70.0'
READ(IOIN,*) WX(I,5)
WRITE(IOOUT,*) WX(I,5)
WRITE(IOOUT,*)
IF(WX(I,5) .LT. 0.0 .OR. WX(I,5) .GT. 70.) THEN
  WRITE(IOOUT,*) 'YOU HAVE ENTERED AN UNREALISTIC VALUE'
  WRITE(IOOUT,*) 'FOR THE SURFACE WIND SPEED. PLEASE CHECK'
  WRITE(IOOUT,*) 'AND REENTER'
GOTO 180
ENDIF

C*******************STX ADDITION*************************************
C STX TCM2 ECR ASL-8-1
C TCM2 REQUIRES WIND DIRECTION (NEW INPUT)
IF (WX(I,5).GT.0.0) THEN
  185 CONTINUE
WRITE(IOOUT,*) 'ENTER THE SURFACE WIND DIRECTION'
WRITE(IOOUT,*) 'FROM 1 TO 360'
READ(IOIN,*) WX(I,18)
WRITE(IOOUT,*) INT(WX(I,18))
WRITE(IOOUT,*)
IF(WX(I,18) .LE. 0.0 .OR. WX(I,18) .GT. 360.0) THEN
  WRITE(IOOUT,*) 'YOU HAVE ENTERED AN UNREALISTIC VALUE'
  WRITE(IOOUT,*) 'FOR THE SURFACE WIND DIRECTION. PLEASE'
  WRITE(IOOUT,*) 'CHECK AND REENTER'
GOTO 185
ENDIF
ENDIF
C*********************STX ADDITION*****************************************
190 CONTINUE
WRITE(IOOUT,*) 'ENTER THE SURFACE VISIBILITY'
WRITE(IOOUT,*) 'FROM 0.0 TO 100.0 KM'
READ(IOIN,*) WX(I,6)
WRITE(IOOUT,*) WX(I,6)
WRITE(IOOUT,*)
IF(WX(I,6) .LT. 0.0 .OR. WX(I,6) .GT. 100.0) THEN
  WRITE(IOOUT,*) 'YOU HAVE ENTERED AN UNREALISTIC VALUE'
  WRITE(IOOUT,*) 'FOR THE SURFACE VISIBILITY. PLEASE CHECK'
  WRITE(IOOUT,*) 'AND REENTER'
GOTO 190
ENDIF
C STX TCM2 ECR ASL-8-1 .++++++++++++++++++++++++++++++++++++++++++++++++
C TCM2 CALCULATES SOLAR IRRADIANCE AND SKY IRRADIANCE EVERY 15 MINUTES
C FOR CONSISTENCY THE USER NO LONGER ENTERS THEIR OWN VALUES.
WX(I,7) = -1.0
WX(I,8) = -1.0
C STX ----------------------------------------------------------------------
260 CONTINUE
WRITE(IOOUT,*) 'ENTER THE TYPE OF HIGH CLOUD ACCORDING'
WRITE(IOOUT,*) 'TO THE CODE BELOW'
WRITE(IOOUT,*) '0 - NONE'
WRITE(IOOUT,*) '1 - THIN'
WRITE(IOOUT,*) '2 - THICK'
READ(IOIN,*) IWX(I,4)
WRITE(IOOUT,*) IWX(I,4)
WRITE(IOOUT,*)
IF(IWX(I,4) .LT. 0 .OR. IWX(I,4) .GT. 2) THEN
  WRITE(IOOUT,*) 'YOU HAVE ENTERED AN ILLEGAL VALUE'
  WRITE(IOOUT,*) 'FOR THE HIGH CLOUD TYPE'
GOTO 260
ENDIF
270 CONTINUE

C-149
WRITE(IOOUT,*) 'ENTER THE TYPE OF MIDDLE CLOUD ACCORDING' WRITE(IOOUT,*) 'TO THE CODE BELOW' WRITE(IOOUT,*) '0 - NONE' WRITE(IOOUT,*) '3 - ANY MIDDLE CLOUD' READ(IOIN,*) IWX(I,5) WRITE(IOOUT,*) IWX(I,5) WRITE(IOOUT,*) IF(IWX(I,5) .NE. 0 .AND. IWX(I,5) .NE. 3) THEN WRITE(IOOUT,*) 'YOU HAVE ENTERED AN ILLEGAL VALUE' WRITE(IOOUT,*) 'FOR THE MIDDLE CLOUD TYPE' GOTO 270 ENDIF CONTINUE WRITE(IOOUT,*) 'ENTER THE TYPE OF LOW CLOUD ACCORDING' WRITE(IOOUT,*) 'TO THE CODE BELOW' WRITE(IOOUT,*) '0 - NONE' WRITE(IOOUT,*) '4 - LOW STRATIFORM' WRITE(IOOUT,*) '5 - LOW CONVECTIVE' READ(IOIN,*) IWX(I,6) WRITE(IOOUT,*) IWX(I,6) WRITE(IOOUT,*) IF(IWX(I,6) .NE. 0 .AND. IWX(I,6) .NE. 4 .AND. IWX(I,6) .NE. 5) THEN WRITE(IOOUT,*) 'YOU HAVE ENTERED AN ILLEGAL VALUE' WRITE(IOOUT,*) 'FOR THE LOW CLOUD TYPE' GOTO 280 ENDIF IF(IWX(I,4) .GT. 0) THEN CONTINUE WRITE(IOOUT,*) 'ENTER THE FRACTIONAL SKY COVERAGE FROM' WRITE(IOOUT,*) '0.0 TO 1.0 FOR HIGH CLOUDS' READ(IOIN,*) WX(I,9) WRITE(IOOUT,*) WX(I,9) WRITE(IOOUT,*) IF(WX(I,9) .LT. 0.0 .OR. WX(I,9) .GT. 1.0) THEN WRITE(IOOUT,*) 'YOU HAVE ENTERED AN ILLEGAL VALUE FOR' WRITE(IOOUT,*) 'THE FRACTIONAL COVERAGE OF HIGH CLOUDS' GOTO 290 ENDIF ELSE WX(I,9) = 0.0 ENDIF IF(IWX(I,5) .GT. 0) THEN CONTINUE WRITE(IOOUT,*) 'ENTER THE FRACTIONAL SKY COVERAGE FROM' WRITE(IOOUT,*) '0.0 TO 1.0 FOR MIDDLE CLOUDS' READ(IOIN,*) WX(I,10) WRITE(IOOUT,*) WX(I,10) WRITE(IOOUT,*) IF(WX(I,10) .LT. 0.0 .OR. WX(I,10) .GT. 1.0) THEN WRITE(IOOUT,*) 'YOU HAVE ENTERED AN ILLEGAL VALUE FOR' WRITE(IOOUT,*) 'THE FRACTIONAL COVERAGE OF MIDDLE CLOUDS' GOTO 300 ENDIF ELSE WX(I,10) = 0.0 ENDIF IF(IWX(I,6) .GT. 0) THEN CONTINUE WRITE(IOOUT,*) 'ENTER THE FRACTIONAL SKY COVERAGE FROM' WRITE(IOOUT,*) '0.0 TO 1.0 FOR LOW CLOUDS' READ(IOIN,*) WX(I,11) WRITE(IOOUT,*) WX(I,11) WRITE(IOOUT,*) IF(WX(I,11) .LT. 0.0 .OR. WX(I,11) .GT. 1.0) THEN
WRITE(IOOUT,'*') 'YOU HAVE ENTERED AN ILLEGAL VALUE FOR'
WRITE(IOOUT,'*') 'THE FRACTIONAL COVERAGE OF LOW CLOUDS'
GOTO 310
ENDIF
ELSE
WX(I,11) = 0.0
ENDIF
IF(IWX(I,4) .GT. 0) THEN
320 CONTINUE
ENDIF
IF(IWX(I,5) .GT. 0) THEN
330 CONTINUE
ENDIF
IF(IWX(I,6) .GT. 0) THEN
340 CONTINUE
ENDIF
WRITE(IOOUT,'*') 'ENTER THE BASE HEIGHT OF HIGH CLOUDS'
WRITE(IOOUT,'*') 'FROM 6.1 TO 13.7 KM. IF -1.0 IS ENTERED'
WRITE(IOOUT,'*') 'A DEFAULT VALUE OF 9.0 KM WILL BE USED.'
READ(IOIN,*) WX(I,12)
WRITE(IOOUT,'*') WX(I,12)
WRITE(IOOUT,'*')
IF(WX(I,12) .LT. 6.1 .OR. WX(I,12) .GT. 13.7) THEN
IF(WX(I,12) .NE. -1.0) THEN
WRITE(IOOUT,'*') 'BASE HEIGHT OF HIGH CLOUDS OUT OF RANGE'
GOTO 320
ELSE
WX(I,12) = 9.0
ENDIF
ENDIF
WRITE(IOOUT,'*')
WRITE(IOOUT,'*') 'ENTER THE BASE HEIGHT OF MIDDLE CLOUDS'
WRITE(IOOUT,'*') 'FROM 2.0 TO 6.1 KM. IF -1.0 IS ENTERED'
WRITE(IOOUT,'*') 'A DEFAULT VALUE OF 4.0 KM WILL BE USED.'
READ(IOIN,*) WX(I,13)
WRITE(IOOUT,'*') WX(I,13)
WRITE(IOOUT,'*')
IF(WX(I,13) .LT. 2.0 .OR. WX(I,13) .GT. 6.1) THEN
IF(WX(I,13) .NE. -1.0) THEN
WRITE(IOOUT,'*') 'BASE HEIGHT OF MIDDLE CLOUDS OUT OF RANGE'
GOTO 330
ELSE
WX(I,13) = 4.0
ENDIF
ENDIF
WRITE(IOOUT,'*')
WRITE(IOOUT,'*') 'ENTER THE BASE HEIGHT OF LOW CLOUDS'
WRITE(IOOUT,'*') 'FROM 0.1 TO 2.0 KM. IF -1.0 IS ENTERED'
WRITE(IOOUT,'*') 'A DEFAULT VALUE OF 1.0 KM WILL BE USED.'
READ(IOIN,*) WX(I,14)
WRITE(IOOUT,'*') WX(I,14)
WRITE(IOOUT,'*')
WRITE(IOOUT,*)
IF(WX(I,14) .LT. 0.1 .OR. WX(I,14) .GT. 2.0) THEN
  IF(WX(I,14) .NE. -1.0) THEN
    WRITE(IOOUT,*), 'BASE HEIGHT OF LOW CLOUDS'
    WRITE(IOOUT,*), 'OUT OF RANGE'
  GOTO 340
ELSE
  WX(I,14) = 1.0
END IF
END IF

C-HSTX --- SCENE SHADOWS

IF (WX(1,15) .LT. 0.01) THEN
  IH = -1.
ELSE
  IH = WX(1,15)
ENDIF
WRITE(21,401) 'XSCL',FLOAT(IAERO),IH,WX(1,2)

WRITE(21,398) 'META',WTME,FLOAT(IWX(I,1)),WX(I,7),WX(I,8),WX(I,15), & WX(I,18)
WRITE(21,397) 'METB',WTME,WX(I,3),WX(I,4),WX(I,5),WX(I,6), & FLOAT(IWX(I,8))
WRITE(21,399) 'HCND',WTME,FLOAT(IWX(I,4)),WX(I,9),WX(I,12)
WRITE(21,399) 'LCLD',WTME,FLOAT(IWX(I,5)),WX(I,10),WX(I,13)
WRITE(21,399) 'MCND',WTME,FLOAT(IWX(I,6)),WX(I,11),WX(I,14)
397 FORMAT(A4,6X,6E10.4)
398 FORMAT(A4,6X,6E10.4)
399 FORMAT(A4,6X,4E10.4)

CD  REV 8/18/89
C STX TCM2 ECR ASL-8-1
C ADDED WX(I,18) WIND DIRECTION TO RECORD META
400 CONTINUE

CD  REV 8/18/89
401 CONTINUE

C  READ THE METEOROLOGICAL DATA FOR TIMES
C  EVERY THREE HOURS BEGINNING (NTIM-1)*3 HOURS PRIOR
TO ITIMOT AND ENDING AT ITIMOT.

DO 500 I=1,NTIM

COMPUTE THE TIME, RELATIVE TO TOT, OF THE I' TH OBSERVATION.

WX(I,1)=-3.*(NTIM-I)
K = I - 1
IWX(I,1) = NINT(RECVAL(IBEGIN(18)+K,2))
CALL INTCHK(3,1,IWX(I,1),'META','2ND',1)
IF (IWX(I,1) .EQ. 2) THEN
   WX(I,2) = RECVAL(14,3)
   CALL REALCK(100.0,0.0,WX(I,2),'XSCL','3RD',0.0)
   WX(I,2) = WX(I,2)/25.4
ELSE
   WX(I,2) = 0.0
ENDIF
IWX(I,4) = NINT(RECVAL(IBEGIN(15)+K,2))
IWX(I,5) = NINT(RECVAL(IBEGIN(17)+K,2))
IWX(I,6) = NINT(RECVAL(IBEGIN(16)+K,2))
IWX(I,8) = NINT(RECVAL(IBEGIN(19)+K,6))
DO 505 N = 3,6
   WX(I,N) = RECVAL(IBEGIN(19)+K,N-1)
505 CONTINUE

SKY AND SOLAR IRRADIANCE ALWAYS -1 IN RECORD
WX(I,7) = -1.
WX(I,8) = -1.
WX(I,9) = RECVAL(IBEGIN(15)+K,3)
WX(I,10) = RECVAL(IBEGIN(17)+K,3)
WX(I,11) = RECVAL(IBEGIN(16)+K,3)
WX(I,12) = RECVAL(IBEGIN(15)+K,4)
WX(I,13) = RECVAL(IBEGIN(17)+K,4)
WX(I,14) = RECVAL(IBEGIN(16)+K,4)
WX(I,15) = RECVAL(IBEGIN(18)+K,5)

ASSIGN WIND DIRECTION FROM RECORD
WX(I,18) = RECVAL(IBEGIN(18)+K,6)
CALL INTCHK(2,0,IWX(I,4),'HCLD','2ND',0)
CALL INTCHK(3,3,IWX(I,5),'MCLD','2ND',0)
CALL INTCHK(5,4,IWX(I,6),'LCLD','2ND',0)
CALL INTCHK(3,1,IWX(I,8),'METB','6TH',1)
CALL REALCK(60.0,-60.0,WX(I,3),'METB','2ND',10.0)
CALL REALCK(WX(I,3),-60.0,WX(I,4),'METB','3RD',8.0)
CALL REALCK(70.0,0.0,WX(I,5),'METB','4TH',8.0)
CALL REALCK(200.0,0.1,WX(I,6),'METB','5TH',10.0)
CALL REALCK(1368.0,-1.0,WX(I,7),'HCLD','3RD',-1.0)
CALL REALCK(600.0,-1.0,WX(I,8),'META','4TH',-1.0)
CALL REALCK(1.0,0.0,WX(I,9),'HCLD','3RD',0.0)
CALL REALCK(1.0,0.0,WX(I,10),'MCLD','3RD',0.0)
CALL REALCK(1.0,0.0,WX(I,11),'LCLD','3RD',0.0)

C-HSTX---SCENE SHADOWS---ECR # HSTX-13-1------------------------
C CHANGED THE CLOUD BASE HEIGHT LIMITS TO MATCH THE VISIBLE INPUT LIMITS
C CALL REALCK(20.0,9.0,WX(I,12),'HCLD','4TH',9.0)
C CALL REALCK(8.0,4.0,WX(I,13),'MCLD','4TH',4.0)
C CALL REALCK(4.0,1.0,WX(I,14),'LCLD','4TH',1.0)
C CALL REALCK(13.7,6.1,WX(I,12),'HCLD','4TH',9.0)
C CALL REALCK(6.1,2.0,WX(I,13),'MCLD','4TH',4.0)
C CALL REALCK(2.0,0.1,WX(I,14),'LCLD','4TH',1.0)
C-HSTX---SCENE SHADOWS---------------------------------------------
C CALL REALCK(3.0,0.0,WX(I,15),'META','5TH',3.0)
C STX TCM2 ECR ASL-8-1********************************************
C RECORD CHECK OF WIND DIRECTION

IF (WX(I,5).GT.0.0)
& CALL REALCK(360.,1.,WX(I,18),'META','6TH',270.0)

500 CONTINUE

ENDIF

TRLMAX = WX(1,1) + 6.0

IF (INTER) THEN

525 CONTINUE

WRITE(IOOUT,*) 'ENTER THE NUMBER OF TIMES DURING THE FORECAST PERIOD THAT TARGET/BACKGROUND TEMPERATURES AND/OR SENSOR PERFORMANCE OUTPUT ARE TO BE CALCULATED. ENTER A NUMBER FROM 1 TO 10'
WRITE(*,*)'THE NUMBER OF TIMES DURING THE FORECAST PERIOD THAT THE BACKGROUND/TARGET TEMPERATURES CAN BE COMPUTED DEPENDS ON THE NUMBER OF MET DATA SETS INPUT. FOR EXAMPLE, IF THREE SETS OF MET DATA ARE INPUT ONLY ONE TEMPERATURE PREDICTION MAY BE MADE AT TIME 0.0.'
READ(IOIN,*) NRUNTM
WRITE(IOOUT,*) NRUNTM
WRITE(IOOUT,*) NRUNTM
WRITE(IOOUT,*) IF(NRUNTM .LT. 1 .OR. NRUNTM .GT. 10) THEN
WRITE(IOOUT,*) 'THE NUMBER OF RUN TIMES MUST BE BETWEEN 1 AND 10'
GOTO 525
ENDIF

CD REV 8/18/89
WRITE(21,526)'TIME',FLOAT(NTIM),FLOAT(NRUNTM)
526 FORMAT(A4,6X,2E10.4)
CD REV 8/18/89
DO 530 I = 1,NRUNTM

527 CONTINUE

WRITE(IOOUT,*) 'ENTER THE TIME RELATIVE TO THE TIME OF INTEREST FOR RUN NUMBER',I,'.'
WRITE(IOOUT,*) 'THE TIME MUST BE LESS THAN OR EQUAL TO 0 AND MUST BE IN HH.HH FORMAT'
READ(IOIN,*) TRLTOT(I)
WRITE(IOOUT,*) TRLTOT(I)
WRITE(IOOUT,*) IF(TRLTOT(I).LT.TRLMAX-0.0001.OR.TRLTOT(I).GT.0.0 ) THEN
WRITE(IOOUT,*) 'YOU HAVE ENTERED A RELATIVE TIME WHICH IS OUT OF RANGE. PLEASE ENTER A NUMBER BETWEEN',TRLMAX,'AND 0.0'
GOTO 527
ENDIF

CD REV 8/18/89
IF (I .EQ. NRUNTM) THEN
WRITE(21,528)'NRUN',TRLTOT(1),TRLTOT(2),TRLTOT(3),TRLTOT(4),TRLTOT(5)
528 FORMAT(A4,6X,5E10.4)
CD REV 8/18/89
DO 530 I = 1,NRUNTM

530 CONTINUE

C PUT TRLTOT(N) IN CHRONOLOGICAL ORDER

C DO 533 J = 2,NRUNTM

M = J

532 IF (M .GE. 2) THEN
IF (TRLTOT(M) .GT. TRLTOT(M-1)) THEN
TEMP = TRLTOT(M)
TRLTOT(M) = TRLTOT(M-1)
TRLTOT(M-1) = TEMP
ENDIF
M = M - 1
GOTO 532
ENDIF

533 CONTINUE

ELSE
C READ IN THE NUMBER OF TIMES FOR WHICH TARGET/BACKGROUND
C TEMPERATURES ARE TO BE COMPUTED.
C
NRUNTM = NINT(RECVAL(13,2))
CALL INTCHK(10,1,NRUNTM,'TIME', '2ND', 1)

C READ IN THE TIMES, RELATIVE TO TOT, FOR WHICH TARGET/
C BACKGROUND TEMPERATURES ARE TO BE COMPUTED.
C
ICOUNT = 1
NPNT = 1BEGIN(20)
DO 535 J = 1,NRUNTM
   IF (ICOUNT .EQ. 8) THEN
      ICOUNT = ICOUNT - 7
      NPNT = IPTR(20)
   ENDIF
   TRLTOT(J) = RECVAL(NPNT,ICOUNT)
   ICOUNT = ICOUNT + 1
535 CONTINUE

C CONVERT WX(I,1) TO GMT IN HOURS AND HUNDRED HOURS AND
C FIND APPROPRIATE GMT DAY.
C
DO 560 I=1,NTIM
   CALL TCONVR(TOT,WX(I,1),IDATE,JD,GMT,ITIME)
   JDATE(I) = JD
   JTIME(I) = ITIME
560 CONTINUE

C STX TCM2 ECR ASL-8-1+..............................................
C WRITE INPUT DATA TO FILE FOR TCM2 TO READ
C WHEN TCM2 IS DONE. (PC VERSION ONLY)
C FOR MAINFRAME COMMENT OUT THE FOLLOWING LINES TO THE STX LABEL
OPEN(UNIT=41, FILE='COMMON.DAT', STATUS='UNKNOWN')
WRITE(41,*),(RLATT,RLONG,ELEV,THEAD,SLAT,SLON)
WRITE(41,*),(NTARID,VSPEED,ASP,ALT,IDTG,ITARG)
WRITE(41,*),(NBKID,GNRICB,TBKG,INTER,GNRICT,TTRG)
WRITE(41,*),(IDATE, ITIMOT,YEAR)
WRITE(41,*),(NTIM,NRUNITM,WTME,TRLTOT,TOT)
WRITE(41,*),(ALB, TBAR, IAERO, RG, IH)
WRITE(41,*),JDATE
WRITE(41,*),JTIME
WRITE(41,*),WX
WRITE(41,*),IXW
C*STX*****************************************************************************
6 FORMAT(A4)
1000 FORMAT(1X,I2,A35,I2,A35)
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: THERMB Old Date: 3/19/93
File Name: THERMB.FOR New Date: 9/14/93

Implemented By: Michael Oberlatz

Reason for Revision: The ceiling height assignment was changed to match the assignment in the visible section.

Description of Revision: The ceiling height is now assigned to the lowest cloud layer with a cloud fraction of greater than 0.7, rather than the lowest layer with a cloud fraction greater than 0.0.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: THERMB  Old Date: 9/14/93
File Name: THERMB.FOR  New Date: 10/20/93

Implemented By: Melanie J. Gouveia

Reason for Revision: The ceiling height assignment was changed to match the Air Force definition, and the assignment in the visible section.

Description of Revision: The ceiling height is now assigned to that height at which clouds at and below the height cover more than 4/8 of the sky. Since the Air Force defines cloud coverage in eighths, this means that the coverage must be at least 5/8, or approximately 0.6.

Notes: This definition of ceiling height was taken from AWSR 105-24, Vol. 1, 1 March 1983.

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
SUBROUTINE THERME

THIS SUBROUTINE WAS BLOCKS 2 & 3 OF THE ORIGINAL THERM1 SUBROUTINE.
THE SPLIT WAS TO ACCOMMODATE THE PC VERSION TO ALLOW TCM2 TO RUN THEN
RETURN TO THIS ROUTINE, THE SECOND HALF OF THERM1 TO WRITE THE INPUT OF
TARGAC AND THE OUTPUT OF TCM2.

+STX TCM2 ECR ASL-9-1 +++++++++++++++++++++++++++++++++++++++++++

RE-DIMENSIONED:

WX(17,19) TO WX(11,19), IWX(17,8) TO IWX(11,8)
JDATE(17) TO JDATE(11), JTIME(17) TO JTIME(11)

TCM2 CAN ONLY HANDLE 30 HOURS OF WEATHER DATA VERSUS 48 FOR TCM1

Commons:

COMMON/IOFILE/IOFILE
COMMON/WPARMI/ATEMP(2), ADPT(2), ANGLE, RANGE1, HTINV, HTSEN, IHAZE,
+ VIS, IWX, RR, KFLAG
COMMON/WEATHER/WX(11,19), IWX(11,8), ALB, TCORE, TBAR
COMMON/TIMES/NTIM, RUNTH, TRLTOT(7), IDATE, ITIMOT, TOT,
+ JDATE(11), JTIME(11)
COMMON/ASCENE/NTARID, NBKID, ITARID, IBKID, RLATT, RLONG, ELEV, VSPEED,
+ THEAD
COMMON/OFILE/TITLE
COMMON/THRM/NRMT, TREG, TBKG, TMPC, DEWP, ALT, VS, IAERO, RNRT, CEIL
+ AINV, WINDS, ELEVA, ASP, IDTG, IOP
COMMON/TEMPS/TEMPSF(10)

+STX TCM2 ECR ASL-9-1 +++++++++++++++++++++++++++++++++++++++++++

NEW COMMON BLOCK
COMMON/TEMPS2/TEMPB(10), RNDELT(5)

C-158
CHARACTER *4 FOVTIT(2) UNUSED NOV 92 PSG
DATA ICDLAY/'H','M','L'/
DATA ZDFALT/9.4,1./

WRITE INPUT DATA

IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
  WRITE(NDIRU,2000)
  WRITE(NDIRU,2001) TITLE
  WRITE(NDIRU,2046)NTARID
  WRITE(NDIRU,2003) IDATE, ITIMOT, INT(YEAR)
  WRITE(NDIRU,2004)ASP
  WRITE(NDIRU,2006) (TRLTOT(N),N=1,NRUNTM)
  WRITE(NDIRU,2007)ALB,TBAR
  WRITE(NDIRU,2008)
  ... ... ...

  WRITE(NDIRU,2017) (WX(N,5),N=1,NTIM)
C STX TCM2 ECR ASL-9-1+......................................
C ADDED TARGET HEADING, TARGET ID, BACKGROUND ID, AND YEAR TO
C THE OUTPUT FILE TAC.OUT
C
C STX TCM2----------------------------------------------------
WRITE(NDIRU,2018) (WX(N,6),N=1,NTIM)
WRITE(NDIRU,2019) (WX(N,7),N=1,NTIM)
WRITE(NDIRU,2020) (WX(N,8),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
  WRITE(IOOUT,2000)
  WRITE(IOOUT,2001) TITLE
  WRITE(IOOUT,2046)NTARID
  WRITE(IOOUT,2003) IDATE, ITIMOT, INT(YEAR)
  WRITE(IOOUT,2004)ASP
  WRITE(IOOUT,2006) (TRLTOT(N),N=1,NRUNTM)
  WRITE(IOOUT,2007)ALB,TBAR
  WRITE(IOOUT,2008)

  WRITE(IOOUT,*)'HIT ENTER TO CONTINUE'
  READ(IOOUT,*)
C
WRITE(IOOUT,2009) (JDAT(N),N=1,NTIM)
WRITE(IOOUT,2010) (JTIME(N),N=1,NTIM)
WRITE(IOOUT,2011) (WX(N,1),N=1,NTIM)
WRITE(IOOUT,2012) (IWX(N,1),N=1,NTIM)
WRITE(IOOUT,2013) (WX(N,2),N=1,NTIM)
WRITE(IOOUT,2014) (WX(N,3),N=1,NTIM)
WRITE(IOOUT,2015) (WX(N,4),N=1,NTIM)
WRITE(IOOUT,2016) (WX(N,5),N=1,NTIM)
C STX TCM2 ECR ASL-9-1+......................................
C ADDED TARGET HEADING, TARGET ID, BACKGROUND ID, AND YEAR TO
C THE OUTPUT DISPLAY (SCREEN)
C
C STX TCM2----------------------------------------------------
WRITE(IOOUT,2017) (WX(N,18),N=1,NTIM)
WRITE(IOOUT,2018) (WX(N,19),N=1,NTIM)
WRITE(IOOUT,2019) (WX(N,20),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
  WRITE(IOOUT,2000)
  WRITE(IOOUT,2001) TITLE
  WRITE(IOOUT,2046)NTARID
  WRITE(IOOUT,2003) IDATE, ITIMOT, INT(YEAR)
  WRITE(IOOUT,2004)ASP
  WRITE(IOOUT,2006) (TRLTOT(N),N=1,NRUNTM)
  WRITE(IOOUT,2007)ALB,TBAR
  WRITE(IOOUT,2008)

  WRITE(IOOUT,*)'HIT ENTER TO CONTINUE'
  READ(IOOUT,*)
C
WRITE(IOOUT,2009) (JDAT(N),N=1,NTIM)
WRITE(IOOUT,2010) (JTIME(N),N=1,NTIM)
WRITE(IOOUT,2011) (WX(N,1),N=1,NTIM)
WRITE(IOOUT,2012) (IWX(N,1),N=1,NTIM)
WRITE(IOOUT,2013) (WX(N,2),N=1,NTIM)
WRITE(IOOUT,2014) (WX(N,3),N=1,NTIM)
WRITE(IOOUT,2015) (WX(N,4),N=1,NTIM)
WRITE(IOOUT,2016) (WX(N,5),N=1,NTIM)

C-159
WRITE(IOOUT,2016) (WX(N,4),N=1,NTIM)
WRITE(IOOUT,2017) (WX(N,5),N=1,NTIM)
C STX TCM2 ECR ASL-9-1 ++++++++++++++++ 
C ADDED WIND DIRECTION TO SCREEN
WRITE(IOOUT,2117) (WX(N,18),N=1,NTIM)
C STX TCM2-------------------------------
WRITE(IOOUT,2018) (WX(N,6),N=1,NTIM)
WRITE(IOOUT,2019) (WX(N,7),N=1,NTIM)
WRITE(IOOUT,2020) (WX(N,8),N=1,NTIM)
ENDIF
DO 570 I=1,3
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU, 2022)ICDLAY(I),(IWX(N,3+I),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT, 2022)ICDLAY(I),(IWX(N,3+I),N=1,NTIM)
ENDIF
570 CONTINUE
DO 580 I=1,3
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU, 2023)ICDLAY(I),(WX(N,8+I),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT, 2023)ICDLAY(I),(WX(N,8+I),N=1,NTIM)
ENDIF
580 CONTINUE
DO 590 I=1,3
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU, 2024)ICDLAY(I),(WX(N,11+I),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT, 2024)ICDLAY(I),(WX(N,11+I),N=1,NTIM)
ENDIF
590 CONTINUE
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,2025) (WX(N,15),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,2025) (WX(N,15),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,2069) (IWX(N,8),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,2069) (IWX(N,8),N=1,NTIM)
WRITE(IOOUT,*)'HIT ENTER TO CONTINUE'
READ(IOOUT,*)
ENDIF
C*******************************************************************************
C
C END BLOCK 1

C*******************************************************************************
C
C*******************************************************************************
C
BEGIN BLOCK 2

C
C SET DEFAULTS, CALCULATE AND WRITE
C
SOLAR AND SKY IRRADIANCE.

C*******************************************************************************
C
C
C
PARAMETERS DEFINED BY DEFAULT.

C-160
DO 651 I=1,NTIM

C DEFINE RESTRICTION TO VISIBILITY INDEX FOR INSOLATION MODEL
  IWX(I,3)=0
  IF(WX(I,6).LE.11.)IWX(I,3)=1
C
C COMPUTE PRECIPITATION INDEX FOR INSOLATION MODEL FOR WEATHER
C PREDICTION TIMES.
  IWX(I,7)=0
  IF(IWX(I,1).GT.1)IWX(I,7)=1
C
C CONVERT RAINFALL RATE FROM INCHES/HR TO MM/HR
  WX(I,2)=WX(I,2)*25.4
C
C CONVERT WIND SPEED FROM KNOTS TO METERS/HOUR
  WINDS = WX(I,5)
  WX(I,5)=WX(I,5)*1851.984
651 CONTINUE

C CONVERT STATION ELEVATION FROM FEET TO METERS
  ELEV=ELEV*3.2808
C
C STX TCM2 ECR ASL-9-1 ++++++++++++++++++++++++++++++++++++++++
C REMOVED CODE THAT DEFINED OR REDINED:
C SENSOR ALTITUDE
C BACKGROUND ALBEDO
C AVERAGE AIR TEMPERATURE
C CORE TEMPERATURE
C STX-TCM2-----------------------------------------------------
C LOOP THROUGH CLOUD HEIGHT DATA, IF ANY DEFAULT VALUES
C ARE REQUIRED, FILL IN WITH THE FOLLOWING DEFAULT CLOUD
C BASE HEIGHTS. HIGH CLOUDS=9KM, MIDDLE CLOUDS=4KM,
C LOW CLOUDS=1KM.
  IPRF=0
  DO 680 N=1,NTIM
    DO 680 I=1,3
      IF(WX(N,I+11).LT.0..AND.WX(N,I+8).GT.0.)THEN
        WX(N,I+11)=ZDFALT(I)
        IPRF=1
      END IF
  680 END IF
  CONTINUE
C
C WRITE OUT CLOUD HEIGHT DEFAULTS FOR USER.
C
  IF(IPRF.EQ.1)THEN
    WRITE(IOOUT,2033)
  END IF
  VS = WX(NTIM,6)
  TMPC = WX(NTIM,3)
  DEWP = WX(NTIM,4)
  RNRT = WX(NTIM,2)
  CEIL = -1
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-14-1-----------------------------
C THE CEILING HEIGHT CHECK WAS CHANGED TO BE THE LOWEST CLOUD LAYER WITH A
C FRACTION GREATER THAN 0.7, RATHER THAN THE LOWEST EXISTING CLOUD LAYER
C-HSTX---SCENE SHADOWS---ECR # HSTX-14-2-----------------------------
C CHANGE THE CEILING HEIGHT TO MATCH THE AIR FORCE DEFINITION: THE
C HEIGHT AT WHICH CLOUDS AT AND BELOW THAT HEIGHT COVER MORE THAN 4/8
C OF THE SKY (FROM AWSR 105-24).  5/8 (0.625) IS THE NEXT INCREMENT.
  FRAC = WX(NTIM,11)
  IF (FRAC.GT.0.6) THEN
    CEIL = WX(NTIM,14)
  ELSE
    FRAC = FRAC + WX(NTIM,10)
C
C-161
IF (FRAC.GT.0.6) THEN
  CEIL = WX(NTIM,13)
ELSE
  FRAC = FRAC + WX(NTIM,9)
ENDIF
ENDIF
C-HSTX---SCENE SHADOWS---------------------------------------------------
C
AINV = WX(NTIM,15)
C STX TCM2 -ECR ASL-9-1+-----------------------------------------------
C REMOVED CALL TO RADCOM. TCM2 WILL COMPUTE SOLAR AND/OR SKY IRRADIANCE
C STX-TCM2--------------------------------------------------------------
C
WRITE OUT A SOLAR AND/OR SKY IRRADIANCE VALUES AS COMPUTED
BY SUBROUTINES SKYRAD AND INSOL AT THE METEOROLOGICAL
INPUT TIMES.

IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
  WRITE(NDIRTU,2034)
  WRITE(NDIRTU,2009)(JDATE(N),N=1,NTIM)
  WRITE(NDIRTU,2010)(JTIME(N),N=1,NTIM)
  WRITE(NDIRTU,2011)(WX(N,1),N=1,NTIM)
  WRITE(NDIRTU,2035)(ZEN(N),N=1,NTIM)
  WRITE(NDIRTU,2019)(SOLRAD(N),N=1,NTIM)
  WRITE(NDIRTU,2020)(SKYRAD(N),N=1,NTIM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
  WRITE(IOOUT,2034)
  WRITE(IOOUT,2009)(JDATE(N),N=1,NTIM)
  WRITE(IOOUT,2010)(JTIME(N),N=1,NTIM)
  WRITE(IOOUT,2011)(WX(N,1),N=1,NTIM)
  WRITE(IOOUT,2035)(ZEN(N),N=1,NTIM)
  WRITE(IOOUT,2019)(SOLRAD(N),N=1,NTIM)
  WRITE(IOOUT,2020)(SKYRAD(N),N=1,NTIM)
ENDIF
C STX TCM2 ECR ASL-9-1 +-----------------------------------------------
C SENSOR DEPRESSION ANGLE IS COMPUTED IN FINDR
C STX TCM2-------------------------------------------------------------
C
C END BLOCK 2
C
C BEGIN BLOCK 3
C
C TCM2 COMPUTES TARGET AND BKGND TEMPS, INCLUDING EFFECT OF RAIN.
C
C STX TCM2 ECR ASL-9-1 +-----------------------------------------------
C REMOVED CALLS TO TARG, ALPHA1, ENVMOD
C STX TCM2-------------------------------------------------------------
C
C * * * OUTPUT DATA FOR TCM * * *
C
C WRITE TARGET TEMPERATURES
C
C STX TCM2 ECR ASL-9-1 +-----------------------------------------------
C REMOVED LOGIC TO DETERMINE STAR
C STX --THE ASTERISK IS NO LONGER NEEDED BECAUSE TCM2

C-162
C TAKES ACCOUNT OF RAIN EFFECT ON TARGET/BACKGROUND TEMPERATURES.
C STX TCM2-----------------------------------------------
C
C DETERMINE GMT DATE AND TIME FROM OUTPUT TIMES WHICH ARE RELATIVE
C TO TOT
C LOOP ON OUTPUT TIMES
    DO 739 N=1,NRUNTM
        CALL TCONVR(TOT,TRLTOT(N),IDATE,JD,GMT,ITIME)
        JDATE(N)=JD
        JTIME(N)=ITIME
    739
C WRITE HEADER FOR TARGET TEMPERATURE DATA
    IF(NTARID.NE.0)THEN
        IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
            WRITE(NDIRTU,2036)
            C STX TCM2 ECR ASL-9-1 +++++++++++++++++++++++
            C ALTITUDE IS NOW ENTERED IN FEET CONVERT TO KM
            WRITE(NDIRTU,2053)ASP,(ALT/3280.8)
            C STX TCM2---------------------------------
            WRITE(NDIRTU,2037)(JDATE(N),N=1,NRUNTM)
            WRITE(NDIRTU,2038)(JTIME(N),N=1,NRUNTM)
            WRITE(NDIRTU,2039)(TRLTOT(N),N=1,NRUNTM)
        ENDIF
        IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
            WRITE(IOOUT,2036)
            C STX TCM2 ECR ASL-9-1 +++÷++++++++++++++++++++
            C ALTITUDE IS NOW ENTERED IN FEET CONVERT TO KM
            WRITE(IOOUT,2053)ASP,(ALT/3281.)
            C STX TCM2-------------------------------------
            WRITE(IOOUT,2037)(JDATE(N),N=1,NRUNTM)
            WRITE(IOOUT,2038)(JTIME(N),N=1,NRUNTM)
            WRITE(IOOUT,2039)(TRLTOT(N),N=1,NRUNTM)
        ENDIF
    ELSE
        IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
            WRITE(NDIRTU,2040)IDTG,(TEMPSF(N),N=1,NRUNTM)
        ENDIF
        IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
            WRITE(IOOUT,2040)IDTG,(TTRG,N=1,NRUNTM)
        ENDIF
    ENDIF
    IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
        WRITE(IOOUT,*)'HIT ENTER TO CONTINUE'
        READ(IOOUT,*)
    ENDIF
C-163
C
C***********************************************************************
C
C  END BLOCK 3A

C***********************************************************************
C
IF(NBKID.NE.0)THEN
C PRINT OUT BACKGROUND TEMPERATURES
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,2041)
WRITE(NDIRTU,2037) (JDATE(N),N=1,NRUNTM)
WRITE(NDIRTU,2038) (JTIME(N),N=1,NRUNTM)
WRITE(NDIRTU,2039) (TRLTOT(N),N=1,NRUNTM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,2041)
WRITE(IOOUT,2037) (JDATE(N),N=1,NRUNTM)
WRITE(IOOUT,2038) (JTIME(N),N=1,NRUNTH)
WRITE(IOOUT,2039) (TRLTOT(N),N=1,NRUNTM)
ENDIF
ID=NBKID
IF(.NOT.GNRICB) THEN
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,2042)ID,(TEMPB(N),N=1,NRUNTM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,2042)ID,(TEMPB(N),N=1,NRUNTM)
ENDIF
ELSE
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,2042)ID,(TBKG,N=1,NRUNTM)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,2042)ID,(TBKG,N=1,NRUNTM)
ENDIF
ENDIF
ENDIF
ENDIF
DO NOT PROCEED TO NEXT SECTION UNDER ANY OF THE FOLLOWING CONDITIONS
IF(NTARID.EQ.0.OR.NBKID.EQ.0) STOP '97'
C***********************************************************************
C  FORMAT STATEMENTS
4 FORMAT('1X,'INSUFFICIENT DATA TO COMPUTE AVERAGE AIR TEMPERATURE')
2000 FORMAT('1X,'INPUT DATA')/
2001 FORMAT('1X,'RUN TITLE ='',A10)
2002 FORMAT('1X,'TARGET LOCATION- LATITUDE ='',F6.1,' DEGREES(NORTH',
+ ' POSITIVE)',/18X,'LONGITUDE ='',F7.1,' DEGREES(WEST POSITIVE)',/
+ 18X,'ELEVATION ='',F8.1,' FEET',/18X,'TARGET HEADING ='',F5.1,
+ ' DEGREES')
2003 FORMAT('1X,'TIME-OVER-TARGET- GMT DAY-OF-THE',
+ '-YEAR ='',I4,'/21X,'GMT TIME ='',IS,' HUNDRED HOURS',
+ '/21X,'YEAR ='',IS)
2004 FORMAT('1X,'ASPECT ANGLE='',F5.1,' DEGREES ')
2006 FORMAT('1X,'OUTPUT TIMES(HOURS RELATIVE TO TOT)'=',I0F6.1)
2007 FORMAT('1X,'ALBEDO='',F5.2,'/1X,'AVERAGE AIR ',
+ 'TEMPERATURE(24 HRS)'=',F6.1,' DEGREES CENTIGRADE')
2008 FORMAT('/13X,'TIME DEPENDENT INPUT DATA')/
2009 FORMAT('1X,'GMT DATE',',4X,17I7)
2010 FORMAT('1X,'GMT TIME',',4X,17I7)
2011 FORMAT('1X,'REL.TIME',',4X,17F7.1)
2012 FORMAT('1X,'PRECIP.',',5X,17I7)
2013 FORMAT('1X,'PREcip. RATE',',17F7.2)
2014 FORMAT('1X,'AEROSOL',',5X,17I7)
2015 FORMAT(1X, 'TEMPERATURE ',17F7.1)
2016 FORMAT(1X, 'DEW POINT',3X,17F7.1)
2017 FORMAT(1X, 'WIND SPEED ',17F7.1)
2117 FORMAT(1X, 'WIND DIRECT.',17F7.1)
2018 FORMAT(1X, 'VISIBILITY ',17F7.1)
2019 FORMAT(1X, 'SOLAR IRR. ',17F7.1)
2020 FORMAT(1X, 'SKY IRR. ',17F7.1)
2022 FORMAT(1X, 'CLD. TYPE',A2,1X,17I7)
2023 FORMAT(1X, 'CLD. FRAC.',A2,17F7.3)
2024 FORMAT(1X, 'CLD. HT.',A3,1X,17F7.1)
2025 FORMAT(1X, 'INVERSION HT',17F7.1)
2033 FORMAT(2X, 'THE FOLLOWING DEFAULT CLOUD BASE HEIGHTS ARE',/
       + ' USED WHERE THE USER DOES DEFINE THEM-- HIGH CLOUDS=9KM,'/
       + ' MIDDLE CLOUDS=4KM, LOW CLOUDS=1KM.'
2034 FORMAT(///2X, 'SOLAR AND SKY IRRADIANCE AT METEOROLOGICAL INPUT',/
       + ' TIMES, COMPUTED BY TCM2.'
2035 FORMAT(1X, 'ZENITH ANGLE',17F7.1)
2036 FORMAT(///1X, 'EQUIVALENT BLACKBODY TEMPERATURES OF TARGETS')
2037 FORMAT(1X, 'GMT DATE',20X,10I7)
2038 FORMAT(1X, 'GMT TIME',20X,10I7)
2039 FORMAT(1X, 'REL. TIME',20X,10F7.1)
2040 FORMAT(1X, 'TARGET',I2,20X,10(2X,F5.1))
2041 FORMAT(///1X, 'EQUIVALENT BLACKBODY TEMPERATURES OF BACKGROUNDS')
2042 FORMAT(1X, 'BACKGROUND',I3,16X,10(2X,F5.1))
2043 FORMAT(///1X,A31,3X,A3,3X,A4,' TARGET ID=',I3,2X,A10/
       + ' TIMES, COMPUTED BY TCM2.')
2046 FORMAT(1X, 'TARGET ID=',4I4)
2052 FORMAT(1X, 'BACKGROUND ID=',6I4)
2053 FORMAT(1X, 'ASPECT ANGLE=',F5.1,' DEGREES ALTITUDE=',F7.2,
       + ' KM')
2069 FORMAT(1X, 'CLUTTER ',17I7)
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: LAYERS Old Date: New
File Name: LAYERS.FOR New Date: 10/20/93

Implemented By: Dan DeBenedictis

Reason for Revision: To find a weighted average of the extinction coefficient (BETAA) and the asymmetry parameter (G) for all atmospheric layers that contain some or all of a cloud layer.

Description of Revision: Using the cloud base heights and thickness values, each atmospheric layer is checked for the presence of clouds. When a layer is found to contain a cloud, a weighted average of the extinction coefficient and asymmetry parameter is computed using the percent of cloud in the layer and percentage of the layer that is clear.

Notes: 

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-166
NEW ROUTINE TO COMPUTE THE EXTINCTION COEFFICIENT AND ASYMMETRY PARAMETERS FOR THE ATMOSPHERIC LAYERS THAT CONTAIN CLOUDS.

**NAME:** LAYERS

**TYPE:** FORTRAN SUBROUTINE

**FILENAME:** LAYERS.FOR

**DESCRIPTION:**
Using the cloud base heights and thickness values, each atmospheric layer is checked for the presence of clouds. When a layer is found to have a cloud, a percentage of how much of the layer contains the cloud is computed. Then a weighted average of the extinction coefficient and asymmetry parameters is computed with percentage of cloud and the percentage of clear in the layer.

**INPUT:**
COMMON BLOCK VARIABLES:
- /ATMOS/ Z Array of base heights of the atmos layers. (km)
- /RADIA/ G(20) Array of layer asymmetry parameters.
- /RADIA/ BETA(20) Array of layer extinction coefficients.

**OUTPUT:**
COMMON BLOCK VARIABLES:
- /RADIA/ G(20) Array of layer asymmetry parameters.
- /RADIA/ BETA(20) Array of layer extinction coefficients.

**CALLED ROUTINES:**
NONE

**CALLING SEQUENCE:**
LAYERS (ZC1, THK1, ZC2, THK2, CLDBTA, CLDG)

**INPUT:**
- ZC1 Cloud base height for top cloud. (km)
- ZC2 Cloud base height for lower cloud. (km)
- THK1 Cloud thickness for top cloud. (km)
- THK2 Cloud thickness for lower cloud. (km)
- CLDBTA Array of cloud extinction coefficients.
  - I = 1 for lower cloud layer
  - I = 2 for upper cloud layer.
- CLDG Array of cloud asymmetry parameters.
  - I = 1 for lower cloud layer.
  - I = 2 for upper cloud layer.

**OUTPUT:**
NONE

**CREATED OCTOBER, 1993. HUGHES STX CORPORATION.**

**LIST OF VARIABLES:**
- AMOUNT Amount of the atmospheric layer that contains a cloud.
- BASE1 Cloud base of the top cloud. (km)
- BASE2 Cloud base of the lower cloud. (km)
- PCLD1 Percentage of the layer that is the top cloud.
- PCLD2 Percentage of the layer that is the lower cloud.
- PCLR Percentage of the layer that is clear.
- TOP1 Top of the top cloud. (km)
- TOP2 Top of the lower cloud. (km)
SUBROUTINE LAYERS (ZC1, THK1, ZC2, THK2, CLDBTA, CLDG)

DIMENSION CLDBTA(2), CLDG(2)
COMMON/ATMOS/PRES(21), Z(21), RHO(21), NLEV, NLAY
COMMON/RADIA/A(20), G(20), GP(20), BETAA(20), BETAR(20),
1 TAU(0:20), IO(20,4), I1(20,4), F(20,4), TF(20), FNOT, L
2 NANG, PF(180,20), ANG(180,20), I2(20), TAUP(0:20),
3 ISTAR(20,30,2), PTHRD(30,2), SING(30), AP(20), PHF(30),
4 ALBEDO, SURF(30,4), SURFO(30,2), TAUSTR, VIS, CAPTP(20),
5 B1, B2
C
BASE1 = ZC1
TOP1 = ZC1 + THK1
BASE2 = ZC2
TOP2 = ZC2 + THK2
IF (TOP2.GT.ZC1.AND.ZC1.NE.0.0) TOP2 = ZC1
C
DO I = 18, 2, -1
AMOUNT = 0.0
PCLR = 0.0
PCLD1 = 0.0
PCLD2 = 0.0
C
IF (BASE1.GE.Z(I).AND.BASE1.LT.Z(I-1).AND.BASE1.NE.0) THEN
 IF (TOP1.LE.Z(I-1)) THEN
 AMOUNT = TOP1 - BASE1
 ELSEIF (TOP1.GT.Z(I-1)) THEN
 AMOUNT = Z(I-1) - BASE1
 BASE1 = Z(I-1)
 ENDIF
 PCLD1 = AMOUNT / (Z(I-1) - Z(I))
ENDIF
C
IF (BASE2.GE.Z(I).AND.BASE2.LT.Z(I-1).AND.BASE2.NE.0) THEN
 IF (TOP2.LE.Z(I-1)) THEN
 AMOUNT = TOP2 - BASE2
 ELSEIF (TOP2.GT.Z(I-1)) THEN
 AMOUNT = Z(I-1) - BASE2
 BASE2 = Z(I-1)
 ENDIF
 PCLD2 = AMOUNT / (Z(I-1) - Z(I))
ENDIF
C
IF ((PCLD1 + PCLD2).GT.0.0) THEN
 PCLR = 1 - (PCLD1 + PCLD2)
 G(I) = PCLR*G(I) + PCLD1*CLDG(1) + PCLD2*CLDG(2)
 BETAA(I) = PCLR*BETAA(I) + PCLD1*CLDBTA(1) + PCLD2*CLDBTA(2)
ENDIF
ENDDO
C
RETURN
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: CNTRAS  Old Date: 5/18/93
File Name: TARGAC.FOR  New Date: 10/8/93

Implemented By: Dan DeBenedictis

Reason for Revision: The background reflectance (BKREF) was not being saved to TAC.SAV when climate data was being used. BKREF is now being saved and accessed with record CONTEXTL.

Description of Revision: Added RECVAL(4,2) to the WRITE statement for writing the record CONTEXTL to TAC.SAV. Update the call to REALCK to reference the CONTEXTL record instead of the METD record for BKREF. Change the lower limit of BKREF from 0.0 to 0.01, as specified in the User’s Guide to prevent a divide by zero error.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-169
SUBROUTINE CNTRAS(KTYPE,CNTRST,BKREF)
C
C THIS SUBROUTINE CALCULATES THE TARGET/BACKGROUND CONTRAST
C FOR DIRECT VIEW OPTICS (KTYPE=1), IMAGE INTENSIFIERS
C (KTYPE=2), TV'S (KTYPE=3), OR TEMPERATURE CONTRAST
C (KTYPE=4)
C
C
C STX TCM2 -- ECR ASL-6-1 --
C REMOVED COMMON BLOCK THRM
C REMOVED CALL TO THERML. THERML IS CALLED BY FINDR.
C STX TCM2--

COMMON /IOFILE/IOFILE
COMMON /INOUT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,IRELH,
1 KSTOR,NPLOTU,STDERR
COMMON /IBLOCK/RECVAL(111,7),RECUSE(21),IPTR(21),IBEGIN(22)
COMMON /CHBLCK/RECFLD(21)
INTEGER IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT
INTEGER IRELH,KSTOR,NPLOTU,STDERR,IOFILE
LOGICAL INTER,EFLAG,RECUSE
CHARACTER*4 RECFLD
DIMENSION TGTREF(3,14),BKGREF(3,41)
DATA TGTREF/.081,.121,.229,.114,.228,.458,.146,.273,.561,.105,
1 .216,.444,.184,.291,.457,.097,.139,.224,.145,.146,.139,.136,
2 .252,.498,.267,.238,.198,.375,.394,.428,.449,.422,.370,.323,
3 .326,.336,.497,.594,.730,.230,.287,.373/
DATA BKGREF/.329,.393,.494,.172,.211,.273,.175,.243,.322,
1 .322,.393,.486,.171,.223,.295,.103,.136,.183,.174,.215,.275,
2 .080,.102,.141,.271,.332,.439,.119,.161,.232,.182,.249,.332,
3 .093,.143,.195,.165,.228,.320,.087,.114,.158,.471,.487,.517,
4 .491,.523,.589,.302,.307,.315,.228,.226,.250,.328,.356,.413,
5 .247,.267,.301,.319,.385,.487,.174,.233,.328,.090,.127,.178,
6 .736,.702,.609,.137,.152,.152,.168,.380,.399,.440,.219,.237,.272,
7 .165,.229,.324,.102,.200,.437,.176,.224,.309,.041,.139,.348,
8 .097,.230,.455,.122,.254,.512,.214,.328,.514,.136,.133,.445,
9 .172,.231,.355,.151,.282,.559,.185,.316,.562,.946,.924,.876,
1 .913,.869,.791,.908,.853,.748/

C
C CHOOSE BASIC FUNCTION -- VISIBLE INHERENT CONTRAST
IF(INTER) THEN
11
  CONTINUE
  WRITE(IOOUT,*,'(A,A)') 'YOU MAY ENTER YOUR OWN VALUE OF CONTRAST,'
  WRITE(IOOUT,*,'(A,A)') 'WHICH YOU MAY PRE-COMPUTE USING THE CODE'
  WRITE(IOOUT,*,'(A,A)') 'FASCAT FOR EXAMPLE, OR YOU MAY SELECT FROM'
  WRITE(IOOUT,*,'(A,A)') 'A MENU OF TARGET AND BACKGROUND REFLECTANCES'
  WRITE(IOOUT,*,'(A,A)') 'AND ALLOW THE MODEL TO CALCULATE THE INHERENT'
  WRITE(IOOUT,*,'(A,A)') 'CONTRAST. IF YOU WISH TO ENTER YOUR OWN VALUE'
  WRITE(IOOUT,*,'(A,A)') 'ENTER A 1 TO THE PROMPT BELOW, IF YOU WISH TO'
  WRITE(IOOUT,*,'(A,A)') 'SEE THE MENU ENTER A 0'
  READ(IOIN,*) ICON
  WRITE(IOOUT,*,'(A,A)') 'ENTER THE NUMBER FOR THE TARGET TYPE OR MATERIAL'
  WRITE(IOOUT,*,'(A,A)') '1 - MAN IN FATIGUES 8 - SUMMER CAMOUFLAGE PAINT'
  WRITE(IOOUT,*,'(A,A)') '2 - OLIVE DRAB PAINT 9 - NAVAL GRAY PAINT'
  WRITE(IOOUT,*,'(A,A)') '3 - LIGHT GREEN PAINT 10 - WEATHERED AIRCRAFT SKIN'
  IF(ICON.EQ.0) THEN
    WRITE(IOOUT,*,'(A,A)') 'YOU MUST ENTER A 0 OR 1.'
    GOTO 11
  ENDIF
  IF(ICON.GT.1.OR.ICON.LT.0) THEN
    WRITE(IOOUT,*) 'YOU MAY ENTER YOUR OWN VALUE OF CONTRAST,'
  ENDIF
  IF(ICON.EQ.0) THEN
    WRITE(IOOUT,100)
100  FORMAT('ENTER THE NUMBER FOR THE TARGET TYPE OR MATERIAL','I10)
    IF(ICON.EQ.0) THEN
      WRITE(IOOUT,100)
    ENDIF
GOTO 11
ENDIF
WRITE(IOOUT,100)
100  FORMAT('ENTER THE NUMBER FOR THE TARGET TYPE OR MATERIAL','I10)
   1 ' 1 - MAN IN FATIGUES 8 - SUMMER CAMOUFLAGE PAINT'/
   2 ' 2 - OLIVE DRAB PAINT 9 - NAVAL GRAY PAINT'/
   3 ' 3 - LIGHT GREEN PAINT 10 - WEATHERED AIRCRAFT SKIN'/
C-170
FOREST GREEN PAINT
FOREIGN PAINT 1
FOREIGN PAINT 2
FOREIGN PAINT 3
READ(IOIN,*), ITARGT
WRITE(IOOUT,*), ITARGT
IF (ITARGT.LT.1.OR.ITARGT.GT.14) THEN
   WRITE(IOOUT,*), 'ILLEGAL TARGET TYPE CHOSEN-- TRY AGAIN'
   GO TO 1
ENDIF
WRITE(IOOUT,101)
101 FORMAT('ENTER THE NUMBER FOR THE TYPE OF BACKGROUND /
   1 1 - BARE SOIL 3 - VEGETATION /
   2 2 - ROAD 4 - SNOW /
   READ(IOIN,*), IBKTYP
WRITE(IOOUT,*), IBKTYP
WRITE(IOOUT,*), IF(IBKTYP.EQ.1) THEN
   WRITE(IOOUT,102)
102 FORMAT('ENTER THE NUMBER FOR THE SPECIFIC BACKGROUND /
   1 1 - DRY SAND 11 - DRY SANDY LOAM /
   2 2 - WET SAND 12 - WET SANDY LOAM /
   3 3 - DRY CLAY 13 - DRY PLOWED FIELD /
   4 4 - WET CLAY 14 - WET PLOWED FIELD /
   5 5 - DRY LOAM 15 - SALT /
   6 6 - WET LOAM 16 - CORAL /
   7 7 - DRY LOAM CLAY 17 - GRANITE /
   8 8 - WET LOAM CLAY 18 - SANDSTONE /
   9 9 - DRY LOAM SILT 19 - LIMESTONE /
   10 - WET LOAM SILT 20 - MIXED ROCKS /
   READ(IOIN,*), IBACKG
WRITE(IOOUT,*), IBACKG
ELSE IF (IBKTYP.EQ.2) THEN
   WRITE(IOOUT,103)
103 FORMAT('ENTER THE NUMBER FOR THE SPECIFIC BACKGROUND /
   1 1 - CLAY ROAD 5 - ASPHALT /
   2 2 - DRY DIRT ROAD 6 - CONCRETE /
   3 3 - WET DIRT ROAD 7 - AIRPORT APRON /
   4 4 - SNOW COVERED ROAD 8 - TANK TRAIL /
   READ(IOIN,*), IBACKG
WRITE(IOOUT,*), IBACKG
ELSE IF (IBKTYP.EQ.3) THEN
   WRITE(IOOUT,104)
104 FORMAT('ENTER THE NUMBER FOR THE SPECIFIC BACKGROUND /
   1 1 - LIVE GRASS 6 - DORMANT BROADLEAF TREES /
   2 2 - DEAD GRASS 7 - ACTIVE SHRUBS /
   3 3 - ACTIVE CONIFERS 8 - DORMANT SHRUBS /
   4 4 - DORMANT CONIFERS 9 - LIVE CROP FOLIAGE /
   5 5 - ACTIVE BROADLEAF TREES 10 - DEAD CROP FOLIAGE /
   READ(IOIN,*), IBACKG
WRITE(IOOUT,*), IBACKG
ELSE IF (IBKTYP.EQ.4) THEN
   WRITE(IOOUT,105)
105 FORMAT('ENTER THE NUMBER FOR THE SPECIFIC BACKGROUND /
   1 1 - FRESH DRY SNOW 3 - METAMORPHOSED SNOW /
   2 2 - WET SNOW /
   READ(IOIN,*), IBACKG
WRITE(IOOUT,*), IBACKG
C-171
IBACKG=IBACKG+38
ELSE
    WRITE(IOOUT,*)’ ILLEGAL BACKGROUND TYPE CHOSEN-- TRY AGAIN’
    GO TO 2
ENDIF
IF(IBACKG.LT.0.OR.IBACKG.GT.41) THEN
    WRITE(IOOUT,*)’ ILLEGAL BACKGROUND TYPE CHOSEN-- TRY AGAIN’
    GO TO 2
ENDIF
CD REV 8/18/89
WRITE(21,201)’CONTINTL’,FLOAT(ITARGT),FLOAT(IBACKG)
CD REV 8/18/89
ELSE
    CONTINUE
    WRITE(IOOUT,*)’YOU HAVE CHOSEN TO ENTER YOUR OWN VALUE OF’
    WRITE(IOOUT,*)’INHERENT CONTRAST. YOU MUST ENTER A NUMBER’
    WRITE(IOOUT,*)’WHOSE ABSOLUTE VALUE IS LESS THAN 1.0’
    WRITE(IOOUT,*)’ENTER THE VALUE BELOW.’
    READ(IOIN,*) CNTRST
    WRITE(IOOUT,*) CNTRST
    WRITE(IOOUT,*)
    CNTRST = ABS(CNTRST)
    IF(CNTRST .GT. 1.0) THEN
        WRITE(IOOUT,*)’THE ABSOLUTE VALUE OF CONTRAST MUST BE’
        WRITE(IOOUT,*)’LESS THAN OR EQUAL TO 1.0’
        WRITE(IOOUT,*)
        GO TO 12
    ENDIF
C-HSTX---SCENE SHADOWS---ECR # HSTX-16-1----------------------------------
C MOVE THIS WRITE STATEMENT TO AFTER THE BACKGROUND REFLECTANCE IS KNOWN.
CD REV 8/18/89
C WRITE(21,202)’CONTEXTL’,CNTRST
CD REV 8/18/89
C-HSTX---SCENE SHADOWS------------------------------------------------------
C MINIMUM VALUE OF BKREF HAS BEEN CHANGED FROM 0.0 TO 0.01, AS SPECIFIED
C IN USER’S GUIDE, TO PREVENT DIVIDE BY ZERO ERROR.
    WRITE(IOOUT,*)’YOU MUST ALSO ENTER A VALUE FOR THE BACKGROUND’
    WRITE(IOOUT,*)’REFLECTANCE. ENTER A VALUE BETWEEN 0.01 AND 1.0’
    READ(IOIN,*) BKREF
    WRITE(IOOUT,*) BKREF
    WRITE(IOOUT,*)
    IF(BKREF .LT. 0.01 .OR. BKREF .GT. 1.0) THEN
        WRITE(IOOUT,*)’YOUR INPUT IS OUT OF RANGE. PLEASE ENTER A’
        WRITE(IOOUT,*)’NUMBER BETWEEN 0.01 AND 1.0’
        GOTO 13
    ENDIF
C-HSTX---SCENE SHADOWS------------------------------------------------------
C C-HSTX---SCENE SHADOWS---ECR # HSTX-16-1----------------------------------
C ADD BACKGROUND REFLECTANCE TO THE CONTEXTL RECORD. IT HAS BEEN REMOVED
C FROM THE METD RECORD.
CD REV 8/18/89
C RECVAL(8,7) = BKREF
WRITE(21,201)’CONTEXTL’,CNTRST,BKREF
CD REV 8/18/89
C-HSTX---SCENE SHADOWS------------------------------------------------------
C RETURN
ENDIF
ELSE
    IF (RECFLD(4) .EQ. ’EXTL’) THEN
C BACKGROUND REFLECTANCE IS NO LONGER IN THE METD RECORD. IT HAS BEEN
C MOVED TO THE CONTEXTL RECORD. MINIMUM VALUE HAS BEEN CHANGED FROM
C 0.0 TO 0.01, AS SPECIFIED IN USER'S GUIDE, TO PREVENT DIVIDE BY ZERO
C ERROR.
CNTRST = RECVAL(4,1)
C
BKREF = RECVAL(8,7)
BKREF = RECVAL(4,2)
CALL REALCK(1.0,0.0,CNTRST,'CONT','1ST',1.0)
CALL REALCK(1.0,0.01,BKREF,'CONT','2ND',.5)
C RETURN
ELSE
ITARGT = NINT(RECVAL(4,1))
IBACKG = NINT(RECVAL(4,2))
CALL INTCHK(14,1,ITARGT,'CONT','1ST',3)
CALL INTCHK(41,1,IBACKG,'CONT','2ND',1)
ENDIF
ENDIF

200 FORMAT(10I5)
CD REV 8/18/89
201 FORMAT(A8,2X,2E10.4)
202 FORMAT(A8,2X,E10.4)
CD REV 8/18/89
BKREF=BKGREF(KTYPE,IBACKG)
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,*)'BACKGROUND REFLECTANCE =',BKREF
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,*)'BACKGROUND REFLECTANCE =',BKREF
ENDIF
CNTRST=(TGTREF(KTYPE,ITARGT)-BKREF)/BKREF
RETURN
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: DIFUSE Old Date: New
File Name: DIFUSE.FOR New Date: 10/20/93

Implemented By: Melanie J. Gouveia

Reason for Revision: New routine to compute the components of diffuse radiance. Removing this code from the CONTST routine makes CONTST more modular.

Description of Revision: Set the values of directionally independent and dependent diffuse radiance and direct radiance (forward scattering around cloud edges) to be used for weighted averages. Call PCDIF to compute the averages.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records

C-174
NEW ROUTINE TO COMPUTE COMPONENTS OF DIFFUSE RADIANCE.

This routine computes the components of diffuse radiance.

CALLING SEQUENCE:
DIFUSE (L,NLOCT,SZA, AVEIO,AVEI1,CORR)

INPUT:
L Target-sensor viewing number.
NLOCT Atmospheric layer containing the target.
SZA Solar zenith angle.

OUTPUT:
AVEIO Weighted average directionally independent diffuse radiance term.
AVEI1 Weighted average directionally dependent diffuse radiance term.
CORR Weighted average forward scattering correction term.

HISTORY
CREATED OCTOBER, 1993. HUGHES STX CORPORATION.
COMMON /CLOUD/ICLDF, NUMCLD, MLOOP, CLDBTA(2), CLDG(2), LYRCLD(2),
1 THK1, THK2, THK3, ITY1, ITY2, ITY3,
2 ZC1, ZC2, ZC3, CF1, CF2, CF3, PCF(2), PSCLD
COMMON/RADIA/A(20), G(20), GP(20), BETAA(20), BETAR(20),
1 BETA(20), K(20), P(20), ALPHA(20), BET(20),
2 TAU(0:20), IO(20,4), I1(20,4), P(20,4), TF(20), FNOT,
3 NANG, PF(180,20), ANG(180,20), I2(20), TAUP(0:20),
4 ISTAR(20,30,2), PTHR(30,2), SING(30), AP(20), PHF(30)
5 , ALBEDO, SURF(30,4), SURF0(30,2), TAUSTR, VIS, CAPTP(20),
6 B1, B2

REAL IO, I1, I2, ISTAR, K
REAL SURFA(4)

C LOAD TEMPORARY VARIABLES WITH DIRECT AND DIFFUSE RADIANCE TERMS FOR
C VARIOUS CLOUD SITUATIONS.
C

IF ( ICLDF .EQ. 2 ) THEN
SURFA(1) = 0
SURFA(2) = SURF(L,1)
SURFA(3) = 0
SURFA(4) = 0
CLR0 = 0
AO = IO(NLOCT,1)
BO = 0
AB0 = 0
CLR1 = 0
A1 = I1(NLOCT,1)
B1 = 0
AB1 = 0
ELSEIF ( ICLDF .EQ. 3 ) THEN
SURFA(1) = 0
SURFA(2) = 0
SURFA(3) = 0
SURFA(4) = SURF(L,1)
CLR0 = 0
AO = 0
BO = 0
AB0 = IO(NLOCT,1)
CLR1 = 0
A1 = 0
B1 = 0
AB1 = I1(NLOCT,1)
ELSEIF ( ICLDF .EQ. 5 ) THEN
SURFA(1) = 0
SURFA(2) = 0
SURFA(3) = 0
SURFA(4) = SURF(L,2)
CLR0 = 0
AO = IO(NLOCT,1)
BO = 0
AB0 = IO(NLOCT,2)
CLR1 = 0
A1 = I1(NLOCT,1)
B1 = 0
AB1 = I1(NLOCT,2)
ELSE
SURFA(1) = SURF(L,1)
SURFA(2) = SURF(L,2)
SURFA(3) = SURF(L,3)
SURFA(4) = SURF(L,4)
CLR0 = IO(NLOCT,1)
AO = IO(NLOCT,2)
BO = IO(NLOCT,3)
AB0 = IO(NLOCT,4)

C-176
CLR1 = I1(NLOCT,1)
A1 = I1(NLOCT,2)
B1 = I1(NLOCT,3)
AB1 = I1(NLOCT,4)
END IF

C COMPUTE THE WEIGHTED AVERAGE (OVER POSSIBLE CLOUD SITUATIONS) OF C DIRECTIONALLY INDEPENDENT DIFFUSE RADIANCE.
C
CALL PCDIF (CLR0, A0, B0, ABO,
+ AVEI0, 1, PCF, SZA, NUMCLD, CF1, CF2, SURFA)
C
C COMPUTE THE WEIGHTED AVERAGE (OVER POSSIBLE CLOUD SITUATIONS) OF C DIRECTIONALLY DEPENDENT DIFFUSE RADIANCE.
C
CALL PCDIF (CLR1, A1, B1, AB1,
+ AVEI1, 1, PCF, SZA, NUMCLD, CF1, CF2, SURFA)
C
C COMPUTE THE CORRECTION DUE TO FORWARD SCATTER OF DIRECT RADIANCE C THROUGH CLOUD EDGES
C
CALL PCDIF (DUM, DUM, DUM, DUM,
+ CORR, 2, PCF, SZA, NUMCLD, CF1, CF2, SURFA)
C
RETURN
END
ENGINEERING CHANGE RECORD FOR TARGAC-3

Routine Name: ILMDAT        Old Date: 8/17/93
File Name: ILMDAT.FOR        New Date: 10/20/93

Implemented By: Melanie J. Gouveia

Reason for Revision: All of the conditions included in the significant weather input are now included as part of other inputs.

Description of Revision: Omitted SIGWX from the WRITE statements for writing the record ILUM to TAC.SAV. A dummy value of 0.0 is written instead.

Notes:

As appropriate, attach the following:

1. Code listing with changes highlighted
2. Test records
C******************************************************************************
SUBROUTINE ILMDAT ( IDEV )
C******************************************************************************
C PARAMETER (IDVOP = 1, INTENS = 2, ITELE = 3, IETHER = 4, IUSR = 5)
C
COMMON /INOUT/ INTER, IRPT, EFLAG
COMMON /IOFILE/ IOFILE
COMMON /IOUNIT/ IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT,
+ IRELI, KSTOR, NPLTU, STDERR
COMMON /IBLOCK/ RECVAL(111,7), RECUSE(21), IPTR(21), IBEGIN(22)
C***REV 1/91
COMMON /CHBLCK/ RECFLD(21)
COMMON /ILLUMI/ AL, ILLUM, L22, ACK, IL1, IL2
COMMON /ILDATA/ FMONTH, DAY, YEAR, GTIME, SLAT, SLON, ILR1, ILR2,
+ ILR3, RG, FR1, FR2, FR3, SIGWX, OBSURF, CEILHT,
+ PRTP, FRC, ITARG
COMMON /ILUMCM/ ALTS, AZIS, ALTMN, AZIM, DPHASE, ELUMI, SUNLIT,
+ MOOLIT, TCLSUN, TCLLUN, RCLSUN, RCDLUN, RCLLUN
C
DIMENSION ALL(3)
REAL MOOLIT
INTEGER IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, IRELH
INTEGER KSTOR, NPLOTU, STDERR, IOFILE
LOGICAL INTER, EFLAG, RECUSE
C***REV 1/91
CHARACTER*4 RECFLD
C
DATA ALL / 0.01, 0.001, 0.0001 /
C
IF (IDEV .EQ. INTENS) THEN
C***REV 1/91
ITARG = 2
C******
IF (INTER) THEN
CONTINUE
C***REV 1/91
ILUMA OPTION ADDED TO INTERACTIVE MENU
WRITE(IOOUT,*)' THE I**2 DEVICE MODULE REQUIRES THE AMBIENT'
WRITE(IOOUT,*)' ILLUMINATION AS INPUT. YOU MAY ENTER THIS'
WRITE(IOOUT,*)' INFORMATION BY '
WRITE(IOOUT,*)' A MENU'
WRITE(IOOUT,*)' 1 - TO INPUT A VALUE FOR ILLUMINATION'
WRITE(IOOUT,*)' 2 - EOSAEL ILUMA MODULE TO COMPUTE ','
WRITE(IOOUT,*)' THE ILLUMINATION'
WRITE(IOOUT,*)' CAUTION SHOULD BE USED IN INTERPRETING'
WRITE(IOOUT,*)' RESULTS UNDER LOW LIGHT CONDITIONS.'
WRITE(IOOUT,*)' RESEARCH SHOWS THAT CUTFURAL LIGHTING CAN'
WRITE(IOOUT,*)' SIGNIFICANTLY CONTRIBUTE TO LIGHT LEVELS'
WRITE(IOOUT,*)' UNDER LOW LIGHT CONDITIONS.'
READ(IOIN,*) ICHILL
WRITE(IOOUT,*)ICHILL
WRITE(IOOUT,*)
IF (ICHILL .LT. 0 .OR. ICHILL .GT. 2) THEN
WRITE(IOOUT,*)' YOU HAVE MADE AN ILLEGAL ENTRY'
WRITE(IOOUT,*)' TRY AGAIN'
GOTO 100
ENDIF
IF (ICHILL .EQ. 0) THEN
CONTINUE
C***REV 1/91
SELECT THE PHASE OF THE MOON'
WRITE(IOOUT,*)' 1 -- FULL MOON'
WRITE(IOOUT,*)' 2 -- QUARTER MOON'
WRITE(IOOUT,*)' 3 -- NO MOON'

C-179
READ(IOIN,*) IMOON
WRITE(IOOUT,*)IMOON
WRITE(IOOUT,*)
IF (IMOON .LT. 1 .OR. IMOON .GT. 3) THEN
WRITE(IOOUT,*)'YOU HAVE MADE AN ILLEGAL ENTRY'
WRITE(IOOUT,*)'FOR THE PHASE OF THE MOON -- TRY AGAIN'
GOTO 110
ELSE
AL = ALL(IMOON)
ENDIF
ELSE IF (ICHILL .EQ. 1) THEN
CONTINUE
WRITE(IOOUT,*)' ENTER THE ILLUMINATION IN FOOT CANDLES:'
WRITE(IOOUT,*)' **NOTE** THE ILLUMINATION MUST BE ',
+ 'STRICTLY LESS THAN 1'
READ(IOIN,*) AL
WRITE(IOOUT,*)AL
WRITE(IOOUT,*+)
IF (AL .GT. 1.0) GOTO 120
ENDIF
C***REV 1/91 CALLS ILUMA MODULE TO CALCULATE ILLUMINATION, IF THE
C "INTL" FLAG IS SET, AND SOLAR AND LUNAR POSITIONS
CALL ILUMA ( 1, IERR )
C
C***REV 1/91 ILLUMINATION INTERNALLY CALCULATED BY ILUMA
C CONVERTED FROM LUMENS/SQ METER TO FOOT CANDLES
IF (ICHILL .EQ. 2) THEN
AL = ELUMI / 10.76
IF (AL .LT. 0.0001 .OR. AL .GT. 0.01) THEN
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,*+)
WRITE(IOOUT,*)'*** ILLUMINATION VALUE OUT OF RANGE'
WRITE(IOOUT,*)' DEFAULT VALUE WILL BE USED'
WRITE(IOOUT,*)
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,*+)
WRITE(NDIRTU,*)'*** ILLUMINATION VALUE OUT OF RANGE'
WRITE(NDIRTU,*)' DEFAULT VALUE WILL BE USED'
WRITE(NDIRTU,*)
ENDIF
AL = ALL(l)
ENDIF
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-18-1-----------------------------
C ELIMINATE SAVE OF SIGWX; THIS VARIABLE IS NO LONGER USED.
C***REV 1/91
DUM = 0.0
IF (ICHILL .EQ. 0) THEN
WRITE(21,1200)'ILUMEXTL',-FLOAT(IMOON),DUM,OBSURF,PRTYP
ELSE IF (ICHILL .EQ. 1) THEN
WRITE(21,1200)'ILUMEXTL',AL,DUM,OBSURF,PRTYP
ELSE
WRITE(21,1200)'ILUMINTL',AL,DUM,OBSURF,PRTYP
ENDIF
C-HSTX---SCENE SHADOWS---------------------------------------------
C
 ELSE
IF (.NOT. RECUSE(7))) THEN
CHILL = -1.0
ELSE
CHILL = RECVAL (7,1)
ENDIF
C***REV 1/91 CALLS ILUMA MODULE TO CALCULATE ILLUMINATION, IF THE
"INTL" flag is set, and solar and lunar positions

CALL ILUMA (1, IERR)

C***REV 1/91 Batch option to calculate illumination internally by ILUMA
or externally supplied by the user

IF (RECFLD(7) .EQ. 'EXTL') THEN
   CHILL = RECVAL(7,1)
ELSE
   CHILL = ELUMI / 10.76
ENDIF

IF (CHILL .LT. 0.0) THEN
   ICHILL = NINT(ABS(CHILL))
   CALL INTCHK(3,1,ICHILL,'ILUM','1ST',1)
   AL = ALL(ICHILL)
ELSE
   AL = CHILL
   IF (RECFLD(7) .EQ. 'EXTL') THEN
      CALL REALCK(0.01,0.001,AL,'ILUM','1ST',99999.0)
      IF (AL .EQ. 99999.0) AL = ALL(1)
   ELSE
      IF (AL .LT. 0.0001 .OR. AL .GT. 0.01) THEN
         IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
            WRITE(IOOUT,*)
            WRITE(IOOUT,*)'*** ILLUMINATION VALUE OUT OF RANGE'
            WRITE(IOOUT,*)' DEFAULT VALUE WILL BE USED'
         ENDIF
         IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
            WRITE(NDIRTU,*)
            WRITE(NDIRTU,*)'*** ILLUMINATION VALUE OUT OF RANGE'
            WRITE(NDIRTU,*)' DEFAULT VALUE WILL BE USED'
         ENDIF
      ELSE
      ENDIF
   ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

ILLUM = -INT(ALOG10(AL) + 1.0)
CUV = -INT(ALOG10(AL))
ACK = 10.00 ** (- (1 + ILLUM))

C***REV 1/91 ELSE IF (IDEV .EQ. IDVOP .OR. IDEV .EQ. ITELE) THEN
C***REV 1/91 ILUMA option added to interactive menu

ITARG = 2
IF (INTER) THEN
   CONTINUE
C***REV 1/91 WRITE(IOOUT,*)' THIS DEVICE MODULE REQUIRES THE AMBIENT'
   WRITE(IOOUT,*)' ILLUMINATION AS INPUT. YOU MAY ENTER THIS'
   WRITE(IOOUT,*)' INFORMATION BY '
   WRITE(IOOUT,*)' 0 - TO INPUT A VALUE FOR ILLUMINATION'
   WRITE(IOOUT,*)' 1 - EOSAEL ILUMA MODULE TO COMPUTE '
   WRITE(IOOUT,*)' THE ILLUMINATION'
   READ(IOIN,*) ICHILL
   WRITE(IOOUT,*)ICHI
   WRITE(IOOUT,*)ICHILL
   WRITE(IOOUT,*)ICHILL
   IF (ICHILL .NE. 0 .AND. ICHILL .NE. 1) THEN
      WRITE(IOOUT,*)' YOU HAVE MADE AN ILLEGAL ENTRY'
      WRITE(IOOUT,*)' TRY AGAIN'
      GOTO 130
   ENDIF
ENDIF
IF (ICHILL .EQ. 0) THEN
   CONTINUE
WRITE(IOOUT,*)' ENTER THE ILLUMINATION IN FOOT CANDLES:'
WRITE(IOOUT,*)' **NOTE** THE ILLUMINATION MUST BE ',
+ 'GREATER THAN 1'
READ(IOIN,*) AL
WRITE(IOOUT,*) AL
WRITE(IOOUT,*)
IF (AL .LT. 1.0) GOTO 140
ENDIF
C***REV 1/91 CALLS ILUMA MODULE TO CALCULATE ILLUMINATION, IF THE
C "INTL" FLAG IS SET, AND SOLAR AND LUNAR POSITIONS
CALL ILUMA ( 1, IERR )
C***REV 1/91 ILLUMINATION INTERNALLY CALCULATED BY ILUMA
CONVERTED FROM LUMENS/SQ METER TO FOOT CANDLES
IF (ICHILL .EQ. 1) THEN
AL = ELUMI / 10.76
IF (AL .LT. 1.0 .OR. AL .GT. 10000.0) THEN
WRITE(IOOUT,*)
WRITE(IOOUT,*)'*** ILLUMINATION VALUE OUT OF RANGE'
WRITE(IOOUT,*)' DEFAULT VALUE WILL BE USED'
WRITE(IOOUT,*)
ENDIF
IF(IOFILE.EQ.0.OR.IOFILE.EQ.2) THEN
WRITE(IOOUT,*)
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,*)
WRITE(NDIRTU,*)'*** ILLUMINATION VALUE OUT OF RANGE'
WRITE(NDIRTU,*)' DEFAULT VALUE WILL BE USED'
ENDIF
AL = 1000.0
ENDIF
C
C-HSTX---SCENE SHADOWS---ECR # HSTX-18-1---------------------------
C ELIMINATE SAVE OF SIGWX; THIS VARIABLE IS NO LONGER USED.
C***REV 1/91 DUM = 0.0
IF (ICHILL .EQ. 0) THEN
WRITE(21,1200)'ILUMEXTL',AL,DUM,OBSURF,PRTYP
ELSE
WRITE(21,1200)'ILUMINTL',AL,DUM,OBSURF,PRTYP
ENDIF
C-HSTX---SCENE
ELSE
C***REV 1/91 BATCH OPTION TO CALCULATE ILLUMINATION INTERNALLY BY ILUMA
C OR EXTERNALLY SUPPLIED BY THE USER
C CALLS ILUMA MODULE TO CALCULATE ILLUMINATION, IF THE
C "INTL" FLAG IS SET, AND SOLAR AND LUNAR POSITIONS
CALL ILUMA ( 1, IERR )
IF (RECFLD(7) .EQ. 'EXTL') THEN
AL = RECVAL(7,1)
CALL REALCK(10000.0,1.0,AL,'ILUM','1ST',1000.0)
ELSE
AL = ELUMI / 10.76
IF (AL .LT. 1.0 .OR. AL .GT. 10000.0) THEN
WRITE(IOOUT,*)
WRITE(IOOUT,*)'*** ILLUMINATION VALUE OUT OF RANGE'
WRITE(IOOUT,*)' DEFAULT VALUE WILL BE USED'
WRITE(IOOUT,*)
ENDIF
IF(IOFILE.EQ.1.OR.IOFILE.EQ.2) THEN
WRITE(NDIRTU,*)
WRITE(NDIRTU,*)'*** ILLUMINATION VALUE OUT OF RANGE'
WRITE(NDIRTU,*)' DEFAULT VALUE WILL BE USED'
ENDIF

C-182
WRITE(NDIRTU,*)
ENDIF
AL = 1000.0
ENDIF
ENDIF
IL1 = 4 - INT(ALOG10(AL))
ACK = 1000.0 * 10.0 ** (1 - IL1)
IL2 = IL1 +1
IF (ALOG10(AL) - FLOAT(INT(ALOG10(AL)))) .EQ. 0.0) THEN
  IL2 = IL1
  ACK = ACK * 10.0
ENDIF
ELSE
  RETURN
ENDIF
1000 FORMAT(10I5)
1100 FORMAT(8F10.7)
C***REV 1/91
1200 FORMAT(A8,2X,4E10.4)
C
RETURN
END
C
APPENDIX D: TARGAC Small-Scale Feature Shadows
Design Document

The report presents an overview of the problem, reviews the scope of this effort, details assumptions, references relevant documents, and provides a design overview. This appendix focuses on the design specifics of the TARGAC-2 code modifications to support small-scale feature shadow-cued detection and shadow-induced clutter. The areas treated include (1) modifications to existing code and (2) the addition of new modules and supporting data structures.

The modifications to TARGAC-2 (the existing TARGAC code) are discussed first with the aid of high-level flowcharts of the TARGAC-2 code, and some high-level dataflow diagrams. The new modules are discussed with the aid of detailed data flow diagrams. The dataflow diagrams specify the information flow and transformations that must be performed to achieve the new functionality. These dataflow representations are further factored into modules as represented in the software hierarchy diagram presented at the end of the appendix.

During the course of this research, it was noted that the PC and UNIX versions of the TARGAC-2 code are different in some ways. First, it seems that the routines are partitioned up among files different between the PC and UNIX versions. Second, both versions seem to have several routines that compute approximately the same functions -- probably due to work by multiple programmers. All of the modifications in this design were applied to the UNIX version.

1. Code Modifications

The modifications to TARGAC-2 include minor changes in the CNTRAS module, modifications to the input codes to support the collection of required scene statistics, and an additional loop inserted into the main routine to support multiple calls for different viewer angles relative to the sun/moon and to display the detection and recognition probabilities.

Figure 1 shows the current overall control flow for the interactive visible and near-IR codes. Data on background scene, atmospherics, etc. are input and various initializing computations are performed. The control then enters the TARGAC routine FINDR where the detection and output are performed.
In FINDR, more initializations are performed and the system enters an outer loop to compute detection probabilities in the first transit and recognition probabilities in the second transit. Inside the detect loop, a smaller convergence loop finds the maximum ranges for detection at various levels of probability. The results are accumulated in a table as the loop is traversed at increasing ranges. As long as the detection probability remains non-zero, the range is increased and the computation is repeated. When the target can no longer be detected, the loop fails and the accumulated tables of ranges are output.

The main control modification (as shown in figure 2) will be to add another loop to compute detection and recognition probabilities for various viewing angles relative to the
source of illumination. An additional check can be made to test if the level of direct to
diffuse light is high enough to cause shadow effects. If no shadows exist in the scene, the
probabilities of detection and recognition will not vary with view angle. Additional minor
modifications will be made in the input and output routines to support the additional scene
information required and to support table outputs for multiple viewing angles.

Input data

Initialize

Set Mode=detect

Compute detection probability at range

Pd > 0.0?
  Increase range

Output tables

Mode=Recognition?
  Set Mode=recognition

All views done?
  Set next view angle

Figure 2. Modified Control Flow.
2. New and Modified Modules

The new modules include routines to support the use of shadow information in computing clutter and shadow-cued probabilities of detection and recognition. They are called SCR, ACQUIR, and SHADOWS. The SCR module computes the shadow-enhanced signal-to-clutter ratios which are passed to the ACQUIR module. ACQUIR computes probabilities of detection and recognition based on Schmeider's clutter experiments. Although the current TARGAC code includes an ACQUIR module, it will undergo a major rewrite and hence is here considered a new module. The SHADOWS module provides data to support the shadow-cued detection algorithms introduced in the new ACQUIR module.

To better understand the new modules in context, consider the excerpt of TARGAC-2 shown in figure 3. Data on the background, atmospherics, and target are combined in the computation labelled CONTRAST to yield the apparent contrast at the target. (The CONTRAST computation in figure 3 includes the TARGAC-2 CNTRAS subroutines and other calls.) The apparent contrast is passed to the RCF module which uses the sensor characteristics and the sky-to-ground ratio (SGR) to compute the resolvable cycles per milliradian of angle subtended by the target given the contrast. The resolved cycles are then used with target range and size to compute the probability of detection.

![Diagram](image-url)

*Figure 3. Existing TARGAC Code (Excerpt).*
Figure 4 shows most of the new code (less some routines for additional data input). There are two major changes: (1) the path through CNTRAS, RCF, and ACQUIR has been augmented by a second parallel path, and (2) two new modules have been added to support shadow-based clutter and detection algorithms. New data areas have also been added to support the new algorithms. In figure 4, the modified modules and data areas are shaded, and the new modules and data areas are shown with heavy outlines.

Figure 4. Modified TARGAC Code (Excerpt).

The two parallel paths are used to compute independent resolvable cycles for target-on-shadow and target-on-background. In actuality, the present TARGAC code computes resolvable cycles for target-on-background. This computation was modified slightly, and a second path will be added to compute resolvable cycles for target-on-shadow. The CNTRAS module was modified to use the appropriate weighting factors that result from splitting the target into separate portions adjacent to shadow and background. Note that for certain view angles, the target completely obscures the shadow so this component would be zero.

The two new modules work together with the rewritten ACQUIR module to combine the target-on-shadow and target-on-background detection probabilities. The SCR
module computes the signal-to-clutter ratio (SCR) based on statistical scene data and shadow effects that arise from viewing angle. The SHADOWS module computes the expected radiance within shadow areas for use in computing target-to-shadow contrast. It also computes a weighting factor that is used to eventually combine the shadow and background detection probabilities within ACQUIR.

We now present these three new modules in detail. Each module description includes a processing narrative and annotated data flow diagram.

2.1 SCR Module

The SCR module computes the signal-to-clutter ratio (SCR) for an expected scene. The SCR is used with empirically derived data (see Schmeider) to alter the maximum ranges for detection and recognition based on expected clutter in the target area. Generally, the detection and recognition ranges decrease with increasing clutter.

The SCR computation presented here addresses two issues related to clutter not previously treated in Tactical Decision Aid codes. First, SCR computes a clutter index for the scene that is based on statistical information about the scene. Second, it includes clutter effects that are due to shadows. The adjustments for shadow effects are fairly sophisticated. The viewed shadow geometries depend both on the sun angle relative to the target and the viewing angle relative to the sun.

The clutter computation requires statistical information about clutter objects in the scene. These include 3-dimensional clutter objects such as trees, bushes, rock outcroppings, etc., and flat clutter objects such as soil types and ground cover. Once obtained, the system computes the expected clutter index.

In the modified TARGAC code, the statistical information may be obtained by one of two methods. The simplest allows the user to choose from a variety of sample scenes similar to those in Chapter 15 of the Electro-Optical Tactical Decision Aid User's Manual [Freni, et al. 1993]. Several menu sessions can be performed to obtain detailed statistical information in many "layers," including large scale ground covers, small scale ground covers, and various types of 3-dimensional clutter objects. The system would then compute the required statistical information keyed to the selected scenes. A second method will allow the more sophisticated user to input statistics derived from imagery. In this case,
precise information could be obtained from several suspected target locations and fused into an accurate model of scene clutter. These approaches contrast with the more traditional approach of selecting low, medium, and high clutter indicators from a set of three sample images.

2.1.1 Overview of the Computation

In this section we present an overview of the SCR computation. Figure 5 shows the basic order of processing. Figure 6 is a data flow diagram with explicit computations called out by index numbers appearing on specific functions in the diagram.

The computation begins with statistical scene parameters elicited in functions 1 and 2 by one of the methods described above. Scene parameters consist of expected object densities in the x-axis and y-axis, mean and variance for size (assuming a gaussian distribution), mean and variance for reflectance, and various aspect and inclination parameters. The scene parameters are referred to as flat parameters and 3-d parameters for 2-dimensional and 3-dimensional clutter respectively. The 3-d parameters are then used to
compute shadow geometries in function 4, assuming spherical clutter objects. These comprise a third set of parameters referred to as *shadow parameters*.

The flat, shadow, and 3-d parameters are then passed through several transformations (described in the main report) to account for viewing angle as determined by the height of the observer above the target and the viewing angle relative to the sun or moon (function 3). All three parameters are compressed in the y-axis (relative to the viewer's frame of reference) to account for scene compression due to low depression angles (function 5). This transformation is required because experiments have shown that detection range decreases with depression angle (see Schmeider et al. 1982).

The shadow and flat parameters are then further transformed to account for compressed aspect ratio (function 7) and angle of inclination for shadows induced by the viewing angle relative to the illumination source (function 6). Note that the 3-d parameters retain the same aspect ratios because they are assumed to be spherical and have the same aspect from all viewing angles.

Once the parameters have been adjusted for viewing angle, they are combined into a single clutter index $CI$ as follows. Conceptually, a sample region is broken into cells roughly the size of the target, and the background radiance is computed for each cell. The clutter index is the root-mean-square of variations in radiance between adjacent cells as depicted in the equation below.

$$CI = \sqrt{\frac{\sum \sigma_i^2}{N}}$$

Here the summation of variance is taken over adjacent cells and divided by the total number of adjacent components in the sum.

The CI computation can be performed in one of two ways. The first is to use the statistical information on scene objects and perform a closed form solution assuming independence between the various components. The second is to perform a fast Monte-Carlo computation on a small synthetically generated scene. This second approach was adopted for the research effort because of its simplicity and flexibility.
In the Monte-Carlo approach, the flat, shadow, and 3-d parameters are used to fill a small 2-dimensional matrix with radiance values. The fill is performed in several passes. First, the flat parameters are filled in function 9 starting with the coarsest information and finishing with the finest grain information. The shadow parameters are then used to attenuate the flat radiances in function 10. Finally, the 3-d parameters are used to overlay the 3-dimensional clutter objects on the background (function 11). It should be noted that the clutter objects are overlaid independent of the shadow information. It was believed that this simplifying assumption would still result in a good approximation of the clutter index for synthetic scenes.

Once the matrix has been filled in, it is used to compute the average background radiance in function 12, and to compute the clutter index in function 13. The final SCR computation is performed in function 14.

2.1.2 Supporting Data Structures

We now describe two language independent data structures required to support the SCR module. The first is a record type holding statistical scene information and the second is a matrix for the Monte-Carlo computation of the clutter index.

The clutter objects are not stored individually; rather, a statistical summary is stored for clutter that is stratified into several layers. There are a total of six layers. The first three layers give statistical parameters for small, medium, and large scale ground coverings. The next two layers consist of small and large scale 3-d clutter objects. Each layer is described by a single "record" or collection of related variables.

All clutter objects are assumed to be circular. Flat objects are described by ellipses, and 3-d objects are described by spheres. The record summarizes the average densities of objects in the x and y planes, parameters for a gaussian distribution (i.e., mean and variance) on reflectance (for trees and ground coverings), and inclination. The size of objects in the layer as measured by the major axis is assumed to follow a gaussian distribution, and the relation between major-axis and minor-axis is summarized by the aspect field.
A language-independent representation of the data stored for each layer is shown below in record format.

**LAYER_REC**

- **X_DENSITY** - density of object spacing in x-axis
- **Y_DENSITY** - density of object spacing in y-axis
- **MEAN_SIZE** - assume gaussian distribution on object sizes in layer
- **VAR_SIZE** - variance of the object sizes in layer
- **ASPECT** - but all objects have the same aspect imposed by view angles
- **MEAN_REF** - assume gaussian distribution on object reflectances in layer
- **VAR_REF** - variance of the object reflectances in layer
- **INCLINATION** - all objects have the same inclination (used for shadows)

**END_REC**

The computation of the clutter index uses a small (50 by 50) table of cells. Each cell covers an area of the scene from the viewer’s frame of reference that is roughly the same area as the expected target.

### 2.1.3 Data flow Representation

The computation appears in data flow representation in figure 6. The diagram consists of three types of object. The **data areas** hold information for repeated use, and are depicted as boxes. **Data transformations** are depicted as circles, and accept **data tokens** on their inputs and produce transformed tokens on the outputs. Descriptions are provided for all data areas, tokens and transformations appearing in the diagram.
Figure 6. Data Flow Representation.
2.1.3.1 Data Areas

A) Viewer data - Viewer height and viewing angle relative to a ray from target to sun.

B) Sun/moon angles - Azimuth and elevation of the source of direct illumination relative to the target. These values are stored in the variables AZISUN, ALTSUN, AZIMOO, ALTMOO once they are calculated in the subroutine ILLUM located in the file AUXLRY/rtdleib.f.

C) Illumination - Illumination intensity at ground level. Sum of direct and diffuse components, which are calculated in subroutine DIRDIF from the file AUXLRY/rtdleib.f and stored separately in the variables FDIR and FDIF. Solar and lunar illumination are available in the variables SUNLIT and MOOLIT from the routine ILLUM in AUXLRY/rtdleib.f.

D) Dirdif ratio - The ratio of direct to diffuse illumination intensities at ground level. The direct and diffuse components are calculated in subroutine DIRDIF from the file AUXLRY/rtdleib.f and stored separately in the variables FDIR and FDIF. Solar and lunar illumination are available in the variables SUNLIT and MOOLIT from the routine ILLUM in AUXLRY/rtdleib.f.

E) Target data - Target location in desired coordinate system, target reflectance, type and size. The target location is stored in the variables RLATT and RLONG inside the subroutine FINDR which is in TARGAC/targac.f. Reflectance for different targets is either read from a table or received from the operator inside the subroutine CNTRAS which is in the file TARGAC/targac.f. Type is held in the variable ITARGT inside CNTRAS, and the size is stored in different locations, one of which is DIM in the subroutine FINDR inside targac.f.

F) Average background radiance - Intensity of background radiance averaged over a large background. Computed from the 2-d, 3-d, and shadow information in scene.

G) SCR - Signal-to-clutter ratio used to compute probability of detection and probability of recognition.

H) View angles - The elevation and azimuth of the observer relative to the target.
2.1.3.2 Data Tokens

Flat parms - Parameters for two categories of ground cover: large scale cover encompassing moderately large land areas and small scale covering areas roughly the size of targets. Parameters include means and variances for reflectance, diameter, and densities in the x and y coordinates.

3d parms - Parameters for two categories of 3-dimensional clutter objects: trees and objects roughly the size of targets (i.e., shrubs, etc). Parameters include means and variances for reflectance, diameter, and densities in the x and y coordinates.

Shadow parms - Parameters describing shadow geometries (i.e., no luminance data) for the two types of 3-d clutter objects provided. These include major axis, minor axis, inclination from the y-axis, and densities in the x- and y-axes.

View angles - The elevation and azimuth of the observer relative to the target.

Obscuration parms - The fraction of shadows obscured by the objects that cast them. Computed from sun and view angles.

Background matrix - A small (roughly 50 by 50) matrix of cells used for a Monte-Carlo computation of the clutter index. Cells are roughly the size of a target, and filled with radiance values derived from the flat, 3-d, and shadow parms above.

Clutter index - The rms clutter index presented in Schneider, computed from the filled background matrix.

2.1.3.3 Data Transformations

1) Input flat data - The information to compute the flat parms are provided by an outside source. This could range from a simple user interface asking the user to choose from a menu of sample pictures, to a sophisticated automatic analysis of imagery. Currently, most of this data is gathered through the subroutine CNTRAS which is located in the file TARGAC/targac.f. The data is either read in from data files or through use of a text menu interface to the operator. Depending upon the source of the data, different approaches are required to extract the appropriate statistical parameters used to compute the SCR.
2) **Input 3-d data** - Same as above but for 3-d clutter objects in the scene.

3) **Compute view angles** - Accepts the viewer height and view angle relative to the sun, and computes the elevation and azimuth of the viewer relative to the target.

4) **Compute shadow parms** - Computes the appropriate shadow parameters assuming spherical clutter objects. Parameters include major and minor axes, inclination, and x and y densities.

5) **Compress view** - Accepts view angles and modifies the densities in the y-axis. The compression represents how objects separated in the y-dimension (relative to viewer's frame of reference) appear closer together when viewed at low depression angles.

6) **Adjust inclination** - Accepts view angles and adjusts the inclination of ellipses in the viewer's frame of reference.

7) **Compress aspect** - Accepts view angles and modifies the y-dimensions (aspect) relative to the viewer frame of reference. The compression represents how flat objects appear compressed in the y-dimension (viewer's frame of reference) when viewed at low depression angles.

8) **Compute obscuration** - Accepts sun and viewer angles and computes the fraction of an object's shadow that would be obscured by the object when viewed from the viewer's position.

9) **Fill matrix** - Fills the matrix with radiance values for flat clutter. Reflectances are first inserted based on the flat parms corrected for the viewer's frame of reference. Large scale clutter is inserted first, and then small scale clutter is overlaid. A second pass then uses the radiance at ground to replace reflectance values with irradiance values.

10) **Shadow attenuation** - The background matrix containing irradiance values is now adjusted to account for shadows. Cells to update are chosen according to the shadow parms and the dirdif ratio is used to attenuate the cell's irradiance by the appropriate factor.

11) **Overlay 3-d** - The background matrix is overlaid with irradiance values for 3-d clutter objects according to the 3-d parms. Spherical Lambertian reflectors are assumed.

D-14
12) *Compute average background* - A pass is made over the background matrix to compute the average irradiance of the background. This value is used in later calculations to compute contrast at the target.

13) *Compute clutter index* - A pass is made over the matrix to compute the rms clutter index presented in Schmeider.

14) *Compute SCR* - The clutter index is combined with the average background irradiance and target irradiance to obtain the SCR presented in Schmeider.

### 2.2 SHADOWS Module

The SHADOWS module computes target shadow information to support the use of shadow information for target cueing. The first output is the target shadow radiance based on the reflectance of the synthetic background computed in the SCR module. The second output is a weighting parameter that expresses how much of the target is adjacent to its shadow, and how much of the target is adjacent to the background. The weighting factor is used by the ACQUIR Module to properly combine the separately computed resolvable contrasts against shadow and background into a single probability of detection.

#### 2.2.1 Overview of the Computation

The computation is fairly straightforward as shown in figure 7. In what follows we also reference the data flow diagram (figure 8) at the end of this section. Each data flow function is referred to by index number.

The target shadow length is computed in function 1. The geometry is adjusted for viewer angle relative to the sun in function 2 and passed to two independent computations. The first of these computes the fraction of target area adjacent to shadow versus the area adjacent to background, or *Pd ratio*. This ratio is computed in functions 4 and 5.

The second independent computation calculates the expected radiance within the target shadow. It uses the ratio of direct to diffuse light computed in function 6 to compute the actual shadow radiance in function 3.
Compute Target Shadow

Compute Shadow Geometry

Compute Pd Ratio

Compute Expected Radiance

Figure 7. Overview of Shadow Module Computation.

2.2.2 Supporting Data Structures

No significant new data structures are required to support the SHADOWS module.

2.2.3 Data Flow Diagram

The major functional components of the shadow computation are broken out by the data flow diagram shown in figure 8. In this diagram persistent data objects are depicted as boxes, data transformations are shown as circles, and data passing between transformations are represented as arcs.

Figure 8. Data Flow Diagram for Shadow Module Computation.
2.2.3.1 Data Areas

A) Target data - Target location in desired coordinate system, target reflectance, type and size. The target location is stored in the variables RLATT and RLONG inside the subroutine FINDR which is in TARGAC/targac.f. Reflectance for different targets is either read from a table or received from the operator inside the subroutine CNTRAS which is in the file TARGAC/targac.f. Type is held in the variable ITARGT inside CNTRAS, and the size is stored in different locations, one of which is DIM in the subroutine FINDR inside targac.f.

B) View angles - The elevation and azimuth of the observer relative to the target.

C) Solar/lunar angles - Azimuth and elevation of the source of direct illumination relative to the target. These values are stored in the variables AZISUN, ALTSUN, AZIMOO, ALTMOO once they are calculated in the subroutine ILLUM located in the file AUXLRY/rtlib.f.

D) Illumination data - Illumination intensity at ground level. Sum of direct and diffuse components, which are calculated in subroutine DIRDIF from the file AUXLRY/rtlib.f and stored separately in the variables FDIR and FDIF. Solar and lunar illumination are available in the variables SUNLIT and MOOLIT from the routine ILLUM in AUXLRY/rtlib.f.

E) Atmospheric data - Atmospheric transmittance factors. These are stored in variables named XSTRN_ in the file TARGAC/targac.f.

F) Average background irradiance - Intensity of background irradiance averaged over a large background. Computed in Compute SCR from the 2-d, 3-d, and shadow information in scene.

G) Pd ratio - The weighting factor used to combine the probability of detection or recognition from target-to-shadow contrast and target-to-background contrast.

H) Shadow radiance - The expected radiance within shadow areas based on the average background reflectances and the direct and diffuse light at ground level.
2.2.3.2 Data Tokens

Shadow parms - Parameters describing target shadow geometry (i.e., no luminance data). This includes major axis, minor axis, and inclination from the y-axis.

Obscure ratio - The fraction of the target shadow that is obscured by the target from the viewer's frame of reference.

Dirdif ratio - The ratio of direct to diffuse illumination intensities at ground level. It is calculated in the subroutine DIRDIF in AUXLRY/rtplib.f and named RATIO.

2.2.3.3 Data Functions

1) Compute shadow parms - Computes the appropriate shadow parameters assuming spherical clutter objects. Parameters include major and minor axes, inclination, and x and y densities.

2) Compute shadow geometry - Computes the appropriate shadow parameters assuming spherical clutter objects. Parameters include major and minor axes, inclination, and x and y densities.

3) Compute radiance - Computes the expected radiance within shadow areas based on the average background reflectances and the direct and diffuse light at ground level.

4) Compute obscuration - Accepts sun and viewer angles and computes the fraction of an object's shadow that would be obscured by the object when viewed from the viewer's position.

5) Compute Pd ratio - Computes the fraction of the target adjacent to shadow and the fraction adjacent to background. The ratio is used to weight the shadow and background detection probabilities within ACQUIR.

6) Dir/dif - Computes the ratio of diffuse to direct light at ground level. Subroutine DIRDIF is located in AUXLRY/rtplib.f.
2.3 ACQUIR Module

ACQUIR computes the probability of detection and recognition given the resolvable cycles of the sensor and transmitted contrast, the object range and size, and the clutter index. The module is a direct replacement for the current module, and implements two improvements. First, the detection and recognition probabilities are computed using a novel technique which combines probabilities for target-on-background and target-on-shadow. Second, the actual detection probabilities are computed by fast data look-up and interpolation routines.

The probability of detection and recognition is computed using dual detection algorithms for target-on-background and target-on-shadow. Target area is split between the portion adjacent to shadow and the portion adjacent to background; the two components are then used in separate detection computations. These independent probabilities of detection (or recognition) are then combined to provide a composite probability of detection (or recognition).

The probability computation is performed by fast table look-up and interpolation routines. This approach has the advantage that the mapping between cycles on target and detection probability can be modified on the fly by swapping in new look-up tables. Furthermore, mappings that are hard to fit by closed-form functions are easily accommodated by the table look-up approach.

2.3.1 Overview of the Computation

The Pd ratio (computed in the SHADOWS Module) is first used to compute the relative target area in function 1 that will be used by the target-on-shadow and target-on-background detection calculations. Both computations then proceed in parallel. Function 2 accepts the range and resolvable cycles and computes the cycles on target for the shadows case and the background case.
Figure 9. Overview of ACQUIR Module Computation.

The signal-to-clutter ratio is binned into coarser categories (if necessary) in function 3 and used to index the proper interpolation vectors from the Cycles Table. The proper interpolation vector is passed to both detection computations and used to interpolate the detection probability as follows. First, a binary search is used to find the proper position of the cycles-on-target in the sorted cycles list. Typically, it will fall between two adjacent components in the list. The position of cycles on target between the two bounding elements in the list is then approximated linearly and this same linear factor is used to approximate the value of the cycles-to-probability mapping in the Pd/Pf table.

These two independently computed probabilities are then passed down to function 5 where they are combined (weighted by the Pd ratio) to produce a composite probability of detection.

2.3.2 Supporting Data Structures

The ACQUIR module replacement uses new data structures to compute the probabilities of detection and recognition by interpolation. This requires two arrays. The first array is 1-dimensional and consists of an ordered list of probability thresholds. The second array is two dimensional. Each row of array 2 corresponds to a particular range of
SCR values and consists of a list of the minimum number of cycles-on-target required for
that level of detection/recognition probability given the level of clutter indicated by SCR.

2.3.3 Data flow Diagram

The major functional components of the ACQUIR computation are broken out by
the data flow diagram shown in figure 10. In this diagram, persistent data objects are
depicted as boxes, data transformations are shown as circles, and data passing between
transformations are represented as arcs.

Figure 10. Data flow Diagram for ACQUIR Module.
2.3.3.1 Data Areas

A) Target data - Target location in desired coordinate system, target reflectance, type and size. The target location is stored in the variables RLATT and RLONG inside the subroutine FINDR which is in TARGAC/targac.f. Reflectance for different targets is either read from a table or received from the operator inside the subroutine CNTRAS which is in the file TARGAC/targac.f. Type is held in the variable ITARGT inside CNTRAS, and the size is stored in different locations, one of which is DIM in the subroutine FINDR inside targac.f.

B) SCR - Signal-to-clutter ratio used to compute probability of detection and probability of recognition.

C) Pd ratio - The weighting factor used to combine the probability of detection or recognition from target-to-shadow contrast and target-to-background contrast.

D) Cycles table - Tables of minimum cycles required on target parameterized by detection and recognition probabilities. (Drawn from Schmeider). TARGAC-2 presently appears to contain the proper arrays in DCYC, located in file targac.f.

E) Pd/Pr Table - Tables of probability values parameterizing the Cycles table. TARGAC-2 presently appears to contain the proper arrays near DCYC, located in file targac.f.

F) RC values - Resolvable cycles given contrast and imaging equipment parameters. Two values are used: one for contrast between target and background and one for contrast between target and its shadow.

2.3.3.2 Data Tokens

Shadow weighted size - Size of target weighted by fraction of target against shadow.

Background weighted size - Size of target weighted by fraction of target against background.

Shadow cycles-on-target - Resolvable cycles on portion of target against shadow.
Background cycles on target - Resolvable cycles on portion of target against background.

Cycles vector - Vector of minimum number of cycles required on target given clutter parameters. Vector is parameterized by probability entries in \( Pd/Pr \) Table.

\( Pd \) - Probability of detection.

2.3.3.3 Data Functions

1) **Split target** - Splits the target size based on fraction of target against background.

2) **Compute cycles-on-target** - Computes the number of cycles on the target given the range, the maximum target dimension, and the sensor characteristics.

3) **Quantize** - Buckets the SCR into coarser grained measurements and extracts proper row from the Cycles table.

4) **Interp** - Accepts two input arrays and a search index. The first array is the "parameter" array, and the second is the "function" array. It locates the position within the parameter array, and interpolates between two adjacent cells if necessary. The interpolation "distance" is then used to interpolate the appropriate value from the same two adjacent cells in the function array.

5) **Combine Pd** - Combines the detection probabilities from shadow and background contrast according to the \( Pd \) ratio.

2.4 Software Structure

We now present the structure of the proposed software modifications to TARGAC. Figure 11 shows the first level of TARGAC. (Note that figure 11 only shows the first level of software decomposition under the FINDR routine.) Figure 12 shows a detailed view of the software structure for the proposed new codes. In both figures, major routines are represented by boxes, and downward arcs in the diagram indicate subroutine and function calls. Hatched boxes are routines devoted exclusively to IR and are not treated.
Heavily outlined boxes indicate TARGAC-2 modules that will be modified or replaced, and shaded boxes represent new modules. The module descriptions for figure 12 are outlined in the following section. The suggested FORTRAN module names are included in parentheses.

Figure 11. Decomposition of the FINDR routine.
2.4.1 New Module Descriptions

ACQUIR (ACQUIR) - Accepts the resolvable cycles across target for target-on-shadow and target-on-background, the shadow detection weighting factors, and Signal-to-Clutter Ratio to determine the probabilities of detection and recognition for the given viewer to target range.

SPLIT TARGET (SPLTGT) - Called by ACQUIR to compute the relative portions of the target against background and against shadow, and returns the two target sizes to be used in the dual detection algorithms.

COMPUTE CYCLES (GETCYC) - Accepts the size of the target and range, and computes the cycles across the target.
COMPUTE Pd (GETPD) - Accepts the number of cycles on target, weighting factors, and SCR to compute the probabilities of detection and recognition using calls to INTERPOLATE and COMBINE.

INTERPOLATE (INTER) - Accepts an argument list, a function list, and an input argument. The argument list is ordered, and the function list contains values of a function evaluated at the discrete points defined in the argument list (i.e., it is a discrete mapping from arguments to some other space). The routine interpolates the value of the function on the input argument in two steps. First, it calculates the relative position of the input argument between two adjacent cells in the argument list. It then uses this information to linearly interpolate within the function list.

COMBINE (CMBINE) - Accepts the Pd ratio and recombines the two detection/recognition probabilities into a single detection/recognition probability using the Pd ratio as a weighting factor. Empirically-derived rules for combining may apply.

SHADOW (SHADW) - Accepts target sizing data and solar/lunar angles (computed in the subroutine ILLUM from AUXLRY/rtplib.f) to compute the Pd ratio and the expected shadow radiance. The Pd ratio designates the relative portion of the target-on-shadow to target-on-background to use in the separate detection algorithms, and the final combining. The expected shadow radiance is used to compute the target-on-shadow contrast.

TARGET SHADOW PARMS (GETTSP) - Computes the size of the target shadow (assuming direct illumination) given the solar/lunar angles (from subroutine ILLUM), target characteristics, and the viewing angles. It calls SHADOW SIZE to compute basic shadow geometry from the target size and solar/lunar angles, and then calls SHADOW GEOMETRY to adjust for viewer angle.

SHADOW SIZE (SHSIZE) - Computes the shadow size parameters given an object size and the solar/lunar angles, which are calculated by the subroutine ILLUM and held in the variables AZISUN, ALTSUN, AZIMOO and ALTMOO.

SHADOW GEOMETRY (SHGEOM) - Adjusts the shadow size parameters for viewer angle relative to the sun and the observed object.
ASPECT (APSECT) - Adjusts the x-axis and y-axis measurements in the viewer's frame of reference for an object based on view angle.

INCLINATION (INCLIN) - Adjusts the inclination in the viewer's frame of reference for an object based on view angle.

Pd RATIO (PRATIO) - Computes the fraction of the target that is adjacent to its shadow and adjacent to its background. The ratio is used to compute cycles on target for the detection algorithms, and for combining the target-on-shadow and target-on-background detection/recognition probabilities into a single probability.

OBSCURE (OBSCUR) - Computes the portion of the shadow that is obscured by the target in the viewer's frame of reference.

SHADOW RADIANCE (SHRAD) - Computes the expected radiance from shadowed areas using the expected background reflectance, which comes from the subroutine CNTRAS, and the ratio of direct to diffuse light at ground level which is computed in the subroutine DIRDIF from AUXLRY/rtplib.f.

SCR (SCR) - Oversees the conversion of input data to statistical background parameters, and uses them to compute a clutter index, signal-to-clutter ratio, and average background radiance.

FLAT PARMS (FLTPRM) - Accepts statistical information on expected cluttering ground cover in the target area and computes the statistics for the flat clutter component. Currently, background information is read in via the subroutine CNTRAS.

3D PARMS (D3PRM) - Accepts statistical information on expected 3-d clutter objects in the target area and computes the statistics for the 3-d clutter component. Currently, most target information is read in by the subroutine CNTRAS.

SHADOW PARMS (SHPRM) - Computes the statistics for the flat clutter component due to shadows from the statistical parameters for 3-d clutter objects.

COMPRESS VIEW (COMPRS) - Adjusts the y-density of objects in a scene (viewer's frame of reference) to account for viewer angle.
PREPARE MATRIX (PREPMTX) - Places radiance values into the 50 by 50 scene matrix for the Monte-Carlo computation of Clutter Index using a three phase algorithm. In the first phase, flat parameters are used to fill in reflectances cell by cell, in several passes. The first pass uses the largest scale covering data and the last pass uses the lowest scale (smallest) covering data. A third pass then uses the average reflectance per cell to fill in average radiances for each cell. In the second phase, the radiances are attenuated by shadow information, and in the third phase, radiances from 3-d clutter objects are overlaid on the matrix.

FILL MATRIX (FILMTX) - Fills a matrix with values based on passed size and density parameters. All objects are assumed to be elliptical. A flag indicates if the values should replace old matrix values, or be averaged with old values.

COMPUTE BACKGROUND (GETBGD) - Accepts a matrix of radiances and computes the average.

COMPUTE SCR (GETSCR) - Computes the signal-to-clutter ratio from the target radiance, average background radiance, and clutter index.

COMPUTE CI (GETCI) - computes the clutter index from a matrix of scene radiances.
<table>
<thead>
<tr>
<th>Distribution</th>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commandant</td>
<td></td>
</tr>
<tr>
<td>U.S. Army Chemical School</td>
<td>1</td>
</tr>
<tr>
<td>ATTN: ATZN-CM-CC (Mr. Barnes)</td>
<td></td>
</tr>
<tr>
<td>Fort McClellan, AL 36205-5020</td>
<td></td>
</tr>
<tr>
<td>NASA Marshal Space Flight Center</td>
<td>1</td>
</tr>
<tr>
<td>Deputy Director</td>
<td></td>
</tr>
<tr>
<td>Space Science Laboratory</td>
<td></td>
</tr>
<tr>
<td>Atmospheric Sciences Division</td>
<td></td>
</tr>
<tr>
<td>ATTN: E501 (Dr. Fichtl)</td>
<td></td>
</tr>
<tr>
<td>Huntsville, AL 35802</td>
<td></td>
</tr>
<tr>
<td>NASA/ Marshall Space Flight Center</td>
<td>1</td>
</tr>
<tr>
<td>Atmospheric Sciences Division</td>
<td></td>
</tr>
<tr>
<td>ATTN: Code ED-41</td>
<td></td>
</tr>
<tr>
<td>Huntsville, AL 35812</td>
<td></td>
</tr>
<tr>
<td>Deputy Commander</td>
<td></td>
</tr>
<tr>
<td>U.S. Army Strategic Defense Command</td>
<td>1</td>
</tr>
<tr>
<td>ATTN: CSSD-SL-L (Dr. Lilly)</td>
<td></td>
</tr>
<tr>
<td>P.O. Box 1500</td>
<td></td>
</tr>
<tr>
<td>Huntsville, AL 35807-3801</td>
<td></td>
</tr>
<tr>
<td>Deputy Commander</td>
<td></td>
</tr>
<tr>
<td>U.S. Army Missile Command</td>
<td>1</td>
</tr>
<tr>
<td>ATTN: AMSMI-RD-AC-AD (Dr. Peterson)</td>
<td></td>
</tr>
<tr>
<td>Redstone Arsenal, AL 35898-5242</td>
<td></td>
</tr>
<tr>
<td>Commander</td>
<td></td>
</tr>
<tr>
<td>U.S. Army Missile Command</td>
<td>1</td>
</tr>
<tr>
<td>ATTN: AMSMI-RD-DE-SE (Mr. Lill, Jr.)</td>
<td></td>
</tr>
<tr>
<td>Redstone Arsenal, AL 35898-5245</td>
<td></td>
</tr>
<tr>
<td>Commander</td>
<td></td>
</tr>
<tr>
<td>U.S. Army Missile Command</td>
<td>1</td>
</tr>
<tr>
<td>ATTN: AMSMI-RD-AS-SS (Mr. Anderson)</td>
<td></td>
</tr>
<tr>
<td>Redstone Arsenal, AL 35898-5253</td>
<td></td>
</tr>
<tr>
<td>Commander</td>
<td></td>
</tr>
<tr>
<td>U.S. Army Missile Command</td>
<td>1</td>
</tr>
<tr>
<td>ATTN: AMSMI-RD-AS-SS (Mr. B. Williams)</td>
<td></td>
</tr>
<tr>
<td>Redstone Arsenal, AL 35898-5253</td>
<td></td>
</tr>
</tbody>
</table>
Commander
U.S. Army Missile Command
Redstone Scientific Information Center
ATTN: AMSMI-RD-CS-R/Documents
Redstone Arsenal, AL 35898-5241

Commander
U.S. Army Aviation Center
ATTN: ATZQ-D-MA (Mr. Heath)
Fort Rucker, AL 36362

Commander
U.S. Army Intelligence Center
and Fort Huachuca
ATTN: ATSI-CDC-C (Mr. Colanto)
Fort Huachuca, AZ 85613-7000

Northrup Corporation
Electronics Systems Division
ATTN: Dr. Tooley
2301 West 120th Street, Box 5032
Hawthorne, CA 90251-5032

Commander
Pacific Missile Test Center
Geophysics Division
ATTN: Code 3250 (Mr. Battalino)
Point Mugu, CA 93042-5000

Commander
Code 3331
Naval Weapons Center
ATTN: Dr. Shlanta
China Lake, CA 93555

Lockheed Missiles & Space Co., Inc.
Kenneth R. Hardy
ORG/91-01 B/255
3251 Hanover Street
Palo Alto, CA 94304-1191

Commander
Naval Ocean Systems Center
ATTN: Code 54 (Dr. Richter)
San Diego, CA 92152-5000

E-2
Commander
Space & Naval Warfare System Command
ATTN: PMW-145-1G
Washington, DC 20362-5100

Director
Naval Research Laboratory
ATTN: Code 4110
(Mr. Ruhnke)
Washington, DC 20375-5000

Commandant
U.S. Army Infantry
ATTN: ATSH-CD-CS-OR (Dr. E. Dutoit)
Fort Benning, GA 30905-5090

USAFETAC/DNE
Scott AFB, IL 62225

Air Weather Service
Technical Library - FL4414
Scott AFB, IL 62225-5458

USAFETAC/DNE
ATTN: Mr. Glauber
Scott AFB, IL 62225-5008

Headquarters
AWS/DOO
Scott AFB, IL 62225-5008

Commander
U.S. Army Combined Arms Combat
ATTN: ATZL-CAW
Fort Leavenworth, KS 66027-5300

Commander
U.S. Army Space Institute
ATTN: ATZI-SI
Fort Leavenworth, KS 66027-5300

Commander
U.S. Army Space Institute
ATTN: ATZL-SI-D
Fort Leavenworth, KS 66027-7300
Dr. Jerry Davis  
North Carolina State University  
Department of Marine, Earth, and  
Atmospheric Sciences  
P.O. Box 8208  
Raleigh, NC 27650-8208

Commander  
U.S. Army CECRL  
ATTN: CECRL-RG (Dr. Boyne)  
Hanover, NH 03755-1290

Commanding Officer  
U.S. Army ARDEC  
ATTN: SMCAR-IMI-I, Bldg 59  
Dover, NJ 07806-5000

Commander  
U.S. Army Satellite Comm Agency  
ATTN: DRCPM-SC-3  
Fort Monmouth, NJ 07703-5303

Commander  
U.S. Army Communications-Electronics  
Center for EW/RSTA  
ATTN: AMSEL-EW-MD  
Fort Monmouth, NJ 07703-5303

Commander  
U.S. Army Communications-Electronics  
Center for EW/RSTA  
ATTN: AMSEL-EW-D  
Fort Monmouth, NJ 07703-5303

Commander  
U.S. Army Communications-Electronics  
Center for EW/RSTA  
ATTN: AMSEL-RD-EW-SP  
Fort Monmouth, NJ 07703-5206

Commander  
Department of the Air Force  
OL/A 2d Weather Squadron (MAC)  
Holloman AFB, NM 88330-5000

PL/WE  
Kirtland AFB, NM 87118-6008
Commander
U.S. Army Field Artillery School
ATTN: ATSF-F-FD (Mr. Gullion)
Fort Sill, OK 73503-5600

Commander
Naval Air Development Center
ATTN: Al Salik (Code 5012)
Warminister, PA 18974

Commander
U.S. Army Dugway Proving Ground
ATTN: STEDP-MT-M (Mr. Bowers)
Dugway, UT 84022-5000

Commander
U.S. Army Dugway Proving Ground
ATTN: STEDP-MT-DA-L
Dugway, UT 84022-5000

Defense Technical Information Center
ATTN: DTIC-OCP
Cameron Station
Alexandria, VA 22314-6145

Commander
U.S. Army OEC
ATTN: CSTE-EFS
Park Center IV
4501 Ford Ave
Alexandria, VA 22302-1458

Commanding Officer
U.S. Army Foreign Science & Technology Center
ATTN: CM
220 7th Street, NE
Charlottesville, VA 22901-5396

Naval Surface Weapons Center
Code G63
Dahlgren, VA 22448-5000

Commander and Director
U.S. Army Corps of Engineers
Engineer Topographies Laboratory
ATTN: ETL-GS-LB
Fort Belvoir, VA 22060