A NUMERICAL MODEL FOR THE COMPUTATION OF RADIANCE DISTRIBUTIONS IN NATURAL WATERS WITH WIND-ROUGHENED SURFACES, PART II: USERS' GUIDE AND CODE LISTING

Curtis D. Mobley

Pacific Marine Environmental Laboratory
Seattle, Washington
July 1988
NOAA Technical Memorandum ERL PMEL-81

A NUMERICAL MODEL FOR THE COMPUTATION OF RADIANCE DISTRIBUTIONS IN NATURAL WATERS WITH WIND-ROUGHENED SURFACES, PART II: USERS' GUIDE AND CODE LISTING

Curtis D. Mobley
Joint Institute for the Study of the Atmosphere and Ocean
University of Washington
Seattle, Washington

Pacific Marine Environmental Laboratory
Seattle, Washington
July 1988
NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA/ERL. Use of information from this publication concerning proprietary products or the tests of such products for publicity or advertising purposes is not authorized.

Contribution No. 1047 from NOAA/Pacific Marine Environmental Laboratory
Contribution No. 44 from the Joint Institute for the Study of the Atmosphere and Ocean

For sale by the National Technical Information Service, 5285 Port Royal Road
Springfield, VA 22161
# CONTENTS

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>PROGRAM 1</td>
</tr>
<tr>
<td></td>
<td>A. Program Description</td>
</tr>
<tr>
<td></td>
<td>B. Input</td>
</tr>
<tr>
<td></td>
<td>C. File Management</td>
</tr>
<tr>
<td></td>
<td>D. Code Listing</td>
</tr>
<tr>
<td></td>
<td>MAIN (all-quad version)</td>
</tr>
<tr>
<td></td>
<td>INISHL (all-quad version)</td>
</tr>
<tr>
<td></td>
<td>EQSANG</td>
</tr>
<tr>
<td></td>
<td>EQTHET</td>
</tr>
<tr>
<td></td>
<td>FINTCP</td>
</tr>
<tr>
<td></td>
<td>GTABC</td>
</tr>
<tr>
<td></td>
<td>GETNOD</td>
</tr>
<tr>
<td></td>
<td>MPINDEX</td>
</tr>
<tr>
<td></td>
<td>P2ARRAY</td>
</tr>
<tr>
<td></td>
<td>P3ARRAY</td>
</tr>
<tr>
<td></td>
<td>PULL</td>
</tr>
<tr>
<td></td>
<td>PUSH</td>
</tr>
<tr>
<td></td>
<td>REFLF</td>
</tr>
<tr>
<td></td>
<td>RSPLIT</td>
</tr>
<tr>
<td></td>
<td>TIP</td>
</tr>
<tr>
<td></td>
<td>TRACE</td>
</tr>
<tr>
<td></td>
<td>TRIADS</td>
</tr>
<tr>
<td></td>
<td>MAIN (one-quad version)</td>
</tr>
<tr>
<td></td>
<td>INISHL (one-quad version)</td>
</tr>
<tr>
<td>3</td>
<td>PROGRAM 2</td>
</tr>
<tr>
<td></td>
<td>A. Program Description</td>
</tr>
<tr>
<td></td>
<td>B. Input</td>
</tr>
<tr>
<td></td>
<td>C. File Management</td>
</tr>
<tr>
<td></td>
<td>D. Code Listing</td>
</tr>
<tr>
<td></td>
<td>MAIN (all-quad version)</td>
</tr>
<tr>
<td></td>
<td>INISHL (all-quad version)</td>
</tr>
<tr>
<td></td>
<td>MAIN (one-quad version)</td>
</tr>
<tr>
<td></td>
<td>INISHL (one-quad version)</td>
</tr>
<tr>
<td>4</td>
<td>PROGRAM 3</td>
</tr>
<tr>
<td></td>
<td>A. Program Description</td>
</tr>
<tr>
<td></td>
<td>B. Input</td>
</tr>
<tr>
<td></td>
<td>C. File Management</td>
</tr>
<tr>
<td></td>
<td>D. Code Listing</td>
</tr>
<tr>
<td></td>
<td>MAIN</td>
</tr>
<tr>
<td></td>
<td>INISHL</td>
</tr>
<tr>
<td></td>
<td>FULLRT</td>
</tr>
</tbody>
</table>
5. PROGRAM 4 ............................................. 63
   A. Program Description ................................ 63
   B. Input ............................................. 63
   C. File Management ................................... 66
   D. Code Listing
       MAIN ............................................. 67
       INISHL .......................................... 70
       ADIPAK .......................................... 74
       AMPA0 .......................................... 74
       AMPAP .......................................... 76
       AMPINT .......................................... 77
       AMPX ............................................ 79
       BOTMBC .......................................... 81
       DRTAB ............................................ 82
       EIGENR .......................................... 84
       EPSINF .......................................... 87
       FALPHA .......................................... 88
       FFMPAK .......................................... 88
       PHASEF (version defining Lake Limne) ............ 89
       PHASEF (version defining the Pelagos Sea) ....... 91
       PHASEF (version for absorption and spherical scattering,
               independent of depth) ....................... 94
       PHASEF (version for absorption and spherical scattering
               as functions of depth) ....................... 95
       PNTAMP .......................................... 96
       QAPHAS .......................................... 97
       QASKY ........................................... 100
       RFMPAK ........................................... 101
       RHOTAU .......................................... 102
       RICATI .......................................... 103
       Y2ZGEO .......................................... 106

6. PROGRAM 5 ............................................. 107
   A. Program Description ................................ 107
   B. Input ............................................. 107
   C. File Management ................................... 110
   D. Code Listing
       MAIN ............................................. 112
       INISHL .......................................... 114
       BFSCAT .......................................... 117
       CONTRM .......................................... 119
       IRRAD ............................................ 121
       KFCN ............................................. 123
       KRAD ............................................. 124
       PRINT ............................................. 126
       RADI ............................................. 128
       RTECK ............................................ 129
       SYNRAD .......................................... 132
7. GRAPHICS PROGRAMS .................................................. 133

A. Plotting Radiance Distributions .................................. 133
   1. Input ............................................................... 133
   2. Code Listing ..................................................... 135
      MPRAD ........................................................... 135
      NCHAR ............................................................ 137
      PPHILOG ......................................................... 138
      PTHELOG ......................................................... 141
      PTHEPLR ......................................................... 144

B. Plotting Chromaticity Diagrams .................................. 148
   1. Input ............................................................... 148
   2. Code Listing ..................................................... 150
      MPCHRO ........................................................... 150
      ATMOS ............................................................ 151
      CHRMXY .......................................................... 152
      PLTBLU ............................................................ 154
      PLTCHR ............................................................ 155

C. Plotting Data as a Function of Wavelength ..................... 157
   1. Input ............................................................... 157
   2. Code Listing ..................................................... 159
      MPWAVE ........................................................... 159
      PLTWVL ............................................................ 160

APPENDIX A: IMSL Routines Used by the NHM ...................... 164
APPENDIX B: A Simple Model for Incident Radiance Distributions 164
APPENDIX C: Computation of Chromaticity Coordinates .......... 166
APPENDIX D: A Simple Model Atmosphere and Solar Spectrum .... 169
A Numerical Model for the Computation of Radiance Distributions in Natural Waters with Wind-Roughened Surfaces, Part II: Users' Guide and Code Listing

Curtis D. Mobley*

ABSTRACT. This report is a users' guide for and listing of the FORTRAN V computer code that implements a numerical procedure for computing radiance distributions in natural waters. The mathematical details of the numerical radiance model are described in a companion report (A Numerical Model for the Computation of Radiance Distributions in Natural Waters with Wind-Roughened Surfaces, by Curtis D. Mobley and Rudolph W. Preisendorfer, NOAA Technical Memorandum ERL PMEL-75). The present report describes how to run the computer model and therefore addresses questions such as which routines perform which calculations, what input is required by the various programs, and what is the file structure of the overall program.

1. INTRODUCTION

General knowledge of the radiance distribution in a natural hydrosol, such as a lake or ocean, is a prerequisite for the solution of more specific problems in underwater visibility, remote sensing, photosynthesis, or climatology. Moreover, since radiance is the fundamental radiometric quantity, if the radiance distribution is known, then all other quantities of interest, such as the irradiances and K-functions, are easily computed.

With the above incentives, a numerical model, called the Natural Hydrosol Model or NHM, was developed, based on the following assumptions:

(1) The water body is a plane-parallel medium which
    (a) has no internal light sources, and is non-fluorescent
    (b) is directionally isotropic,
    (c) is laterally homogeneous, but is inhomogeneous with depth.

(2) The upper boundary is the random air-water interface, which is wind-ruffled, laterally homogeneous, and azimuthally anisotropic.

(3) The lower boundary is a surface whose reflectance is azimuthally isotropic. This boundary may be either the physical bottom of an optically shallow water body, or a plane in an optically infinitely deep water body, below which the water is homogeneous with depth.

(4) There is radiant flux incident downward on the upper boundary. There is no radiant flux incident upward on the lower boundary.

* Joint Institute for the Study of the Atmosphere and Ocean, University of Washington, AK-40, Seattle, WA 98195.
§1. INTRODUCTION

(5) The radiance field is monochromatic and unpolarized.

The exact meaning of these assumptions and their mathematical consequences are described in the following two reports.

(1) "The NHM report." This technical memorandum is the companion to the present one, and should be studied prior to reading this report. The NHM report describes the overall computational structure of the Natural Hydrosol Model and contains all the mathematical details. The full reference is:

A Numerical Model for the Computation of Radiance Distributions in Natural Waters with Wind-Roughened Surfaces, by Curtis D. Mobley and Rudolph W. Preisendorfer, NOAA Tech. Memo. ERL PMEL-75, Pacific Marine Environmental Laboratory, Seattle, WA 98115, January 1988, 195 pages. (Also available from the National Technical Information Service, 5285 Port Royal Road, Springfield VA 22161, as report number PB88-192703.)

(2) "The ray-tracing report." This technical memorandum describes mathematical algorithms for simulating random air-water surfaces and for tracing light rays as the rays interact with the simulated water surface. This ray-tracing procedure is used in computing the surface boundary conditions for the radiance computations (cf. assumption 2, above). The full reference is:


Comments throughout the computer code and in the descriptive sections of this report make frequent reference to the NHM report (reference 1, just cited), enabling the user of the code to trace in detail the implementation of the mathematical procedures. Thus, in the computer code, the comment "compute forward scattering by 11.7" refers to equation 11.7 in report ERL PMEL-75. Comments referring to the ray-tracing report, ERL PMEL-63, are prefaced by "63/". Thus a reference to "63/3.20" refers to equation 3.20 in the ray-tracing report. To avoid confusion in the present report, references to the NHM report, ERL PMEL-75, are prefaced by "75/".

The various computations performed by the NHM are grouped into five separate programs, which are run in sequence to obtain the solution of a given problem. The first three programs compute the surface boundary reflectance and transmittance functions. The fourth program solves for the radiance amplitudes at all depths, and the fifth program then reconstitutes the radiances and analyzes the results. A sixth set of programs for graphical analysis of the numerical results is included for convenience although, strictly speaking, these programs are not a part of the NHM.

The following six sections of this report describe in turn the NHM programs. Each section begins with a brief description of the program. Then there are sections on the user-supplied input required to run the program, and on file management. Each program consists of a main program.
named MAIN, which controls overall program flow, and a subroutine named INISHL, which reads the user-supplied data and performs other initialization tasks. The reader wishing to see the actual statements that read the user-supplied input can always find them in subroutine INISHL. Each section ends with a listing of MAIN, INISHL, and then the other subroutines of that program in alphabetical order. There are several subroutines (e.g. utility routines for printing arrays) which are used in two or more of the NHM programs. These are listed with the program in which they are first used.

The numerical computations make frequent use of the IMSL library (9th edition)\(^1\) of FORTRAN-callable subroutines. These subroutines are used to perform standard mathematical operations such as random number generation, matrix inversion, and solving ordinary differential equations. The IMSL library is likely to be available at any scientific computing center. However, any comparable mathematical software library, such as NAGLIB\(^2\), could be used after minor rewriting of the code. Appendix A lists the required IMSL subroutines. The graphics routines use standard "CalComp Basic Software\(^3\)" for plotting data.

**Acknowledgments.** The author was supported in this work by the Oceanic Biology Program of the Office of Naval Research, under contract number N00014-87-K-0525. This is Contribution No. 44 from the Joint Institute for the Study of the Atmosphere and Ocean and Contribution No. 1047 from the Pacific Marine Environmental Laboratory/NOAA. Ryan Whitney did the word processing.

---

1. Licensed and distributed by
   IMSL, Inc.
   2500 Park West Tower One
   2500 City West Boulevard
   Houston, TX 77042-3020

2. Licensed and distributed by
   Numerical Algorithms Group, Inc.
   1101 31st Street, Suite 100
   Downers Grove, IL 60515-1263

3. Licensed and distributed by
   California Computer Products, Inc.
   2411 West La Palma
   Anaheim, CA 92801
2. PROGRAM I

A. Program Description

This program does the ray tracing described in 75/§9a and charted in 75/Fig. 9 (i.e. in §9a and Fig. 9 of the NHM report).

It is convenient to run the program twice. The first run is used to generate and save a file of random air-water surfaces: no ray tracing is performed. The second run then reads the file of surface realizations and performs the ray tracing. This two-step procedure allows the same set of realized surfaces to be repeatedly used in the ray tracing, as follows. Each initial ray directed toward a particular input quad Q_{rs} requires an independent realization of the random water surface. However, the rays directed toward different input quads Q_{rs} and Q_{pq} can use the same set of surface realizations. Moreover, the symmetry of the water surface for capillary waves (see 75/§3f and 75/Fig. 5) allows a given surface realization to be rotated by $\phi = 180^\circ$ in order to get another independent surface realization. One can also turn a capillary wave surface "upside down" and get yet another independent surface realization. Thus each generated and stored capillary-wave surface can be used four times: two azimuthal orientations, each "right side up" or "upside down." The code, as listed in this report (see statements 55 to 506 in the MAIN program), makes use of these symmetries so that if, say, 10000 rays are to be traced for each input quad, only 2500 surfaces need be generated and saved in the first run. Note, however, that if a gravity-wave spectrum is used, one can no longer turn the surface upside down and get a new gravity-wave surface realization. And if the wave spectrum has different wave-slope statistics in the downwind and upwind directions, then one cannot rotate the surface by $\phi = 180^\circ$. Thus for a fully realistic, mixed gravity-capillary wave spectrum, one must generate 10000 surface realizations if 10000 rays are to be traced for each input quad Q_{rs}. However, these surface realizations (which can be very expensive to generate for a mixed gravity-capillary wave spectrum) can still be recycled for different input quads.

The net result of Program I is then to repeatedly obtain a random surface realization, randomly select a direction in Q_{rs}, and send a parent ray toward Q_{rs} and the realized surface. All the reflected and refracted daughter rays are traced to completion, and the quads receiving the final daughter rays are determined. One parent ray is sent toward each quad Q_{rs} in the first quadrant (of the wind-based system shown in 75/Fig. 1) for each surface realization, until the desired number of surface realizations has been made. For each (parent ray)-(daughter ray) pair, the program records the values of r, s, u, v, and the radiant flux of the daughter ray (u,v labels the output quad Q_{uv} receiving the final daughter ray). These ray-tracing computations can form a significant part of the entire work of the NHM.

Two versions of MAIN and INISHL are included in the code listing for Programs 1 and 2. The regular version of these two routines (listed first) automatically loops over all first-quadrant input quads Q_{rs}, sending rays toward each quad in turn (but using the same surface realizations
§2. PROGRAM 1

for each quad, as noted above), and thereby generating all the ray data required to compute the entire quad-averaged geometric reflectance and transmittance arrays. This version of Program 1 is to be used for production runs.

The second version of MAIN and INISHL (listed last) is a “one-quad” version, which sends rays toward only one input quad selected by the user (in the one-quad version of record 2, below). The ray data so generated lead to the evaluation of only one row of the reflectance and transmittance arrays. If the rays are air-incident, one row of \( r(a,x) \) and \( t(a,x) \) is computed; if the rays are water-incident, one row of \( r(x,a) \) and \( t(x,a) \) is computed.

The one-quad versions of Programs 1 and 2 are useful for determining how many rays must be traced to achieve a given accuracy in the elements of the quad-averaged \( r \) and \( t \) arrays, for a given quad resolution and wind speed. This determination must be empirically made, and the individual elements of the \((r,slu,v)\) arrays approach their final values at differing rates as more and more rays are tabulated. (Here \( f(r,slu,v) \) represents \( r(a,x); r,slu,v \), or any of the other three \( r \) and \( t \) arrays.) For a given input quad \( Q_r \), the output quads \( Q_{uv} \) which are near the specular (still water) reflection or refraction directions of the parent rays in \( Q_r \) will receive far more reflected or transmitted daughter rays than those quads which are in directions far from the specular directions. Thus after only a few hundred surface realizations, some elements of \( f(r,slu,v) \) may have achieved their final values with great accuracy, whereas other elements may not have had a single ray path connect the particular \( Q_r \) and \( Q_{uv} \) quads. However, those elements which are largest in magnitude dominate the behavior of the light field in the sea, so it is not necessary to know all matrix elements to the same degree of accuracy. The user of the NHM is thus faced with making a decision regarding the desired accuracy of the elements of the \( r \) and \( t \) arrays. The larger matrix elements can and must be determined with great accuracy, but the smaller matrix elements, which are many orders of magnitude smaller than the larger elements, cannot be accurately estimated unless a tremendously large number of rays is traced.

Thus, using the one-quad version, one can make a series of runs with, say, 1000, 5000 and 10000 air-incident rays being traced for a particular input quad \( Q_r \). The computed values of \( r(a,x) \) and \( t(a,x) \) at \( x = 1, \ldots, n \), and \( x = 1, \ldots, 2n \), can then be studied to see how quickly these array elements achieve stable values.

Other specialized studies can be economically performed with the one-quad version. For example, for a fixed number of rays incident on the surface toward a given quad \( Q_r \), one can study the effects of wind speed on the directions of the reflected and refracted rays, and so on.

B. Input

Four parameters, which determine maximum array dimensions, must be set at compilation time. These parameters are (in the first PARAMETER statement in the MAIN program)
§2. PROGRAM 1

<table>
<thead>
<tr>
<th>parameter</th>
<th>value in listed code</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MXMU</td>
<td>10</td>
<td>the maximum number of quads in the ( \mu )-direction in a hemisphere, including the polar cap</td>
</tr>
<tr>
<td>MXPHI</td>
<td>24</td>
<td>the maximum number of quads in the ( \phi )-direction, ( 0 \leq \phi \leq 2\pi )</td>
</tr>
<tr>
<td>MXSTAK</td>
<td>10</td>
<td>the maximum number of rays in the push-pull stack at once</td>
</tr>
<tr>
<td>MXNHEX</td>
<td>7</td>
<td>the maximum order of the hexagonal grid used for ray tracing</td>
</tr>
</tbody>
</table>

Refering to 75/§3a, MXMU gives the maximum allowed value of \( m (\equiv \text{NMU}, \text{below}) \), and MXPHI gives the maximum value of \( N(\equiv \text{NPHI}, \text{below}) \). Figures 75/4a, 75/4b and 75/4d show quad partitions for which \( m = 10 \) and \( N = 24 \). A run using the quad partitioning of 75/Fig. 4c has \( m = 23 \) and \( N = 60 \), as so would require \( \text{MXMU} \geq 23 \) and \( \text{MXPHI} \geq 60 \). For efficient use of computer storage, one should pick MXMU and MXPHI to be the same as the actual number of \( \mu \) and \( \phi \) cells in the quad partitioning, \( \text{NMU} \) and \( \text{NPHI} \), respectively, to be specified in record 2, below. The value of \( \text{MXSTAK} = 10 \) should be sufficient for any problem (see 63/page 11, i.e. page 11 in the ray-tracing report). \( \text{MXNHEX} = 7 \) is sufficient for simulation of capillary wave surfaces. 75/Fig. 8 and 63/Fig. 5 each show hexagonal grids of order two (\( \text{NHEX} = 2 \)). Proper resolution of mixed gravity-capillary waves requires high-order hexagonal grids (\( \text{NHEX} \) of 100 or more), and so \( \text{MXNHEX} \) must be increased accordingly if such studies are to be made.

Two or three free-format data records are read at execution time (see subroutine INISHL). In essence, the first record specifies the water surface; the second (and optional third) specifies the quad partitioning and the number of rays to be traced. The records are as follows:

**Record 1:** IDBUG, IGENSF, NHEX, WNDSPD, DSEFD

- **IDBUG**
  - \( = 0 \) for minimal output (production runs)
  - \( = 1 \) for greater output
  - \( = 2 \) for full debugging output

- **IGENSF**
  - \( = 0 \) if a file of random surfaces already exists, and is to be used for ray tracing
  - \( > 0 \) if this is an initial run for generating and saving a file of random surfaces. The value of IGENSF gives the number of surfaces to be generated (IGENSF = 2500, say).

- **NHEX**
  - gives the order of the hexagonal surface grid (\( \text{NHEX} = 7 \) is adequate for capillary waves)
§2. PROGRAM 1

WNDSPD gives the wind speed for use in the wave spectrum, e.g. WNDSPD = 10.0 for a 10 m s⁻¹ wind at 12.5 m elevation (see 63/page 15).

DSEED is a double precision seed for the IMSL random number generators, e.g. DSEED = 18762203.D0

If IGENSF > 0, only record 1 is required.

Record 2: NMU, NPHI, MUPART, NREAD0, NUMRAY

NMU gives the number of µ-cells in one hemisphere in the quad partitioning (the value of m on 75/page 20).

NPHI gives the number of φ-cells in the quad partitioning (the value of N on 75/page 20). NPHI must be a multiple of 4.

MUPART selects the scheme for µ-partitioning of the unit sphere (see 75/page 22-24), as follows:

  = 1 if all quads are to have equal solid angles
  = 2 if all quads are to have equal Δθ values

The user may write subroutines to define other quad partitions, using other values of this variable to select the desired subroutine.

NREAD0 = 1 if the file of stored surface realizations is to be read from the start (the usual case)

= 2, 3 or 0 if the file is to be read starting with a rotation or inversion of the stored surfaces (this can be useful if additional rays are to be traced and complete use of the stored surfaces has not yet been made)

NUMRAY gives the number of rays to be traced from each input quad

if NUMRAY < 0, then a third record is used to give the number of rays to be traced from quads in each µ-band.

Record 2a: NRAYQD(1), ..., NRAYQD (NMU)

This record is read only if NUMRAY < 0 in record 2.

NRAYQD(1) gives the number of rays to be traced from each input quad in the µ-band nearest the equator (r = 1), and so on until

NRAYQD(NMU) gives the number of rays to be traced from the polar cap (r = m = NMU)

Record 2a can be used if, for example, one wants to trace a certain number of initial rays per steradian, but the quads have different solid angles in the various µ-bands. Or, if it is found in preliminary studies (e.g. with the one-quad version) that more rays must be traced from quads
near the equator than from quads near the polar caps, in order to achieve the desired accuracy, then record 2a must be used.

Record 2, one-quad version: NMU, NPHI, MUPART, IR, JS, NUMRAY
NMU, NPHI, MUPART and NUMRAY are as above. IR and JS give the values of \( r \) and \( s \), respectively, specifying the input quad \( Q_{rs} \). If IR is positive, \( 1 \leq IR \leq NMU \), the rays are air-incident. If IR is negative, \( -NMU \leq IR \leq -1 \), the rays are water-incident. JS must be in the first quadrant, i.e. \( 1 \leq JS \leq NPHI/4+1 \).

C. File Management
Throughout the NHM, files are given a symbolic (alphanumeric) filename beginning with "NU" (e.g. NUSFC for the file containing the surface realizations), as well as an external filename of the form "TAPEXX", where XX is a FORTRAN logical unit number (e.g. TAPE15 is the file NUSFC). This naming scheme is appropriate for CDC computers, but may require minor modification on other machines. User-supplied input is always read from unit 5 (INPUT, or TAPE5) and printout is written to unit 6 (OUTPUT, or TAPE6), in accordance with standard FORTRAN conventions.

Program 1 creates the following five files:

<table>
<thead>
<tr>
<th>symbolic filename</th>
<th>external filename</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUSFC</td>
<td>TAPE15</td>
<td>the file of random surface realizations; created in the initial run of Program 1, and read in the ray-tracing run of Program 1</td>
</tr>
<tr>
<td>NUDU</td>
<td>TAPE16</td>
<td>a ray-data file, containing initial and final ray direction and radiance information, for initial rays downward and final rays upward. Created in the second run of Program 1 and used to compute ( r(a,x; r,slu,v) )</td>
</tr>
<tr>
<td>NUDD</td>
<td>TAPE17</td>
<td>ray data file for initial rays downward, final rays downward; used to compute ( t(a,x; r,slu,v) )</td>
</tr>
<tr>
<td>NUUD</td>
<td>TAPE18</td>
<td>ray data file for initial rays upward, final rays downward; used to compute ( r(x,a; r,slu,v) )</td>
</tr>
<tr>
<td>NUUU</td>
<td>TAPE19</td>
<td>ray data file for initial rays upward, final rays upward; used to compute ( t(x,a; r,slu,v) )</td>
</tr>
</tbody>
</table>

Program 2 reads the four ray-data files and tallies the information to generate the quad-averaged, geometric reflectance and transmittance arrays.
D. Code Listing
Each subroutine begins with a brief description of its purpose.

```fortran
PROGRAM MAIN(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE15.1
    TAPE16, TAPE17, TAPE18, TAPE19)
******
*   *  THIS IS PROGRAM 1 OF THE NATURAL HYDROSOL MODEL
   *
******

ON NHM1/M1ALI

THIS PROGRAM BEGINS COMPUTATION OF THE GEOMETRIC REFLECTANCE AND
TRANSMITTANCE ARRAYS WHICH DESCRIBE THE AIR-WATER INTERFACE FOR
A GIVEN WIND SPEED.

NOTE: THIS VERSION OF THE CODE ATTEMPTS TO MINIMIZE THE EXECUTION
TIME, AT THE EXPENSE OF MODULARITY AND READABILITY OF THE CODE.
SOME SECTIONS OF FREQUENTLY EXECUTED CODE ARE WRITTEN AS STRAIGHT
LINE CODE WITH SIMPLE VARIABLES, RATHER THAN BEING GROUPED IN
SUBROUTINES OR DC LOOPS WITH ARRAYS, IN ORDER TO AVOID CALLING
AND INDEXING OVERHEAD. ALMOST ALL ERROR CHECKING AND INTERMEDIATE
OUTPUT HAS BEEN REMOVED.

THIS PROGRAM USES THE MONTE CARLO RAY TRACING TECHNIQUE
DESCRIBED IN NOAA TECH MEMO ERL-PMEL-75. COMMENTS REFERRING TO
THIS REPORT ARE PREFACED BY $3. THIS 63/7.5 REFERENCES TO
EQUATION 2.12 IN "TECH MEMO M1.

REFERENCES WITHOUT THE $3/ REFER TO NOAA TECH MEMO ERL-PMEL-75.

NUSFC = TAPE15...CONTAINS THE RANDOM SURFACE REALIZATIONS

RESULTS OF COMPUTED RAY PATHS ARE WRITTEN TO FILES AS FOLLOWS:

NUDD = TAPE16...INITIAL RAY DOWNWARD, FINAL RAY UPWARD: R = R(A,X)
NUDD = TAPE17...INITIAL RAY DOWNWARD, FINAL RAY DOWNWARD: T = T(A,X)
NUDD = TAPE18...INITIAL RAY UPWARD, FINAL RAY DOWNWARD: R = R(A,X)
NUDD = TAPE19...INITIAL RAY UPWARD, FINAL RAY UPWARD: T = T(A,X)

PROGRAM 2 READS THESE FILES AND CALCULATES THE RESULTS TO GENERATE
THE ACTUAL R AND T ARRAYS.

PARAMETER(MXNUD=4, MPH1=24, MXSTAP=1, MXNHEX=7)
PARAMETER(MXNODE=4*MXNHEX*(MXNHEX+1)+1, MXNTIP=4*MXNHEX+1)
COMMON/CNJODES/NUOE, NODE(2, MXNODE), CNODE(MXNODE)
COMMON/CHEXGR/EX*(MAXR+1, MAXA+1), R2HAT(2), R1AT, TARGET(2)
COMMON/STAP/CIP, SMIN, YTIP,2, MAXTIP, SIMNTIP, KS(MXNTIP), ZMIN, 1
ZMAX
COMMON/STACK/NSTACK, STACK(3, MAXSTACK, 1
COMMON/CMISC/CMISC(20), FMISC(10),
IQM, SMI, XMIN(3), P31, XLRM, XLRP(3)
DIMENSION NRAYD(MXNUD), NBRANCH(10))
DOUBLE PRECISION OSSEED

DATA NREAU/NREAU, NRAYD, NBRANCH, NRAYD(3), 3*0/
DATA NSTACK/0/, KTRACE/0/, NBRANCH/10/, NRAYD, NRAYD(3), NTIR/3*0/

INITIALIZE THE PROGRAM

CALL INIT(NHMS, NREAU, NRAYD, OSSEED)

NMU = IMUL(11)
```

9
§2. PROGRAM 1

NPHI = IMISC(2)
P1 = FMISC(1)
TWOP = 2.*P1
JP2 = NPHI/4 + 1
DPhiNP = BNDPHI(2) - BNDPHI(1)
NUMDU = 0
NUMDD = 0
NNU = 0
NUMTP1 = 0

C***** BEGIN COMPUTATIONS *****

C N.B. I AND J LABEL THE INPUT QUAD, WHICH IS THE QUAD RECEIVING
C THE PHOTONS (XI PRIME IS THE DIRECTION OF PHOTON TRAVEL).

C LOOP OVER MU PRIME CELLS (THETA = -PI/2 TO +PI/2)
DO 1001 I=-NNU,NNU
   IF(I.EQ.0) GO TO I001
   C GET MU PRIME BOUNDARIES OF THE INCUMING QUAD
   FMUMIN = 0.
   IA = IABS(I)
   IF(AA.GT.1) FMUMIN = BNDMU(IA-1)
   DMU = BNDMU(IA) - FMUMIN
   NUMRAY = NRAYQU(IA)
   C IF(I.EQ.NMU) THEN
   IF(I IS A POLAR CAP, DO ONLY J = 1 INDEX
   JCOMP = 1
   PHIMIN = 0.
   DPHI = TWOP
   ELSE
   FOR NON POLAR QUADS, DO ONLY FIRST QUADRANT (0 .LE. PHI PRIME .LE. PI/2)
   JCOMP = JP2
   PHIMIN = BNDPHI(NPHI)
   DPHI = DPhiNP
   ENDIF

C LOOP OVER PHI PRIME CELLS (0 .LE. PHI PRIME .LE. PI/2, OR 2*PI IF POLAR CAP)
DO 1000 J=1,JCOMP
   IF(J.GT.1) PHIMIN = BNDPHI(J-1)
   C LOOP OVER THE RANDOM STARTING POINTS WITHIN THE QUAD
   C NOTE THAT DIFFERENT QUADS MAY HAVE DIFFERENT NUMBERS OF RAYS TRACED
   REWIND NUSFC
   READ(NUSFC) HEADER
   READ(NUSFC) HEADER
   NREAD = NREADU
   DO 1000 NRAV=1,NUMRAY
   C SELECT A SURFACE REALIZATION
   55 CONTINUE
   IF(NREAD.EQ.1) THEN
   READ A SURFACE REALIZATION AS GENERATED
   READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(IZ),IZ=1,NNODE)
   ELSEIF(NREAD.EQ.2) THEN
   READ THE SURFACE AS ROTATED BY 180 DEGREES
   READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(IZ),IZ=NNODE,1,1)
   ELSEIF(NREAD.EQ.3) THEN
   READ THE SURFACE AND INVERT
   READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(IZ),IZ=NNODE,1,1)
   DO 502 IZ=1,NNODE
   502 ZNODE(I2) = -ZNODE(I2)
   ELSEIF(NREAD.EQ.0) THEN
   READ THE SURFACE AS ROTATED BY 180 DEGREES AND THEN INVERT
   READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(IZ),IZ=NNODE,1,1)
   DO 504 IZ=1,NNODE
   504 ZNODE(I2) = -ZNODE(I2)
   ENDIF
§2. PROGRAM 1

GO TO 506

END OF FILE PROCESSING FOR THE STORED FILE OF CAPILLARY SURFACES

50 WRITE(6,514) NREAD,NUSFC
NREAD = NREAD + 1
NREAD = MOD(NREAD,4)
REWIND NUSFC
READ(NUSFC) HEADER
READ(NUSFC) HEADER
GO TO 55

506 CONTINUE

CHOOSE A RANDOM MU PRIME VALUE
I POSITIVE (NEGATIVE) GIVES UPWARD (DOWNWARD) RAYS WITH MU PRIME =
XI PRIME(3) = XI IN(3) POSITIVE (NEGATIVE)

777 RMU = (FMUMIN + DM * GGUBFS(DSEEU)) * SIGN(1.0,FLOAT(I))
NO RAYS FROM THE POLE ITSELF
IF(ABS(RMU).GT.1.0) GO TO 777
ROOT = SQRT(1.0 - RMU*RMU)

CHOOSE A RANDOM PH. VALUE
SPHI = AMOD(PHI,MIN + GGUBFS(DSEEU)*PHI,TWOPI)

DEFINE A TEMPORARY RAY AS -XI PRIME, AND FOLLOW THIS RAY TO
THE BOUNDARY TO GET SM IN
XI IN(1) = -ROOT*SIN(SP HI)
XI IN(2) = -ROOT*SIGN(SP HI)
XI IN(3) = -RMU

CALL TP(TARGET, XI IN, 0)
DEFINE THE INITIAL POINT ON THE HEXAGON BOUNDARY
TEMP = SM IN/CM 25(20)
PIN(1) = TARGET(1) * TEMP*XI IN(1)
PIN(2) = TARGET(2) * TEMP*XI IN(2)
PIN(3) = TEMP*XI IN(3)

RESET XI IN TO THE DESIRED INCIDENT RAY DIRECTION, XI PRIME
THE DIRECTION OF PHOTON TRAVEL
XI IN(1) = -XI IN(1)
XI IN(2) = -XI IN(2)
XI IN(3) = -XI IN(3)
RAD = 1.0
INRAY = 1

PERFORM RAY TRACING COMPUTATIONS
*** THIS IS THE RECURSIVE TREE FOR A GIVEN INITIAL RAY ***

KBRNCH = 0
999 CALL TRAC E(INRAY,RAD,PIN,XI IN, OUT,P,REFL,XIREFL,REFR,XIREFR)
KTRACE = KTRACE + 1
KBRNCH = KBRNCH + 1
INRAY = 0

CHECK FOR RAY GOING LEFT THE HEXAGON
IF(OUT.EQ.1) THEN
RAY HAD NO F AIL INTERCEPTS.

GET THE QUAD INDICES OF THE FINAL RAY DIRECTION
PHIFIN = AMOD(6*TAN2(XI IN(2),XI IN(1))) * TWOPI,TWOPI)
AMUFIN = XI IN:
CALL MPINDX(AMUFIN,P HIFIN,K,L)

RECORD THE RESULT FOR THE APPROPRIATE R OR T CONTRIBUTION
IF(I.LT.0) THEN
DOWNWARD INITIAL RAY
IF(AMUFIN.GT.0.0) THEN
UPWARD FINAL RAY
NUMOU = NUMOU + 1
WRITE(NUMOU,11,F,L,RAD)
ELSE
DOWNWARD FINAL RAY
NUMDD = NUMDD + 1
WRITE(NUMDD,11,F,L,RAD)
ENDIF

ENDIF
§2. PROGRAM 1

C
C ELSE
C C UPWARD INITIAL RAY
C IF(AMUPIN,GT,0,0) THEN
C C UPWARD FINAL RAY
C IF(RAD.EQ.1.0) THEN
C C ERROR RAY, DUE TO FINITE HEXAGON
C NUMTP1 = NUMTP1 + 1
C ELSE
C NUMUU = NUMUU + 1
C WRITE(NUUU) I,J,K,L,RAD
C ENDIF
C C ELSE
C C DOWNWARD FINAL RAY
C NUMUD = NUMUD + 1
C WRITE(NUUD) I,J,K,L,RAD
C ENDIF
C ELSE
C C RAY INTERSECTED A FACET.  PUSH REFLECTED AND REFRACTED RAYS INTO
C STACK FOR FURTHER TRACING.  (DISCARD RAYS WITH RADIANCE LE. RADEPS)
C C IF(RREFL.GT.RADEPS) THEN
C CALL PUSH(RREFL,P,XIREFL)
C ELSE
C NREFLO = NREFLO + 1
C ENDIF
C C IF(RREFR.GT.RADEPS) THEN
C CALL PUSH(RREFR,P,XIREFR)
C ELSEIF(RREFR.LE.0.0) THEN
C NTIR = NTIR + 1
C ELSE
C NREFRO = NREFRO + 1
C ENDIF
C ENDIF
C C HAVE ALL RAYS BEEN FOLLOWED TO TERMINATION
C IF(NSTACK.GT.0) THEN
C C READ A NEW RAY FROM THE STACK AND TRACE
C C CALL PULL(RAD,PIN,XIIN)
C GO TO 999
C ENDIF
C C***** THIS IS THE END OF THE RECURSIVE TREE FOR THE GIVEN INITIAL RAY *****
C C IF(KBRNCH.LT.10) THEN
C NBRNCH(KBRNCH) = NBRNCH(KBRNCH) + 1
C ELSE
C NBRNCH(10) = NBRNCH(10) + 1
C ENDIF
C 1000 CONTINUE
C 1001 CONTINUE
C C***** END OF COMPUTATIONS *****
C C ENDFILE NUDD
C ENDFILE NUDD
C ENDFILE NUDD
C ENDFILE NUDD
C NRAYTL = MISC(17)
C WRITE(6,600) NRAYTL,KTRACE
C WRITE(6,601) NREFLO,RADEPS,NREFR,RADEPS,NTIR
C WRITE(6,602) NUMUU,NUMDD,NUMUD,NUMUU,NUMTP1
C WRITE(6,604) (K,K=2,10),(NBRNCH(K),K=2,10)
C WRITE(6,605)
§2. PROGRAM 1

FORMATS

514 FORMAT(1H7 , ' NREAD = ', I2 , 'X.
1 "FILE OF SURFACE REALIZATIONS EXHAUSTED: UNIT ', I3 , ' REWOUND."
560 FORMAT(1H7 , ' END OF RAY TRACING COMPUTATIONS.
21H7 , ' TOTAL RAYS WERE STARTED IN THIS RUN.'
31H7 , ' TOTAL RAYS WERE TRACED TO COMPLETION'.
601 FORMAT(1H7 , ' REFLECTED RAYS WITH RADIANCE .LT. ', 1PE9.1,
71H7 , ' WERE DISCARDED.'
81H7 , ' REFRACTED RAYS WITH RADIANCE .LT. ', 2PE9.1,
91H7 , ' WERE DISCARDED'.
10H7 , ' TOTAL INTERNAL REFLECTIONS OCCURRED:
602 FORMAT(1H7 , ' RAYS STARTED DOWNWARD AND FINISHED UPWARD'.
11H7 , ' RAYS STARTED DOWNWARD AND FINISHED DOWNWARD'.
12H7 , ' RAYS STARTED UPWARD AND FINISHED UPWARD'.
13H7 , ' RAYS STARTED UPWARD AND FINISHED DOWNWARD WITH RAD = 1
5.0 (DISCARDED)'.
604 FORMAT(1H7 , ' BRANCH OCCURRENCE TALLY'.
18H7 , ' NUM BRANCHES: ', 18H7 , ' OR MORE'.
19H7 , ' NUM OCCURRENCES: ', 18H7 , '.
605 FORMAT(1H7 , ' NORMAL EXIT FROM NHM, PROGRAM I."

SUBROUTINE INITMH (NREAD, NRAYQD, ISEED)

ON NHM/INITIAL

THIS ROUTINE INITIALIZES NHM/INITIAL

TWO INPUT RECORDS ARE READ:

RECORD 1 (DEFINE THE HEXAGON GRID AND THE WATER SURFACE):

I0BUG = 0 FOR MINIMAL OUTPUT
 = 1 FOR GREATER OUTPUT
 = 2 FOR FULL DEBUGGING OUTPUT
NGENSF = 0 IF A FILE OF RANDOM SURFACES ALREADY EXISTS (USUAL CASE)
 = GT.0 IF "GENSF" IS A SPECIAL RUN FOR GENERATING AND SAVING A
FILE OF RANDOM SURFACES. IGENSF SURFACES WILL BE GENERATED.
NHEx = THE ORDER OF THE HEXAGONAL SURFACE GRID ( = MXNHEX FOR EFFICIENCY)
WINDSPD = THE WIND SPEED IN M/SEC AT 12.5 M ELEVATION
ISEED = THE SEED FOR RANDOM NUMBER GENERATION

RECORD 2 (DEFINE THE QUAD GRID AND SELECTS RAY PARAMETERS):

NMU = THE NUMBER OF MU CELLS IN ONE HEMISPHERE (0 TO PI/2)
NPHI = THE NUMBER OF PHI CELLS (0 TO 2*PI)
MUPART = 1 IF ALL QUADS ARE TO HAVE EQUAL SOLID ANGLES
 = 2 IF ALL QUADS ARE TO HAVE EQUAL DELTA THETA VALUES
NREADD = 1, IF THE SURFACE REALIZATION FILE (NUSFC) IS TO BE READ
 FROM THE BEGINNING
 = 2, 3, OR 4, IF NUSFC IS TO BE READ STARTING WITH A ROTATION
 OR INVERSION (SEE LOOPS 55 IN MAIN)
NUMRAY = IF NUMRAY.GT.0, NUMRAY IS THE NUMBER OF RAYS
 TO BE SENT FROM EACH INPUT QUAD (NRAYQD(IR) = NUMRAY)
 = IF NUMRAY.LT.0, THE NEXT RECORD GIVES
NRAYQD(IR), IR=1,2,...,NMU

13
§2. PROGRAM 1

PARAMETER(MXMu=10, MXPHI=24)
PARAMETER(MXNHEX=7, MXNUDE=3*MXNHEX*(MXNHEX+1)+1)
COMMON/CMPHII, BNDMU(MXMu), BNDPHI(MXPHI)
COMMON/CHEXGR, MXHEX, R1(2), R2(2), R1HAT(2), R2HAT(2), R1RAT, TARGET(2)
COMMON/CNODES, MXNODE, NNODE(2), MMODE, ZNODE(2), NODE(2), MNODE(2)
COMMON/CMISS, IMISS(20), FMISS(20)
DIMENSION FMU(MXMU), PHI(MXPHI), OMEGA(MXMU), DELTMU(MXMU), NRAYQD(MXMU)

C DATA PI, DEGRAD, RADEG/3.141592654, 0.017453292525, 5.2719577957/
DATA REFR/1.3333333333/
DATA DELTA, EPS/1.0, 1.1111, TARGET/0.5, 0.37033333333333/
DATA NUSFC, NUDU, NUDD, NUJD, NNUU0/1.006, 17, 18, 19/

C READ THE INPUT RECORDS

READ(5.*) IDBUG, IGENSF, NHEX, WNDSPD, DSEED
WRITE(6, 300) NHEX, WNDSPD, DSEED

C IF(IGENSF.EQ.0) THEN
READ(5.*) NMU, NPHI, MUPART, NREADO, NUMRAY
C IF(NUMRAY.LT.0) THEN
READ(5.*) (NRAYQD(IR), IR=1, NMU)
ELSE DO 40 IR=1, NMU
40 NRAYQD(IR) = NUMRAY
ENDIF

C GET THE TOTAL NUMBER OF RAYS TO BE THAIEd
NUMRAY = 0
DO 100 I=1, NMU
100 NUMRAY = NUMRAY + NRAYQD(I)

C WRITE(6, 301) NMU, NPHI, NREADO, NUMRAY

ENDIF

C STORE THE NEEDED PARAMETERS

IMISS(1) = NMU
IMISS(2) = NPHI
IMISS(9) = IDBUG
IMISS(17) = NUMRAY
FMISS(1) = PI
FMISS(2) = DEGRAD
FMISS(3) = RADEG
FMISS(15) = WNDSPD
FMISS(16) = DELTA
FMISS(17) = EPS
FMISS(18) = REFR

C RAD48 IS THE CRITICAL ANGLE FOR TOTAL INTERNAL REFLECTION
RAD48 = ASIN(1.0/REFR)
FMISS(19) = RAD48

C IF(IGENSF.GT.0) THEN

C **** THIS IS AN INITIAL RUN FOR GENERATION OF A FILE OF RANDOM SURFACES

WRITE(6, 304)
REWIND NUSFC

C CHECK TO SEE IF NUSFC IS EMPTY
READ(NUSFC, END=50) DUMMY
STOP SURFACE FILE ALREADY EXISTS

REWIND NUSFC

C DEFINE GRID VECTORS AS IN 63/PAGES 24-26

Gamma1 = 1.0/SQRT(0.25*DELTA*DELTA + EPS*EPS)
R1(1) = 0.5*DELTA*Gamma1
R1(2) = EPS*Gamma1
R2(1) = -R1(1)
R2(2) = R1(2)
R1HAT(1) = -R1(1)/R1(2)
R1HAT(2) = R1(1)/R1(2)
§2. PROGRAM 1

R2HAT(1) = -R2(2)
R2HAT(2) = R2(1)
RIRAT = -2.0*EPS/DELTA

DEFINE THE HEXAGONAL SURFACE GRID NODE LOCATIONS
FMISC(16) = DELTA
FMISC(17) = EPS
CALL TRIADS(NMEX)

WRITE THE HEADER RECORDS
WRITE(NUSFC) NUSFC,NMEX,NNODE,WINDSPD,DSEED
WRITE(NUSFC) R1,R2,RIRAT,R2HAT,RIRAT,NNODE

DEFINE THE STANDARD DEVIATION FOR SURFACE HEIGHTS BY 63/2.12
SIGUSFC = 0.0397*SORT(WINDSPD)
WRITE(6,302) DELTA, EPS, SIGUSFC

GENERATE AND SAVE THE CAPILLARY WAVE SURFACE REALIZATIONS,
63/SECTION 2C

DO 55 NSFC=1,IGENSF

DRAW N(0,1) RANDOM NUMBERS
CALL GGNML(DSEED,NNODE,ZNODE)

CONVERT TO N(0 SIGSFC**2) RANDOM NUMBERS
ZMAX = -1.0E30
ZMIN = 1.0E30
DO 99 IRAN=1,NNODE
ZN = SIGSFC*ZNODE(IRAN)
ZNODE(IRAN) = ZN
IF(ZN.GT.ZMAX) ZMAX = ZN
IF(ZN.LT.ZMIN) ZMIN = ZN
99 CONTINUE

55 WRITE(NUSFC) NSFC,ZMIN,ZMAX,ZNODE(I),I=1,NNODE

ENDIF IF(SIGUSFC

***** THIS IS A PRODUCTION RUN FOR RAY TRACING

READ THE EXISTING FILE OF SURFACE REALIZATIONS AND TEST FOR
COMPATABILITY WITH REQUESTED PARAMETERS
WRITE(6,70B) REWIND NUSFC
READ(NUSFC) NSFC,NMEX,NNODE,WIND1
READ(NUSFC) R1,R2,RIRAT,R2HAT,RIRAT,NNODE
IF(NMEX.NE.NMEX .OR. WIND1.NE.WINDSPD) THEN
WRITE(6,70) NMEX, WIND1
STOP
ENDIF

DEFINE THE N+1 AND PHI VALUES WHICH FORM THE QUAD BOUNDARIES FOR
GEOMETRIC discretization (SECTION 3).
IF(MUPART.EQ.1) THEN
PARTITION THE UNIT SPHERE SO THAT ALL QUADS HAVE EQUAL SOLID ANGLES
CALL EQSANG(NM1,NPHI,DELMU)
ELSEIF(MUPART.EQ.2) THEN
PARTITION THE UNIT SPHERE INTO EQUALLY SPACED THETA VALUES
CALL EOTHET(NM1,DELMU)
ENDIF

15
§2. PROGRAM 1

C DEFINE THE BOUNDARY MU VALUES BY SUMMING THE DELTA MU VALUES
BNDMU(I) = DELTMU(I)
DO 101 I = 2,NMU-1
101 BNDMU(I) = BNDMU(I-1) + DELTMU(I)
BNDMU(NMU) = 1.

C DEFINE THE MU VALUES AT THE QUAD CENTERS
FMU(I) = 0.5*DELMU(I)
DO 104 I = 2,NMU
104 FMU(I) = 0.5*(BNDMU(I-1) + BNDMU(I))

C DEFINE THE BOUNDARY PHIS BY PHI = PHI/2 TO PHI + DPHI/2
DELPHI = 2.0*PI/FLOAT(NPHI)
BNDPHI(I) = 0.5*DELPHI
DO 102 J = 2,NPHI
102 BNDPHI(J) = BNDPHI(J-1) - DELPHI

C DEFINE THE PHI VALUES AT THE QUAD CENTERS
DO 103 J = 1,NPHI
103 PHI(J) = DELPHI*FLOAT(J-1)

C DETERMINE THE SOLID ANGLE OF THE QUADS
C DO 400 I = 1,NMJ-1
400 OMEGA(I) = DELPHI*DELMU(I)
OMEGA(NMU) = 2.0*PI*DELMU(NMU)

C WRITE HEADER RECORDS FOR OUTPUT FILES
REWIND NNUU
REWIND NNUDD
REWIND NNUUD
REWIND NNUUU
WRITE(NNUU) NNUU,'DOWN UP',NRAVQD
WRITE(NNUUD) NNUUD,'DOWN DOWN',NRAVQD
WRITE(NNUUU) NNUUU,'UP DOWN',NRAVQD
WRITE(NNUUDD) NNUUDD,'UP UP',NRAVQD
WRITE(NNUUUU) NNUUUU,'UP UP',NRAVQD

RETURN
C FORMATS

60 FORMAT(1HO,110,'SURFACE REALIZATIONS GENERATED')
70 FORMAT(1HO,'SURFACE REALIZATION FILE NOT COMPATABLE WITH REQUESTE'
10 PARAMETERS')
100 FORMAT(1HO,'MONTE CARLO AIR-WATER SURFACE RAY TRACING')
20 FORMAT(1HO,' THE HEXAGON GRID PARAMETERS FOR THIS RUN ARE')
45X,NMREX = I1.' = ORDER OF THE WAVE FACET HEXAGON'
45X,WINDSPD = '7.8'. = THE WIND SPEED IN M/SEC AT 12.5 M'
55X,USESEP = '.7000.10.' = THE SEED FOR RANDOM NUMBER GENERATION'
301 FORMAT(1HO,' THE QUAD GRID PARAMETERS FOR THIS RUN ARE')
305X,NMU = I1.' = NUMBER OF MU CELLS IN (0,PI/2)'
25X,NEP = I1.' = NUMBER OF PHI CELLS IN (0,2*PI)'
35X,NRAYS = I1.' = THE TOTAL NUMBER OF INPUT RAYS TO BE TRACED'
50 FORMAT(1HO,' WAVE FACET PARAMETERS ARE')
15X,DELTA = .PE10.3/5X, EPS = '.E10.3/5X, SIGSFC = .E10.3)
§2. PROGRAM 1

04 FORMAT(1HO, ' THIS IS AN INITIAL RUN FOR GENERATING A FILE OF CAPILLARY WAVE SURFACE REALIZATIONS')
08 FORMAT(1HO, ' THIS IS A PRODUCTION RUN FOR RAY TRACING')
10 FORMAT(1HO, ' THE MU AND THETA VALUES DEFINING THE QUADS ARE' //
12 15X,' I, CNT MU, THETA',8X,' BNU MU THETA',7X,
14 ' DELTA MU, SOLID ANGLE, NRAYQD')
14 FORMAT(1H0, ' NRAYQD')
16 FORMAT(1HO, ' THE QUADS HAVE A WIDTH OF DELTA PHI =',F7.3,
18 ' DEGREES')
END

SUBROUTINE EQSANG(NMU,NPHI,DELMU)

ON NHM1/EQSANG

THIS ROUTINE PARTITIONS THE UNIT SPHERE INTO MU BANDS WHICH HAVE
EQUAL SOLID ANGLES FOR ALL QUADS, INCLUDING THE POLAR CAP, AS
ON PAGE 22.

DIMENSION DELMU(NMU)

WRITE(6,200)
DMU = FLOAT(NPHI)/FLOAT(NMU*NPHI - NPHI + 1)
00 100 I=1,NMU-1
100 DELMU(I) = DMU
DELMU(NMU) = DMU/FLOAT(NPHI)
RETURN

200 FORMAT(1HO, ' THE UNIT SPHERE IS PARTITIONED SO THAT ALL QUADS HAVE
210 1 EQUAL SOLID ANGLES')
END

SUBROUTINE EQTHET(NMU,DELMU)

ON NHM1/EQTHET

THIS ROUTINE PARTITIONS THE UNIT SPHERE INTO MU BANDS WHICH HAVE
EQUAL DELTA THETA SPACINGS, PLUS A POLAR CAP OF HALF-ANGLE DTHETA/2.
AS ON PAGE 24.

COMMON/CMISC/ IMISC(20),FMISC(20)
DIMENSION DELMU(NMU)

WRITE(6,200)
P12 = 0.5*FMISC(1)

DTHETA = P12/(FLOAT(NMU) - 0.5)
00 100 I=1,NMU-1
100 DELMU(I) = COS(P12 - FLOAT(I)*DTHETA) - COS(P12 - FLOAT(I-1)*DTHETA)
DELMU(NMU) = 1.0 - COS(P12 - FLOAT(NMU-1)*DTHETA)
RETURN

200 FORMAT(1HO, ' THE UNIT SPHERE IS PARTITIONED INTO MU BANDS WHICH HAVE
210 1 EQUAL DELTA THETA SPACING')
END
SUBROUTINE FINTCP(INRAY,A,B,C,PIN,XIIN,SIMI,SI,INTCP,UON)

ON NHMI/FINTCP

THIS ROUTINE DETERMINES IF THE TRACK INTERCEPTS A PARTICULAR FACET.

INPUT IS
INRAY = 1 FOR AN INITIAL RAY, = 0 FOR A DAUGHTER RAY
A, B, C...THE 2-D TRIAD NODE LOCATIONS
PIN...THE INITIAL POINT OF THE CURRENT TRACK
XIIN...THE DIRECTION OF THE CURRENT TRACK
SIMI AND SI...THE DISTANCES S(I-1) AND S(I) ALONG THE TRACK,
MEASURED FROM PIN. SIMI IS T.SI BY CONSTRUCTION.

OUTPUT IS
INTCP = 0 IF THERE IS NO INTERCEPT
1 IF THE TRACK DOES INTERCEPT THE FACET
P = THE 3-D FACET INTERCEPT POINT, IF INTCP = 1

PARAMETER(MXNHEX=7., MXNODE=J*MNHEX*(MXNHEX+1)+1)
DIMENSION A(2),B(2),C(2),PIN(3),PIN(3),UON(3)
COMMON/CNODES/ NNODE,FNODE(2,MXNODE)ZNODE(MXNODE)
COMMON/CMISC/ IMISC(20),FMISC(20)

GET THE NODES ASSOCIATED WITH A, B AND C
CALL GETNOA(INRAY, NA)
CALL GETNOB(INRAY, NB)
CALL GETnoch(INRAY, NC)

IF((NA.EQ.0) .OR. (NA.EQ.NC) .OR. (NB.EQ.NC)) THEN
WRITE(*,300) NA, A, NB, NC, C
STOP
ENDIF

DEFINE THE FACET VERTEX BY
V1A = A(1)
V1B = A(2)
V1C = ZNODE(INRAY)
V2A = B(1)
V2B = B(2)
V2C = ZNODE(INRAY)
V3A = C(1)
V3B = C(2)
V3C = ZNODE(INRAY)

GET THE UNIT OUTWARD NORMAL
UON1 = (V3A - V1A)**2 + (V3B - V1B)**2 + (V3C - V1C)**2
UON2 = (V2A - V1A)**2 + (V2B - V1B)**2 + (V2C - V1C)**2
UON3 = (V1A - V1A)**2 + (V1B - V1B)**2 + (V1C - V1C)**2
SGN = SIGN(SIMI, INRAY)**2/SQRT(UON1*UON1 + UON2*UON2 + UON3*UON3)
JON1 = SGN*UON2
JON2 = SGN*UON3
JON3 = SGN*UON1
JON(1) = JON2
JON(2) = JON2
JON(3) = JON3

GET SQ = S(I)**2
SQ = (XIIN**2 + PIN**2)**2 + (XIIN**2 + PIN**2)**2 + (XIIN**2 + PIN**2)**2

CHECK FOR FACET INTERCEPT BY
IF(SIMI.LT.SQ*XIDXIHM) AND. SQ*XIDXIM(E.SI) THEN
HAVE A FACET INTERCEPT
CHECK INITIAL RAYS TO SEE IF THE RAY IS COMING IN UNDER THE GRID

§2. PROGRAM 1

IF(INRAY.EQ.1) THEN
  IF(XDOTN.EQ.0.0) XDOTN = XIIN(1)*UON(1) + XIIN(2)*UON(2) + XIIN(3)*UON(3)
  IF(XDOTN.LE.0.0 .AND. XDOTN.GT.0.0) IOK = 0
  IF(XDOTN.GT.0.0 .AND. XDOTN.LT.0.0) IOK = 0
ENDIF
C
IF(IOK.EQ.1) THEN
  IF(XIIN(3).LE.0.0 .AND. XDOTN.GT.0.0) IOK = 0
  IF(XIIN(3).GT.0.0 .AND. XDOTN.LT.0.0) IOK = 0
ENDIF
C
IF(IOK.EQ.1) THEN
  IF(PILOT INTERCEPT IS OK)
    INTCP = 1
    P(1) = PIN(1) + SQ*XIIN(1)
    P(2) = PIN(2) + SQ*XIIN(2)
    P(3) = PIN(3) + SQ*XIIN(3)
  ELSE
    INTCP = 0
  ENDIF
C
IF(IOK.EQ.1) THEN
  IF(PILOT INTERCEPT IS OK)
    INTCP = 1
    P(1) = PIN(1) + SQ*XIIN(1)
    P(2) = PIN(2) + SQ*XIIN(2)
    P(3) = PIN(3) + SQ*XIIN(3)
  ELSE
    INTCP = 0
  ENDIF
C
RETURN
C
JOB FORMAT(1,0), SUB FINTCP: ILL-DEFINED FACE!
110X, NODE, A = .15,1P2E12.3//1UX, NODE, B = .15,2E12.3//21UX, NODE, C = .15,2E12.3//
END

SUBROUTINE GETAB(INTRIP, YTM1, YTM2, K1, K2, A, B, C)
C
GIVEN TWO TRIAD INTERCEPT POINTS, YTM1 AND YTM2, AND THEIR K
VALUES, K1 AND K2, THIS ROUTINE RETURNS THE TRIAD VERTICES A, B AND C
C
NOTATION USED: YJK = YJ(K)
C
DIMENSION YTM1(2), YTM2(2), A(2), B(2), C(2)
COMMON/CHKGR1, R11, R12, R21, R22, JMAT(4), R1RAT
COMMON/C12, E2, S(2), FMISC(20), FMISC(120)
C
DELTA = FMISC(1)
EPS = FMISC(11)
C
IF(K1.EQ.K2.EQ.0 THEN
  HAVE CASE 1.6. SEE 03/FIGURE 11. USE 03/3.8.3.12
  IF(K1.EQ.0 THEN
    Y1 = YTM1(2)
    Y2 = YTM1(2)
    Y3 = YTM1(2)
  ELSE
    Y1 = YTM2(2)
    Y2 = YTM2(2)
    Y3 = YTM2(2)
  ENDIF
C
RETURN
C
SUBROUTINE GETAB(INTRIP, YTM1, YTM2, K1, K2, A, B, C)
§2. PROGRAM

D1 = V11*R1*RAT + V12
D2 = -V21*R1*RAT + V22
A1 = 0.25*(D2-D1)*DELTA/EPS
A2 = 0.5*(D1+D2)
A(1) = A1
A(2) = A2

SGN1 = SIGN(1.0, (V11-A1)*R11 + (V12-A2)*R12)
SGN2 = SIGN(1.0, (V21-A1)*R21 + (V22-A2)*R22)

B(1) = A1 + SGN1*DELTA*0.5
B(2) = A2 + SGN1*EPS
C(1) = A1 - SGN2*DELTA*0.5
C(2) = A2 + SGN2*EPS

ELSEIF(KS1*KS2.EQ.1) THEN
HAVE CASE I-R1 (SEE 63/FIGURE B). USE 63/3.13

IF(KS1.EQ.0) THEN

Y0 IS VTIP1, Y1 IS VTIP2
Y01 = VTIP1(1)
Y02 = VTIP1(2)
Y11 = VTIP2(1)
Y12 = VTIP2(2)
ELSE

V0 IS VTIP2, V1 IS VTIP1
V01 = VTIP2(1)
V02 = VTIP2(2)
V11 = VTIP1(1)
V12 = VTIP1(2)
ENDIF

A1 = 0.5*(V02 - V11*R1*RAT - V12)*DELTA/EPS
A(1) = A1
A(2) = V02

SGN1 = SIGN(1.0, (V11-A1)*R11 + (V12-V02)*R12)

B(1) = A1 + SGN1*DELTA*0.5
B(2) = V02 + SGN1*EPS
C(1) = A1 + SIGN(1.0, Y01 - A1)*DELTA
C(2) = V02

ELSEIF(KS1*KS2.EQ.2) THEN
HAVE CASE I-R2 (SEE 63/FIGURE B). USE 63/3.14

IF(KS1.EQ.0) THEN

V0 IS VTIP1, Y2 IS VTIP2
V01 = VTIP1(1)
V02 = VTIP1(2)
V21 = VTIP2(1)
V22 = VTIP2(2)
ELSE

V0 IS VTIP2, . . . IF VTIP1
V01 = VTIP2(1)
V02 = VTIP2(2)
V21 = VTIP1(1)
V22 = VTIP1(2)
ENDIF

A1 = 0.5*(-V02*R1*RAT + V22 - V02)*DELTA/EPS
A(1) = A1
A(2) = V02

SGN2 = SIGN(1.0, (V21-A1)*R21 + (V22-V02)*R22)
§2. PROGRAM 1

\begin{verbatim}
  \begin{verbatim}
  B(1) = A1 - SGN2*DELTA*0.5
  B(2) = Y02 + SGN2*EPS
  C(1) = A1 + SGN(1.0, Y01 - A1)*DELTA
  C(2) = Y02
  \end{verbatim}

  ELSE
  \begin{verbatim}
  ERROR IN INPUT
  WRITE(6,100) YTIP1, YTIP2, KS1, KS2
  STOP
  END IF
  RETURN
  \end{verbatim}

  \begin{verbatim}
  100 FORMAT(1HO.', ' ERROR IN SUB GETABC //1H ', ' YTIP1 =.', IP2E12.3,4X,
  1'YTIP2 ='.2E12.3,4X,'K(1), K(2) ='.213)
  END
  \end{verbatim}

  SUBROUTINE GETNCH(A, NODE)
  \begin{verbatim}
  COMMON/CHEXGR/ NHEX
  COMMON/CNODES/ NNODES, FNODE(2, MXNODE)
  COMMON/CMISC/ IMISC(20), FMISC(20)
  \end{verbatim}

  \begin{verbatim}
  ON NHM1/GETNOD
  GIVEN A VECTOR A, WHICH LOCATES ANY POINT IN THE HEXAGON, THIS
  ROUTINE RETURNS THE INDEX, NODE, OF THE NEAREST TRIAD NODE.
  \end{verbatim}

  \begin{verbatim}
  PARAMETER(MXN=EX+7, MXNODE=3*MXN+EX*(MXN+EX+1)+1)
  \end{verbatim}

  \begin{verbatim}
  AY = A(2) - 0.5*EPS
  K = 1
  IF(FNODE(2,K).GT.AY) GO TO 100
  DO 200 J=1, NHEX-1
  K = K + NHEX - J
  IF(FNODE(2,K).LT.AY) GO TO 100
  200 CONTINUE
  DO 204 J=NHEX, 2, -1
  K = K + NHEX + J
  IF(FNODE(2,K).LT.AY) GO TO 100
  204 CONTINUE
  \end{verbatim}

  \begin{verbatim}
  NOW CHECK X VALUES ALONG CONSTANT Y ROW
  AX = A(1) - 0.5*DELTA
  DO 204 J=1, NNODES
  IF(FNODE(1,J).GT.AX) GO TO 102
  204 CONTINUE
  WRITE(6,206) A
  STOP
  \end{verbatim}

  \begin{verbatim}
  102 NODE = J
  RETURN
  \end{verbatim}

  \begin{verbatim}
  206 FORMAT(1HO.', ' SUB GETNOD: POINT A = ('IP1E12.3, ',',E12.3,1') NOT WITHIN HEXAGON )
  END
  \end{verbatim}
\end{verbatim}

21
§2. PROGRAM 1

SUBROUTINE MPINDX(FMU,PHI,I,J)
C
C ON NHMI/MPINDX
C
C GIVEN A (MU,PHI) POINT, THIS ROUTINE RETURNS THE INDICES (I,J)
C OF THE QUAD QIJ WHICH CONTAINS THE POINT.
C
C -1.0 .LE. FMU .LE. 1.0 AND 0.0 .LE. PHI .LE. 2*PI
C
PARAMETER (MXMU=10, MXPHI=24)
COMMON/CMISC/ IMISC(20)
COMMON/CMUPHI/ BNDMU(MXMU),BNDPHI(MXPHI)
C
NMU = IMISC(1)
NPHI = IMISC(2)
ABSMU = ABS(FMU)
C
C SEARCH THE MU BOUNDARY VALUES
C
DO 400 IB=1,NMU
  IF(ABSMU.LE.BNOMU(IB)) GO TO 402
400 CONTINUE
402  I = IB
C
C SEARCH THE PHI BOUNDARY VALUES
C
DO 404 JB=1,NPHI
  IF(PHI.LT.BNDPHI(JB)) GO TO 406
404 CONTINUE
406  J = JB
C
RETURN
END

SUBROUTINE P2ARAY(A,NR,NC,IDIM,IDFMT,TITLE)
C
C THIS ROUTINE PRINTS OUT AN ARRAY A OF NR ROWS AND NC COLUMNS ON ANY
C OF A NUMBER OF FORMATS. IDIM IS THE ROW DIMENSION OF A IN THE
C CALLING PROGRAM. THE VALUE OF IDFMT SPECIFIES THE FORMAT:
C
IDFMT = 1 FOR 10F12.4
   2 FOR 1P10E12.4
   3 FOR 10I12
   4 FOR 12A10
   5 FOR 1P6E20.8
   6 FOR 20F5.1
   7 FOR 1P5E25.15
C
THE ARRAY IS PARTITIONED BY COLUMNS INTO BLOCKS, AND ROWS AND
COLUMNS ARE NUMBERED. THE CHARACTER STRING TITLE CONTAINS ANY
DESIRED TITLE (UP TO 130 CHARACTERS FOR PRINTER OUTPUT).
C
DIMENSION A(IDIM,NC)
CHARACTER TITLE(*)
§2. PROGRAM 1

C SET UP THE PROPER FORMATS
C
KSIZE = 10
ASSIGN 910 TO IFMT1
IF(IDFMT.EQ.1) THEN
ASSIGN 11 TO IFMT2
ELSEIF(IDFMT.EQ.2) THEN
ASSIGN 21 TO IFMT2
ELSEIF(IDFMT.EQ.3) THEN
ASSIGN 31 TO IFMT2
ELSEIF(IDFMT.EQ.4) THEN
KSIZE = 12
ASSIGN 912 TO IFMT1
ASSIGN 41 TO IFMT2
ELSEIF(IDFMT.EQ.5) THEN
KSIZE = 6
ASSIGN 906 TO IFMT1
ASSIGN 51 TO IFMT2
ELSEIF(IDFMT.EQ.6) THEN
KSIZE = 20
ASSIGN 920 TO IFMT1
ASSIGN 61 TO IFMT2
ELSEIF(IDFMT.EQ.7) THEN
KSIZE = 5
ASSIGN 905 TO IFMT1
ASSIGN 71 TO IFMT2
ELSE
WRITE(6,100) IDFMT
RETURN
ENDIF

C PARTITION ARRAY
C
KMANY = (((NC-1)/KSIZE) + 1)*KSIZE
IBLOCK = 0
NBLOCK = 1
IF(NR.LE.25) NBLOCK = 60/(NR+4)

C PRINT ARRAY
C
DO 210 L=KSIZE,KMANY,KSIZE
L1 = L - (KSIZE - 1)
L2 = L
IF(L.GE.KMANY) L2 = NC

C PRINT TITLE AND COLUMN HEADINGS IF NEW PAGE
I BLOCK = 1
WRITE(6,110) TITLE
WRITE(6,IFMT1) (K,K=L1,L2)
ELSE

WRITE A LINE OF DATA
210 WRITE(6,IFMT2) I,(A(I,J),J=L1,L2)
C RETURN

C FORMATS
C
IFMT1 FOR COLUMN LABELS
Y15 FORMAT(1H//10X,12S)
Y90 FORMAT(1H//10X,12I)
Y10 FORMAT(1H//10X,10I)
Y12 FORMAT(1H//10X,12I)
Y20 FORMAT(1H//10X,15I)
SUBROUTINE P3ARAY(A, NR, NC, NP, IDIM, JDIM, IDFMT, TITLE)

C THIS ROUTINE PRINTS OUT AN ARRAY A OF NR ROWS, NC COLUMNS AND
C NP PLANES ON ANY OF A NUMBER OF FORMATS. IDIM AND JDIM ARE THE
C ROW AND COLUMN DIMENSIONS OF A IN THE CALLING PROGRAM. THE VALUE
C OF IDFMT SPECIFIES THE FORMAT:
C
IDFMT = 1 FOR 10F12.4
2 FOR 1P10E12.4
3 FOR 10F12.4
4 FOR 1P10E12.4
5 FOR 1P6E20.8
6 FOR 20F5.1
7 FOR 1P5E25.15
C
THE ARRAY IS PRINTED BY PLANES. FOR EACH PLANE
THE ARRAY IS PARTITIONED BY COLUMNS INTO BLOCKS, AND ROWS AND
COLUMNS ARE NUMBERED. THE CHARACTER STRING TITLE CONTAINS ANY
C DESIRED TITLE (UP TO 130 CHARACTERS FOR PRINTER OUTPUT).
C
DIMENSION A(IDIM, JDIM, NP)
CHARACTER TITLE(*)
C
SET UP THE PROPER FORMATS
C
KSIZE = 10
ASSIGN 910 TO IFMT1
IF(IDFMT.EQ.1) THEN
ASSIGN 11 TO IFMT2
ELSEIF(IDFMT.EQ.2) THEN
ASSIGN 21 TO IFMT2
ELSEIF(IDFMT.EQ.3) THEN
ASSIGN 31 TO IFMT2
ELSEIF(IDFMT.EQ.4) THEN
KSIZE = 12
ASSIGN 912 TO IFMT1
ASSIGN 41 TO IFMT2
ELSEIF(IDFMT.EQ.5) THEN
KSIZE = 6
ASSIGN 906 TO IFMT1
ASSIGN 51 TO IFMT2

24
ELSEIF(IDFMT.EQ.6) THEN
  KSIZE = 20
  ASSIGN 920 TO IFMT1
  ASSIGN 61 TO IFMT2
ELSEIF(IDFMT.EQ.7) THEN
  KSIZE = 5
  ASSIGN 905 TO IFMT1
  ASSIGN 71 TO IFMT2
ELSE
  WRITE(6,100) IDFMT
  RETURN
ENDIF

C
PARTITION ARRAY
C
KMANY = (((NC-1)/KSIZE) + 1)*KSIZE
IBLOCK = 0
NBLOCK = 1
IF(NR.LE.25) NBLOCK = 60/(NR+4)
C
PRINT ARRAY
DO 210 IP=1,NP
  DO 210 L=KSIZE,KMANY,KSIZE
    L1 = L - (KSIZE - 1)
    L2 = L
    IF(L.GE.KMANY) L2 = NC
    DO 210 I=1,NR
      IF(MOD(I-1,50).NE.0) GO TO 210
      IF(IBLOCK.EQ.0 .OR. IBLOCK.GE.NBLOCK) THEN
        WRITE(6,110) TITLE,IP
        WRITE(6,IFMT1) (K,K=L1,L2)
      ELSE
        WRITE(6,IFMT1) (K,K=L1,L2)
      ENDIF
    210 WRITE(6,IFMT2) 1.(A(I,J*IP),J=LI,L2)
  END
RETURN
C
FORMATS
C
IFMT1 FOR COLUMN LABELS
  905 FORMAT(1H //10X.5125)
  906 FORMAT(1H //10X.8120)
  910 FORMAT(1H //10X.9112)
  912 FORMAT(1H //10X,121I0)
  920 FORMAT(1H //10X,2015)
C
IFMT2 FOR DATA
  11 FORMAT(1H ,19,10F12.4)
  21 FORMAT(1H ,19,10E12.3)
  31 FORMAT(1H ,19,10I12)
  41 FORMAT(1H ,19,12A10)
  51 FORMAT(1H ,19,12A20.8)
  61 FORMAT(1H ,19,2GF5.1)
  71 FORMAT(1H ,19,1P5E25.15)
C
100 FORMAT(1H0,"INVALID FORMAT OPTION IN PARRAY, IDFMT = ",I5)
110 FORMAT(1H1,A//" THREE-DIMENSIONAL ARRAY, PLANE (THIRD INDEX)",13)
§2. PROGRAM 1

SUBROUTINE PULL(R,P,XI)
C
ON NHMI/PULL
C
C
AS DESCRIBED IN 63/PAGE 11.
C
PARAMETER (MXSTAK=10)
COMMON /CSTACK/ NSTACK,STACK(MXSTAK,7)
DIMENSION P(3),XI(3)
C
GET THE BOTTOM ELEMENTS
C
R = STACK(NSTACK,1)
DO 200 I=1,3
P(I) = STACK(NSTACK,I+1)
200 XI(I) = STACK(NSTACK,I+4)
NSTACK =NSTACK - 1
C
RETURN
END

SUBROUTINE PUSH(R,P,XI)
C
ON NHMI/PUSH
C
THIS ROUTINE PUSHES R, P AND XI ONTO THE BOTTOM OF THE STACK
C
AS DESCRIBED IN 63/PAGE 11.
C
PARAMETER (MXSTAK=10)
COMMON /CSTACK/ NSTACK,STACK(MXSTAK,7)
DIMENSION P(3),XI(3)
C
TEST FOR OVERFLOW OF STACK
C
IF(NSTACK.GE.MXSTAK) THEN
WRITE(6,100) NSTACK
RETURN
ENDIF
C
ADD NEW ELEMENTS AT THE BOTTOM
C
NSTACK = NSTACK + 1
STACK(NSTACK,1) = R
DO 200 I=1,3
STACK(NSTACK,I+1) = P(I)
200 STACK(NSTACK,I+4) = XI(I)
RETURN
C
100 FORMAT(1HO,' STACK FULL, NSTACK = ',I6,' RAY DISCARDED')
END
FUNCTION REFLF(THETAP, THETA)
ON NHMI/REFL

THIS FUNCTION RETURNS THE REFLECTANCE, GIVEN THE REFLECTED AND
REFRACTED ANGLES, THETA PRIME AND THETA

COMMON/CMISC/ IMISC(20), FMISC(20)
DATA EPSO/1.0E-5/

PI = FMISC(1)
REFR = FMISC(18)
TPMT = THETAP - THETA
TPPT = THETAP + THETA

CHECK FOR NORMAL INCIDENCE
IOK = 0
IF(ABS(TPMT).GT.EPSO .AND. ABS(TPMT-PI).GT.EPSO) IOK = IOK + 1
IF(ABS(TPPT).GT.EPSO .AND. ABS(TPPT-PI).GT.EPSO) IOK = IOK + 1
IF(IOK.EQ.2) THEN
63/3.20
REFL = 0.5*((SIN(TPMT)/SIN(TPPT))**2 + (TAN(TPMT)/TAN(TPPT))**2)
ELSE
USE LIMITING CASE FOR NORMAL INCIDENCE
REFL = ((REFR - 1.0)/(REFR + 1.0))**2
ENDIF

RETURN
END

SUBROUTINE RSPLIT(RIN, XIIN, UON, RREFL, XIREFL, RREFR, XIREFR)

NHMI/RSPLIT

THIS ROUTINE DETERMINES THE REFLECTED AND REFRACTED DIRECTIONS
AND THE ASSOCIATED RADIANCES AT THE INTERCEPTED FACET.
FACTORS OF REFR**2 AND 1/REFR**2 ARE NOT INCLUDED IN THE
TRANSMITTED RADIANCES

INPUT
RIN... THE RADIANCE OF THE INCOMING RAY
XIIN... THE DIRECTION OF THE INCOMING RAY
UON... THE UNIT OUTWARD NORMAL OF THE INTERSECTED FACET

OUTPUT
RREFL... THE RADIANCE OF THE REFLECTED RAY
XIREFL... THE DIRECTION OF THE REFLECTED RAY
RREFR... THE RADIANCE OF THE REFRACTED RAY
XIREFR... THE DIRECTION OF THE REFRACTED RAY

DIMENSION XIIN(3), UON(3), XIREFL(3), XIREFR(3)
COMMON/CMISC/ IMISC(20), FMISC(20)
§2. PROGRAM 1

```c
C
REFR = FMISC(18)
RAD48 = FMISC(19)
XPDOTN = XIIN(1)*UON(1) + XIIN(2)*UON(2) + XIIN(3)*UON(3)
C
IF(XPDOTN.LT.0.0) THEN
C
AIR-INCIDENT CASE
C
REFLECTED AND REFRACTED DIRECTION BY 63/3.18
C
C = XPDOTN + SQRT(XPDOTN*XPDOTN + REFRR*REFRR-1.0)
DO 100 J=1,3
XIREFL(J) = XIIN(J) - 2.0*XPDOTN*UON(J)
100
XIREFR(J) = (XIIN(J) - C*UON(J))/REFR
C
ANGLES BY 63/3.18
C
THETAP = ACOS(ABS(XPDOTN))
THETA = ASIN(SIN(THETAP)/REFR)
R = REFLP(THETAP,THETA)
C
COMPUTE RADIANCES BY 63/3.30 AND 3.31A
RREFL = RIN*R
RREFR = RIN*(1.0-R)
C
ELSE
C
WATER-INCIDENT CASE
C
REFLECTED AND REFRACTED DIRECTIONS BY 63/3.19
C
ARG = (REFR*XPDOTN)**2 - REFRR*REFRR + 1.0
IF(ARG.GE.0.0) THEN
C
C = REFR*XPDOTN - SQRT(ARG)
ELSE
C = 0.0
ENDIF
DO 102 J=1,3
XIREFL(J) = XIIN(J) - 2.0*XPDOTN*UON(J)
102
XIREFR(J) = REFR*XIN(J) - C*UON(J)
C
ANGLES BY 63/3.19
C
THETAP = ACOS(ABS(XPDOTN))
C
COMPUTE THE REFLECTANCE
C
IF(THETAP.GT.RAD48) THEN
C
HAVE TOTAL INTERNAL REFLECTION
R = 1.0
ELSE
C
REFLECTION AND REFRACTION
THETA = ASIN(REFR*SN(THETAP))
R = REFLP(THETAP,THETA)
ENDIF
C
RADIANCES BY 63/3.30 AND 3.31B
RREFL =RIN*R
RREFR = RIN*(1.0 - R)
C
ENDIF
C
RETURN
END
```
§2. PROGRAM 1

SUBROUTINE TIP(P, XI, IALL)

ON NHM1/TIP

GIVEN A POINT P AND A DIRECTION XI, THIS ROUTINE FIRST COMPUTES THE TRACK OF THE RAY P + S*XI AS IN 63/SECTION 3B.

IF IALL = 0, THE COMPUTATIONS ARE CARRIED ONLY TO 63/3.6, AND SMIN IS RETURNED.

IF IALL = 1, THE TRIAD INTERCEPT POINTS YTIP ARE ALSO COMPUTED.

PARAMETER(MXNHEX=7, MXNTIP=4*MXNHEX+1)
DIMENSION P(3), XI(3)
DIMENSION SO(-MXNHEX:MXNHEX), S1(-MXNHEX:MXNHEX), S2(-MXNHEX:MXNHEX)
DIMENSION IR(MXNTIP), WORK(MXNTIP), KWORK(MXNTIP)
COMMON/CHEXGR/NH-EX.R1(2), R2(2), R1HAT1, R1HAT2, R2HAT1, R2HAT2
COMMON/CTIP/ NTIP, SMIN, YTIP(2, MXNTIP), S(MXNTIP), KS(MXNTIP)
COMMON/CMIISC/ IMISC(20), FMISC(20)

DATA EPSDOT/1.0E-8/

EPS = FMISC(17)

COMPUTE THE HORIZONTAL UNIT VECTOR, XIH, AND XI DOT XIH
FMISC(20) = SQRT(XI(1)*XI(1) + XI(2)*XI(2))
XIH1 = XI(1)/FMISC(20)
XIH2 = XI(2)/FMISC(20)
P1 = P(1)
P2 = P(2)
DO 100 L=-NHEX, NHEX
100 SO(L) = 1.E30
S1(L) = 1.E30
100 S2(L) = 1.E30

COMPUTE SO VALUES BY 63/3.5
IF(ABS(XIH2).GT.EPSDOT) THEN
SS = -P2/XIH2
A = EPS/XIH2
DO 110 L=-NHEX, NHEX
110 SO(L) = SS + FLOAT(L)*A
ENDIF

COMPUTE S1 BY 63/3.5
D = XIH1*R1HAT1 + XIH2*R1HAT2
IF(ABS(D).GT.EPSDOT) THEN
SS = -(P1*R1HAT1 + P2*R1HAT2)/D
A = 2.0*EPS*R1HAT2/D
DO 112 L=-NHEX, NHEX
112 S1(L) = SS + FLOAT(L)*A
ENDIF

COMPUTE S2 BY 63/3.5
IF(ABS(XIH1).GT.EPSDOT) THEN
D = XIH1*R2HAT1 + XIH2*R2HAT2
IF(ABS(D).GT.EPSDOT) THEN
SS = -(P1*R2HAT1 + P2*R2HAT2)/D
A = 2.0*EPS*R2HAT2/D
DO 114 L=-NHEX, NHEX
114 S2(L) = SS + FLOAT(L)*A
ENDIF

FIND THE MINIMUM POSITIVE END POINT BY 63/3.6
SMIN = AMIN1(AMAX1(SO(-NHEX)), SO(NHEX)), AMAX1(S1(-NHEX), S1(NHEX)),
1
AMAX1(S2(-NHEX), S2(NHEX))
IF(IALL.EQ.0) RETURN

SELECT THE NON-NEGATIVE S VALUES, LE. SMIN

29
NS = 0
DO 200 L=-NHEX,NHEX
  IF(S0(L).GT.-EPSDOT .AND. S0(L).LT.SMIN+EPSDOT) THEN
    NS = NS + 1
    S(NS) = S0(L)
    KS(NS) = 0
  ENDIF
C
  IF(S1(L).GT.-EPSDOT .AND. S1(L).LT.SMIN+EPSDOT) THEN
    NS = NS + 1
    S(NS) = S1(L)
    KS(NS) = 1
  ENDIF
C
  IF(S2(L).GT.-EPSDOT .AND. S2(L).LT.SMIN+EPSDOT) THEN
    NS = NS + 1
    S(NS) = S2(L)
    KS(NS) = 2
  ENDIF
200 CONTINUE
C
C ORDER THE S VALUES
C
DO 210 I=1,NS
  IR(I) = I
  CALL VSRTS(S,NS,IR)
C
C CORRESPONDINGLY PERMUTE THE ASSOCIATED K VALUES
C
DO 212 I=1,NS
  KWORK(I) = KS(I)
DO 214 I=1,NS
  KS(I) = KWORK(IR(I))
C
C CHECK THE SORTED S VALUES FOR EQUAL ENTRIES. DISCARD DEGENERATE VALUES AND RELABEL THE REMAINING S VALUE WITH AN APPROPRIATE KS VALUE.
C
DO 240 I=1,NS
  WORK(I) = S(I)
240  KWORK(I) = KS(I)
  NTIP = 1
  IF(S(I).LT.0.0) S(I) = 0.0
C
DO 250 I=2,NS
  IF(ABS(WORK(I)-WORK(I-1)).GT.EPSDOT) THEN
    NTIP = NTIP + 1
    S(NTIP) = WORK(I)
    KS(NTIP) = KWORK(I)
  ELSE
    KS(NTIP) = I
  ENDIF
250 CONTINUE
C
C COMPUTE THE TRIAD INTERCEPT POINTS FROM THE NON-DEGENERATE S VALUES
C USING 63/3.7
C
DO 300 I=1,NTIP
  YTIP(1,I) = P1 + S(I)*XIM1
  YTIP(2,I) = P2 + S(I)*XIM2
300 RETURN
C
END
SUBROUTINE TRACE(INRAY,RIN,PIN,XIIN, IOUT,P,RREFL,XIREFL,RREFR, 
1 XIREFR)

ON NHM/TRACE

GIVEN AN INITIAL RADIANCE, RIN, STARTING POINT, PIN, AND 
DIRECTION, XIIN. THIS ROUTINE TRACES THE RAY UNTIL IT EITHER 
LEAVES THE HEXAGON REGION OR INTERCEPTS A FACET.

IF THE RAY LEAVES THE HEXAGON BEFORE INTERSECTING A FACET, 
IOUT = 1 AND RETURN IS MADE. 

IF THE RAY INTERCEPTS A FACET BEFORE LEAVING THE HEXAGON, 
P = THE INTERCEPT POINT 
RREFL = THE REFLECTED RADIANCE 
XIREFL = THE DIRECTION OF THE REFLECTED RAY 
RREFR = THE REFRACTED RADIANCE 
XIREFR = THE DIRECTION OF THE REFRACTED RAY 
ARE RETURNED.

PARAMETER(MXNHEX=7, MXNTIP=4*MXNHEX+1)  
COMMON/CTIP/ NTIP,SMIN,VTIP(2,MXNTIP),S(MXNTIP),KS(MXNTIP),ZMIN, 
1 ZMAX 
DIMENSION PIN(3),XIIN(3),P(3),XIREFL(3),XIREFR(3) 
DIMENSION A(2),B(2),C(2),UON(3)

CALL TIP(PIN, XIIN, 1) 
LOCATE THE TRIAD INDICES FOR WHICH AN INTERCEPT IS POSSIBLE 
IF(XIIN(3).LE.U.U) THEN 
  DOWNWARD RAY 
  TANTHP = TAN(ACOS(-XIIN(3))) 
  GET FIRST FACET TO BE CHECKED 
  IF(PIN(3).GT.ZMAX) THEN 
    INITIAL POINT ABOVE THE MAXIMUM SURFACE (INITIAL RAY) 
    D1 = (PIN(3) - ZMAX)*TANTHP 
    DO 50 I=2,NTIP 
    IF(S(I).GE.D1) GO TO 55 
    50 CONTINUE 
  ELSE 
    INITIAL POINT BELOW THE MAXIMUM SURFACE (DAUGHTER RAY OR LOW-ANGLE RAY) 
    I1 = 2 
  ENDIF 
  GET THE LAST FACET TO BE CHECKED 
  D2 = (PIN(3) - ZMIN)*TANTHP 
  DO 60 I=I1,NTIP 
  IF(S(I).GE.D2) GO TO 65 
  60 CONTINUE 
  65 I2 = 1 
  ELSE 
    UPWARD RAY 
    TANTHP = TAN(ACOS(XIIN(3))) 
    GET FIRST FACET 
    IF(PIN(3).LT.ZMIN) THEN 
      INITIAL POINT BELOW THE MINIMUM SURFACE (INITIAL RAY) 
      D1 = (ZMIN - PIN(3))*TANTHP 
      DO 70 I=2,NTIP 
      IF(S(I).GE.D1) GO TO 75 
      70 CONTINUE 
    ELSE 
      75 I1 = 1 

C
ELSE
C INITIAL POINT ABOVE THE MINIMUM SURFACE (DAUGHTER OR LOW-ANGLE RAY)
I1 = 2
ENDIF
C
C GET LAST FACET TO BE CHECKED
D2 = (ZMAX - PIN(3))*TANTHP
DO 80 I=11,NTIP
IF(S(I).GE.D2) GO TO 85
80 CONTINUE
85 I2 = I
ENDIF
I2 = MINO(I2,NTIP)
C
C CHECK POSSIBLE PAIRS OF TRIAD INTERCEPT POINTS FOR A FACET INTERCEPT
C
DO 100 I=11,12
C GET THE TRIAD NODE VECTORS CORRESPONDING TO INTERCEPT POINTS I AND I-1
C CALL GETABL(VTIP(1,I-I),VTIP(I,),KS(I-1),KS(I), A,B,C)
C SEE IF THE RAY TRACK INTERCEPTS THIS FACET
C CALL FINTCP(INRAY,A,B,C,PIN,XIIIN,S(I-1),S(I), INTCP,P,UON)
C IF(INTCP.EQ.1) GO TO 200
C 100 CONTINUE
C IF HERE, NO INTERCEPT WAS FOUND
IOUT = 1
RETURN
C IF HERE, AN INTERCEPT WAS FOUND. COMPUTE THE REFLECTED AND
C REFRACTED RAYS
C 200 IOUT = 0
CALL RSPLIT(RIN.XIIIN,UON, RREFL,XIREFL,RREFR,XIREFR)
C RETURN
C END
SUBROUTINE TRIADS(NHEx)

ON NHM1/STRIADS

GIVEN THE ORDER OF THE HEXAGON, NHEx, THIS ROUTINE DEFINES THE
VECTOR NODES, FNODE, OF THE HEXAGON TRIADS IN UNITS OF DELTA AND
EPSILON, AS IN 63/PAGE 26.

PARAMETER(MXNHEx=7, MXNODE=3*MXNHEx*(MXNHEx+1)+1)
COMMON/CNODES/ NNODE,FNODE(2,MXNODE)
COMMON/CMISC/ IMISC(20),FMISC(20)

DELA = FMISC(16)
EPS = FMISC(17)
NF = 0
IPRINT = 0
DO 100 IC=-NHEx,NHEx
CEPS = FLOAT(IC)*EPS
IF(MOD(IC,2).EQ.0) THEN
C C IS EVEN
MXB = NHEx - IABS(IC)/2
DO 200 IB=-MXB,MX3
NF = NF + 1
FNODE(1,NF) = FLOAT(IB)*DELA
200 FNODE(2,NF) = CEPS
ELSE
C C IS ODD
MXB = NHEx - (IABS((IC)*1)/2
DO 210 IB=-MXB,0
NF = NF + 1
FNODE(1,NF) = (-0.5+FLOAT(IB))*DELA
210 FNODE(2,NF) = CEPS
ENDIF
100 CONTINUE

NNODE = 3*NHEx*(NHEx+1) + 1
IF(NF.EQ.NNODE) THEN
IF(IPRINT.EQ.1) WRITE(6,300) NNODE,NHEx
ELSE
WRITE(6,302) NHEx,NF,NNODE
STOP
ENDIF

IF(IPRINT.EQ.1) THEN
WRITE(6,304)
DO 306 I=1,NNODE,5
306 WRITE(6,308) I,1*4,1(FNODE(1,I+J),FNODE(2,I+J),J=0,4)
ENDIF

RETURN

300 FORMAT(1H0,1H ' SUB TRIADS: 13,11, NODES DEFINED FOR AN ORDER', 112, ' HEXAGONAL GRID')
302 FORMAT(1H0,1H ' ORDER IN SUB TRIADS 1HM ', NHEx = ',12,4X', NF = ', 114,4X', NNODE = ',14)
304 FORMAT(1H0,1H ' THE HEXAGON GRID N0DES ARE CULATED AT/')
308 FORMAT(1H ', NODES',13,1H AT',5I0,1H ',F7.2','F7.2'))
PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE16,TAPE17,TAPE18,TAPE19,TAPE15)

**********************************************************************
+ THIS IS PROGRAM 1 OF THE NATURAL HYDROSOL MODEL +
+ **********************************************************************

ON NHMI/MLIQD FTN5/FTN200

THIS PROGRAM BEGINS COMPUTATION OF THE QUAD-AVERAGED GEOMETRIC REFLECTANCE AND TRANSMITTANCE ARRAYS WHICH DESCRIBE THE AIR-WATER INTERFACE FOR A GIVEN WIND SPEED.

THIS SPECIAL VERSION OF MAIN1 DOES ONLY ONE INPUT QUAD (ONE ROW OF R OR T).

NOTE: THIS VERSION OF THE CODE STRIVES TO MINIMIZE THE EXECUTION TIME. AT THE EXPENSE OF MODULARITY AND READABILITY OF THE CODE. SOME SECTIONS OF FREQUENTLY EXECUTED CODE ARE WRITTEN AS STRAIGHT LINE CODE WITH SIMPLE VARIABLES, RATHER THAN BEING GROUPED IN SUBROUTINES OR DO LOOPS WITH ARRAYS. IN ORDER TO AVOID CALLING AND INDEXING OVERHEAD. ALMOST ALL ERROR CHECKING AND INTERMEDIATE OUTPUT HAS BEEN REMOVED.

THIS PROGRAM USES THE MONTE CARLO RAY TRACING TECHNIQUE DESCRIBED IN NOAA TECH MEMO ERL-PMEL-63. COMMENTS REFERRING TO THIS REPORT ARE PREFACED BY 63/. THIS 63/2.12 REFERS TO EQUATION 2.12 IN TECH MEMO 63. REFERENCES WITHOUT THE 63/ REFER TO NOAA TECH MEMO ERL-PMEL-75.

NUFC = TAPE15...CONTAINS THE RANDOM SURFACE REALIZATIONS

RESULTS OF COMPUTED RAY PATHS ARE WRITTEN TO FILES AS FOLLOWS:

NUDU = TAPE16...INITIAL RAY DOWNWARD, FINAL RAY UPWARD: R = \( R(A,X) \)
NUDD = TAPE17...INITIAL RAY DOWNWARD, FINAL RAY DOWNWARD: T = \( T(A,X) \)
NUU = TAPE18...INITIAL RAY UPWARD, FINAL RAY DOWNWARD: R = \( R(X,A) \)
NUUL = TAPE19...INITIAL RAY UPWARD, FINAL RAY UPWARD: T = \( T(X,A) \)

PROGRAM 2 READS THESE FILES AND TALLIES THE RESULTS TO GENERATE THE ACTUAL R AND T ARRAYS.

PARAMETER(MXNU=10, MXPHI=24, MASTAK=10, MXNHEX=7)
PARAMETER(MXNODE=3*MXNHEX*(MXNHEX+1)+1, MXNTIP=4*MXNHEX+1)

COMMON/CMPHI/, BNDMU(MXNU), BNDPHI(MXPHI)
COMMON/CNODES/, NODE, NODENODE(NXNODE), NODE(NXNODE)
COMMON/CHEUX/, CHEUX(NXHEX), CHEUX(NXHEX)
COMMON/CHAT/, CHAT(NXHEX), CHAT(NXHEX), CHAT(NXHEX), CHAT(NXHEX)
COMMON/CTIP/, CTIP(NXNTIP), CTIP(NXNTIP), CTIP(NXNTIP), CTIP(NXNTIP)
COMMON/CPH/, CMAT(2), CMAT(2), CMAT(2), CMAT(2)
COMMON/CS/, CMAT(2), CMAT(2), CMAT(2), CMAT(2)

COMMON/CMISC/, CMISC(20), CMISC(20)

DIMENSION PI(NMISC), XI(NMISC), XI(NMISC), XI(NMISC)

DOUBLE PRECISION NXNUM

DATA MUSKIPV1, E+11, NUSFC, NUSFO, NUSF1, NUSF2, NUSF3, NUSF4, NUSF5, NUSF6, NUSF7, NUSF8, NUSF9, NUSF10, NUSF11, NUSF12, NUSF13, NUSF14, NUSF15, NUSF16, NUSF17, NUSF18, NUSF19, NUSF20, NUSF21, NUSF22, NUSF23, NUSF24, NUSF25, NUSF26, NUSF27, NUSF28, NUSF29, NUSF30, NUSF31, NUSF32, NUSF33, NUSF34, NUSF35, NUSF36, NUSF37, NUSF38, NUSF39, NUSF40, NUSF41, NUSF42, NUSF43, NUSF44, NUSF45, NUSF46, NUSF47, NUSF48, NUSF49, NUSF50, NUSF51, NUSF52, NUSF53, NUSF54, NUSF55, NUSF56, NUSF57, NUSF58, NUSF59, NUSF60, NUSF61, NUSF62, NUSF63, NUSF64, NUSF65, NUSF66, NUSF67, NUSF68, NUSF69, NUSF70, NUSF71, NUSF72, NUSF73, NUSF74, NUSF75, NUSF76, NUSF77, NUSF78, NUSF79, NUSF80, NUSF81, NUSF82, NUSF83, NUSF84, NUSF85, NUSF86, NUSF87, NUSF88, NUSF89, NUSF90, NUSF91, NUSF92, NUSF93, NUSF94, NUSF95, NUSF96, NUSF97, NUSF98, NUSF99, NUSF100

INITIALIZE THE PROGRAM

CALL INSCARK(14,5,NRAQO,0,SEED)
NM = 1IMISC1)
N2PI = IMISC2)
PI = IMISC3)
T2PI = 2.0*PI

GET MU BOUNDARIES OF THE INCOMING QUAD
FMUMIN = 0;
IA = 1ABS(1R)
IF(IA.GT.1) FMUMIN = BNDMU(IA-1)
DMU = BNDMU(IA) - FMUMIN
§2. PROGRAM 1

IF(IA.EQ.NMU) THEN
  ALL PHI VALUES FOR A POLAR CAP
  PHIMIN = 0.
  DPHI = TWOPI
ELSE
  PHIMIN = BNDPHI(NO2PI)
  IF(JS.GT.1) PHIMIN = BNDPHI(JS-1)
  DPHI = BNDPHI(2) - BNDPHI(1)
ENDIF

C NUMDU = 0
NUMDD = 0
NUMU = 0
NUMUU = 0
NUMTP1 = 0

***** BEGIN COMPUTATIONS *****
C EACH RAY GETS A NEW SURFACE REALIZATION, BUT EACH STORED SURFACE
C REALIZATION IS USED FOUR WAYS TO EXPLOIT SYMMETRY
C
NREAD = 1
NRAYTL = NRAYQD(IA)
DO 1000 NRAY=1,NRAYTL
C SELECT A SURFACE REALIZATION
C
55 CONTINUE
IF(NREAD.EQ.1) THEN
  READ A SURFACE REALIZATION AS GENERATED
  READ(NUSFC,END=50) NSF,ZMIN,ZMAX.(ZNODE(I),I=1,NNODE)
ELSEIF(NREAD.EQ.2) THEN
  READ THE SURFACE AS ROTATED BY 180 DEGREES
  READ(NUSFC,END=50) NSF,ZMIN,ZMAX.(ZNODE(I),I=NNODE,1,-1)
ELSEIF(NREAD.EQ.3) THEN
  READ THE SURFACE AND INVERT
  READ(NUSFC,END=50) NSF,ZMIN,ZMAX.(ZNODE(I),I=1,NNODE)
DO 502 I=1,NNODE
502 ZNODE(I) = -ZNODE(I)
ELSEIF(NREAD.EQ.4) THEN
  READ THE SURFACE AS ROTATED BY 180 DEGREES AND THEN INVERT
  READ(NUSFC,END=50) NSF,ZMIN,ZMAX.(ZNODE(I),I=NNODE,1,-1)
DO 504 I=1,NNODE
504 ZNODE(I) = -ZNODE(I)
ENDIF
C GO TO 506

END OF FILE PROCESSING FOR THE STORED FILE OF CAPILLARY SURFACES
WRITE(6,514) NREAD
NREAD = NREAD + 1
NREAD = NREAD * 1
READ(NUSFC,HEADER)
READ(NUSFC,HEADER)
GO TO 55
C
SElecT CONTINUE
C
C SELECT A RANDOM MU VALUE
177 RMU = (PHIMIN + GGUDBS(DSEED)*MU)*SIGN(1.0, FLOAT(IR))
C NO RAYS FROM THE POLE ITSELF
1F ABS(RMU) GT 1.0-KADEPS) GO TO 777
ROOT = SQRT(1.0 - RMU*RMU)
C
C CHOOSE A RANDOM PHI VALUE
SPHI = AMOD(PHIMIN + GGUDBS(DSEED)*PHI,TWOPI)
C
LOCATE THE INITIAL STARTING POINT FOR THIS TARGET AND DIRECTION
C
C FOLLow THE TRACK BACKWARDS TO THE BOUNDARY TO GET SMIN

35
§2. PROGRAM

C DEFINE THE INITIAL RAY DIRECTION TO BE -XI PRIME
XIN(1) = ROOT*COS(SPHI)
XIN(2) = ROOT*SIN(SPHI)
XIN(3) = RMU
CALL TIP(TARGET.X: LN.O)
C DEFINE THE INITIAL POINT ON THE HEXAGON BOUNDARY
TEMP = SMIN/FMISC(20)
PIN(1) = TARGET(1) + TEMP*XIN(1)
PIN(2) = TARGET(2) + TEMP*XIN(2)
PIN(3) = TEMP*XIN(3)
C
C RESET XIN TO THE INCOMING DIRECTION, XI PRIME
XIN(1) = -XIN(1)
XIN(2) = -XIN(2)
XIN(3) = -XIN(3)
RAD = 1.0
INRAY = 1
C
C PERFORM RAY TRACING COMPUTATIONS
C
C ----- THIS IS THE RECURSIVE TREE FOR A GIVEN INITIAL RAY -----
C
KBRNCH = 0
999 CALL TRAC(T(INRAY,RAD,PIN,XIN, IUUL,P,RREFL,XIREFL,RREFR,XIREFR)
KTRACE = KTRACE + 1
KBRNCH = KBRNCH + 1
INRAY = 0
C
C CHECK FOR RAY HAVING LEFT THE HEXAGON
C
IF(IOUT.EQ.1) THEN
C RAY HAD NO FACET INTERCEPTS.
C GET THE QUAD INDICES OF THE FINAL RAY DIRECTION
PHIFIN = AMODI(ATAN2(XIN(2),XIN(1)) + 2*60,2*60)
AMUFIN = XIN(3)
CALL MPINDEX(AMUFIN,PHIFIN,KU,LV)
C
C RECORD THE RESULT FOR THE APPROPRIATE R OR T CONTRIBUTION
C
IF(IR.GT.0) THEN
C DOWNWARD INITIAL RAY
IF(AMUFIN.GT.0.0) THEN
UPWARD FINAL RAY
NUMDU = NUMDU + 1
WRITE(NUMDU) IR,JS,KU,LV,RAD
ELSE
DOWNWARD FINAL RAY
NUMDD = NUMDD + 1
WRITE(NUMDD) IR,JS,KU,LV,RAD
ENDIF
ELSE
UPWARD INITIAL RAY
IF(AMUFIN.GT.U) THEN
UPWARD FINAL RAY
IF(RAD.EQ.1.0) THEN
ERROR RAY, DUE TO INFINITE HEXAGON
NUMPI = NUMPI + 1
ELSE
NUMPI = NUMPI + 1
WRITE(NUMPI) IR,JS,KU,LV,RAD
ENDIF
ELSE
DOWNWARD FINAL RAY
NUMDU = NUMDU + 1
WRITE(NUMDU) IR,JS,KU,LV,RAD
ENDIF
ENDIF
ENDIF
ELSE
DOWNWAR RAY
NUMDD = NUMDD + 1
WRITE(NUMDD) IR,JS,KU,LV,RAD
ENDIF
ENDIF
ENDIF
ELSE

§2. PROGRAM 1

C RAY INTERSECTED A FACET. PUSH REFLECTED AND REFRACTED RAYS INTO
C STACK FOR FURTHER TRACING. (DISCARD RAYS WITH RADIANCE L.E. RADEPS)

C IF (RREFL.GT.RADEPS) THEN
CALL PUSH (RREFL,P,XIREFL)
ELSE
NREFLO = NREFLO + 1
ENDIF

C IF (RREFR.GT.RADEPS) THEN
CALL PUSH (RREFR,P,XIREFR)
ELSEIF (RREFR.LE.0.0) THEN
NTIR = NTIR + 1
ELSE
NREFRO = NREFRO + 1
ENDIF

C ENDIF

C HAVE ALL RAYS BEEN FOLLOWED TO TERMINATION
C IF (NSTACK.GT.0) THEN
C READ A NEW RAY FROM THE STACK AND TRACE
C CALL PULL (RADPINXIIN)
GO TO 999
ENDIF

C ..... THIS IS THE END OF THE RECURSIVE TREE FOR THE GIVEN INITIAL RAY .....  
C IF (KBRNCH.LT.10) THEN
NBRNCH (KBRNCH) = NBRNCH (KBRNCH) + 1
ELSE
NBRNCH (10) = NBRNCH (10) + 1
ENDIF

1000 CONTINUE
C END OF COMPUTATIONS
C ENDFILE NUDD
ENDFILE NUDD
ENDFILE NUDD
ENDFILE NUDD
WRITE (6,600) NRAVTL, KTRACE
WRITE (6,601) NREFL, RADEPS, NREFR, RADEPS, NTIR
WRITE (6,602) NUMUP, NUMDD, NUMUD, NUMUU, NUMTP1
WRITE (6,604) (K,K=2,10), (NBRNCH(K),K=1,10)
C WRITE (6,1002)
C
C FORMATS

514 FORMAT (1HO, ' NREAD = ',12,3X,  
1 'FILE OF SURFACE REALIZATIONS EXHAUSTED. FILE REWOUND. ')
600 FORMAT (1HO, ' END OF COMPUTATIONS'//  
1 ' TOTAL RAYS WERE STARTED FROM THE SELECTED QUAD'//  
2 ' TOTAL RAYS WERE TRACED TO COMPLETION')
601 FORMAT (1HO, ' REFLECTED RAYS WITH RADIANCE L.T. 1PE9.1,'  
1 ' WERE DISCARDED'// ' REFRACTED RAYS WITH RADIANCE L.T. 2PE9.1,'  
2 ' WERE DISCARDED'// ' THERE WERE '16,' 3 ' TOTAL INTERNAL REFLECTIONS')
602 FORMAT (1HO, ' RAYS STARTED DOWNWARD AND FINISHED UPWARD'//  
1 ' RAYS STARTED DOWNWARD AND FINISHED DOWNWARD'//  
2 ' RAYS STARTED UPWARD AND FINISHED DOWNWARD'//  
3 ' RAY STARTED UPWARD AND FINISHED UPWARD'//  
4 ' RAYS STARTED UPWARD AND FINISHED UPWARD WITH RAD = 1 5.6 (DISCARDED) ')
604 FORMAT (1HO, ' BRANCH OCCURRENCE TALLY'// ' NUM BRANCHES: '  
1 ' NUM OCCURRENCE: '9110)
1002 FORMAT (1HO, ' NORMAL EXIT FROM NNM1')
END
SUBROUTINE INISHL(IR,JS,NRAYQD,DSEED)
ON NHMI/INIIQD

THIS ROUTINE Initializes NHMI/M1QD

TWO INPUT RECORDS ARE READ:
RECORD 1 (DEFINED THE HEXAGON GRID AND THE WATER SURFACE):

IDBUG = 0 FOR MINIMAL OUTPUT
= 1 FOR GREATER OUTPUT
= 2 FOR FULL DEBUGGING OUTPUT

IGENSF = 0 IF A FILE OF RANDOM SURFACES ALREADY EXISTS (USUAL CASE)
.GT. 0 IF THIS IS A SPECIAL RUN FOR GENERATING AND SAVING A
FILE OF RANDOM SURFACES. IGENSF SURFACES WILL BE GENERATED.

NHEX = THE ORDER OF THE HEXAGONAL SURFACE GRID (= MXNHEX FOR EFFICIENCY)
WNDSPD = THE WIND SPEED IN M/SEC AT 12.5 M ELEVATION
DSEED = THE SEED FOR RANDOM NUMBER GENERATION

RECORD 2 (DEFINED THE QUAD GRID AND SELECTS THE INCOMING RAY QUAD):

NMU = THE NUMBER OF MU CELLS IN ONE HEMISPHERE (0 TO PI/2)
NPHI = THE NUMBER OF PHI CELLS (0 TO 2*PI). MUST BE A MULTIPLE OF 4
MUPART = 1 IF ALL QUADS ARE TO HAVE EQUAL SOLID ANGLES
2 IS ALL QUADS ARE TO HAVE EQUAL DELTA THETA VALUES
IR = THE INDEX OF THE INPUT MU QUAD (-NMU,...,-1,1,...,NMU)
JS = THE INDEX OF THE INPUT PHI QUAD (1,...,NPHI/4 + 1)
NUMRAY = THE NUMBER OF RAYS TO BE TRACED FROM THE INPUT QUAD

PARAMETER(MXMU=10, MXPHI=24)
PARAMETER(MXXHEX=7, MXXNODE=3*MXXHEX*(MXXHEX+1)+1)
COMMON/CMPH/MNNU(MXMU),MNPHI(MMphi)
COMMON/CNODI/MNODE,FNODE(7, MXXNODE),ZNODE(MXXNODE)
COMMON/CHEX/MNHEX,R1(2),R2(2),R1HAT(2),R2HAT(2),R1AT,TD/(TAR/2)
COMMON/CMSC/ IMISC(1), IMISC(2), IMISC(9) = IDBUG
COMMON/DIMETJ, FMU(MXMU), FMU(MXMU), PHIM1(MMphi), OMEGA(MMXMU)
DIMENSION NRAYQD(MXMU)

DOUBLE PRECISION DSEED

DATA P[.RAD,REFR,3.141592654, 57.2957795, 1.333333333/
DATA DELTA, EPF1, 1.111, TARGET/0.5, 0.370333333/
DATA NUSFC,NUDD,NUDD,NUDD,NUDD/15,16,17,18,19/

READ THE INPUT RECORDS
READ(5,*) IDBUG,IGENSF,NHEX,WNDSPD,DSEED
WRITE(6,300) NHEX,WNDSPD,DSEED
IF (IGENSF.EQ.0) THEN
READ(5,*) NMU, NPHI, MUPART, IR, JS, NUMRAY
WRITE(6,301) NMU, NPHI, IR, JS, NUMRAY
ENDIF

STORE THE NEEDED PARAMETERS
IMISC(1) = NMU
IMISC(2) = NPHI
IMISC(9) = IDBUG
IMISC(17) = NUMRAY
FMISC(11) = PI
FMISC(3) = RADI
FMISC(15) = WINDSP
FMISC(16) = DELTA
FMISC(17) = EPF
FMISC(18) = RE/R
RADI0 IS THE CRITICAL ANGLE FOR TOTAL INTERNAL REFLECTION
RAD0 = ASIN1.0/REFR
FMISC(19) = RADI0
IF (IGENSF.GT.0) THEN
****THIS IS AN INITIAL RUN FOR GENERATION OF A FILE OF RANDOM SURFACES
C

38
WRITE(6,304)
REWIND NUSFC
C
CHECK TO SEE IF NUSFC IS EMPTY
READ(NUSFC,END=50) DUMMY
STOP 'SURFACE FILE ALREADY EXISTS'
50 REWIND NUSFC
C
DEFINE THE GRID VECTORS AS IN 63/PAGES 24-26
GAMMA1 = 1.0/SQRT(0.25*DELTA*DELTA + EPS*EPS)
R1(1) = 0.5*DELTA*GAMMA1
R2(1) = EPS*GAMMA1
R2(2) = R1(2)
R1HAT(1) = -R1(1)
R1HAT(2) = R1(1)
R2HAT(1) = -R2(2)
R2HAT(2) = R2(1)
RIRAT = -2.0*EPS/DELTA
C
DEFINE THE HEXAGONAL SURFACE GRID NODE LOCATIONS
FMISC(16) = DELTA
FMISC(17) = EPS
CALL TRIADS(NHEX)
C
WRITE THE HEADER RECORDS
WRITE(NUSFC) 1GENSP,NHEX,NNODE,WNDSPD,DSEED
WRITE(NUSFC) R1,R2,R1HAT,R2HAT,RIRAT,FNODE
C
DEFINE THE STANDARD DEVIATION FOR SURFACE HEIGHTS BY 63/2.12
SIGSFC = 0.0397*SQR(WNDSPD)
WRITE(6,302) DELTA,EPS,SIGSFC
C
GENERATE AND SAVE THE CAPILLARY WAVE SURFACE REALIZATIONS.
63/SECTION 2C
DO 55 NSFC=1,1GENSF
C
DRAW N(0,1) RANDOM NUMBERS
CALL GGNML(DSEED,NNODE,ZNODE)
C
CONVERT TO N(0, SIGSFC**2) RANDOM NUMBERS
ZMAX = -1.0E30
ZMIN = 1.0E30
DO 99 IRAN=1,NNODE
ZN = SIGSFC*ZNODE(IRAN)
ZNODE(IRAN) = ZN
IF(ZN.GT.ZMAX) ZMAX = ZN
IF(ZN.LT.ZMIN) ZMIN = ZN
99 CONTINUE
C
55 WRITE(NUSFC) NSFC,ZMIN,ZMAX,I2*NODE(1),I=1,NNODE)
C
ENDFILE NUSFC
WRITE(6,60) 1GENSF
STOP
END

****THIS IS A PRODUCTION RUN FOR RAY TRACING
C
READ THE EXISTING FILE OF SURFACE REALIZATIONS AND TEST FOR
C COMPATIBILITY WITH REQUESTED PARAMETERS
C
WRITE(6,308)
REWIND NUSFC
READ(NUSFC) NSF1,NHEX1,NNODE,WIND1
READ(NUSFC) R1,R2,R1HAT,R2HAT,RIRAT,FNODE
C
IF(NHEX1.NE.NHEX .OR. WIND1.NE.WNDSPO) THEN
WRITE(6,70) NHEX1,WIND1
STOP
ENDIF
§2. PROGRAM 1

DEFINE THE MU AND PHI VALUES WHICH FORM THE QUAD BOUNDARIES FOR
GEOMETRIC DISCRETIZATION, SECTION 3.

IF(MUPART.EQ.1) THEN

PARTITION THE UNIT SPHERE SO THAT ALL QUADS, INCLUDING THE POLAR
CAP, HAVE EQUAL SOLID ANGLES

CALL EQSANG(NMU,NPHI,DELTMU)

ELSEIF(MUPART.EQ.2) THEN

PARTITION THE UNIT SPHERE INTO EQUALLY SPACED THETA VALUES

CALL EQTHET(NMU,DELTMU)

ENDIF

DEFINE THE BOUNDARY MU VALUES BY SUMMING THE DELTA MU VALUES

BNDMU(1) = DELTMU(1)
DO 101 I=2,NMU-1
101 BNDMU(I) = BNDMU(I-1) + DELTMU(I)
BNDMU(NMU) = 1.

DEFINE THE MU VALUES AT THE QUAD CENTERS

FMU(1) = 0.5*DELTMU(1)
DO 104 I=2,NMU-1
104 FMU(I) = 0.5*(BNDMU(I-1) + BNDMU(I))
FMU(NMU) = 0.5*DELTMU(NMU)

DEFINE THE PHI VALUES AT THE QUAD CENTERS, AND
DEFINE THE BOUNDARY PHIS BY PHI = PHID/2 TO PHI + PHID/2

DELPHI = 2.0*PI/FLOAT(NPHI)
PHI(1) = 0.
BNDPHI(1) = 0.5*DELPHI
DO 102 J=2,NPHI
102 PHI(J) = PHI(J-1) + DELPHI
BNDPHI(J) = BNDPHI(J-1) + DELPHI

DETERMINE THE SOLID ANGLE OF THE QUADS

DO 400 I=1,NMU
400 OMEGA(I) = DELPHI*DELTMU(I)
OMEGA(NMU) = 2.0*PI*DELTMU(NMU)
IA = IABS(IR)
NRAYQD(IA) = NUMRAY

WRITE(6,310)
WRITE(6,311) 1,J,FMU(I),THETAC,BNDMU(I),THETAB,DELTMU(I),
1 OMEGA(I),NRAYQD(I)
WRITE(6,316) DELPHI*RADEG

WRITE HEADER RECORDS FOR OUTPUT FILES

REWIND NUDD
REWIND NUDO
REWIND NUDD
REWRITE NUDD

NUDD,'DOWN UP','IR,JS,NRAYQD
NUDO,'IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
NUDD,'UP DOWN','IR,JS,NRAYQD
NUDO,'IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
NUDD,'UP DOWN','IR,JS,NRAYQD
NUDO,'IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU

RETURN

REWIND NUDD
REWIND NUDO
REWIND NUDD
REWRITE NUDD
§2. PROGRAM 1

C FORMATS

60 FORMAT(1HO,110,' SURFACE REALIZATIONS GENERATED')
70 FORMAT(1HO,' SURFACE REALIZATION FILE NOT COMPATABLE WITH REQUESTED PARAMETERS'
   1 HO ' MONT CARLO AIR-WATER SURFACE RAY TRACING PROGRAM'/
   2 HO ' THE HEXAGON GRID PARAMETERS FOR THIS RUN ARE'
   3 X ' NMEX = ',I3, ' = ORDER OF THE SURFACE GRID HEXAGON'
   4 X ' WINDSPD = ',F7.3, ' = THE WIND SPEED IN M/SEC AT 12.5 M'
   5 X ' GSEED = ',1PD20.10, ' = THE SEED FOR RANDOM NUMBER GENERATION'
301 FORMAT(1HO,' THE QUAD GRID PARAMETERS FOR THIS RUN ARE'
   15X ' NMU = ',I3, ' = NUMBER OF MU CELLS IN (0,PI/2)
   25X ' NPHI = ',I3, ' = NUMBER OF PHI CELLS IN (0,2*PI)
   45X ' NUMRAY = ',I10, ' = THE TOTAL NUMBER OF RAYS TO BE TRACED'
J02 FORMAT(1HO,' THE WAVE FACET PARAMETERS ARE'
   15X ' DELTA = ',1PE10.3//5X,' EPS = ',E10.3//5X,' SIGSFC = ',E10.3)
304 FORMAT(1HO,' THIS IS AN INITIAL RUN FOR GENERATING A FILE OF CAPILLARY WAVE REALIZATIONS'
308 FORMAT(1HO,' THIS IS A PRODUCTION RUN FOR RAY TRACING (1 QUAD)
310 FORMAT(1HO,' THE MU VALUES DEFINING THE QUADS ARE'
   15X ' I CNT MU THETA',8X,' BND MU THETA',7X,
   2* ' DELTA MU SOLID ANGLE NRAYQD'
314 FORMAT(1H,15,2(F9.4,F9.4X),F9.4,F12.4,I10)
316 FORMAT(1H,' THE QUADS HAVE A WIDTH OF DELTA PHI = ',F7.3,
   1 ' DEGREES'
END

41
§3. PROGRAM 2

3. PROGRAM 2

A. Program Description

This program tallies the ray information from Program 1 and computes the four quad-averaged geometric reflectance and transmittance arrays, using 75/9.1a-d and 75/9.7a-d. Once again, there is an "all-quad" and a "one-quad" version of Program 2, to be run with the ray-data files of the corresponding versions of Program 1.

Program 1 creates all four ray-data files (Tapes 16, 17, 18 and 19) in one run. Program 2 processes these files one at a time, in four separate runs, generating four separate output files.

After running Programs 1 and 2 and studying the resultant quad-averaged geometric r and t arrays, the user may decide that still more rays should be traced in order to increase the accuracy of the computed array elements. In this case, Program 1 can be run again to generate a new batch of rays. Program 2 can then read the new ray-data files from Program 1, read the output files from the previous run of Program 2, and merge the new and old information to create an updated set of r and t arrays. This repetition of Program 1 and 2 can be repeated until a satisfactory number of rays has been traced and the r and t array elements have been declared sufficiently accurate.

B. Input

Only one free-format data record is required:

Record 1: NEWRUN, IDBUG

```
NEWRUN = 1      if this is the first run of Program 2
               = 0      if Program 2 has already been run, and new ray data are to be merged with existing r and t files from the previous run of Program 2

IDBUG = 0, 1 or 2, as in record 1 of Program 1
```

C. File Management

File management for Program 2 depends on whether this is an initial run (NEWRUN = 1) or a continuation run to incorporate additional ray data (NEWRUN = 0). In either case, four separate runs must be made in order to process the four output ray-data files from Program 1. The file names are as follows:

Initial run (NEWRUN = 1)

There is one input file, always named TAPE20. This file is either of TAPE16, TAPE17, TAPE18 or TAPE19 from Program 1, locally renamed as TAPE20. There is one output file with symbolic filename of NUOUT. The external file name for NUOUT is
§3. PROGRAM 2

{TAPE22, TAPE23, TAPE24, TAPE25} if TAPE20 is {TAPE16, TAPE17, TAPE18, TAPE19}.

These external filenames are generated automatically by Program 2. The user should save NUOUT with an appropriate descriptive filename, to avoid confusion if more than one set of runs of Programs 1 and 2 is made.

Continuation run (NEWRUN = 0)

There are now two input files, always named TAPE20 and TAPE21. As above, TAPE20 is either of TAPE16,...,TAPE19 containing \textit{new} ray data from a second run of Program 1. TAPE21 is the corresponding output file from the \textit{previous} run of Program 2, i.e. TAPE21 is the renamed TAPE22,...,TAPE25 from the previous run. The output file, NUOUT, is corresponding TAPE22,...,TAPE25 and contains the updated r or t array. In other words,

if TAPE20 is the new \{TAPE16, TAPE17, TAPE18, TAPE19\}, then TAPE21 is \{TAPE22, TAPE23, TAPE24, TAPE25\} from the previous run,

and NUOUT is the updated \{TAPE22, TAPE23, TAPE24, TAPE25\}.

The most convenient manner for keeping track of these files, if multiple runs of Programs 1 and 2 are made, depends on the particular computer system.

The final versions of TAPE22,...,TAPE25 contain the quad-averaged geometric arrays as follows:

{TAPE22, TAPE23, TAPE24, TAPE25} contain \{r(a,x), t(a,x), r(x,a), t(x,a)\}.

All four of these files are read by Program 3. TAPE22 (r(a,x)) and TAPE25 (t(x,a)) also are read by Program 5, if the contrast transmittance is computed.
§3. PROGRAM 2

D. Code Listing

```
PROGRAM MAIN(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE20,
  TAPE21, TAPE22, TAPE23, TAPE24, TAPE25)

.........................
+ THIS IS PROGRAM 2 OF THE NATURAL HYDROSOL MODEL +
+ 

ON NHM2/M2ALL

THIS PROGRAM READS AN OUTPUT FILE WRITTEN BY NHM1/M1ALL AND
TALLEYS THE SCATTERED RAYS TO COMPUTE THE CORRESPONDING
GEOMETRIC REFLECTANCE OR TRANSMITTANCE ARRAY, AS DESCRIBED IN
SECTION 9.

THIS PROGRAM COMPUTES AND STORES THE "TOP HALF" OF RTGEO.
SEE SECTION 128 FOR THE BLOCK SYMMETRIES USED.

INPUT:
NEWRUN = 1, IF THIS RUN STARTS FROM SCRATCH
  0, IF THIS IS A CONTINUATION RUN

TAPE20 = A FILE OF RAY DATA WRITTEN BY NHM1/M1ALL AS
TAPE16, 17, 18, OR 19

TAPE21, IF NEWRUN = 0, THE FILE 22, 23, 24, OR 25 WRITTEN BY THE
PREVIOUS RUN OF NHM2/M2ALL. CONTAINING THE RTGEO ARRAY

OUTPUT:
NUOUT = TAPE22 IF TAPE20 IS A FILE OF NHM1
= TAPE23 IF TAPE20 IS TAPE17 OF NHM1, ETC.

PARAMETER (MXMU=10, MXPHI=24)
MXROW AND MXCOL ARE FOR THE TOP HALF OF RTGEO
PARAMETER (MXROW=MXMU*MXPHI/2, MXCOL=MXMU*MXPHI)
COMMON/CMUPHI/ FMU(MXMU), PHI(MXPHI), OMEGA(MXMU)
COMMON/CMISC/ IMISC(20), FMISC(120)
DIMENSION RTGEO(MXROW, MXCOL), NRAYQD(MXMU)
CHARACTER RTLABL*6

INITIALIZE

CALL INISHL(RTGEO, RTLABL, NUOUT, NEWRUN, NRAYQD)

NMU = IMISC(1)
NPHM = IMISC(2)
IDBUG = IMISC(9)
RADEG = FMISC(3)
NUMCOL = NMU*NPHM
NUMROW = NUMCOL/2
NREC = 0

READ AND ACCUMULATE RAY CONTRIBUTIONS. THIS IS THE SUM OVER OMEGA
IN 9.1 BUT WITHOUT THE 1/5 FACTOR. THE SUM OVER J IN 9.1
WAS DONE AUTOMATICALLY AS THE RAY WAS TRACED TO COMPLETION.
THE INPUT QUAD (I,J,S) IS (I,J); THE OUTPUT QUAD (U,V) IS (K,L)

IOLD = 0
IF(NEWRUN.EQ.1) WRITE(6,102)
NPNT = 0

200 READ(20,END=250) I,J,K,L,RAD
  NREC = NREC + 1
```

44
§3. PROGRAM 2

C ANY FINAL RAYS GOING INTO A POLAR CAP ARE STORED IN COLUMN NMU
IF(K.EQ.NMU) THEN
    JCOL = NMU
ELSE
    JCOL = K + (L-1)*NMU
ENDIF

C ANY INITIAL RAYS XI PRIME GOING TOWARD A POLAR CAP ARE STORED IN
C ROW NMU, COLUMNS 1, 2, ..., NUMCOL
IF(I.EQ.NMU) THEN
    IROW = NMU
ELSE
    IROW = I + (J-1)*NMU
ENDIF

C IF (NEWRUN.EQ.'YES', I.EQ.IOLD .AND. NPNT.LT.25) THEN
    IOLD = I
    NPNT = NPNT + 1
ENDIF

C RTGEO(IROW,JCOL) = RTGEO(IROW,JCOL) + RAD
GO TO 200

C 250 WRITE(6,110) NREC

C RTGEO IS NOW PROPORTIONAL TO THE RADIANT FLUX TRANSFER FUNCTION
C CONVERT THE RAY-TALLY ARRAY INTO A GEOMETRIC R OR T ARRAY BY 9.7
C (INPUT RAYS XI PRIME ARE IN THE FIRST QUADRANT ONLY)

C JPI2 = NPHI/4 + 1
DO 252 JS=1,JPI2
    MAXIR = NMU - JS
    IF(JS.EQ.1) MAXIR = NMU
    DO 252 IR=1,MAXIR
        IROW = IR + (JS - 1)*NMU
    ENDIF
    C NRAYQD(IR) IS 5 OF 9.1
    FACT1 = FMU(IR)*OMEGA(IR)/FLUAI(NUSTY(IR))
    C NUN POLAR QUADS
    DO 253 KU=JPI2,NPHI-1
    C FACT2 CONTAINS THE MU AND OMEGA FACTORS OF 9.7, AND 1/S OF 9.1
    FACT2 = FACT1*FMU(KU)*UMEA(KU)
    DO 253 LV=1,NPHI
        JCOL = KU + (LV-1)*NMU
        RTGEO(IROW,JCOL) = FACT2*RTGEO(IROW,JCOL)
    ENDIF
    C POLAR CAPS: KU = NMU
    252 RTGEO(IROW,NMU) = FACT1*RTGEO(IROW,NMU)/UMEA(NMU)

C RTGEO IS NOW THE QUAD-AVERAGED GEOMETRIC R OR T ARRAY

C FILL OUT THE REMAINING ROWS (THE SECOND QUADRANT) OF THE "TOP HALF"
C OF RTGEO BY SYMMETRY (SEE PAGE 130).
C (IP,IP) ARE THE 'ROW,COL' BLOCK INDICES OF THE KNOWN BLOCK
C (IBP,IB) ARE THE BLOCK INDICES OF THE BLOCK TO BE DEFINED

C N34 = (NPHI + 3)/4
Nopi = NPHI/2
DO 300 IP=2,N34
    IBP = Nopi + 2 - IP
    IRTP = NMU*(IP-1)
    IRTBP = NMU*(IP-1)
    DO 300 KP=1,N34
        IB = Nopi + 2 - 1
        IF(IP.LE.0) IB = IB + NPHI
        IRT = NMU*(I-1)
        IRTB = NMU*(I-1)
    ENDIF
    DO 300 RTGEO(IPTP+KP,IRT+K) = RTGEO(IPTP+KP,IRT+K)
300 CONTINUE

45
§3. PROGRAM 2

C

RESET THE INPUT SECOND QUADRANT, OUTPUT POLAR CAP DIRECTION
COLUMN NMU), WHICH HAS PICKED UP ZERO VALUES FROM THE INPUT
FIRST QUADRANT, PHI = 180 BLOCKS

DO 310 IP=2,N34
  IBP = NOPI + 2 - IP
  IRTBP = NMU*(IP - 1)
  IRTP = NMU*(IP - 1)
DO 310 K=1,NMU
310 RTGEO(IRTP,K,NMU) = RTGEO(IRBP,K,NMU)

C

RE-ZERO THE INPUT SECOND QUADRANT, OUTPUT PHI = 180 COLUMN, WHICH
HAS PICKED UP NON-ZERO VALUES FROM THE INPUT FIRST QUADRANT.
C
OUTPUT POLAR CAP (PHI = 0) COLUMN

JCOL = NMU*(NUPI + 1)
DO 312 I=1,NUMROW
312 RTGEO(I,JCOL) = RTGEO(I,NUMROW)

C

WRITE THE FINAL ARRAY TO THE OUTPUT FILE. ONLY THE "TOP HALF" IS
STORED (SEE PAGE 190).

DO 270 JCOL=1,NUMCOL
270 WRITE(NUOUT) (RTGEO(IR,JCOL),IR=1,NUMROW)

C

PRINT SELECTED PARTS OF THE NEW RTGEO
C
THE SPECULAR BLOCK FOR PHI PRIME = 0
IS = 1
WRITE(6,113)IS,RADEG*PHI(IS),IS,RADEG*PHI(IS),RTLABL,(J,J=1,NMU)
DO 114 I=1,NMU
  THET = RADEG*ACOS(FMU(I))
114 WRITE(6,115) I,THET,(RTGEO(I,J)*J1,NMUJ)

C

THE SPECULAR BLOCK FOR PHI PRIME = 90
IS = NPHI/4 + 1
IR1 = NMU*(IS - 1) + 1
IR2 = IR1 + NMU - 1
WRITE(6,115) I,IR1,IR2
DO 116 I=IR1,IR2
  IR = MOD(I,NMU)
  IF(IR.EQ.0) IR = NMU
  THET = RADEG*ACOS(FMU(IR))
116 WRITE(6,115) I,IR,THET,(RTGEO(I,J),J=1,IR1,IR2)

C

IF(DBG.EQ.0) CALL P2ARRAY(RTGEO,2*NMU,2*NMU,NUROW,2.
1' THE UPPER LEFT BLOCKS OF THE NEW RTGEO ARRAY')
IF(DBG.EQ.1) CALL P2ARRAY(RTGEO,NUROW,NUMCOL,NUMROW,2.
1' THE TOP HALF OF THE FULL RTGEO ARRAY')
WRITE(6,605)

C

FORMATS
C

102 FORMAT(1HO,' SELECTED RAY DATA: ')
104 FORMAT(1H1,' NREC IR JS KU: ')
106 FORMAT(1H1,' ROW COL: ')
108 FORMAT(1H1,' ROW COL ')
110 FORMAT(1H1,' DATA RECORDS READ FROM UNIT 20:
112 FORMAT(1H0,' (SPECULAR) BLOCK FOR PHI PRIME= ')
113 FORMAT(1H0,' OF THE NEW 'AB' ARRAY: ')
115 FORMAT(1H0,' ROW: ')
271 FORMAT(1H0,' EOF WRITTEN ON FILE NUOUT = TAPE',12)
605 FORMAT(1H0,' NORMAL EXIT FROM NMIN, PROGRAM II: ')
SUBROUTINE INISHL(RTGEO,RTLABL,NUOUT,NEWRUN,NRAYQD)

ON NHM2/IN2ALL

THIS ROUTINE Initializes PROGRAM NH-M2/M2ALL

PARAMETER (MXMU=10, MXPHI=24)
PARAMETER(MXROW=MXMU*MXPHI/2, MXCOL=MXMU*MXPHI)
DIMENSION RTGEO(MXROW,MXCOL)
DIMENSION NRAYQD(MXMU),BNDMU(MXMU),BNDPHI(MXPHI),DELMU(MXMU)
DIMENSION IMISC2(20),FMISC2(20),NRAVQD(20)
CHARACTER UPDOWN*9,RTLABL*6,UPDN2*9,RTLAB2*6
READ(5.*) NEWRUN, IDEBUG
READ(20) NU2O,UPDOWN,NRAYQD
READ(20) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELMU
NMU = IMISC(1)
NPHI = IMISC(2)
NUMRAV = IMISC(17)
RADEG = FMISC(3)
WNDSPD = FMISC(15)
REFR = FMISC(18)
IMISC(9) = IDEBUG
NUOUT = NU20 + 6
1P12 = NPHI/4 + 1
IF(UPDOWN.EQ.'DOWN DOWN') THEN
   RTLABL = 'T(A,X)
ELSEIF(UPDOWN.EQ.'UP UP') THEN
   RTLABL = 'T(A,X)
ELSEIF(UPDOWN.EQ.'DOWN UP') THEN
   RTLABL = 'R(A,X)
ELSEIF(UPDOWN.EQ.'UP DOWN') THEN
   RTLABL = 'R(A,X'
ELSE
   WRITE(6,118) UPDOWN
STOP
ENDIF
NUMCOL = NMU*NPHI
NUMROW = NUMCOL/2
WRITE(6,100) RTLABL,UPDOWN,NU2O,NPHI,WNDSPD,REFR
WRITE(6,110) NMU,NRAYQD

IF(NEWRUN.EQ.1) THEN
  THIS IS A NEW RUN, ZERO RTGEO
  ZERO ONLY THOSE ARRAY ELEMENTS WHICH ARE ACTUALLY USED FOR STORAGE,
  AS AN AID TO DEBUGGING ON THE B55
DO 98 JCOL=1,NUMCOL
C 98 (RTGEO(IROW,JCOL) = 0.
C NON POLAR OUTPUT QUADS
DO 99 (J=1,NPHI)
C 99 (RTGEO(IROW,JCOL) = 0.
C NON POLAR INPUT QUADS
DO 99 (J=1,NPHI)
C 99 (RTGEO(IROW,JCOL) = 0.

47
§3. PROGRAM 2

L POLAR CAP INPUT QUAD
98 RTGEO(NMU,JCOL) = 0.
C NON POLAR INPUT QUADS, POLAR CAP OUTPUT QUAD
DO 97 IS=1,IP12
DO 97 IR=1,NMU-1
IROW = IR + (IS-1)*NMU
97 RTGEO(IROW,NMU) = 0.
C POLE TO POLE QUADS
RTGEO(NMU,NMU) = 0.
C ELSE
C THIS IS A CONTINUATION RUN, READ EXISTING RTGEO (TAPE21 = NUOUT OF PREVIOUS RUN)
REWIND 21
READ(21) NU21, NRAQO2, IMISC2, FMISC2
C
NMU2 = IMISC2(1)
NPHI2 = IMISC2(2)
NUMRA2 = IMISC2(17)
WNDSP2 = FMISC2(15)
C
CHECK FOR COMPATIBLE FILES
IF (NU21 .NE. NU21 .OR. NMU .NE. NMU2 .OR.
NPHI .NE. NPHI2 .OR. WNDSPD .NE. WNDSP2) THEN
WRITE(6,200)
WRITE(6,202) NU21, NMU, NPHI, WNDSPD
WRITE(6,202) NU21, NMU2, NPHI2, WNDSP2
STOP
ENDIF
C
THE "TOP HALF" OF RTGEO IS STORED
DO 130 JCOL=1,NUMCOL
130 READ(21) (RTGEO(I,JCOL),I=1,NMU)
WRITE(6,112) RTLABL, NUMRA2
C
PRINT SELECTED PARTS OF THE EXISTING RTGEO
C
THE SPECULAR BLOCK FOR PH1 PRIME = 0
IS = 1
IV = IS + NPHI/2
JC1 = NMU*(IV - 1) + 1
JC2 = JC1 + NMU - 1
WRITE(6,113)IS,RADEG*PHI(IS),IV,RADEG*PHI(IV),RTLABL,(J,J=JC1,JC2)
DO 114 I=1,NMU
THET = RADEG*(RAD*(FMU(I))
114 WRITE(6,115) I,THET,(RTGEO(I,J),J=JC1,JC2)
C
THE SPECULAR BLOCK FOR PH1 PRIME = 90
IS = NPHI/4 + 1
IV = IS + NPHI/2
IR1 = NMU*(IV - 1) + 1
IR2 = IR1 + NMU - 1
JC1 = NMU*(IV - 1) + 1
JC2 = JC1 + NMU - 1
WRITE(6,113)IS,RADEG*PHI(IS),IV,RADEG*PHI(IV),RTLABL,(J,J=JC1,JC2)
DO 116 I=1,IR1,IR2
IR = MOD(I,NMU)
IF (IR.EQ.0) IR = NMU
THET = RADEG*(RAD*(FMU(IR))
116 WRITE(6,115) I,IR,THET,(RTGEO(I,J),J=JC1,JC2)
1F(156.EQ.0) CALL P2RAY(RTGEO,NMU,NUMCOL,MXROW,2,
1'THE PH1 PRIME = 0 BLOCKS OF THE EXISTING RTGEO ARRAY')
C
CONVERT THE GEOMETRIC R OR T ARRAY BACK INTO A RAY-TALLY ARRAY,
C I.E. UNDO 9.7
§3. PROGRAM 2

DO 120 IROW=1, NUMROW
  IR = MOD(IROW, NMU)
  IF(IR.EQ.0) IR = NMU
  F1 = FLOAT(NRAQD2(IR)) / (FMU(IR) * OMEGA(IR))
C Non-polar quads
  DO 121 KU=1, NMU
  F2 = F1 * FMU(KU) * OMEGA(KU)
  DO 121 LV=1, NPHI
    JCOL = KU - (LV-1) * NMU
    RTGEO(IROW, JCOL) = F2 * RTGEO(IROW, JCOL)
C Polar caps: KU = NMU
  RTGEO(IROW, NMU) = F1 * OMEGA(NMU) * RTGEO(IROW, NMU)
  NUMRAY = NUMRAY + NUMRA2
  IMISC(17) = NUMRAY
  DO 122 I=1, NMU
  NRAYQD(I) = NRAYQD(I) + NRAQD2(I)
END
C Write header on output file
C
  WRITE(NOUT) NUMOUT, NRAYQD, IMISC, FMU, PHIR, BNDMU, BNDPHI, OMEGA,
  DELTMU
C RETURN
C
100 FORMAT(IHO, ' NATURAL HYDROSOIL MODEL, PROGRAM 2''/
  ' RAY TALLY AND COMPUTATION OF ''/A6.' FROM ''/A9.'/
  ' NMU = ''/I3.' NPHI = ''/I3.' WNDSPD = ''/F8.3,
  ' M/SEC''/I6.' ' REF = ''/F7.4)
110 FORMAT(IHO, ' FOR THE CURRENT RUN, NUMRAY = ''/I10,
  ' TOTAL RAYS TRACED''/
112 FORMAT(IHO, ' THE RAISING GEOMETRY ''/A6.'/
  ' ARRAY WAS ACCUMULATED FROM ''/I10.' RAYS'')
113 FORMAT(IHO, ' (SPECULAR) BLOCK FOR PHI PRIME(''/I2,'') = ''/F6.1,
  ' AND PHI(''/I2,'') = ''/F6.' OF THE EXISTING ''/A6.' ARRAY''/
  ' column: ''/I10 I15)
  MU(1) MU(2) MU(3) MU(4) MU(5) MU(6)
  M(1) M(2) M(3) M(4) M(5) M(6)
  M(7) M(8) M(9) M(10)
115 FORMAT(' ROW: ''/I4.' MUL: ''/I2.' ''/I12.' ''/F6.1,4X,10(2X',F8.5),(/24X,
  ' 1 10(X2,FB 51))
118 FORMAT(IHO, ' UPON ''/A4.' ERROR STOP')
200 FORMAT(IHO, ' FILES 20 AND 21 INCOMPATIBLE''/
  ' NUM, NPHI, WNDSPD =''/I1M,12X,21A1,F1..3)
END

49
§3. PROGRAM 2

PROGRAM MAIN(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE20.
1 TAPE21, TAPE22, TAPE23, TAPE24, TAPE25)

**************************************************************************
+ THIS IS PROGRAM 2 OF THE NATURAL HYDROSOL MODEL *
+**************************************************************************

ON NHM2/M21QD

THIS PROGRAM READS AN OUTPUT FILE WRITTEN BY NHM1/M11QD AND
TALLEYS THE SCATTERED RAYS TO COMPUTE THE CORRESPONDING ROW
OF THE GEOMETRIC REFLECTANCE OR TRANSMITTANCE ARRAY, AS
DESCRIBED IN SECTION 9.

THIS SPECIAL VERSION OF MAIN2 DOES ONLY ONE INPUT QUAD (ONE ROW
OF R OR T).

INPUT:
NEWRUN = 1, IF THIS RUN STARTS FROM SCRATCH
0, IF THIS IS A CONTINUATION RUN

TAPE20 = A FILE OF RAY DATA WRITTEN BY NHM1/M11QD AS
TAPE16, 17, 18, OR 19

TAPE21, IF NEWRUN = 0, THE FILE 22, 23, 24, OR 25 WRITTEN BY THE
PREVIOUS RUN OF NHM2/M21QD, CONTAINING THE RTGEO ARRAY

OUTPUT:
NUOUT = THE FILE WITH THE COMPUTED RTGEO ARRAY

NUOUT = TAPE22 IF TAPE20 IS TAPE16 OF NHM1/M11QD
= TAPE17 IF TAPE20 IS TAPE17, ETC.

PARAMETER (MXMU = 13, MXPHI = 24, MXCOL = MXMU*MXPHI)
COMMON/CMUPHI/ FMU(MXMU), PHI(MXPHI), OMEGA(MXMU)
COMMON/CMISC/ IMISC(20), FMISC(26)
DIMENSION RTGEO(MXCOL), KNTRAY(MXCOL)
CHARACTER RTLAB_.6

INITIALIZE

CALL INISHL(RTGE0, KNTRAY, IMPH, RTLAB, NUOUT)
NNU = IMISC(1)
NPHI = IMISC(2)
NUMHAY = IMISC(17)
RADEG = FMISC(3)
NUMCOL = NNU*NPHI
NREC = 0

READ AND ACCUMULATE RAY CONTRIBUTIONS - THIS IS THE SUM OVER OMEGA
IN 9.1, BUT WITHOUT THE 1/2 FACTOR. THE SUM OVER J IN 9.1 WAS
DONE AUTOMATICALLY AS THE RAY WAS TRACED TO COMPLETION.

WRITE(6,102)
200 READ(20, END=250) I,J,K,L, P,AD
NREC = NREC + 1
ANY RAYS GOING INTO A POLAR CAP ARE STORED IN COLUMN NNU
L = L + 1
IF (K.EQ.NNU) L = 1
JCOL = K + (L - 1)*NNU
IF (NREC.LE.25) WRITE(16,104) NREC, I, J, K, L, P, AD, JCOL
KNTRAY(JCOL) = KNTRAY(JCOL) + 1
RTGE0(JCOL) = RTGE0(JCOL) + P, AD
GO TO 200

250 WRITE(16,110) NREC

RTGE0 IS NOW PROPORTIONAL TO THE RADIANT FUX TRANSFER FUNCTION
CONVERT THE RAY-TALLY ARRAY INTO A GEOMETRIC R OR T ARRAY BY 9.7

50
§3. PROGRAM 2

IR = MOD(IROW,NMU)
IF(IR .EQ. 0) IR = NMU
NUMRAY IS S OF 9.1
FACT1 = FMU(IR)*OMEGA(IR)/FLOAT(NUMRAY)
DO 252 KU=1,NMU-1
JCOL = KU + (LV-1)*NMU
RTGEO(JCOL) = FACT2*RTGEO(JCOL)
C
C POLAR CAP: KU = NMU
RTGEO(NMU) = FACT1*RTGEO(NMU)/OMEGA(NMU)
C
RTGEO IS NOW THE QUAD-AVERAGED GEOMETRIC R OR T ARRAY
C
C PRINTOUT OF SELECTED COLUMNS NEAR THE SPECULAR DIRECTION
C
WRITE(6,112) IROW,RTLABL,NUMRAY
IV = 1
WRITE(6,262) IV,RADEG*PHI(IV),(RTGEO(JCOL),JCOL=1,NMU)
IS = (IROW - 1)/NMU + 1
IV = IS + NPHI/2
IV1 = MAX0(2,IV-3)
IV2 = MIN0(NPHI,IV+3)
DO 260 IV=IV1,IV2
JC1 = 1 + (IV-1)*NMU
JC2 = IV*NMU
260 WRITE(6,262) IV,RADEG*PHI(IV),(RTGEO(JCOL),JCOL=JC1,JC2)
C
C PRINT COUNTS OF RAYS CONNECTING THE QUADS
WRITE(6,112) IROW,*KNTRAY*,NUMRAY
IV = 1
WRITE(6,272) IV,RADEG*PHI(IV),(KNTRAY(JCOL),JCOL=1,NMU)
IS = (IROW - 1)/NMU + 1
IV = IS + NPHI/2
IV1 = MAX0(2,IV-3)
IV2 = MIN0(NPHI,IV+3)
DO 270 IV=IV1,IV2
JC1 = 1 + (IV-1)*NMU
JC2 = IV*NMU
270 WRITE(6,272) IV,RADEG*PHI(IV),(KNTRAY(JCOL),JCOL=JC1,JC2)
C
C COMPUTE SUM OVER U,V FOR IRRAD CHECK
C
SUM = RTGEO(NMU)*OMEGA(NMU)
DO 300 IU=1,NPHI
DO 300 IU=1,NMU-1
300 SUM = SUM + RTGEO(IU) + (IV-1)*NMU)*FMU(IU)*OMEGA(IU)
SUM = SUM/(FMU(IR)*OMEGA(IR))
WRITE(6,302) SUM
302 FORMAT(/' SUM(U,V) OF R1*MU(U)*OMEGA(U))/(MU(R)*OMEGA(R)) = ',F8.6)
C
C WRITE FINAL ARRAYS TO OUTPUT FILE
C
WRITE(NOUT), (RTGEO(JCOL),JCOL=1,NMU)
WRITE(NOUT), (KNTRAY(JCOL),JCOL=1,NMU)
ENDFILE NOUT
C
C FORMATS
C
102 FORMAT(' HO: SELECTED RAY DATA: ',NREC IR JS KU',
1' LV FRESNEL RT COL'/)
104 FORMAT(' I10.15,F10.5,18)
118 FORMAT(' H0.1B DATA RECORDS READ FROM UNIT 20'
122 FORMAT(' HO: SELECTED COLUMNS OF ROW ',I4,' OF THE ',AG. 
1' ARRAY: ',ACCUMULATED FROM I8. INITIAL RAYS: //
325X. MU(1),MU(2),MU(3),MU(4),MU(5),MU(6) M
4C(7) MU(9) MU(10),//
262 FORMAT(' PHIL ',I4,' = ',F6.1,4X,10(2X,F8.5)),(/24X,10(2X,F8.5)))
272 FORMAT(' PHIL ',I4,' = ',F6.1,4X,10(2X,F8.5)),(/24X,10(2X,F8.5)))
END
§3. PROGRAM 2

SUBROUTINE INISHL(RTGEO,KNTRAY,IROW,RTLABL,NUOUT)

ON NHM2/M21QD

THIS ROUTINE INITIALIZES PROGRAM NHM2/M21QD

PARAMETER(MXMU=10, MXPHI=24)
COMMON/CMPHI/ FMU(MXMU), PHI(MXPHI), OMEGA(MXMU)
COMMON/CMISC/ IMISC(20), FMISC(20)
DIMENSION RTGEO(I), KNTRAY(1)
DIMENSION BNDMU(MXMU), BNDPHI(MXPHI), DELMU(MXMU), NRAYQ0(MXMU)
DIMENSION IMISC2(20), FMISC2(20), NRAYQ2(MXMU)
CHARACTER UPDOWN*9, RTLABL*6, UPDN2*9, RTLAB2*6

READ(5, *) NEWRUN

READ HEADER RECORDS OF RAY DATA FILE
REWIND 20
READ(20) NU20, UPDOWN, IR, JS, NRAYQ0
READ(20) IMISC, FMISC, FMU, PHI, BNDMU, BNDPHI, OMEGA, DELMU

NMU = IMISC(1)
NPHI = IMISC(2)
NUMRAY = IMISC(17)
WNDSPD = FMISC(15)
REFR = FMISC(16)
NUOUT = NU20 - 6
IA = IABS(IR)
IF (IA.EQ.NMU) JS = 1
IROW = IA + (JS-1)*NMU

Determine the type of array being processed

IF (UPDOWN.EQ. 'DOW NDOWN') THEN
RTLABL = 'S(I,A)' ELSE IF (UPDOWN.EQ. 'UP UP' ) THEN
RTLABL = 'S(X,A)' ELSE IF (UPDOWN.EQ. 'DOW NDUP') THEN
RTLABL = 'S(A,X)' ELSE IF (UPDOWN.EQ. 'UP DOW ND') THEN
RTLABL = 'S(A,X)' ELSE
WRITE(6, 118) UPDOWN
STOP
ENDIF

NUMCOL = NMU*NPHI
WRITE(6, 100) IROW, RTLABL, UPDOWN, NMU, NPHI, WNDSPD, REFR, IR, JS
NRAQDT = NRAYQ0(IA)
WRITE(6, 110) NRAQDT

IF (NEWRUN.EQ.1) THEN

This is a new run, zero RTGEO and KNTRAY

Only those elements actually used for storage are set to zero.

As an aid to debugging
DO 98 J=1, NPHI
DO 98 I=1, NMU
JCOL = I + (J-1)*NMU
KNTRAY(JCOL) = 0
RTGEO(JCOL) = 0.
END DO

Polar (AP)
KNTRAY(NMU) = I
RTGEO(NMU) = 0.

ELSE

This is a continuation run, read existing RTGEO
REWIND 21
READ(21) NU20, UPDN2, RTDN2, IRZ, JS2, NRAQD2
READ(21) IMISC2, FMISC2
NMU2 = IMISC2(1)
NPHI2 = IMISC2(2)
NUMRAY2 = IMISC2(17)
WNDSP2 = FMISC2(15)

52
§3. PROGRAM 2

C

CHECK FOR COMPATIBLE FILES
IF(UPDOWN.NE.UPDN2 .OR. RTLABL.NE.RTIAB2 .OR. IR.NE.IR2 .OR.
1 JS.NE.JS2 .OR. NMU.NE.NMU2 .OR. NPHI.NE.NPHI2 .OR.
2 WNDSPD.NE.WNDSP2 .OR. NUOUT.NE.NUOUT2) THEN
WRITE(6,200)
FILE = 20
WRITE(6,202) IFILE,UPDOWN,IR,JS,NMU,NPHI,WNDSPD,NUOUT
FILE = 21
WRITE(6,202) IFILE,UPDN2,IR2,JS2,NMU2,NPHI2,WNDSPD2,NUOUT2
STOP
ENDIF

C

READ(21) (RTGEO(JCOL),JCOL=1,NUMCOL)
READ(21) (KNTRAY(JCOL),JCOL=1,NUMCOL)
WRITE(6,112) IROW,RTLABL,NUMRA2
DO 114 JCOL=1,NUMCOL,10
114 WRITE(6,116) JCOL,JCOL+9,(TGEO(JCOL),M=0.9)
WRITE(6,112) IROW,'KNTRAY.',NUMRA2
DO 113 JCOL=1,NUMCOL,10
113 WRITE(6,119) JCOL,JCOL+9,(KNTRAY(JCOL+M),M=0.9)
C

CONVERT THE GEOMETRIC P OR T ARRAY BACK INTO A RAY-TALLY ARRAY,
I.E. UNDO 9.7
C

F1 = FLOT(NUMRA) .(FMU(IAJ)*UMGUA(IAJ))
DO 120 KU=1,NMU-1
F2 = F1*FMU(KU)*UMGUA(KU)
DO 120 LV=1,NPHI
JCOL = KU + (LV-1)*NUM
RTGEO(JCOL) = F2*RTGEO(JCOL)
C
POLAR CAP: KU - NMU
RTGEO(NMU) = F1*UMGUA(NMU)*RTGEO(NMU)
NRAQDT = NRAQDT + NUMRA2
IMISC(17) = NRAQDT
ENDIF
C

WRITE HEADER ON OUTPUT FILE
C

REWIND NUOUT
WRITE(NUOUT) NU0,J,UPDOWN,RTLABL,IR,JS,NRAQDT
WRITE(NUOUT) IMISC,FMISC,FMU,PHI,BDMU,BNPHI,OMEGU,DELMU
C

RETURN
C

FORMATS
C

100 FORMAT(1H1, NATURAL HYDROSOL MODEL, PROGRAM 2 (1-QUAD VERSION)//
1 RAY TALLY FOR COMPUTATION OF //
1 NN = .13, //
2 NPHI = .13. //
3 THE FIXED INPUT QUAD Q(R,S) WAS (.12, .12, .12.) )
110 FORMAT(IHO, FOR THE CURRENT RUN, NRAQDT = .16)
112 FORMAT(IHO, ROW I4, OF THE EXISTING A6, ARRAY //
116 FORMAT(IHO, COLUMN I4, TO I4, OF (2X,FB.5))
119 FORMAT(IHO, COLUMN I4, TO I4, OF (2X,FB.5))
18 FORMAT(IHO, UPDOWN = .A9. ERROR STOP)//
200 FORMAT(IHO, FILES 23 AND 21 INCOMPATIBLE)//
202 FORMAT(IHO, FILL I4, UPDOWN, IR, J, NMU, NPHI, WNDSPD, NU OUT = .1//
35
53
4. PROGRAM 3

A. Program Description

This program reads the four quad-averaged geometric reflectance and transmittance arrays computed by Program 2 (\(r(a,x)\) on TAPE22, etc.). The corresponding spectral arrays \(\hat{r}_1(a,x)\), \(\hat{r}_2(a,x)\), etc. are computed using 75/5.31c, 75/5.32, 75/5.34, and 75/5.36. All arrays are processed in one run of Program 3.

Recall that Programs 1, 2 and 3 are concerned only with the air-water surface boundary conditions. We have so far specified only the quad partitioning and the wind speed. The surface boundary condition computations are thus completely independent of the inherent optical properties of the water body, of the incident lighting, etc. (all to be specified in Program 4). The output from Program 3 can therefore be run with many different versions of Program 4, i.e. with many different water bodies. Only a few runs of Programs 1-3 are necessary (say at two or three different wind speeds) in order to study a wide range of ocean optics problems in which the water type, bottom boundary condition, or incident lighting are varied.

B. Input

Only one user-supplied input record is required:

Record 1: IDEBUG
where IDEBUG = 0, 1, or 2 as in Program 1.

C. File Management

Program 3 reads the four output files from Program 2 and creates one output file, as follows:

<table>
<thead>
<tr>
<th>symbolic name</th>
<th>external name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NURAX</td>
<td>TAPE22</td>
<td>the quad-averaged geometric (r(a,x)) array</td>
</tr>
<tr>
<td>NUTAX</td>
<td>TAPE23</td>
<td>the (\hat{r}(a,x)) array</td>
</tr>
<tr>
<td>NURXA</td>
<td>TAPE24</td>
<td>the (\hat{r}(x,a)) array</td>
</tr>
<tr>
<td>NUTXA</td>
<td>TAPE25</td>
<td>the (\hat{r}(x,a)) array</td>
</tr>
<tr>
<td>NUOUT</td>
<td>TAPE30</td>
<td>the four spectral (\hat{r}) and (\hat{t}) arrays, written in the order in which they are needed in Program 4, namely (\hat{t}(a,x)), (\hat{t}(x,a)), (\hat{r}(x,a)), and (\hat{r}(a,x))</td>
</tr>
</tbody>
</table>

TAPE30 contains all of the surface boundary condition information needed by Program 4.
§4. PROGRAM 3

D. Code Listing

PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT, 1 TAPE22,TAPE23,TAPE24,TAPE25,TAPE30)

******************************************************************************
+ THIS IS PROGRAM 3 OF THE NATURAL HYDROSOL MODEL +
******************************************************************************

ON NhM3/MAIN3

THIS PROGRAM COMPUTES THE UPPER BOUNDARY SPECTRAL REFLECTANCE AND
TRANSMITTANCE ARRAYS WHICH DESCRIBE THE AIR-WATER INTERFACE.
THE GOVERNING EQUATIONS ARE 5.31C TO 5.36.

THE ARRAYS ARE COMPUTED IN THE ORDER IN WHICH THE SPECTRAL ARRAYS
ARE NEEDED BY PROGRAM 4, NAMELY
THAT(A,X), RHAT(X,A), THAT(X,A), RHAT(A,X)

THE GEOMETRIC ARRAYS ARE READ FROM THE OUTPUT FILES WHICH
WERE WRITTEN BY PROGRAM 2 (TAPES 22, 23, 24, AND 25)

THE SPECTRAL ARRAYS ARE WRITTEN TO NUOUT (TAPE30)

RTHAT1 AND RTHAT2 ARE EACH NMU*(NL+1) BY NMU*INT((NL+2)/2) WORDS
THE STORED RT ARRAY IS NMU*NPHI/2 BY NMU*NPHI WORDS (THE TOP HALF)

PARAMETER(MXMU=10, MXP1=24)
PARAMETER(ID1HAT=MXMU*(MXL+1), ID2HAT=MXMU*((MNL+2)/2))
DIMENSION RT1(ID1RT,ID1RT)
DIMENSION RTHAT1(ID1HAT,ID2HAT), RTHAT2(ID1HAT,ID2HAT)
COMMON/CPHI/ PHI(MXPHI)
COMMON/CMISC/ IMISC(20), FMISC(14)
DATA NURAX,NUTAX,NURAX/NUTAX/NUTAX/Z,

INITIALIZE THE PROGRAM

CALL INISHL

NMU = IMISC(1)
NPHI = IMISC(2)
IDBUG = IMISC(9)

NROWRT = NMU*NPHI/2
NCOLRT = NMU*NPHI;
NRHAT = IMISC(15)
NC1HAT = IMISC(11)
IFPR = MINTO(20,NROWRT)
IFCR = MINTO(20,NCOLRT)
IFRHAT = MINTO(4,NRHAT)
IFCHAT = MINTO(20,NC1HAT)
IF(IDBUG.EQ.2) GO TO 888

C-----DOWNWARD TRANSMITTANCE T(A,X)
RT CONTAINS THE GEOMETRIC THAT1(A,X;R,S;L,U,V)
RTHAT1 CONTAINS THE SPECTRAL THAT1(A,X;R,L/U,X)
RTHAT2 CONTAINS THE SPECTRAL THAT2(A,X;R,L/U,X)

READ THE GEOMETRIC T(A,X)

READ(NUTAX) NUNIT
IF(NUNIT.EQ.NUTAX) THEN
WRITE(1,702) THAT1(A,X), THAT2(A,X), TH1A,X
ELSE
WRITE(17,702) NUNIT,'NUTAX',NUTAX
STOP
ENDIF

55
§4. PROGRAM 3

DO 710 J=1,NCOLRT
710 READ(NUTAX) (RT(I,J),I=1,NCOLRT)
C
C DEFINE A FULL RT ARRAY TO AVOID SPECIAL INDEXING IN RTSPEC
C
CALL FULLRT(RT,NMU,NPHI,IDRT)
C
CALL P2ARRAY(RT,IPRT,IPCRT,IDRT,2,'GEOMETRIC R(X,R,S/U,V)')
C
C COMPUTE THAT(A,X)
CALL RTSPEC(PHI,RT,IDRT, RTHAT1,RTHAT2,IDHAT)
C
C WRITE THE SPECTRAL ARRAYS TO FILE NUOUT
DO 221 J=1,NCHAT
221 WRITE(NUOUT) (RTHAT1(I,J),I=1,NRHAT)
DO 222 J=1,NCHAT
222 WRITE(NUOUT) (RTHAT2(I,J),I=1,NRHAT)
C
C COMPUTE THAT(A,X)
CALL P2ARRAY(RTHAT1,IPRTHAT,IPCHAT,IDHAT,2,'AMP ARRAY THAT1(A,X)')
CALL P2ARRAY(RTHAT2,IPRTHAT,IPCHAT,IDHAT,2,'AMP ARRAY THAT2(A,X)')
C
C ****** UPWARD REFLECTANCE R(X,A)
C RT CONTAINS THE GEOMETRIC R(X,A;R,S,U,V)
C RTHAT1 CONTAINS THE SPECTRAL RTHAT1(X,A;R,L/U,K)
C RTHAT2 CONTAINS THE SPECTRAL RTHAT2(X,A;R,L/U,K)
C
C READ THE GEOMETRIC R(X,A)
READ(NURX) NUNIT
IF(NUNIT.EQ.NURX) THEN
WRITE(6,700) 'RTHAT1(X,A)', 'RTHAT2(X,A)', 'R(X,A)' ELSE
WRITE(6,702) NUNIT, 'NURX', NURX
STOP
ENDIF
DO 720 J=1,NCOLRT
720 READ(NURX) (RT(I,J),I=1,NCOLRT)
CALL FULLRT(RT,NMU,NPHI,IDRT)
CALL P2ARRAY(RT,IPRT,IPCRT,IDRT,2,GEOMETRIC R(X,R,S/U,V))
C
C COMPUTE THAT(A,X)
CALL RTSPEC(PHI,RT,IDRT, RTHAT1,RTHAT2,IDHAT)
C
DO 211 J=1,NCHAT
211 WRITE(NUOUT) (RTHAT1(I,J),I=1,NRHAT)
DO 212 J=1,NCHAT
212 WRITE(NUOUT) (RTHAT2(I,J),I=1,NRHAT)
C
C ****** UPWARD TRANSMITTANCE T(X,A)
C RT CONTAINS THE GEOMETRIC T(X,A;R,S,U,V)
C RTHAT1 CONTAINS THE SPECTRAL THAT1(X,A;R,L/U,K)
C RTHAT2 CONTAINS THE SPECTRAL THAT2(X,A;R,L/U,K)
C
C READ THE GEOMETRIC T(X,A)
READ(NUTX) NUNIT
IF(NUNIT.EQ.NUTX) THEN
WRITE(6,700) 'THAT1(X,A)', 'THAT2(X,A)', 'T(X,A)' ELSE
WRITE(6,702) NUNIT, 'NUTX', NUTX
STOP
ENDIF
DO 730 J=1,NCOLRT
730 READ(NUTX) (RT(I,J),I=1,NCOLRT)
CALL FULLRT(RT,NMU,NPHI,IDRT)
CALL P2ARRAY(RT,IPRT,IPCRT,IDRT,2,GEOMETRIC R(X,R,S/U,V))
C
C COMPUTE THAT(X,A)
CALL RTSPEC(PHI,RT,IDRT, RTHAT1,RTHAT2,IDHAT)
§4. PROGRAM 3

DO 231 J=1,NCHAT
231 WRITE(NOUT) (RTHAT1(I,J),I=1,NRTHAT)
DO 232 J=1,NCHAT
232 WRITE(NOUT) (RTHAT2(I,J),I=1,NRTHAT)
CALL P2ARRAY(RTHAT1,IPR1AT,IPCHAT,1DIMAT,2,'AMP ARRAY THAT1(X,A)')
CALL P2ARRAY(RTHAT2,IPR1AT,IPCHAT,1DIMAT,2,'AMP ARRAY THAT2(X,A)')
C
C*******DOWNWARD REFLECTANCE R(A,X)
C RT CONTAINS THE GEOMETRIC R(A,X;R,S;U,V)
C RTHAT1 CONTAINS THE SPECTRAL RHAT1(A,X;R,L/U,K)
C RTHAT2 CONTAINS THE SPECTRAL RHAT2(A,X;R,L/U,K)
C
ENDF

C
C READ THE GEOMETRIC R(A,X)
READ(NURAX) NUNIT
IF(NUNIT.EQ.NURAX) THEN
WRITE(6,700) 'RHAT1(X,A)', 'RHAT2(X,A)', 'R(A,X)'
ELSE
WRITE(6,702) NUNIT, 'NURAX', NURAX
STOP
ENDIF

DO 740 J=1,NCOLRT
740 READ(NURAX) (RT(I,J),I=1,NROWRT)
C
CALL FULLRT(RT,NMU,NPHI,1DRT)
CALL P2ARRAY(RT,IPRT,IPCRT,1D1RT,2,GEOMETRIC R(X,R,S/U,V)')
C
C COMPUTE RHAT(A,X)
CALL RTSPEC(PHI,RT,1DRT, RTHAT1,RTHAT2,1DIMAT)
C
DO 201 J=1,NCHAT
201 WRITE(NOUT) (RTHAT1(I,J),I=1,NRTHAT)
DO 202 J=1,NCHAT
202 WRITE(NOUT) (RTHAT2(I,J),I=1,NRTHAT)
CALL P2ARRAY(RTHAT1,IPR1AT,IPCHAT,1DIMAT,2,'AMP ARRAY THAT1(X,A)')
CALL P2ARRAY(RTHAT2,IPR1AT,IPCHAT,1DIMAT,2,'AMP ARRAY THAT2(X,A)')
C
ENDFILE NOUT
WRITE(6,750) NOUT
C
FORMATS
C
700 FORMAT(1HI,' NOW COMPUTING ',A10, ' AND ',A10, ' FROM ',A6)
702 FORMAT(1HI, ' ERROR: NUNIT ',I3, ' AND ',A6, ' = ',I3)
750 FORMAT(1HI, ' NORMAL EXIT FROM PROGRAM 3. TAPE ',I3, ' WRITTEN')
END
SUBROUTINE INISHL

ON NHM3/INISHL3

THIS ROUTINE INITIALIZES PROGRAM 3

PARAMETER (MXMU=10, MXPHI=24)
DIMENSION FMU(MXMU).BNDMU(MXMU).BNDPHI(MXPHI).
1 OMEGA(MXMU).DELMU(MXMU)
COMMON/CPHI/ PHI(MXPHI)
COMMON/CMISC/ IMISC(20).FMISC(20)
DATA NURAX/22/, NUOUT/30/

READ(5,*) IDBUG

READ HEADER RECORD OF ONE OF THE GEOMETRIC ARRAYS

REWIND NURAX
READ(NURAX) NUNIT,NRAYQO,IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,
1 DELMU
REWIND NURAX

NMU = IMISC(1)
NPHI = IMISC(2)
NL = NPHI/2
IMISC(3) = NL
IMISC(4) = IDBUG
NRHAT = NMU*(NL+1)
IMISC(10) = NRHAT
NLHAT = NMU*((NL+2)/2)
IMISC(11) = NLHAT
WNDSPD = FMISC(15)
REFR = FMISC(18)

WRITE(6,300) NMU,NPHI,NL,WNDSPD,REFR

WRITE HEADER RECORDS ON OUTPUT FILE

REWIND NUOUT
WRITE(NUOUT) NUOUT,IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELMU

FORMATS

300 FORMAT(1H1,' PROGRAM 3 OF THE NATURAL HYDROSOL MODEL'//
1HM 'COMPUTATION OF UPPER BOUNDARY SPECTRAL REFLECTANCE AND TRANSMITTANCE ARRAYS'//1H,' NMU = ',I3//1H,' NPHI = ',I3//1H,
3' NL = ',I3//1H,' WNDSPD = ',F7.3//1H,' REFR = ',F6.3)
END
§4. PROGRAM 3

SUBROUTINE FULLRT(RTNMUNPHI,IDRT)
C
ON NHM3/FULLRT
C
THIS ROUTINE CREATES A FULL (SQUARE) GEOMETRIC R OR T ARRAY FROM
THE "TOP HALF" DESCRIBED IN SECTION 12B. USE OF THE FULL ARRAY
MEANS THAT NO SPECIAL INDEXING CALCULATIONS (SEE PAGE 191) NEED
TO BE DONE
C
DIMENSION RT(IDRT,1)
C
NFULL = NMU*NPHI
NHALF = NFULL/2
DO 100 IROW=NHALF+1,NFULL
   DO 102 JCOL=1,NHALF
   102 RT(IROW,JCOL) = RT(IROW-NHALF,JCOL*NHALF)
   DO 100 JCOL=NHALF+1,NFULL
   100 RT(IROW,JCOL) = RT(IROW-NHALF,JCOL-NHALF)
C
RESET THE POLAR CAP OUTPUT FOR THE BOTTOM HALF (ZERO VALUES
CAME FROM THE B BLOCK, SEE PAGE 190)
C
DO 110 IROW=NHALF+1,NFULL
   110 RT(IROW,NMU) = RT(IROW-NHALF,NMU)
C
RE-ZERO THE POLAR CAP OUTPUT COLUMN AT PHI = 180, WHICH HAS
PICKED UP NON-ZERO VALUES FROM THE A-BLOCK
C
JCOL = NMU + NHALF
DO 104 IROW=1,NFULL
   104 RT(IROW,JCOL) = 0.
C
RETURN
END

SUBROUTINE RTSPEC(PHI,RT,IDRT, RTHAT1,RTHAT2,ID1HAT)
C
ON NHM3/RTSPEC
C
THIS ROUTINE FIRST COMPUTES THE SPECTRAL AMPLITUDES FROM THE
VARIOUS SPECIAL CASES, 5.31C TO 5.36, GIVEN RT = R OR T IN
GEOMETRIC FORM.
C
THE AMPLITUDES RTHAT1 = RTHAT1 OR THAT1 AND RTHAT2 = RTHAT2 OR THAT2,
ARE STORED ON THE COMPRESSED SPECTRAL ARRAY FORMAT OF (12.4).
C
THE SPECTRAL AMPLITUDES ARE THEN CHECKED USING RAYLEIGH'S EQUALITY 4.17
C
FINALLY, THE MATRIX ELEMENTS DEFINED BY (5.41) AND (5.43) ARE
COMPUTED FROM THE ARRAYS OF AMPLITUDES.
C
N.B. IN THIS ROUTINE, K AND L ARE REVERSED FROM THE NOTATION
USED IN THE TECH REPORT ERL-PWEL-75.
C
DIMENSION PHI(1), RT(IDRT,1), RTHAT1,ID1HAT,1),RTHAT2(ID1HAT,1)
COMMON/CMIISC/ I3MISC(20)
C
59
§4. PROGRAM 3

NMU = IMISC(1)
NPHI = IMISC(2)
NL = IMISC(3)
IDBUG = IMISC(9)
NRHAT = IMISC(10)
NCHAT = IMISC(11)

C DO 100 K=0,NL
AK = FLOAT(K)
IF(K.EQ.0 .OR. K.EQ.NL) THEN
EPSK = FLOAT(NPHI)
ELSE
EPSK = FLOAT(NL)
ENDIF

C DO 100 L=0,NL
C SKIP THE COMPUTATION IF (K + L) IS ODD
IF(MOD(K+L,2).NE.0) GO TO 100

C AL = FLOAT(L)
IF(L.EQ.0 .OR. L.EQ.NL) THEN
EPSL = FLOAT(NPHI)
ELSE
EPSL = FLOAT(NL)
ENDIF

C DO 102 IU=1,NMU
C DO 102 I=1,NMU
C STORAGE INDICES FOR SPECTRAL ARRAYS
IUS = NMU*L + IU
IVS = NMU*K + IU - NMU*((K+L)/2 - L/2)
C
C IF(IR.LT.NMU .AND. IU.LT.NMU) THEN
C GENERAL CASE: INPUT QUAD IS NONPOLAR, OUTPUT QUAD IS NONPOLAR; USE 5.31C
C
SUM1 = 0.
SUM2 = 0.
DO 204 IS=1,NPHI
COSLPS = COS(AL*PHI(IS))
SINLPS = SIN(AL*PHI(IS))
IROW = NMU*(IS-1) + IR
DO 204 IV=1,NPHI
SUM1 = SUM1 + RT(IROW,NMU*(IV-1)+IU)*COSLPS*COS(AK*PHI(IV))
SUM2 = SUM2 + RT(IROW,NMU*(IV-1)+IU)*SINLPS*SIN(AK*PHI(IV))
RTHAT1(IUS,IVS) = SUM1/(EPSL*EPSK)
IF(L.EQ.0 .OR. L.EQ.NL .OR. K.EQ.0 .OR. K.EQ.NL) THEN
RTHAT2(IUS,IVS) = 0.
ELSE
RTHAT2(IUS,IVS) = SUM2/(EPSL*EPSK)
ENDIF

C SPECIAL CASES FOR THE POLAR CAPS
C ELSEIF(IR.EQ.NMU .AND. IU.LT.NMU) THEN
C INPUT QUAD IS THE POLAR QUAD, OUTPUT IS NONPOLAR; USE 5.32
C
IRT = NMU
IF(L.EQ.0) THEN
SUM1 = 0.
DO 200 IV=1,NPHI
SUM1 = SUM1 + RT(IRT,NMU*(IV-1)+IU)*COSLPS*COS(AK*PHI(IV))
RTHAT1(IUS,IVS) = SUM1/EPSK
RTHAT2(IUS,IVS) = 0.
ELSE
RTHAT1(IUS,IVS) = 0.
RTHAT2(IUS,IVS) = 0.
ENDIF
ELSEIF (IR.LT.NMU .AND. J.EQ.NMU) THEN

C INPUT QUAD IN NONPOLAR, OUTPUT QUAD IN THE POLAR CAP; USE 5.34

C

JRT = NMU

IF (K.EQ.0) THEN

SUM1 = 0.

DO 202 IS=1,NPHI

SUM1 = SUM1 + RT(NMU*(IS-1)+IR,JRT)*COS(AL*PHI(IS))

RTMAT1(IUS,IVS) = SUM1/EPSL

RTMAT2(IUS,IVS) = 0.

ELSE

RTMAT1(IUS,IVS) = 0.

RTMAT2(IUS,IVS) = 0.

ENDIF

C

ELSEIF (IR.EQ.NMU .AND. IU.EQ.NMU) THEN

C INPUT QUAD IS THE POLAR CAP, OUTPUT QUAD IS THE POLAR CAP; USE 5.36

C

IF (K.EQ.0 .AND. L.EQ.0) THEN

RTMAT1(IUS,IVS) = RT(NMU,NMU)

RTMAT2(IUS,IVS) = 0.

ELSE

RTMAT1(IUS,IVS) = 0.

RTMAT2(IUS,IVS) = 0.

ENDIF

C

END IF

C

102 CONTINUE

100 CONTINUE

C

CHECK THE COMPUTED SPECTRAL AMPLITUDES

IF (IDBUG.NE.0) THEN

IPRHAT = 40

IPCHAT = 20

CALL P2ARAY(RTHAT,IPRHAT,IPCHAT,IU1IHAT,2.

1. THE SPECTRAL AMPLITUDES RTHAT1)

CALL P2ARAY(RTHAT2,IPRHAT,IPCHAT,IU1IHAT,2.

1. THE SPECTRAL AMPLITUDES RTHAT2)

CALL SPECHK(RTHAT1,RTHAT2,IPRHAT,IPCHAT,2.

IDIHAT)

ENDIF

C

CONVERT THE SPECTRAL AMPLITUDES TO THE SPECTRAL ARRAYS DEFINED BY

5.41 AND 5.43. THE ARRAY ELEMENTS ARE THE AMPLITUDES

MULTIPLIED BY FACTORS OF 1, NL OR NPHI, AS SEEN IN TABLES 1 AND 2

ON PAGES 90 AND 91.

C

EPSL = FLOAT(NPHI)

DO 300 IROW=1,NMU-1

DO 300 JCOL=1,NCHAT

RTMAT1(IROW,JCOL) = EPSL*RTHAT1(IROW,JCOL)

300 RTMAT1(IROW,JCOL) = EPSL*RTHAT1(IROW,JCOL)

EPSL = FLOAT(NCHAT)

DO 302 IROW=NCHAT+1,NMU

DO 302 JCOL=1,NCHAT

RTMAT1(IROW,JCOL) = EPSL*RTHAT1(IROW,JCOL)

302 RTMAT2(IROW,JCOL) = EPSL*RTHAT2(IROW,JCOL)

EPSL = FLOAT(NMU)

DO 304 IROW=NCHAT+1,NMU

DO 304 JCOL=1,NCHAT

RTMAT1(IROW,JCOL) = EPSL*RTHAT1(IROW,JCOL)

304 RTMAT1(IROW,JCOL) = EPSL*RTHAT1(IROW,JCOL)

RETURN

END
§4. PROGRAM 3

SUBROUTINE SPECHK(RT,IDRT,RTHAT1,RTHAT2,IDIHAT)
ON NHM3/SPECHK

THIS ROUTINE CHECKS THE COMPUTED SPECTRAL R AND T AMPLITUDES
BY SEEING IF THE WEIGHTED SUM OF THE SPECTRAL AMPLITUDES SQUARED EQUALS
THE SUM OF THE GEOMETRIC ELEMENTS SQUARED (RAYLEIGH'S EQUALITY, 4.17)

THIS CHECK HOLDS ONLY FOR NON-POLAR QUADS.

PARAMETER(MXMU=10)
DIMENSION RT(IDRT,1),RTHAT1(IDIHAT,1),RTHAT2(IDIHAT,1)
DIMENSION GEOSUM(MXMU,MXMU),SPCSUM(MXMU,MXMU)
COMMON/Cmisc/ IMISC(20)

NMU = IMISC(1)
NPHI = IMISC(2)
NL = IMISC(3)
NRTGEO = NMU*NPHI

DO 100 I=1,NMU-1
DO 100 J=1,NMU-1

COMPUTE THE SUM OF SQUARES OF THE GEOMETRIC ARRAY ELEMENTS

SUM = 0.
DO 110 IROW=1,NRTGEO,NMU
DO 110 ICOL=J,NRTGEO,NMU
110 SUM = SUM + RT(IROW,ICOL)**2
GEOSUM(I,J) = SUM
IF(I.EQ.NMU .AND. J.EQ.NMU) WRITE(6,333) RT(I,J)
333 FORMAT(1F18.10)

COMPUTE THE WEIGHTED SUM OF SQUARES OF THE SPECTRAL AMPLITUDES.
The AMPLITUDES ARE STORED ON THE ARRAY FORMAT OF (12.4).

SUM = 0.
DO 120 K=0,NL
IF(K.EQ.0 .OR. K.EQ.NL) THEN
EPSK = FLOAT(NPHI)
GAMK = 0.
ELSE
EPSK = FLOAT(NL)
GAMK = FLOAT(NL)
ENDIF

DO 120 L=0,NL
IF(MOD(K+L,2).NE.0) GO TO 120
IF(L.EQ.0 .OR. L.EQ.NL) THEN
EPSL = FLOAT(NPHI)
GAML = 0.
ELSE
EPSL = FLOAT(NL)
GAML = FLOAT(NL)
ENDIF

COMPUTE ROW AND COLUMN INDICES OF THE COMPRESSED AMPLITUDE ARRAYS.

IROW = I + NMU*K
ICOL = J + NMU*L - NMU*(K+L)/2 - K/2

SUM = SUM + EPSK*EPSL*RTHAT1(IROW,ICOL)**2 + 
1 GAML*GAML*RTHAT2(IROW,ICOL)**2
120 CONTINUE

SPCSUM(I,J) = SUM

100 CONTINUE
(CALL P2ARRAY(GEOSUM,NMU-1,NMU-1,MXMU-1,..
1 SUMS OF SQUARES OF THE NON-POLAR GEOMETRIC R/T ARRAY ELEMENTS)
CALL P2ARRAY(SPCSUM,NMU-1,NMU-1,MXMU-2,..
1 RAYLEIGH SUMS OF SQUARES OF THE NON-POLAR SPECTRAL R/T AMPLITUDES)
RETURN
END
5. PROGRAM 4

A. Program Description

This program performs the remaining initialization steps of 75/7a.3-7a.5 and then assembles the solution amplitudes as described in 75/§7b. The internal structure of Program 4 is essentially that shown in 75/Fig. 7. This program is the other main consumer of computer power in the NHM, owing to the discretization of the phase function.

It is usually convenient to run Program 4 in two different modes. In the first mode (ICPHAS ≠ 0 in record 3, below), the program computes and stores the quad-averaged phase function as described in 75/§11. These calculations can be very expensive if the phase function is highly peaked in the forward direction, as is the case in natural waters. However, these calculations need be done only once for a given phase function (and a given quad partition). In the second mode (ICPHAS = 0 in record 3), it is assumed that the phase function has already been discretized; the file containing this information is read and the radiance amplitudes are then computed. In the case of a spherically symmetric phase function, which may be of interest for comparison purposes, the discretization calculations are trivial. In this case, it may be convenient to run Program 4 to completion each time (i.e. both modes 1 and 2); the discretized spherical phase function is not worth saving.

B. Input

Two more parameters, which determine maximum array dimensions, must be set at compilation time. These parameters are (see the first PARAMETER statement in MAIN).

<table>
<thead>
<tr>
<th>parameter</th>
<th>value in listed code</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MXY</td>
<td>30</td>
<td>the maximum number of optical depths ( y_j ), ( j = 1,\ldots,YOUT ), at which the final output is desired (see 75/Fig. 6)</td>
</tr>
<tr>
<td>MXSIGY</td>
<td>2</td>
<td>the maximum number of optical depths ( y_i ), ( i = 1,\ldots,YOP ), at which the inherent optical properties are specified (see 75/Fig. 6)</td>
</tr>
</tbody>
</table>

Referring to 75/pages 132-135, MXY gives the maximum allowed value of YOUT and MXSIGY gives the maximum allowed value of YOP.

Six free-format data records are read at execution time, as follows.

Record 1: ITITLE

This is an alphanumeric title for the run, used to identify the printout. Up to 80 characters are allowed.
§5. PROGRAM 4

Record 2: IDEBUG, WAVENM, ABSORB

IDBUG = 0, 1, or 2, as in Program 1

WAVENM is the wavelength in nanometers of the monochromatic radiance. This wavelength is used in subroutine PHASEF (see the version for the Pelagos Sea in the code listing) to select the correct wavelength dependent absorption and scattering functions. In the listed code, WAVENM must be one of the 13 wavelengths 400.0, 425.0,..., 675.0, 700.0, although this is not a restriction of the NHM algorithms.

ABSORB If ABSORB < 0.0, then the value of the absorption coefficient returned by PHASEF is used.
If ABSORB ≥ 0.0, then the absorption coefficient is set to ABSORB. This overrides the value returned by PHASEF. (This is useful for studies in which only the scattering-to-absorption ratio changes.)

Record 3: ICPHAS, NUQB, NVQB, INCBAS

This record gives information for the discretization of the phase function.

ICPHAS = 0 if the phase function has already been discretized in a previous run of Program 4. File NUPHAS will be read.

≠ 0 if this run is to discretize the phase function.
If ICPHAS < 0, the run stops after file NUPHAS has been written.
If ICPHAS > 0, the run discretizes the phase function, writes file NUPHAS, and continues with the amplitude computations.

NUQB the value of \( \mu_1 \) in 75/11.3. A value of 1 can be used for a spherically symmetric phase function. Use NUQB = 3 or 4 for the quad resolution of 75/Fig. 4a or 4b and phase functions typical of natural waters.

NVQB the value of \( \mu_0 \) in 75/11.3. Use values like those for NUQB.

INCBAS the factor for increasing the base numbers of subcells (\( \mu_1 \) and \( \mu_0 \)) in 75/11.3, for quad pairs which involve forward (or near forward) scattering. A value of 10 is reasonable for natural waters (use 1 if the phase function is spherically symmetric). If NUQB = 3 and INCBAS = 10, say, then in 75/11.3, \( \mu_1 \) is increased to 30 for quads involving forward scattering. This gives a more accurate evaluation of 75/11.1.

Record 4: IBOTM, RFLBOT

This record specifies the bottom boundary condition.

IBOTM = 0 if the bottom is to be a matte surface at a finite depth. The surface has a reflectance of \( \rho_\infty = \text{RFLBOT} \) (see 75/3.26)
§5. PROGRAM 4

= 1 if the bottom is infinitely deep, and the water is homogeneous below depth \( z \) (see 75/§10).

**RFLB**OT

The bottom reflectance (used only if \( IBOTM = 0 \)). \( 0.0 \leq RFLBOT \leq 1.0 \).

**Record 5: IYOP, NY, YOUT(1),...,YOUT(NY)**

This record specifies the depths at which output is desired.

**IYOP**

= 0 if the YOUT values as read are geometric depths in meters (in the listed code, this option is available only for attenuation functions which are independent of depth).

= 1 if the YOUT values as read are optical depths (this option is valid for inhomogeneous waters).

**NY**

the number of \( y \)-levels where output is required (NY is YOUT in 75/Fig. 6. \( NY \leq MXY \)).

**YOUT(1)**

the depths where output is desired. Always set \( YOUT(1) = 0.0 \equiv x \).

The value of \( YOUT(NY) \) is \( z \) (see 75/Fig. 6).

A convenient set of optical depths for printout in infinitely deep water, homogeneous below \( z = 20.0 \) optical depths, might be

\[ 0.0, 0.5, 1.0, 2.0, 5.0, 10.0, 15.0, \text{and} 20.0. \]

Here \( YOUT(1) \equiv x = 0.0, \ YOUT(2) = 0.5, \ldots, YOUT(NY) \equiv z = 20.0, \) with \( NY = 8 \). See input records 3 and 5 of Program 5 for special choices of \( y_j = YOUT(j) \) which are often convenient for checking the results, computing K-functions, etc.

**Record 6: RSKY, CARD, SHTOTL, THETAS, PHIS**

This record specifies the incident (sky + sun) radiance distribution, using the model described in Appendix B.

**RSKY**

The ratio of sky to total (sky + sun) input scalar irradiance; \( 0.0 \leq RSKY \leq 1.0 \). \( RSKY = 0.0 \) for a black sky (sun only); \( RSKY = 1.0 \) for a background sky only (no sun).

**CARD**

The cardioidal parameter for the sky radiance distribution (see Appendix B of this report): \( CARD = 0.0 \) for a uniform sky; \( CARD = 2.0 \) for a cardioidal sky.

**SHTOTL**

The total (sky + sun) spectral scalar irradiance on the water surface at the given wavelength, in \( \text{W m}^{-2} \text{nm}^{-1} \).

**THETAS**

The polar angle, \( \theta_s \), in degrees of the sun's location in the sky. \( THETAS = 0.0 \) for the sun at the zenith; \( THETAS = 90.0 \) for the sun at the horizon.

65
PHIS

The azimuthal angle, $\phi_s$, in degrees of the sun's location, measured counterclockwise from $\phi_s = 0.0$ in the downwind direction (e.g. PHIS = 90.0 places the sun in the crosswind direction).

In addition to the above data records, the user must make sure that the desired version of subroutine PHASEF is being used. This routine specifies the inherent optical properties of the water body. Four versions of PHASEF are listed in this report. Two of these define absorption and scattering functions typical of natural waters: "Lake Limne" and "the Pelagos Sea" which are, respectively, typical of lakes and open ocean waters. The other two examples of PHASEF are for a spherical scattering function: one is depth independent and one is for depth dependent absorption and scattering. The user wishing to make runs with his own absorption and scattering functions must write a corresponding version of PHASEF, mimicking the listed examples.

Likewise, a user wishing to specify an input radiance distribution other than the ones obtainable from the formulas in Appendix B must write a corresponding version of subroutine QASKY. This would be the case if, for example, the user had measured the sky radiance distribution with a few cumulus clouds present in an otherwise clear sky, and wished to include the cloud effects in the computed radiances.

C. File Management

Three permanent files are either read or written by Program 4; an additional three temporary files are used for scratch storage.

<table>
<thead>
<tr>
<th>internal name</th>
<th>external name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUSRT</td>
<td>TAPE30</td>
<td>The file of spectral $\hat{r}$ and $\hat{t}$ arrays for the air-water surface, from Program 3.</td>
</tr>
<tr>
<td>NUPHAS</td>
<td>TAPE39</td>
<td>The file containing the quad-averaged phase function. It is written if ICPHAS $\neq 0$ and read if ICPHAS = 0.</td>
</tr>
<tr>
<td>NUOUT</td>
<td>TAPE40</td>
<td>The file containing the radiance amplitudes (and other information) generated by Program 4.</td>
</tr>
<tr>
<td>NUSCR1</td>
<td>TAPE45</td>
<td>Temporary scratch files used in integrating the Riccati equations. NUSCR1 holds $R(y,x;\ell)$; NUSCR2 holds $T(x,y;\ell)$; NUSCR3 holds $R_1(y,b;\ell)$ and $R_2(y,b;\ell)$.</td>
</tr>
</tbody>
</table>
§5. PROGRAM 4

D. Code Listing

```plaintext
PROGRAM MAIN (INPUT, OUTPUT, TAPE6=INPUT, TAPE6=OUTPUT, TAPE30, TAPE39,
1 TAPE40, TAPE42, TAPE46, TAPE47)

ON NIMAI

******************************************************************************
* THIS IS PROGRAM 4 OF THE NATURAL HYDROSOIL MODEL *
******************************************************************************
PARAMETER (MXMU=10, MXPHI=24, MXV=30, MXSIGV=3)
PARAMETER (MXL=MXPHI/2, MXALG=MXMU*(MXL+1), MXAMP=2*MXMU*(MXL+1))
PARAMETER (MXRRTH=MXMU*(MXL+1), MXCRTH=MXMU*((MXL+2)/2))
PARAMETER (MXWCK=MXMU*(1 + 3*(MXL+1)*(1 + (MXL+2)/2)))

COMMON/CAMP/ AAM(MXAMP), APM(MXAMP), AYM(MXAMP, MXV), AVP(MXAMP, MXV)
COMMON/CSIGY/ VSIG(MXSIGY), ALBE(TM(XSIGY), TOTALS(MXSIGY)
COMMON/CRADO/ JMUO(MXSRC), KP-1O(MXSRC), RAO(MXSRC)
COMMON/CRTA/ RXO(MXMU, MXMU, MXV), RXO(MXMU, MXV, MXV), RXO(MXMU, MXV)
COMMON/CRTRA/ FMU(MXMU), PHM1(MXPHI), Y(MXV), TNHDM1(MXMU),
1 1NPHD1(MXPHI), CMEGA(MXPHI), DELTM1(MXMU, ZGEU(MXV)
COMMON/CRTSIC/ PHOHAT(MXMU, MXMU, MXSIGV), TAUHAT(MXMU, MXMU, MXSIGV),
1 ALGPM(MXMU, MALGPM(MXV), ALGPM(MXMU, MALGPM, MXSIGV)
COMMON/CRTBW/ RHA1(MXPHI, MXPHI, MXPHI, MXPHI)
COMMON/CRTHAT/ T1=1(MXRRTH, MXCRTH), T1=1(MXRRTH, MXCRTH),
1 RHA1(MXRRTH, MXCRTH, MXCRTH),
1 RHA1(MXRRTH, MXCRTH, MXCRTH)
COMMON/CWTR/ IMIC(20), PMISC(20)
COMMON/CWTR/ WERK, MAWERK

DATA NUSRT/30, NUSRT/40, NUSRT/45, NUSRT/46, NUSRT/47

******* INITIALIZATION *******

CALL INISHL

NSIGY = IMICS(5)
IMISC(18) = NUSRT
IMISC(19) = NUSRT
IMISC(20) = NUSRT

NMU = IMICS(1)
ML = IMICS(14)
MV = IMICS(14)
MBUG2 = IMICS(14)
NRHAT = IMICS(14)
NCHAT = IMICS(14)
NMP = 2*NHAT

C: INITIAL LOAD OF THE SPECTRAL DISHARGE ARRAYS
C
C THAT1(A, X) INTO THAT1
C THAT2(A, X) INTO THAT2
C RHA1(A, X) INTO RHA1
C RHA2(A, X) INTO RHA2
C
C CC 101 = 0, NCH
100 READ(NUSRT) (THAT1(1,J), I=1, NHA1)
100 IF (J.LT.1) GO TO 100
100 READ(NUSRT) (THAT2(1,J), I=1, NHA1)
100 IF (J.LT.1) GO TO 100
100 READ(NUSRT) (RHA1(1,J), I=1, NHA1)
100 IF (J.LT.1) GO TO 100
100 READ(NUSRT) (RHA2(1,J), I=1, NHA1)
```

67
§5. PROGRAM 4

C IF(IDBUG2.GT.0) THEN
CALL P2ARAY(THAT1,2*NMU,NMU,MRKTH,2,'THAT1(A,X) AS LOADED')
CALL P2ARAY(THAT2,2*NMU,NMU,MRKTH,2,'THAT2(A,X) AS LOADED')
CALL P2ARAY(RHAT1,2*NMU,NMU,MRKTH,2,'RHAT1(X,A) AS LOADED')
CALL P2ARAY(RHAT2,2*NMU,NMU,MRKTH,2,'RHAT2(X,A) AS LOADED')
ENDIF

********** BEGIN COMPUTATIONS **********

C COMPUTE THE DIRECT BEAM AMPLITUDES AO(Y,-) AT ALL LEVELS Y = A, X, ..., Z FROM THE QUAD-AVERAGED SKY RADIANCES WHICH ARE STORED IN /CWORK/

C CALL AMPAD
/CWORK/ IS NOW ENTIRELY FREE

C SAVE AO(Y,-) (STORED IN AAM) AT ALL Y LEVELS
WRITE(NUOUT) (AAM(I),I=1,NRAMP)
DO 150 J=1,NY
WRITE(NUOUT) (AYM(I,J),I=1,NRAMP)
IF(IDBUG2.NE.0) THEN
WRITE(6,1038)
ENDIF
END IF

C SET DEBUGGING OUTPUT FOR SELECTED L VALUES
IF(IDBUG2.GT.0) THEN
I九年.L.LE.1 .OR. L.GE.NL+1 THEN
IDBUG = IDBUG2
ELSE
IDBUG = 0
ENDIF
ELSE
IDBUG = IDBUG2
ENDIF
IMISC(9) = IDBUG

C**** SOLUTION STEP 1 (SEE PAGE 133 AND FIGURE 7 ON PAGE 140)

C COMPUTE RHOHAT AND TAUHAT AT EACH Y LEVEL WHERE SIGMA AND ALPHA ARE GIVEN

C CALL RHOHAT(L)
IF(IDBUG2.GT.0) THEN
WRITE(6,202)
CALL P2ARAY(RHOLAT,NMU,NMU,NMUL,2,'RHOHAT(L)')
CALL P2ARAY(TAUHAT,NMU,NMU,NSIGY,MKM,2,'TAUHAT(L)')
ENDIF

C**** SOLUTION STEP 2
C COMPUTE RHAT1(Z,B) FOR THE DESIRED BOTTOM BOUNDARY CONDITION

C CALL BOTMBC(L)
IF(IDBUG2.GT.0) THEN
CALL P2ARAY(RHATZB,NMU,NMU,MKM,2,'RHAT1(Z,B,L)')
ENDIF

C**** SOLUTION STEPS 3 AND 4
C INTEGRATE THE RICATTI EQUATIONS TO GET R(Y,X), T(X,Y), AND RP(Y,B)
C
C CALL RICATI(L)
C WRITE R(Y,X), T(X,Y) AND RP(Y,B) FOR THIS L VALUE TO SCRATCH FILES

DO 220 IV=1,NV
WRITE(NUSCR1) ((RX(I,J,IV),I=1,NMU),J=1,NMU)
WRITE(NUSCR2) ((TX(I,J,IV),I=1,NMU),J=1,NMU)
WRITE(NUSCR3) ((RYB(I,J,IV),I=1,NMU),J=1,NMU)
WRITE(NUSCR4) ((R2YB(I,J,IV),I=1,NMU),J=1,NMU)
IF(IDBUG EQ.2) THEN
CALL P2ARAY(RX,NMU,NMU,NV,MKM,MKM,2,'R(Y,X,L)')
CALL P2ARAY(TX,NMU,NMU,NV,MKM,MKM,2,'T(X,Y,L)')
CALL P2ARAY(RYB,NMU,NMU,NV,MKM,NMUL,2,'R1(Y,B,L)')
CALL P2ARAY(R2YB,NMU,NMU,NV,MKM,MKM,2,'R2(Y,B,L)')
ENDIF

220
§5. PROGRAM 4

C 200 CONTINUE
C
C***** SOLUTION STEPS 5 AND 6
C COMPUTE THE AMPLITUDES \( A(x,-) \) AND \( A(x,+). \)
C
CALL AMPX
C
C***** SOLUTION STEPS 7 AND 8
C COMPUTE THE INTERIOR AMPLITUDES \( A(v,-) \) AND \( A(v,+) \), \( x \lt v \leq z \).
C
CALL AMPINT
C
FINAL LOAD OF SPECTRAL STORAGE ARRAYS
THAT1(X,A) INTO THAT1
THAT2(X,A) INTO THAT2
RHA1(T,A,X) INTO RHAT1
RHA2(T,A,X) INTO RHAT2
C
DO 400 J=1,NCHAT
400 READ(NUSRT) (THAT1(I,J),I=1,NRHA1)
DO 401 J=1,NCHAT
401 READ(NUSRT) (THAT2(I,J),I=1,NRHA2)
DO 402 (NUSRT) (RHA1(I,J),I=1,NRHA1)
402 READ(NUSRT) (RHA2(I,J),I=1,NRHA1)
C
IF(IDBUG2.GT.0) THEN
CALL P2ARAY(THAT1.2*NMU,NMU,MXRRTH,2,'THAT1(X,A) AS LOADED')
CALL P2ARAY(THAT2.2*NMU,NMU,MXRRTH,2,'THAT2(X,A) AS LOADED')
CALL P2ARAY(RHA1.2*NMU,NMU,MXRRTH,2,'RHA1(T,A,X) AS LOADED')
CALL P2ARAY(RHA2.2*NMU,NMU,MXRRTH,2,'RHA2(T,A,X) AS LOADED')
ENDIF
C
C***** SOLUTION STEP 9
C COMPUTE THE AMPLITUDE \( A(a,+) \).
C
CALL AMPAP
C
WERK(1) NOW CONTAINS \( A(D) \), THE REFLECTED DIRECT BEAM.
C
******** END OF COMPUTATIONS ********
C
SAVE THE COMPUTED AMPLITUDES
C
WRITE(NUOUT) (WERK(I),I=1,NRAMP)
WRITE(NUOUT) (AAM(I),I=1,NRAMP)
WRITE(NUOUT) (AAP(I),I=1,NRAMP)
DO 450 J=1,NV
450 WRITE(NUOUT) (AAM(I),I=1,NRAMP)
DO 451 J=1,NV
451 WRITE(NUOUT) (AAP(I),I=1,NRAMP)
ENDFILE NUOUT
C
IF(IDBUG2.NE.0) THEN
WRITE(6,1039)
CALL PNTAMP(Y,AAM,AVM,MA3M)
WRITE(6,1040)
CALL PNTAMP(Y,AAM,AVM,MA3M)
ENDIF
C
WRITE(6,1042)
CALL PNTAMP(Y,AAM,AVM,MA3M)
END
C
202 FORMAT(1H1,' ***** BEGINNING THE L = 1, 1, LOOP *****')
501 FORMAT(1H1,' NORMAL EXIT FROM PROGRAM 4. TAPE 12,' WRITTEN.')
1038 FORMAT(1H1,' THE DOWNWARD DIRECT BEAM RADIANCE AMPLITUDES ARE:'//
 1 11X, MU',7X, A(D,-'), 8X, A(D,+')')
1039 FORMAT(1H1,' THE UPWARD DIRECT BEAM RADIANCE AMPLITUDES ARE:'//
 1 11X, MU',7X, A(D,+')')
1040 FORMAT(1H1,' THE DOWNWARD TOTAL RADIANCE AMPLITUDES ARE:'//
 1 11X, MU',7X, A(D,+')',9X, A(D,-')')
1041 FORMAT(1H1,' THE UPWARD TOTAL RADIANCE AMPLITUDES ARE:'//
 1 11X, MU',7X, A(D,+')',9X, A(D,-')')
END
§5. PROGRAM 4

SUBROUTINE INISHL

ON NHM4/INISHL4

THIS ROUTINE INITIALIZES PROGRAM 4 OF THE NHM.

PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
PARAMETER(MXGEOP=MXMU*(MXPHI/2+1))

COMMON/CMISC/ IMISC(20),FMISC(20)
COMMON/CRTSIG/ RHOHAT(MXMU,MXMU,MXSIGY),TAUHAT(MXMU,MXMU,MXSIGY),
1 GEOPP(MXMU,MXGEOP,MXSIGY),GEOPM(MXMU,MXGEOP,MXSIGY)
COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),YOUT(MXY).BNDMU(MXMU),BNDPHI(MXPHI),
1 OMEGA(MXPHI),DELMU(MXMU),ZGEO(MXY)
COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
COMMON/CWORK/ RADSKY(MXMU,MXPHI),PHASE(2701,MXSIGY).PSITAB(2701)

DIMENSION ITITLE(10)

DATA NUSRT,NUPHAS,NUOUT/30,39,40/
KINV......THE NUMBER OF TERMS IN THE SUM (7.4)
TOL........THE TOLERANCE FOR THE RICCATI EQUATION SOLVER (IMSL ROUTINE
OVERK)
DATA KINV/3.0/

READ INPUT RECORDS

RECORD 1:
ITITLE....A RUN TITLE, UP TO 80 CHARACTERS

RECORD 2:
IDBUG.....0 FOR NO INTERMEDIATE OUTPUT (PRODUCTION RUNS)
1 FOR MINIMAL OUTPUT FOR CHECKING (RECOMMENDED)
2 FOR FULL DEBUGGING OUTPUT
WAVENM.....THE WAVELENGTH IN NANOMETERS, ONE OF THE 13 VALUES 400,425,
675,700. (SEE PHASEF FOR THE PELAGOS SEA)
ABSORB....IF.GE.0., THEN THE ABSORPTION COEF A IS RESET TO THIS VALUE
(USED FOR LAKE LIMNE RUNS TO VARY A WITH WAVELENGTH)

RECORD 3:
ICPHAS.....0 IF THE QUAD-AVERAGED PHASE FUNCTIONS ARE TO BE READ
.NE.0 IF THE QUAD-AVERAGED PHASE FUNCTIONS ARE TO BE COMPUTED
(BY SUBROUTINE QAPHAS) AND STORED ON UNIT NUPHAS
INCBAS.....THE FACTOR FOR INCREASING THE BASE NUMBER OF SUBCELLS
FOR QUAD PAIRS WHICH INCLUDE FORWARD SCATTERING

RECORD 4:
IBOTM......0 FOR A MATTE BOTTOM AT Y = Z, OF REFLECTANCE R = RFLBOT
1 FOR AN INFINITELY DEEP BOTTOM, WITH HOMOGENEOUS WATER
BELOW DEPTH Y = Z
RFLBOT.....THE BOTTOM REFLECTANCE, (USED ONLY IF IBOTM = 0)
0.0 .LE. RFLBOT .LE. 1.0

RECORD 5:
IVOP......0 IF YOUT AS READ CONTAINS GEOMETRIC DEPTHS IN METERS
1 IF YOUT AS READ CONTAINS OPTICAL DEPTHS
NY.....THE NUMBER OF Y LEVELS WHERE OUTPUT IS DESIRED

RECORD 6:
RSKY.....THE RATIO OF SKY TO TOTAL INPUT SCALAR IRRADIANCE
0.0 FOR A BLACK SKY (SUN ONLY), RSKY = 1.0 FOR A
BACKGROUND SKY ONLY (NO SUN)
CARD.....THE CARDIOIDAL PARAMETER FOR THE SKY RADIANCE
DISTRIBUTION, CARD = 0. FOR A UNIFORM SKY, CARD = 2. FOR A
CARDIOIDAL SKY
§5. PROGRAM 4

SHTOTL..... THE TOTAL (SKY + SUN) SCALAR IRRADIANCE ON THE WATER
SURFACE, WATTS PER SQUARE METER
THEtas, PhiS...... THE SKY (SOURCE) LOCATION OF THE SUN, IN DEGREES.
THEtas IS 0. AT THE ZENITH, 90. AT THE HORIZON. Phi IS
MEASURED COUNTERCLOCKWISE FROM PHI = 0. IN THE
DOWNWIND DIRECTION

READ(5,1004) ITITLE
READ(5,*) IDBUG,WAVENM,ABSORB
READ(5,*) ICPHAS,NUQB,NVQB,INCBAS
READ(5,*) IBOTM,RFLOBOT
READ(5,*) IVOP,NY,(OUTH(IY),NY=1,NY)
READ(5,*) RSKV,CDRD,SHTOTL,THEtas,PhiS

READ HEADER RECORDS FROM THE SPECTRAL DATA FILE, NUSR

REWIND NUSR
READ(NUSR) NUNIT,IMISC,FMISC,FMU,PHI,BNDPHI,OMEGA,DELMU

NNU = IMISC(1)
NPNI = IMISC(2)
NL = IMISC(3)
TW0PI = 2.0*FMISC(1)
DEGRAD = FMISC(2)
RADEG = FMISC(3)
WINDSPD = FMISC(15)
KCOL = NNU*(NL + 1)

WRITE(6,1000)
WRITE(6,1005) ITITLE
WRITE(6,1010) NNU,NPNI,NU,NL,WINDFD,WAVENM,KINV,TOL
IF(ICPHAS.NE.O) WRITE(6,1014) NUQB,NVQB,INCBAS
IF(IBOTM.EQ.O) WRITE(6,1030) RFLOBOT
IF(IBOTM.EQ.1) WRITE(6,1031)

IMISC(4) = NY
IMISC(8) = KINV
IMISC(9) = IDBUG
IMISC(12) = IBOTM
FMISC(7) = TOL
FMISC(13) = WAVENM
FMISC(14) = RFLOBOT

COMPUTE THE <IN/>^<P>/QUAD-AVERAGED RADIANCE FOR THE SKY
CALL QASKV(RSKV,CARD,SHTOTL,THEtas,PhiS)

RADSKY IS IN /CWORK/ AND MUST BE SAVED UNTIL AMPAO IS CALLED IN MAIN
IF(ICPHAS.NE.0) THEN

COMPUTE AND SAVE THE QUAD-AVERAGED PHASE FUNCTIONS
INITIALIZE THE POINT GEOMETRIC SCATTERING FUNCTION

NOTE: MAKE SURE THE DESIRED VERSION OF PHASEF HAS BEEN LOADED
INTO THE EXECUTABLE ELEMENT (ABSOLUTE RUN FILE)

**X = PHASEF(0..J)
**SIG = IMISC(6)

GENERATE A TABLE OF PHASE FUNCTION VALUES FOR LOOKUP IN QAPHAS
/CWORK/ IS USED TO HOLD THE TABLE OF PHASE VALUES
DO 100 TVY=1,NV
  V = **SIG(TV)
  C
  DO 102 TVS=1,100
    PSITAB(V) = FLOAT(TVS-1)*DPSI*RADEG
    C(TVS) = COS(FLOAT(TVS-1)*DPSI)
102 C(TVS) = PHASEF(V,C(TVS))

71
§5. PROGRAM 4

C 10 .LT. PSI .LE. 180 DEGREES, BY 0.1 DEGREE STEPS
DPSI = DEGRAD*0.1
PSIO = DEGRAD*10.0
DO 100 I=100,2,2701
PSIO = PSIO + DPSI
PSITAB(I) = PSI(0)*RADEG
COSPSI = COS(P*)
100 PHASE(I,IV) = PHASEF(V,COSPSI)
C
IF(IDBUG.GE.1) THEN
WRITE(6,1050)
DO 110 I=1,20
I2 = I + 990
I3 = I + 2681
110 WRITE(6,1052) IPSITAB(I), PHASE(I,1), .12.PSITAB(I2), PHASE(I2,1), .13.PSITAB(13), PHASE(13,1)
C CHECK INTEGRAL OF PHASE FUNCTION BY SUM OF TABULATED VALUES
C SEE PAGE 11, EQ 2.7.
C
DPSI = DEGRAD*0.01
SUM = PHASE(2,1)*SIN(PSITAB(2)*DEGRAD)*0.5*DPSI
DO 120 I=3,1000
PSI = 0.01 TO 0.1
IF(I.EQ.1) SUM1=SUM + PHASE(1,1)*SIN(PSITAB(1)*DEGRAD)*0.5*DPSI
PSI = 0.01 TO 1.0
IF(I.EQ.101) SUM1=SUM + PHASE(1,1)*SIN(PSITAB(1)*DEGRAD)*0.5*DPSI
120 SUM = SUM + PHASE(1001,1)*SIN(PSITAB(1001)*DEGRAD)*0.5*DPSI
C PSI = 0.01 TO 10.0
SUM10 = SUM
DPSI = DEGRAD*0.1
SUM = SUM + PHASE(1001,1)*SIN(PSITAB(1001)*DEGRAD)*0.5*DPSI
DO 122 I=1002,2700
PSI = 0.01 TO 20.0
IF(I.EQ.1101) SUM20 = SUM + PHASE(1101,1)*0.5*DPSI
PSI = 0.01 TO 90.0
IF(I.EQ.1801) SUM90 = SUM + PHASE(1801,1)*0.5*DPSI
122 SUM = SUM + PHASE(1,1)*SIN(PSITAB(1)*DEGRAD)*DPSI
SUM01 = TWOPI*SUM01
SUM1 = TWOPI*SUM1
SUM10 = TWOPI*SUM10
SUM20 = TWOPI*SUM20
SUM90 = TWOPI*SUM90
SUM = TWOPI*SUM
SUM80 = SUM - SUM90
WRITE(6,1054) SUM01, SUM1, SUM10, SUM20, SUM90, SUM80, SUM
ENDIF
C
C COMPUTE THE QUAD-AVERAGED GEOMETRIC PHASE FUNCTION AS IN SECTION 11
C CALL QAPHAS(NUQB,NVQB,INCBA)
C
C STORE THE COMPUTED PHASE FUNCTIONS FOR LATER USE
C REWIND NUPHAS
WRITE(NUPHAS) NSIGY, VSIG, ALBESS, TOTALS
WRITE(NUPHAS) (:,(GEOPP(I,J,K),I=1,NUM),J=1,KCOL),K=1,NSIGY
WRITE(NUPHAS) (:,(GEOPM(I,J,K),I=1,NUM),J=1,KCOL),K=1,NSIGY
ENDFILE NUPHAS
WRITE(6,1060) NUPHAS
IF(1CPHAS.LT.0) STOP
C
ELSE
C READ EXISTING QUAD-AVERAGED PHASE FUNCTIONS FROM NUPHAS
C REWIND NUPHAS
READ(NUPHAS) NSIGY, VSIG, ALBESS, TOTALS
READ(NUPHAS) (:,(GEOPP(I,J,K),I=1,NUM),J=1,KCOL),K=1,NSIGY
READ(NUPHAS) (:,(GEOPM(I,J,K),I=1,NUM),J=1,KCOL),K=1,NSIGY
IMISC(5) = NSIGY
C
C RESET THE ALBEDO OF SINGLE SCATTERING (THE SCATTERING TO ATTENUATION
C RATIO) IF DESIRED
C IF(ABSORB.GE.0.1) THEN
ALBESS(I) = TOTALS(I)/TOTALS(1) + ABSORB
ENDIF
§5. PROGRAM 4

\begin{verbatim}
C COMPUTE TOTALS (1) = TOTALS (1) * (1.0/ALBESS(1) - 1.0)
C ENDIF (DBUG .GT. 0) THEN
C IF (IVOP.EQ.1) THEN
C ELSE
C ENDIF
C END

1000 FORMAT (1H1, ' PROGRAM 4 OF THE NATURAL HYDROSOL MODEL', 1H1, ' SOLUTION OF THE RADIATIVE TRANSFER EQUATION IN A PLANE-PARALLEL ', 1H1, ' MEDIUM')
1001 FORMAT (10A8)
1010 FORMAT (1H1, ' THE GRID PARAMETERS ARE', 1H1, ' T5, NMU = ', F5.3, 1H1, ' OTHER PARAMETERS ARE', 1H1, ' DO 200 IY=1, NY
200 VOUT(IY) = ALFA*ZGEO(IY)
END
C WRITE HEADER RECORDS ON THE AMPLITUDE DATA FILE
C REMIND NUOUT TRAP WRITE (I, YOUT(IY)) YOUT(IY)
C WRITE (I, YOUT(IY), YOUT(IY), YOUT(IY), YOUT(IY))
C WRITE (NUOUT) (QFGEOPP(I, J, K), I=1, NMU, J=1, KCOL, K=1, NSIGV)
C RETURN
C END PROGRAM 4

1020 FORMAT (1H1, ' THE PHASE FUNCTION IS QUAD-AVERAGED USING NUQB = ', F5.3, 1H1, ' OTHER PARAMETERS ARE', 1H1, ' DO 200 IY=1, NY
200 VOUT(IY) = ALFA*ZGEO(IY)
END
C FORMATTED STATEMENTS
C 1000 FORMAT (1H1, ' PROGRAM 4 OF THE NATURAL HYDROSOL MODEL', 1H1, ' SOLUTION OF THE RADIATIVE TRANSFER EQUATION IN A PLANE-PARALLEL ', 1H1, ' MEDIUM')
1001 FORMAT (10A8)
1010 FORMAT (1H1, ' THE GRID PARAMETERS ARE', 1H1, ' T5, NMU = ', F5.3, 1H1, ' OTHER PARAMETERS ARE', 1H1, ' DO 200 IY=1, NY
200 VOUT(IY) = ALFA*ZGEO(IY)
END
C WRITE HEADER RECORDS ON THE AMPLITUDE DATA FILE
C REMIND NUOUT TRAP WRITE (I, YOUT(IY)) YOUT(IY)
C WRITE (I, YOUT(IY), YOUT(IY), YOUT(IY), YOUT(IY))
C WRITE (NUOUT) (QFGEOPP(I, J, K), I=1, NMU, J=1, KCOL, K=1, NSIGV)
C RETURN
C END PROGRAM 4
\end{verbatim}
§5. PROGRAM 4

SUBROUTINE ADIPAK(X, Y, IROW, NMUL, L)

C THIS ROUTINE COMPUTES I + X = Y, WHERE I IS THE IDENTITY MATRIX AND
C X AND Y ARE BLOCK MATRICES STORED ON THE PACKED FORMAT OF 12.4.

C DIMENSION X(IROW, 1), Y(IROW, 1)

C MLR = NMUL*(L+1)
C MLC = NMUL*((L+2)/2)
D0 99 I=1,MLR
D0 99 J=1,MLC
99 Y(I,J) = X(I,J)

C ADD 1.0 TO THE DIAGONAL ELEMENTS

LP1 = L+1
DO 100 IB=1,LP1
JXB = (IB+1)/2
I1 = (JXB-1)*NMU
J1 = (JXB-1)*NMU
DO 100 I=1,NMU
100 Y(I+I1,J1+I) = X(I+I1,J1+I) + 1.

C RETURN
END

SUBROUTINE AMPAO

C ON NHM4/AMPAO

C THIS ROUTINE FOURIER ANALYZES THE QUAD-AVERAGED SKY RADIANCES
C RADSKY = NO(A,-) TO GENERATE THE DIRECT BEAM SPECTRAL AMPLITUDE
C AO(A,-). AO(A,-) IS THEN TRANSMITTED
C THROUGH THE UPPER BOUNDARY TO GET AO(X,-). AO(X,-) IS THEN
C ATTENUATED EXPONENTIALLY TO GET AO(Y,-) AT ALL DEPTHS.
C COSINE AMPLITUDES ARE IN ADAM(I), I=1,2,...,NRHAT
SINE AMPLITUDES ARE IN ADAM(I + NRHAT)
C SPECTRAL STORAGE ARRAYS MUST BE LOADED WITH
C THAT1(A,X) IN THAT1
C THAT2(A,X) IN THAT2
C IN THIS ROUTINE, /CAMP/ IS USED TO STORE AO(Y,-), ALE.V.LE.Z

PARAMETER(MXMU=10, MXPHI=24, MXY=30)
PARAMETER(MXL=MXPHI/2, MXRRTH=MXMU*(MXL+1),
MCRTH=MXMU*(MXL+2)/2, MXAMP=2*MXRRT/H)

COMMON/CGRID/ FMU(MXMU), PHI(MXPHI), Y(MXY)
COMMON/CAMP/ ADAM(MXAMP), DUMMY(MXAMP), AOYM(MXAMP, MXY)
COMMON/CRTHAT/ THAT1(MXRTH, MXRTH), THAT2(MXRTH, MXCRTH)
COMMON/CMISC/ IMISC(20)
COMMON/CWORK/ RADSKY(MXMU, MXPHI)

74
§5. PROGRAM 4

NMU = IMISC(1)
NPHI = IMISC(2)
NL = IMISC(3)
NY = IMISC(4)
IDBUG = IMISC(9)
NRHAT = IMISC(10)

IF(IDBUG.EQ.2) THEN
  CALL P2ARAV(THAT1,2*NMU,NMU,MXRRTH,2,'THAT1(A.X) IN AMPAD')
  CALL P2ARAV(THAT2,2*NMU,NMU,MXRRTH,2,'THAT2(A.X) IN AMPAD')
  CALL P2ARAV(RADSKY,NMU,NPHI,MXRRTH,'RADSKY IN AMPAD')
ENDIF

LOOP OVER L AND MU TO DEFINE AO(A,-) VIA (4.8) AND (4.9)

DO 100 I=1,NMU-1

DEFINE THE AMPLITUDES FOR EACH L VALUE FROM (4.8) AND (4.9)

L = 0 SPECIAL CASE
SUM = 0.
DO 310 J=1,NPHI
  310 SUM = SUM + RADSKY(I,J)
AOAM(I) = SUM/FLOAT(NPHI)
AOAM(I+NRHAT) = 0.

L = NL SPECIAL CASE
SUM = 0.
DO 320 J=1,NPHI
  320 SUM = SUM + RADSKY(I,J)*COS(FLOAT(NL)*PHI(J))
AOAM(NMU*NL+I) = SUM/FLOAT(NPHI)
AOAM(NMU*NL+I+NRHAT) = 0.

0 .LT. L .LT. NL GENERAL CASE

DO 330 L=1,NL-1
  330 SUM1 = 0.
  SUM2 = 0.
  DO 332 J=1,NPHI
    332 SUM1 = SUM1 + RADSKY(I,J)*COS(FLOAT(L)*PHI(J))
    SUM2 = SUM2 + RADSKY(I,J)*SIN(FLOAT(L)*PHI(J))
  AOAM(NMU*NL+I) = SUM1/FLOAT(NL)
  AOAM(NMU*NL+I+NRHAT) = SUM2/FLOAT(NL)

100 CONTINUE

POLAR CAP SPECIAL CASE

THE COSINE AMP IS JUST THE VALUE OF THE POLAR CAP QUAD-AVERAGED
RADIANCE, EQ. (5.4)

AOAM(NMU) = RADSKY(NMU,1)
AOAM(NMU+NRHAT) = 0.
DO 340 L=1,NL
  340 AOAM(NMU*L+NMU) = 0.

340 AOAM(NMU*L+NMU+NRHAT) = 0.

TRANSMIT AO(A,+) THROUGH THE UPPER BOUNDARY VIA 6.55 TO GET
AO(A,+) = AOY*M(+,1) (NOTE IN 6.55 THAT AU(A,+) = 0. SEE PAGE 137)

CALL RFMPAK(AOAM,THAT1,AOY,MXRRTH,NMU,NL)
CALL RFMPAK(AOAM(NRHAT+1),THAT2,AOY(NRHAT+1,1),MXRRTH,NMU,NL)

TRANSMIT AO(A,+) TO ALL LOWER Y LEVELS, X .GT. Y .GE. Z, VIA 8.22

IR0W = 0
DO 400 I=1,NY
  400 IR0W = IR0W + 1
  DO 400 IV=2,NY
    TEP = EXP((V(IV)-V(IV))/FMU(J))
    AOYM(IR0W,IV) = TEMP*AOY*M(IR0W,IV)
  400 AOYM(IR0W+NRHAT,IV) = TEMP*AOY*M(IR0W+NRHAT,IV)

RETURN
END
SUBROUTINE AMPAP
ON NHM4/AMPAP
THIS ROUTINE DEFINES A(A,\(*\)) USING 6.56
SPECTRAL STORAGE ARRAYS MUST BE LOADED WITH
THAT1(X,A) IN THAT1
THAT2(X,A) IN THAT2
RHAT1(A,X) IN RHAT1
RHAT2(A,X) IN RHAT2
PARAMETER(MXMU=10, MXPHI=24, MXY=30)
PARAMETER(MXL=MXPHI/2, MXRRTH=MXMU*(MXL+1),
1 MXCRTH=MXMU*(MXL+2)/2, MXAMP=2*MXRRTH)
COMMON/CAMP/ AOAM(MXAMP), AAP(MXAMP), AVM(MXAMP, MXY), AYP(MXAMP, MXY)
COMMON/CRTHAT/ THAT1(MXRRTH, MXCRTH), THAT2(MXRRTH, MXCRTH),
1 RHAT1(MXRRTH, MXCRTH), RHAT2(MXRRTH, MXCRTH)
COMMON/CMISC/ IMISC(20)
COMMON/CWORK/ AGAP(MXAMP), TEMP1(MXRRTH), TEMP2(MXRRTH),
1 RHAT(MXRRTH, MXCRTH), THAT(MXRRTH, MXCRTH)
DIMENSION AXP(MXAMP)
EQUIVALENCE (AXP(1), AYP(1,1))
NMX = IMISC(1)
NL = IMISC(3)
IDBUG = IMISC(9)
NRHAT = IMISC(10)
NCHAT = IMISC(11)
P = 1 (COSINE AMPLITUDES)
IP = 1
999 CONTINUE
IPOFF = IP*RHAT*(IP-1)
LOAD THATP(X,A) INTO THAT
LOAD RHATP(A,X) INTO RHAT
IF(IP.EQ.1) THEN
DO 800 J=1,NCHAT
DO 800 I=1,NRHAT
THAT(I,J) = THAT1(I,J)
800 THAT(I,J) = THAT1(I,J)
ELSE
DO 802 J=1,NCHAT
DO 802 I=1,NCHAT
RHAT(I,J) = RHAT2(I,J)
802 RHAT(I,J) = RHAT2(I,J)
ENDIF
IF(IDBUG.GE.2) THEN
WRITE(6,310) :P
CALL P2ARAY(THAT,2*NMX, NL, NMU, RHAT, 2, 'THATP(X,A) IN AMPAP')
CALL P2ARAY(RHATP,2*NMX, NL, NMU, RHAT, 2, 'RHATP(A,X) IN AMPAP')
ENDIF
EVALUATE 6.5b AND SAVE AOP FOR WRITING INTO NUOUT
CALL RFMPA(AAP,IPOFF+1), THAT, TEMP1, MXRRTH, NMU, NL)
CALL RFMPA(AAP,IPOFF+1), RHAT, TEMP2, MXRRTH, NMU, NL)
GO 100
100 AAP(IP+IPOFF) = TEMP1(IP) + TEMP2(IP)
IF(IP.GT.1) RETURN
REPEAT FOR P = 2 (SINE AMPLITUDES)
IP = 2
GO TO 999
310 FORMAT(1H1, ' SUBROUTINE AMPAP, P=', IP,12)
END
SUBROUTINE AMPINT
C
ON NHM4/AMPINT
C
THIS ROUTINE FINDS THE AMPLITUDES A(Y,-;) AND A(Y,+) AT ALL INTERIOR
DEPTHS X .LT. Y .LE. Z USING 7.6 AND 7.7.
C
PARAMETER(MXMU=10, MXPHI=24, MXY=30)
PARAMETER(MXL=MXPHI/2, MXAMP=2*MXMU*(MXL+1))
C
COMMON/CAMP/ AAM(MXAMP),AAP(MXAMP),AYM(MXAMP, MXY), AYP(MXAMP, MXY)
COMMON/CRTR/ RYX(MXMU, MXMU, MXY),TXY(MXMU, MXMU, MXY),
1 RYB(MXMU, MXMU, MXY),RYB(MXMU, MXMU, MXY)
COMMON/CMI SC/ IMISC(20)
COMMON/CWORK/ TXVB(MXMU, MXMU), TEMP1(MXMU, MXMU), TEMP2(MXMU, MXMU),
1 RPVB(MXMU, MXMU, MXY)
C
DIMENSION AXM(MXAMP)
EQUIVALENCE (AXM(1), AYM(1,1))
C
DATA IDGT/10/, NUSCR1, NUSCR2, NUSCR3/43, 46, 47/
C
NMU = IMISC(1)
NL = IMISC(3)
NY = IMISC(4)
1DBUG2 = IMISC(9)
NRHAT = IMISC(10)
C
REWIND NUSCR1
REWIND NUSCR2
REWIND NUSCR3
C
DO 100 L=1, NL
C
1 IF(IDBUG2.EQ.2 .OR. L.GE.NL) THEN
2 IDBUG = 2
3 ELSE
4 IDBUG = 0
5 ENDIF
ELSE
6 IDBUG = IDBUG2
7 ENDIF
8 LUPSET = NMU*L
C
READ IN RXY = R(Y,X,L), TXY = T(L,Y,X) AND RPVB = RP(Y,V,L) FOR ALL
C
Y LEVELS FOR THIS L VALUE.
C
DO 300 IY=1, NY
READ(NUSCR1) RXY(IY, IY, 1, 1, 1, NMU, L-1, 1, NMU)
READ(NUSCR2) TXY(IY, IY, 1, 1, 1, NMU, L-1, 1, NMU)
READ(NUSCR3) RPV(IY, IY, 1, 1, 1, NMU, L-1, 1, NMU)
ENDIF
C
IF(IDBUG.EQ.2) THEN
WRITE(3, 300)
C
CALL PARA(VXV, VM, NML, NV, MXMU, MXY, 2, RH(Y,K,L))
CALL PARA(TXY, VXV, XM, NV, MXMU, MXY, 2, RH(Y,K,L))
CALL PARA(RPB, VXV, XM, NV, MXMU, MXY, 2, RH(Y,K,L))
ENDIF
C
77


§5. PROGRAM 4

\[
\text{IOFSET} = \text{LOFSET} \times \text{NRHAT}
\]

DO 206 IV = 1, NY
DO 206 J = 1, NMU
DO 206 I = 1, NMU
206 RPXB(I, J, IV) = R2YB(I, J, IV)
ENDIF

C COMPUTE THE AMPLITUDES AT EACH Y LEVEL
C
DO 102 IV = 2, NY
IF(IBMBUG.EQ.2) WRITE(6,311) IV, IP
C
COMPUTE TXYB = TP(X,Y,B,L) USING b.33
C
COMPUTE TEMP1 = 1 - RP(Y,B) * R(Y,X)
DO 210 I = 1, NMU
DO 210 J = 1, NMU
SUM = 0.
DO 211 K = 1, NMU
211 SUM = SUM + RPXB(I, K, IV) * RXY(K, J, IV)
ENDIF
IF-Bold.EQ.2 CALL P2ARRAY(TEMP1, NMU, NMU, MAXM, 2.
1 'I - RP(Y,B) * R(Y,X)')
C
INVERT 1 - RP(Y,B) * R(Y,X)
CALL LINV1F(TEMP1, NMU, MAXM, TEMP2, IDBUG, TXYB, IER)
IF(IBMBUG.EQ.2) CALL P2ARRAY(TEMP2, NMU, NMU, MAXM, 2.
1 'I - RP(Y,B) * R(Y,X)') INVERSE'
C
COMPUTE SCRIPT TP(X,Y,B,L)
DO 220 I = 1, NMU
DO 220 J = 1, NMU
SUM = 0.
DO 221 K = 1, NMU
221 SUM = SUM + TXY(I, K, IV) * TEMP2(K, J)
ENDIF
IF-Bold.EQ.2 CALL P2ARRAY(TMP1, NMU, NMU, MAXM, 2.
1 'SCRIPT TP(X,Y,B,L)')
C
COMPUTE AP(Y,+) USING 7.6
C
DO 240 J = 1, NMU
SUM = 0.
DO 231 I = 1, NMU
231 SUM = SUM + AYM(I+IOFSET) * TXYB(I, J)
230 AYM(I+IOFSET, IV) = SUM
C
COMPUTE A(Y,+) USING 7.7
C
DO 240 J = 1, NMU
SUM = 0.
C
DO 241 I = 1, NMU
241 SUM = SUM + AYM(I+IOFSET, IV) * RPXB(I, J, IV)
240 AYM(I+IOFSET, IV) = SUM
C
102 CONTINUE
C
REPEAT FOR THE MANE AMPLITUDES, IP = 2
C
IP = IP + 1
IF(IP.LT.3) GO TO 999
C
100 CONTINUE
C
IF(IBMBUG.EQ.2) THEN
CALL P2ARRAY(AYM, 2*NRHAT, NY, MAXM, 2. 'A(Y,+)' )
CALL P2ARRAY(AYM, 2*NRHAT, NY, MAXM, 2. 'A(Y,+)')
ENDIF
310 FORMAT(1H1, ' SUBROUTINE AMPI NT, L = ', I3)
311 FORMAT(1H1, ' Y = ', 12X, ' IP = ', I2)
C
RETURN
END

78
SUBROUTINE AMPX

ON NHM4/AMPX

THIS ROUTINE COMPUTES A(X,-) AND A(X,+) USING 7.3 AND 7.5B

SPECTRAL STORAGE ARRAYS MUST BE LOADED WITH

THAT1(A,X) IN THAT1
THAT2(A,X) IN THAT2
RHAT1(X,A) IN RHAT1
RHAT2(X,A) IN RHAT2

PARAMETER(MXMU=10, MXPHI=24, MXV=30)
PARAMETER(MXL=MXPHI/2, MXRTH=MXMU*(MXL+1),
1 MXCRTH=MXMU*((MXL+1)/2), MXAMP=2*MXRTH)

COMMON/CAMP/AOMP(MXAMP), AYMP(MXAMP, MXV), AYP(MXAMP, MXV)
COMMON/CRTHAT/THAT1(MXRTH, MXCRTH), THAT2(MXRTH, MXCRTH),
1 RHAT1(MXRTH, MXCRTH), RHAT2(MXRTH, MXCRTH)
COMMON/LMISC/IMISC(20)
COMMON/WORK/ WORK(MXMU, MXMU), TEMP(MXRTH, MXCRTH),
1 RHAT(MXRTH, MXCRTH), THAT(MXRTH, MXCRTH), RXBL(MXMU, MXMU, 0:MXL)
2 RXBL(MXMU, MXMU, 0:MXL), RXBL(MXMU, MXMU, 0:MXL)

DIMENSION AXM(MXAMP), AXP(MXAMP)

EQUIVALENCE (AXM(1), AYM(1,1)), (AXP(1), AYP(1,1))

NMU = IMISC(1)
NL = IMISC(3)
NV = IMISC(4)
KINV = IMISC(8)
IDBUG = IMISC(9)
NRHAT = IMISC(10)
ORTHAT = IMISC(11)
NUSCR3 = IMISC(15)
IEVEN = (NL+2)/2
IODD = (NL+1)/2

READ RI(X,B,L) AND R2(X,B,L) FROM SCRATCH FILE NUSCR3

REWIND NUSCR3
DO 100 I=0,NL

READ R1(NUSCR3)
DO 100 I=0,NL

READ R2(NUSCR3)
ENDDO

DO 100 I=0,NL

IF (IDBUG.EQ.2) THEN
CALL P3ARRAY(R1XBL, NMU, NMU, NL+1, MAMU, MAMU, 0:R1(X,B,L) IN AMPX*)
CALL P3ARRAY(R2XBL, NMU, NMU, NL+1, MAMU, MAMU, 0:R2(X,B,L) IN AMPX*)
ENDIF

INITIALIZE FOR I = 1 (COSINE AMPLITUDES)

IP = 1
IPSET = 0

999 CONTINUE

LOAD RHAT1(A,X) INTO RHAT AND RH1(X,B,L) INTO RXBL

IF (IP.EQ.1) THEN
DO 800 J=1,NMU
DO 800 I=1,ORTHAT
800 RHAT(I,J) = RHAT1(I,J)
DO 800 I=0,NL
DO 800 J=1,NMU
DO 800 I=1,ORTHAT
ELSE
801 RXBL(I,J,L) = R1XBL(I,J,L)
END
§5. PROGRAM 4

```
DO 802 J=1,NCHAT
DO 802 I=1,NRHAT
802 RHAT(I,J) = RHAT2(I,J)
DO 803 L=0,NL
DO 803 J=1,NMU
DO 803 I=1,NMU
803 RXBL(I,J,L) = R2XBL(I,J,L)
ENDIF
IF(IDBUG.EQ.2) THEN
   WRITE(6,901) IP
   CALL P2ARRAY(RHAT,2*NMU,NMU,MXRRTH,2, 'RHATP(X,A) IN AMPX')
ENDIF

C COMPUTE TEMP = RP(X,B) * RHATP(X,A) AS NMU BY NMU BLOCKS
C
DO 200 L=0,NL
  IROFF = NMU*L
  NCOL = 100D
  IF(MOD(L,2).EQ.0) NCOL = IEVEN
DO 200 IR=1,NCOL
C
EXTRACT AN NMU BY NMU BLOCK FROM RHATP
  ICOFF = NMU*(IR-1)
DO 210 I=1,NMU
DO 210 J=1,NMU
210 RXBL(I,J,L) = RXBL(I,J,L)
ENDIF

C MULTIPLY RP(X,B,L) TIMES THIS BLOCK AND STORE THE RESULT IN RHAT
C
DO 212 I=1,NMU
DO 212 J=1,NMU
   SUM = 0.
   DO 214 K=1,NMU
       SUM = SUM + RXBL(I,K,L)*WORK(K,J)
214 WORK(I,J) = RXBL(I,J,L)
   ENDIF
212 RHAT(I*IROFF,J+ICOFF) = SUM
200 CONTINUE

C RHAT NOW CONTAINS RP(X,B) * RHATP
C
C COMPUTE THE INVERSE FOR (7.3) USING THE APPROXIMATION (7.4)
C
(1 + X) INVERSE = I + X + X**2 + ... + X**KINV
C
IF(IDBUG.EQ.2) CALL P2ARRAY(RHAT,2*NMU,NMU,MXRRTH,2, 'RHATP(X,A) IN AMPX')
CALL ADIPAK(RHAT,TEMP,MXRRTH,NMU,NL)
DO 250 K=2,KINV
   CALL FFMPAK(TEMP,RHAT,MXRRTH,NMU,NL,WORK)
250 CALL ADIPAK(THAT,TEMP,MXRRTH,NMU,NL)
C
TEMP NOW CONTAINS THE INVERSE
C
IF(IDBUG.EQ.1) THEN
DO 804 I=1,NKHA-T
DO 804 J=1,NKHA-T
804 RHAT(I,J) = -RHAT(I,J)
ENDIF
ENDIF
```

C LOAD THATP(A,X) INTO THAT AND COMPUTE SCRIPT T(A,X,B) BY 6.33
C
IF(IP.EQ.1) THEN
   DO 810 J=1,NKHA-T
   DO 810 I=1,NKHA-T
810 THAT(I,J) = THAT(I,J)
ELSE
   DO 812 J=1,NKHA-T
   DO 812 I=1,NKHA-T
812 THAT(I,J) = THAT2(I,J)
ENDIF
ENDIF
IF(IDBUG.EQ.2) THEN
   CALL P2ARRAY(THAT,2*NMU,NMU,MXRRTH,2, 'THATP(A,X) IN AMPX')
   CALL P2ARRAY(THAT,2*NMU,NMU,MXRRTH,2, (1 - WP*RHATP) INVERSE')
ENDIF
§5. PROGRAM 4

CALL FFMPAK(THAT, TEMP, RHAT, MXRRTH, NMU, NL, WORK)
IF(IDBUG.EQ.2) CALL P2ARRAY(RHAT, 2*NNU, NMU, MXRRTH, 2,
1 'SCRIPT T(A,X,B)')

RHAT NOW CONTAINS SCRIPT T(A,X,B)

COMPUTE AP(X,-) BY (7.3)

CALL RFMPAK(AOAM(IOFSET+1), RHAT, AXM(IOFSET+1), MXRRTH, NMU, NL)

COMPUTE AP(X,+) BY (7.58)

DO 400 L=0, NL
DO 400 I=1, NMU
SUM = 0.
DO 402 K=1, NMU
  SUM = SUM + AXM(K, IOFSET+1)*RXBL(K, I)
400 AXP(I+IOFSET) = SUM

IF(IDBUG.GE.2) THEN
  CALL P2ARAY(AOAM(IOFSET*1), 1, NRHAT, 1, 2, AO(A,-))
  CALL P2ARAY(AXM(IOFSET+1), 1, NRHAT, 1, 2, A(X,-))
  CALL P2ARAY(AXP(IOFSET-1), 1, NRHAT, 1, 2, A(X,+))
ENDIF

IF(IP.EQ.1) RETURN

REPEAT FOR P = 2 (SINE AMPLITUDES)
IP = 2
IOFSET = NRHAT
GO TO 999

901 FORMAT(1H1, 'SUBROUTINE AMPX: P = ', I2)

END

SUBROUTINE BOTMBC(L)

ON NHM4/BOTMB:

THIS ROUTINE COMPUTES THE DISCRETE SPECTRAL RHATZB = RHAT1(Z, B, L)
FOR THE DESIRED BOTTOM BOUNDARY CONDITION.

IF IBOTM = 0, USE 3.26, 5.47, 5.50, 5.51 AND 5.53
IF IBOTM = 1, SET UP AND SOLVE THE EIGENVALUE PROBLEM 10.2
AND THEN USE 10.8 AND 10.9 FOR RHAT(Z, INFINITY)

PARAMETER(MXMU=ID, MXPHI=24, MXY=30)

COMMON/CBOTBC/ RHATZB(MXMU, MXMU)
COMMON/CGRID/ FMU(MXMU), PHI(MXPHI), Y(MXY), BNDMU(MXMMU),
1 BNDPHI(MXPHI), OMEGA(MXMU)
COMMON/CMISC/ IMISC(20), FMISC(20)

NMU = IMISC(1)
NPHI = IMISC(2)
IBOTM = IMISC(12)
RMINUS = FMISC(14)
CONST = RMINUS/FMISC(1)

81
§5. PROGRAM 4

IF(IBOTM.EQ.0) THEN
  IF(L.EQ.0) THEN
    RHAT ELEMENT BY 5.53, RHAT ARRAY BY 5.50
    DO 100 IR=1,NMU
    VAL = FLOAT(NPHI)*CONST*FMU(IR)*OMEGA(IR)
    C SPECIAL CASE FOR POLAR CAP QUAD
    IF(IR.EQ.NMU) VAL = CONST*FMU(NMU)*OMEGA(NMU)
    DO 100 IU=1,NMU
    100 RHATZB(IR,IU) = VAL
  ELSE
    FOR MATTE BOTTOM AND L.GT.U, RHAT1(Z,B) = 0 BY 5.53
    DO 102 IR=1,NMU
    DO 102 IU=1,NMU
    102 RHATZB(IR,IU) = 0.
  ENDIF
ELSE
  SET UP AND SOLVE THE EIGENVALUE PROBLEM FOR R(INFINITY,L)
  EIGENR SETS RHATZB R(INFINy,L)
  IF RUNS ARE BEING MADE WITH A MATTE BOTTOM ONLY, THE CALL TO EIGENR
  CAN BE COMMENTED OUT TO PREVENT LOADING THE LARGE IMSL ROUTINES IT CALLS
  CALL EIGENR(L)
ENDIF
RETURN
END

SUBROUTINE DRTAB(NRTAB,YNOW,RT,DERIV)

ON NNM4/DRTAB

THIS SUBROUTINE EVALUATES DERIV = D(RT)/DY AT Y = YNOW (THE RIGHT
HAND SIDE OF 6.43, 6.44 AND 6.48) FOR USE BY THE IMSL ROUTINE DVERK
RECALL THAT, RYX AND TXY ARE STORED IN RT:
RYX(I,J) = RT(I + (J-1)*NIJ)
TXY(I,J) = RT(I + (J-1)*NIJ + NIJ*NIJ)
PARAMETER(MXMU=10, MXSIGY=3)
REAL RT(NRTAB),DERIV(NRTAB)
DIMENSION WORK(MXMU,MXMU),PHCY(MXMU,MXMU),TAUV(MXMU,MXMU)
COMMON/CRTSIG/RHOHAT(MXMU,MXMU,MXSIGY),TAUHAT(MXMU,MXMU,MXSIGY)
COMMON/CSIGY/YSIG(MXSIGY)
COMMON/CMISC/IMISC(20)
NMU = IMISC(1)
NSIGY = IMISC(5)
IDE = IMISC(13)
NSQ = NMU*NMU

82
§5. PROGRAM 4

DETERMINE RHOHAT AND TAUHAT AT THE CURRENT Y VALUE

IF(NSIGY.EQ.1 .OR. YNOW.LE.YSIG(1)) THEN
  THE WATER IS UNIFORM, OR YNOW IS AT OR ABOVE THE FIRST DEPTH WHERE SIGMA IS KNOWN

DO 50 J=1,NMU
  DO 50 I=1,NMU
    RHOV(I,J) = RHOHAT(I,J,1)
    TAUV(I,J) = TAUHAT(I,J,1)
  ENDIF

ELSEIF(YNOW.GE.YSIG(NSIGY)) THEN
  YNOW IS AT OR BELOW THE LAST DEPTH WHERE SIGMA IS KNOWN

DO 52 J=1,NMU
  DO 52 I=1,NMU
    RHOV(I,J) = RHOHAT(I,J,NSIGY)
    TAUV(I,J) = TAUHAT(I,J,NSIGY)
  ENDIF

ELSE
  DEFINE RHOMAT AND TAUHAT BY LINEAR INTERPOLATION OF THE VALUES FROM THOSE DEPTHS WHERE SIGMA IS KNOWN

DO 55 JY=2,NSIGY
  IF(YNOW.LT.YSIG(JY)) GO TO 56
ENDIF

56 DY = (YNOW - YSIG(JY-1))/(YSIG(JY) - YSIG(JY-1))

DO 58 J=1,NMU
  DO 58 I=1,NMU
    RHOV(I,J) = (1.0 - DY)*RHOHAT(I,J,JY-1) + DY*RHOHAT(I,J,JY)
    TAUV(I,J) = (1.0 - DY)*TAUHAT(I,J,JY-1) + DY*TAUHAT(I,J,JY)
  ENDIF

END

COMPUTE K = TAUV + RHOV*RYX

DO 100 J=1,NMU
  DO 100 I=1,NMU
    WORK(I,J) = TAUV(I,J)
    DO 101 K=1,NMU
      TEMP1 = TEMP1 + RT(I,J)*WORK(K,J)
    ENDIF
  ENDIF
  DERIV(I,J) = TEMP1*(J-1)*NMU
END

ELSE
  CHANGE OF SIGN TO GET EQ. 6.48

DO 700 I=1,NRTAB
  DERIV(I) = -DERIV(I)
ENDIF

RETURN
END
SUBROUTINE EIGENR(L)
C
ON NHM4/EIGENR
C
THIS ROUTINE SETS UP AND SOLVES THE EIGENMATRIX PROBLEM KE = EL
AS DESCRIBED IN SECTION 10.
C
THE SUBMATRICES EP = E(+) AND EM = E(-) ARE EXTRACTED, AND
R(INFINITY,L) = -E(-) * E(+) INVERSE IS COMPUTED.
C
THE ASYMPTOTIC RADIANCE DISTRIBUTION AND ASSOCIATED QUANTITIES
ARE ALSO FOUND USING FORMULAS FROM TECH MEMO ERL-PMEL-76.
C
IF L = 0, THE FULL RHO AND TAU MATRICES ARE USED TO DEFINE K
IF L.GT.0, ROW NMU AND COLUMN NMU OF RHO AND TAU IS ZERO, AND
THUS IS OMITTED FROM K (SEE PAGE 174)
C
PARAMETER(MXMU=10, MXPHI=24, MXY = 30, MXSIGY=3)
PARAMETER(MXMU2=2*MXMU, XMUSQ=MXMU*MXMU)
C
DIMENSION IP(MXMU2), EVALS(MXMU2), EIGv(MXMU)
COMPLEX WEV(MXMU2)
C
COMMON/CGRID/ FMU(MXMU), PHI(MXPHI), YOUT(MXY), BNDMU(MXMU),
1 BNDPHI(MXPHI), OMEGA(MXMU), DELTMU(MXMU)
COMMON/CRSIG/ RH0HAT(MXMU,MXMU, MXSIGY), TAUHAT(MXMU,MXMU, MXSIGY)
COMMON/CSIGY/ YSIG(MXSIGY), ALBESS(MXSIGY), TOTALS(MXSIGY)
COMMON/CDBTBL/ RHATZB(MXMU,MXMU)
COMMON/CMISC/ IMISC(20)
COMMON/CWWORK/ WERK(MXMUSQ,12), RPINF(MXMU), RMINF(MXMU), WORK(1)
C
ARRAY WORK(1) MUST HAVE 4*NMU*(NMIU+1) WORDS AVAILABLE
C
DIMENSION FP(MXMU,MXMU), FM(MXMU, MXY), TEM1(MXMU, MXMU),
1 TEM2(MXMU, MXY), EMNV(MXMU, MXMU)
C
DIMENSION AK(MXMU2, MXMU2), EMK(MXMU2, MXMU2), EP(MXMU, MXMU),
1 EM(MXMU, MXY), EPNV(MXMU, MXMU)
C
COMPLEX ZEV(MXMU2, MXMU2)
C
EQUIVALENCE (WERK( 1, 1), AK( 1, 1)), (WERK( 1, 5), ZEV( 1, 1))
EQUIVALENCE (WERK( 1, 1), EMK( 1, 1)), (WERK( 1, 5), EP( 1, 1)),
1 (WERK( 1, 6), EMV( 1, 1)), (WERK( 1, 7), EPNV( 1, 1))
C
NMU = IMISC(1)
NSIGY = IMISC(5)
IDBUG = IMISC(9)
ALPHA = TOTALS(NSIGY)/ALBESS(NSIGY)
C
DETERMINE THE ARRAY SIZES
IF(L.EQ.0) THEN
M = NMU
ELSE
M = NMU - 1
ENDIF
M2 = 2*M
C
C
INITIALIZE THE K MATRIX, USING 5.21 OR 5.24 IN 6.8
C
DO 100 I=1,M2
DO 100 J=1,M2
IF(J.LE.M) INI:
IF(J.LE.M) AK(I,J) = -TAUHAT(I,J,NSIGY)
IF(M.LT.J) AK(I,J) = RH0HAT(I,J-1,NSIGY)
ELSE
IF(J.LE.M) AK(I,J) = -RH0HAT(I,J-1,M,NSIGY)
IF(M.LT.J) AK(I,J) = TAUHAT(I-1,J,M,NSIGY)
ENDIF
100 CONTINUE
C
CALL EIGRF(AK,M2, MXMU2, 2, Wev, ZEv, XMUSQ, WORK, IER)
IF(IDBUG.GT.1) THEN
WRITE(6,301) (WEV(I), I=1,M2)
WRITE(6,304) WORK(1)
ENDIF
C
84
§5. PROGRAM 4

C
C SORT POSITIVE EIGENVALUES
C
KP0S = 0
DO 600 I=1,M2
TMP = WEV(1)
IF(TMP.LT.0.) GO TO 600
KP0S = KP0S + 1
EVALS(KP0S) = TMP
600 CONTINUE
C
CALL VSRTA(EVALS,KP0S)
C DEFINE ORDERED EIGENVALUES
DO 601 I=1,KP0S
EIGV(I) = EVALS(I)
601 CONTINUE
C
CONSTRUCMT PERMUTATION MATRIX IP BY COMPARING WEV AND EVALS
DO 610 I=1,M2
TMP = WEV(I)
DO 610 J=1,M2
IF(ABS(EVALS(J) - TMP).LT.1.E-8) IP(I) = J
610 CONTINUE
IF(IDBUG.GT.1) WRITE(6,681) (J,IP(J),J=1,M2)
IF(L.EQ.O OR .IDBUG.GT.L) WRITE(6,680) L,(EIGV(I),ALPHA*EIGV(I), 1 I=1,M)

C DEFINE REAL, ORDERED EIGENVECTOR MATRIX EM
DO 620 J=1,M2
JJ = IP(J)
DO 620 I=1,M2
EM(I,J) = ZEV(I,J)
DO 620 JJ=1,J
DO 620 I=1,M2
EM(I,J) = EM(I,J)
DO 620 J=1,M2
EM(I,J) = -EM(I,J)
620 CONTINUE
IF(L.EQ.O OR .IDBUG.GT.L) WRITE(6,682) L,IP(J)
IF(I-RUM.L.EQ.1) WRITE(6,682) L,IP(J)
END IF
C INVERT E(+) AND DEFINE R(INFINI.), USING 10.8 OR 10.9
C
IDGT = 6
CALL LINv2F(EP,M,MXMU,EPNV,IDGT,WORK,IER)
CALL VMULFF(EM,M,M,MXMU,PHATZB,MXMU,IER)
C
FILL THE LAST -5W AND LAST COLUMN OF RHAT(Z,B) WITH ZEROS IF
C L.GT.0
IF(L.GT.0) THEN
DO 649 I=1,NM:
RHATZBI(NMU,I) = 0.
649 CONTINUE
END IF
C
CONSTRUCT THE F(+) AND F(-) MATRICES AND GET THE ASYMPOTIC
C RADIANCE DISTRIBUTION USING 76/18.2. NOTE THAT RADINF(+I) IS OBTAINED
C FROM F(+) AND THAT RADINF(-) IS OBTAINED FROM F(-).
C
IF(L.EQ.0) THEN
EM = -E(-)
ELSE IF(L.EQ.1) THEN
DO 652 J=1,NM:
EM(J) = -EM(J)
652 CONTINUE
ENDIF
§5. PROGRAM 4

IDGT = 6
CALL LINV2F(EM, NMU, XMUMU, EPMV, IDGT, WORK, IER)
IF (IDBUG.GT.1) THEN
    CALL P2ARAY(EPNV, NMU, NMU, XMUMU, 2, 'E(+)')
    CALL P2ARAY(EMNV, NMU, NMU, XMUMU, 2, 'E(-)')
ENDIF
CALL VMULFF(EM, EPMV, NMU, NMU, XMUMU, XMUMU, TEMP1, XMUMU, IER)
CALL VMULFF(TEMP1, EM, NMU, NMU, XMUMU, XMUMU, TEMP2, XMUMU, IER)
DO 650 I = 1, NMU
    DO 650 J = 1, NMU
        TEMP1(I, J) = EM(I, J) - TEMP2(I, J)
    END DO
    IDGT = 6
CALL LINV2F(TEMP1, NMU, XMUMU, FP, IDGT, WORK, IER)
CALL VMULFF(EM, EPMV, NMU, NMU, XMUMU, XMUMU, TEMP1, XMUMU, IER)
CALL VMULFF(TEMP1, EM, NMU, NMU, XMUMU, XMUMU, TEMP2, XMUMU, IER)
DO 651 I = 1, NMU
    DO 651 J = 1, NMU
        TEMP1(I, J) = EM(I, J) - TEMP2(I, J)
    END DO
    IDGT = 6
CALL LINV2F(TEMP1, NMU, XMUMU, FM, IDGT, WORK, IER)
C
C NORMALIZE THE NADIR ASYMPTOTIC RADIANCE TO ONE
ANORM = 1.0/FP(1, NMU)
WRITE (6, 655)
DO 656 I = 1, NMU
    RPINF(I) = ANORM*FM(1, I)
    RMINF(I) = ANORM*FP(1, I)
END DO
WRITE(6, 657) I, RMINF(I), RPINF(I)
C
C USE THE ASYMPTOTIC RADIANCE DISTRIBUTION TO GET THE ASYMPTOTIC
C D+, D-, R-, EPS AND EPS-
C
C ACCUMULATE IRRADIANCE SUMS
SHP = 0.
SHM = 0.
CHP = 0.
CHM = 0.
DO 670 I = 1, NMU
    DMU = DELTMU(I)
    SHP = SHP + RPINF(I)*DMU
    SHM = SHM + RMINF(I)*DMU
    CHP = CHP + RPINF(I)*FMU(I)*DMU
    CHM = CHM + RMINF(I)*FMU(I)*DMU
END DO
DPINF = SHP/CHP
DMINF = SHM/CHM
K(INFINITY) BY 76/19.2
FKINF = ALPHA*E**(1/12)
R(INFINITY) BY 76/19.5
ABSORB = ALPHA - TOTALS(1)
RINF = (FKINF - ABSORB*DMINF)/(FKINF + ABSORB*DPINF)
CALL EPSINF(RPINF, RMINF, EPSP, EPSM)
WRITE(6, 672) DPINF, DMINF, RINF, EPSP, EPSM
C
C ENDIF
C
C RETURN
C
C FORMATS
C
301 FORMAT(/// THE COMPLEX EIGENVALUES OF K ARE /// (1P2E25.15))
304 FORMAT(/// THE PERFORMANCE INDEX IS /// E15.5)
655 FORMAT(/// THE SHAPE OF THE ASYMPTOTIC RADIANCE DISTRIBUTION IS GI
VEN BY /// 1 RADINF(-) RADINF(+))
657 FORMAT(/// 14, 1PE15.4)
672 FORMAT(/// OTHER ASYMPTOTIC VALUES ARE /// D*(INFINITY) = ',
1 F7.4/ 1 D*(INFINITY) = ', F7.4/ R-(INFINITY) = ', E1PE14.7/
7 EPS+(INFINITY) = ', DPFB.5/ EPS-(INFINITY) = ', FB.5
680 FORMAT(/// THE ORDERED POSITIVE EIGENVALUES OF K(L= .12,
1 I') ARE /// NODIMEN PER METER /// (1P2E15.6))
681 FORMAT(/// J 1P /// (125))
C
END
SUBROUTINE EPSINF(RPINF,RMINF, EPSP,EPSM)
C
ON NHM4/ EPSINF
C
THIS ROUTINE COMPUTES THE ASYMPTOTIC BACKSCATTER ECCENTRICITIES
EPSILONB(+ ) AND EPSILONB(-) USING (8.15A) AND THE NORMALIZED
ASYMPTOTIC RADIANCE DISTRIBUTION.
C
PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
PARAMETER(MXL=MXPHI/2, MXGEOP=MXMU*(M XL+1))
C
DIMENSION RPINF(MXMU), RMINF(MXMU)
C
COMMON/CGRID/ FMU(MXMU), PHI(MXPHI), Y(MXY), BNDMU(MXMU),
1 BNDPHI(MXPHI), OMEGA(MXMU), DELTMU(MXMU), ZGEO(MXY)
1 COMMON/CRTSIG/ RHOHAT(MXMU, MXMU, MXSIGY), TAUHAT(MXMU, MXMU, MXSIGY),
1 GEOF(MXMU, M XGEOP, MXSIGY), GEOPM(MXMU, M XGEOP, MXSIGY).
C
COMMON/CMISC/ IMISC(20), FMISC(20)
C
NMU = IMISC(1)
NPHI = IMISC(2)
NSIGY = IMISC(5)
NOPI = NPHI/2
TWOPI = 2.0*FMISC(1)
C
SHPINF = 0.
SHMINF = 0.
EPSP = 0.
EPSM = 0.
C
DO 100 I=1,NMU
C
ACUMULATE SCALAR IRRADIANCE SUMS
SHPINF = SHPINF + RPINF(I)*DELTMU(I)
SHMINF = SHMINF + RMINF(I)*DELTMU(I)
C
QUV = OMEGA(I).
IVMAX = NPHI
IF(I.EQ.NMU) I.MAX = 1
DO 100 IV=1,IVMAX
C
DO 100 IR=1,NMU
RP = RPINF(IR)
RM = RMINF(IR)
ISMAX = NPHI
IF(IR.EQ.NMU) ISMAX = 1
DO 100 IS=1,NMAX
C
COMPUTE THE STORAGE INDEX FOR P-(R,J,.) BY 12.7
IVS = IABS(IS-1)
IF(IR.EQ.NMU, THEN
KCOL = IJ
ELSE IF(IU.EQ.NMU) THEN
KCOL = NNU
ELSE IF(IS.LE.NOPI) THEN
KCOL = IJ + NMU.*IVS
ELSE
KCOL = IJ + NMU.*NPHI + MUH(IVS,NOPI))
ENDIF
ENDIF
C
PM = GEOPM(IR,KCOL,NSIGY)
C
EPSP = EPSP + QUVRP*PM
EPSM = EPSM + QUVRM*PM
100 EPSP = EPSP/(TWOP1*SHPINF)
EPSM = EPSM/(TWOP1*SHMINF)
C
RETURN
END
FUNCTION FALPHA(Y)
ON NHM4/FALPHA
GIVEN AN OPTICAL DEPTH V, THIS FUNCTION RETURNS THE VALUE OF
1.0/ALPHA(Y), WHERE ALPHA IS THE ATTENUATION COEFFICIENT. FOR
USE IN INTEGRATING DV/ALPHA(Y) TO GET GEOMETRIC DEPTHS (SEE
SUBROUTINE Y2ZGEO).
PARAMETER(MXSIGY=3)
COMMON/CSSIGY/ YSIG(MXSIGY), ALBESS(MXSIGY), TOTALS(MXSIGY)
COMMON/CMISC/ IMISC(20)
NSIGY = IMISC(5)
IF(NSIGY.EQ.1 OR. V.LE.YSIG(1)) THEN
ALPHA1 = TOTALS(1)/ALBESS(1)
ELSEIF(Y.GT.YSIG(NSIGY)) THEN
ALPHA1 = TOTALS(NSIGY)/ALBESS(NSIGY)
ELSE
DO 55 JY=2,NSIGY
IF(V.LT.YSIG(JY)) GO TO 56
ALPHAI = (1.0-DV)*TOTALS(JY-1)/ALBESS(JY-1) +
JY*TOTALS(JY)/ALBESS(JY)
55 CONTINUE
56 DV = (V - YSIG(JY-1))/(YSIG(JY) - YSIG(JY-1))
ALPHAI = (1.0 - DV)*TOTALS(JY-1)/ALBESS(JY-1) +
DV*TOTALS(JY)/ALBESS(JY)
END
FALPHA = ALPHA1
RETURN
END

SUBROUTINE FFMPAK(X,Y,Z,IROW,NMU,WK)
ON NHM4/FFMPAK
THIS ROUTINE FORMS THE MATRIX PRODUCT X * Y = Z, WHERE X, Y, AND Z
ARE BLOCK MATRICES STORED ON THE PACKED FORMAT OF 12.4.
INDEXING IS SOMEWHAT COMPLICATED, DUE TO THE PACKING FORMAT.
THE VARIOUS INDICES USED ARE
IXB,... BLOCK ROW INDEX OF X
J1,... ELEMENT ROW OFFSET OF BLOCK ROW 1XB OF X
JYB,... BLOCK COLUMN INDEX OF Y
J1,... ELEMENT COLUMN OFFSET OF BLOCK COLUMN XX OF X
J2,... ELEMENT COLUMN OFFSET OF BLOCK JVB OF Y
KK,... BLOCK COLUMN INDEX OF X
KY,... BLOCK ROW INDEX OF X
I,J,K,... ELEMENT INDICES WITHIN AN NMU BY NMU BLOCK
WORK MUST HAVE AT LEAST NMU*NMU WORDS
DIMENSION X(IROW,1),Y(IROW,1),Z(IROW,1),WK(NMU,1)
LP1 = L + 1
DO 100 IXB=1,LP1
  I1 = (IXB-1)*NMU
  JXB2 = L/2 + 1
  IF(MOD(IXB,2).EQ.0) JXB2 = (L+1)/2
C
DO 100 JYB = 1,JXB2
  J2 = (JYB-1)*NMU
C
ZERO THE ACCUMULATION BLOCK
DO 110 I=1,NMU
DO 110 J=1,NMU
110 WORK(I,J) = 0.
C
MULTIPLY BLOCK ROW IXB OF X BY BLOCK COLUMN JYB OF Y
C
DO 300 KX=1,JXB2
  KX = 2*KX - 1
  IF(MOD(IXB,2).EQ.0) KX = 2*KX
  J1 = (KX - 1)*NMU
  I2 = (KX - 1)*NMU
C
MULTIPLY BLOCK (IXB,KX) OF X BY BLOCK (KY,JYB) OF Y
DO 300 I=1,NMU
DO 300 J=1,NMU
SUM = 0.
DO 301 K=1,NMU
  SUM = SUM + X(I+I,J+J)*Y(I2+K,J2+J)
300 WORK(I,J) = WORK(I,J) + SUM
C
STORE THE BLOCK IN THE PACKED Z ARRAY AS BLOCK (IXB,JYB)
C
DO 400 I=1,NMU
DO 400 J=1,NMU
400 Z(I1+I,J2+J) = WORK(I,J)
100 CONTINUE
C
RETURN
END

FUNCTION PHASEF(V,COSPHI)
C
ON NHM4/PFLINE
C
THIS FUNCTION RETURNS THE VALUE OF THE PHASE FUNCTION AS DEFINED
FOR LAKE LIMN. (SEE RADIATIVE TRANSFER IN NATURAL WATERS, CHAPTER 5,
TABLE 5.3. WHEN PUBLISHED. MEANWHILE, REGARD THIS PHASE FUNCTION
AS TYPICAL OF MODERATELY TURBID LAKE WATER.)
C
SINCE THE PHASE FUNCTION IS
NEARLY LINEAR ON A LOG-LOG PLOT, LINEAR INTERPOLATION IS
PERFORMED IN LOG(PHASE)-LOG(PSI).
C
PARAMETER(MXPTS=22, MXSIGY=J)
C
DIMENSION SIGMA(MXPTS),PSI(MXPTS),PSILOG(MXPTS),PSILOGSIGY(MXPTS)
COMMON/CSIGY/ VSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
COMMON/CMISC/ IMISC(20),FMISC(20)
§5. PROGRAM 4

DATA PSI/0.0,0.01,0.1,1.0,
     1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,110.0,120.0,130.0,
     2.140.,150.,160.,170.,180. /
DATA SIGMA/9.9609E6,315543,31562,315,55,90,62,130.89,13.2,6.41,3.47,2.08,1.37,1.0,0.811,0.716,0.691,
     0.693,0.707,0.741,0.766,0.782,0.789 /
DATA KALL/0/, NSIGY/1/, S,ALPHA/0.5, 0.8/
DATA SIG90/0.0021401/, PSIO/0.01/, APSI/43.4197/, PPSI/1.4/
C
C IF(KALL.EQ.0) THEN
C
C THE FIRST CALL IS USED FOR INITIALIZATION
C
PI = FMISC(1)
DEGRAD = FMISC(2)
RADEG = FMISC(3)
[MISC(5) = NSIGY
VSIG(1) = 0.
TOTALS(1) = S
ALBESS(1) = S/ALPHA
C
C CONVERT TABULATED VALUES TO LOGS
S1 = SIG90/S
DO 100 I=2,MXPTS
   PLOG(I) = ALOG(S1*SIGMA(I))
100   PSILOG(I) = ALOG(PSI(I))
   PLOG(1) = PLOG(2)
   PSILOG(1) = -1.0E200
C
WRITE(6,200)
WRITE(6,202) ALPHA,S
WRITE(6,204)
DO 102 I=1,MXPTS
   PHASE = 10.0**PLOG(I)
102   WRITE(6,206) PSILOG(I),SIGMA(I),PHASE
C
GET THE ANALYTIC INTEGRAL FROM PSI = 0 TO PSI = PSI0
C
APSI = APSI*S1/1000/S
SOPSI = 2.0*APSI/(2.0 - PPSI)
SOPSI0 = SOPSI*(PSI0*DEGRAD)**(2.0 - PPSI)
WRITE(6,210) PSI0,SOPSI0,APSI,PPSI
C
KALL = 1
PHASEL = 0.
C
ELSE
C
CONVERT COS(PSI) TO LOG(PSI) AND INTERPOLATE
C
IF(ABS(COSPSI).GT.1.0) THEN
   COSPSI = SIGN(1.0,COSPSI)
ENDIF
C
PSIDEG = RADEG*ACOS(COSPSI)
IF(PSIDEG.LT.1.0E-8) THEN
   PSI = -8.0
ELSEIF(PSIDEG.GT.1.8E.0) THEN
   PSI = ALOG(1.8E.0)
ELSE
   PSI = ALOG(PSIDEG)
ENDIF
C
IF(PSI.LE.PSILOG(2)) THEN
   PHASEL = PLOG(2)
ELSE
   DO 300 I=2,MXPTS
      IF(PSI.LT.PSILOG(I)) GO TO 301
300 CONTINUE
301   PHASEL = PSILOG(I-1) + (PSILOG(I) - PSILOG(I-1))
         * (PSI - PSILOG(I-1))/(PSILOG(I) - PSILOG(I-1))
ENDIF
ENDIF
C
90
§5. PROGRAM 4

C

PHASEF = 10.0**PHASEL
KALL = KALL + 1
RETURN

C

200 FORMAT(1H1, 'THE VOLUME SCATTERING FUNCTION FOR LAKE LIMNE IS USED
1 AT ALL DEPTHS.
')
201 FORMAT(1H1, 'THE VOLUME ATTENUATION COEFFICIENT ALPHA IS',
1F6.3, ' PER METER.' '
')
202 FORMAT(1H0, ' THE TOTAL VOLUME SCATTERING FUNCTION S IS',
2F6.3, ' PER METER.' )
203 FORMAT(1H0, ' THE TABULATED VALUES DEFINING SIGMA(PSI) ARE',
1PHASE/)
204 FORMAT(I10, 'THE TABULATED VALUES DEFINING SIGMA(PSI) ARE',
1PHASE/)
205 FORMAT(1H0, 'SIGMA(90) = ',F18.6)
206 FORMAT(1H0, 'LINEAR INTERPOLATION IS DONE IN LOG(PSI)-LOG(PHASE)'),
207 FORMAT(1H0, 'THE ANALYTIC INTEGRAL OF 2*PI*PHASE(PSI)*SIN(PSI)',
1FROM PSI TO PSI =',F5.3, ' FOR A = ',E14.6,
2AND P = ',PFPF5.2)
END

FUNCTION PHASEF(X,COSPSI)
ON NHM/PELAG:

C

THIS FUNCTION RETURNS THE VALUE OF THE PHASE FUNCTION AS DEFINED
FOR THE PELAGOS SEA. (SEE RADIATIVE TRANSFER IN NATURAL WATERS,
CHAPTER 5, TABLE 5.5, WHEN PUBLISHED.) MEANWHILE, REGARD THIS
SCATTERING FUNCTION AS TYPICAL OF CLEAN, OPEN OCEAN WATERS.)

C

GIVEN THE WAVELENGTH IN NANOMETERS, WAVENM, THE FIRST CALL TO
THE ROUTINE LINEARLY INTERPOLATES IN LOG(NORM SIGMA) TO GET A NORMALIZED
SIGMA FUNCTION FOR THE DESIRED WAVELENGTH AT EACH TABULATED SCATTERING
ANGLE, PSI. VALUES OF THE ABSORPTION AND TOTAL SCATTERING ARE
ALSO DETERMINED FOR THE REQUESTED WAVELENGTH.

C

N.B.: THE REQUESTED WAVELENGTH, WAVENM, SHOULD BE ONE OF THE
DISCRETE NHM WAVELENGTHS (NAMELY: 400., 425., ..., 675. OR 700. NM).
THIS IS BECAUSE THE SIGMA(90) VALUES CANNOT BE OBTAINED BY
INTERPOLATION OF THE TABULATED VALUES.

C

SINCE THE PHASE FUNCTION IS NEARLY LINEAR ON A LOG-LOG PLOT,
LINEAR INTERPOLATION IS PERFORMED IN LOG(PHASE)-LOG(SIGMA) IN ORDER
TO DEFINE VALUES OF THE PHASE FUNCTION AT ARBITRARY PSI VALUES.

PARAMETER(MXPTS=12, MXSIGV=3)

C

DIMENSION SIGV(MXPTS), SIGL(MXPTS), SIGLX(MXPTS)
DIMENSION MXPTS(MXPTS), SIGLX(MXPTS)
COMMON/MSIGV/YSIGV(SIGV, SIGL, SIGLX, MXSIGV)
COMMON/MXPTS/MXPTS, FMXPTS

F

DATA PSI(MXPTS)=
1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0,
2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0,
3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0,
4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5.0,
5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6.0,
6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 7.0,
7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.0,
8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 9.0,
9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9, 10.0,
10.1, 10.2, 10.3, 10.4, 10.5, 10.6, 10.7, 10.8, 10.9, 11.0,
11.1, 11.2, 11.3, 11.4, 11.5, 11.6, 11.7, 11.8, 11.9, 12.0,
12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, 12.9, 13.0,
13.1, 13.2, 13.3, 13.4, 13.5, 13.6, 13.7, 13.8, 13.9, 14.0,
14.1, 14.2, 14.3, 14.4, 14.5, 14.6, 14.7, 14.8, 14.9, 15.0,
15.1, 15.2, 15.3, 15.4, 15.5, 15.6, 15.7, 15.8, 15.9, 16.0,
16.1, 16.2, 16.3, 16.4, 16.5, 16.6, 16.7, 16.8, 16.9, 17.0,
17.1, 17.2, 17.3, 17.4, 17.5, 17.6, 17.7, 17.8, 17.9, 18.0,
18.1, 18.2, 18.3, 18.4, 18.5, 18.6, 18.7, 18.8, 18.9, 19.0,
§5. PROGRAM 4

DATA WVLTAB/400., 425., 450., 475., 500., 525., 550., 575.,
1. 600., 625., 650., 675., 700./
DATA ABSORB/ 0.05, 0.05, 0.05, 0.05, 0.06, 0.08, 0.14, 0.25, 0.30,
1. 0.37, 0.43, 0.60/
DATA TSCAT/ 0.04, 0.0325, 0.0275, 0.0234, 0.02, 0.0175, 0.0152, 0.0136,
1. 0.0122, 0.0115, 0.0108, 0.0102, 0.01/
DATA S90/9.98348E-4, 7.45989E-4, 5.78771E-4, 4.50275E-4,
1. 3.50922E-4, 2.79280E-4, 2.20106E-4, 1.78304E-4, 1.44517E-4,
1. 2.28485E-4, 2.13858E-4, 1.75634E-5/

C DATA KALL/0., NSIGY/1/
DATA APSI400,APS1700/4.791616, 21.562268/, PSIO/0.01/, PPSI/1.4/

C IF (KALL.EQ.0) THEN
THE FIRST CALL IS USED FOR INITIALIZATION

PI = FMISC(1)
DEGRO = FMISC(2)
RADEG = FMISC(3)
WAVENM = FMISC(13)
IMISC(5) = NSIGY

C LINEARLY INTERPOLATE IN LOG(NORMALIZED SIGMA)-WAVELENGTH TO
DEFINE A NORMALIZED SIGMA AT THE REQUESTED WAVELENGTH, AT EACH
TABULATED PSI VALUE

IF (WAVENM.LE.400.) THEN
WAVEF = 0.
ELSEIF (WAVENM.GE.700.) THEN
WAVEF = 1.
ELSE
WAVEF = (WAVENM - 400.)/300.
ENDIF

C DO 400 I=1,MXPTS
SIGLOG(I) = (1.0 - WAVEF)*ALOGIO(S400(1)) + WAVEF*ALOGIO(S700(1))
C 400 SIGMA(I) = 10.0**SIGLOG(I)

C LOOK UP THE ABSORPTION, TOTAL SCATTERING, AND SIGMA(90) VALUES
FOR THE REQUESTED WAVELENGTH

IWAVE = IFIX(1.5 + AMOD(WAVENM,400.)/25.)
ABSR = ABSORB(IWAVE)
S = TSCAT(IWAVE)
SIG90 = S90(IWAVE)
ALPHA = ABSR + S

C VSIG(1) = 0.
TOTALS(1) = S
ALBESS(1) = S/ALPHA

C CONVERT THE DEFINED SIGMA TO LOGS OF THE PHASE FUNCTION
S1 = SIG90/S
DO 100 I=2,MXPTS
PSILOG(I) = ALOGIO(S1*SIGMA(I))
100 PSILOG(1) = ALOGIO(SIGMA(1))
C
C WRITE(6,201)
WRITE(6,202) WAVENM,ABSR,S,ALPHA,ALBESS(1)
WRITE(6,204)
DO 102 I=2,MXPTS
PHASE = 10.0**PSILOG(I)
102 WRITE(6,206) PSI(1),SIGMA(1),PHASE
WRITE(6,207) SIG90
WRITE(6,208)
GET THE ANALYTIC INTEGRAL FROM \( \psi = 0 \) TO \( \psi = \psi_{10} \)

\[
\begin{align*}
\psi_{10} &= (1.0 - \text{WAVEF}) \times \text{ALOG10}(\psi_{10}/400) + \text{WAVEF} \times \text{ALOG10}(\psi_{10}/1700) \\
\psi_{10} &= 10.0 \times \psi_{10} \\
\psi_{10} &= \psi_{10} \times \text{SIG90}/5 \\
\psi_{10} &= 2.0 \times \pi \times \psi_{10}/(2.0 - \psi_{10}) \\
\psi_{10} &= \psi_{10} \times (\psi_{10}/\text{DEGRAD})^{*}(2.0 - \psi_{10}) \\
\end{align*}
\]

WRITE(6,210) \( \psi_{10}, \psi_{10}/\text{SOPSI}, \psi_{10}/\text{APS1} \)

KALL = 1
PHASEL = 0.

ELSE

CONVERT COS(\( \psi_{10} \)) TO LOG(\( \psi_{10} \)) AND INTERPOLATE

IF(ABS(COSPSI) \( \gt \) 1.0) THEN
COSPSI = SIGN(1.0, COSPSI)
ENDIF

PSIDEG = RADEG*ACOS(COSPS1)
IF(PSIDEG.LT.-1.0E+8) THEN
PSIL = -8.0
ELSEIF(PSIDEG.GT.180.) THEN
PSIL = ALOG10(180.)
ELSE
PSIL = ALOG10(\( \psi_{10} \))
ENDIF

IF(PSIL.LE.PSILOG(2)) THEN
PHASEL = PLOG(2)
ELSE
DO 300 I=2,MAXPTS
IF(PSIL.LT.PSILOG(I)) GO TO 302
300 CONTINUE
302 PHASEL = PLOG(I-1) * (PLOG(I) - PLOG(I-1)) * 
(PSIL - PSILOG(I-1))/(PSILOG(I) - PSILOG(I-1))
ENDIF
ENDIF

PHASEF = 10.0**PHASEL
KALL = KALL + 1
RETURN

200 FORMAT(1X,'THE VOLUME SCATTERING FUNCTION DEFINED FOR THE PELAGIUS SEA IS USED AT ALL DEPTHS')
201 FORMAT(1X,'WAVELENGTH IS LAMBDA = ',F6.1, ' NANOMETERS')
202 FORMAT(1X,' THE VOLUME ABSORPTION FUNCTION IS A = ',F7.4, ' PER METER')
203 FORMAT(1X,' THE TOTAL VOLUME SCATTERING FUNCTION IS S = ',F7.4, ' PER METER')
204 FORMAT(1X,' THE VOLUME ATTENUATION FUNCTION IS ALPHA = ',F7.4, ' PER METER')
205 FORMAT(1X,' THE ALBEDO OF SINGLE SCATTERING IS OMEGA = ',F7.4)
206 FORMAT(1X,' THE TABULATED VALUES DEFINING SIGMA(\( \psi_{10} \)) ARE')
207 FORMAT(1X,' PSI  SIGMA/SIG90 9X PHASE')
208 FORMAT(1X,' THE LINEAR INTERPOLATION IS DONE IN LOG(\( \psi_{10} \)) - LOG(PHASE)')
209 FORMAT(1X,' THE ANALYTIC INTEGRAL OF 2*PI*PHASE(\( \psi_{10} \)) SIN(\( \psi_{10} \))')
210 FORMAT(1X,' FROM \( \psi_{10} = 0 \) TO \( \psi_{10} = \psi_{10} \))')
211 END
FUNCTION PHASEF(V,COSPSI)

ON NHM4/PFSpher

THIS FUNCTION RETURNS A VALUE OF THE CONTINUOUS, POINT GEOMETRIC
PHASE FUNCTION \( P(y, \mu, \phi', \phi') = P(\cos \psi, y) = \sigma(\cos \psi, y)/s(y) \) FOR ANY \( \cos \psi \) AND \( y \) VALUES.

PHASEF IS FOR USE IN THE COMPUTATION OF THE QUAD-AVERAGED,
GEOMETRIC SCATTERING FUNCTIONS \( P(y, r, s/u,v) = P(y, r/uv) \) VIA
11.3.

THIS VERSION FOR FOR ISOTROPIC SCATTERING: \( \sigma = s/(4\pi) \)
INDEPENDENT OF SCATTERING ANGLE AND DEPTH

PARAMETER(MXSIGY=3)
COMMON/CSIGY/ VSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
COMMON/CMISC/ IMISC(20),FMISC(20)
DATA KALL/O/, NSIGY/1/, S,ALPHA/0.125, 0.736/

IF(KALL.EQ.0) THEN
    THE FIRST CALL TO PHASEF IS USED FOR INITIALIZATION ONLY

    PI = FMISC(1)
    SIG = 0.25*S/PI
    IMISC(5) = NSIGY
    VSIG(1) = 0.
    TOTALS(1) = S
    ALBESS(1) = S/ALPHA
    WRITE(6,100) SIG
    WRITE(6,102) ALPHA,S
    SIG = 0.25/PI
    KALL = 1
    PHASEF = 0.
    RETURN

ELSE
    PHASEF = SIG
    RETURN
    ENDF

100 FORMAT(IH1,' A SPHERICALLY SYMMETRIC VOLUME SCATTERING FUNCTION IS
    1 USED:
        \( \sigma = s/(4\pi) \) = ',F8.5,' FOR ALL ANGLES AND DEPTHS'
)
102 FORMAT(IH10,'THE VOLUME ATTENUATION COEFFICIENT ALPHA IS ',F6.3,
        1' PER METER'/', THE TOTAL VOLUME SCATTERING FUNCTION S IS ',
        2F6.3,' PER METER'
)
END

94
FUNCTION PHASEF(V, COSPSI)  
ON NHM4/PFSVPY  

THIS FUNCTION RETURNS A VALUE OF THE CONTINUOUS, POINT GEOMETRIC  
PHASE FUNCTION P(Y, MU PRIME, PHI PRIME/ MU, PHI) = P(Y, COSPSI) =  
SIGMA(Y, COSPSI)/S(Y) FOR ANY Y AND COS(Psi) VALUES.  

PHASEF IS FOR USE IN THE COMPUTATION OF THE QUAD-AVERAGED,  
GEOMETRIC SCATTERING FUNCTIONS P(Y, R/S/ U,V) = P(Y, R/ U,V) VIA  
(11.3)  

THIS VERSION IS FOR DEPTH DEPENDENT SPHERICAL SCATTERING:  
SIGMA(Y) = S(Y)/(4*PI), INDEPENDENT OF SCATTERING ANGLE  
BUT DEPENDENT ON DEPTH Y.  

PARAMETER(MXSIGY=3)  
DIMENSION ALPHA(MXSIGY)  
COMMON/CSIGY/ YSIG(MXSIGY), ALBESS(MXSIGY), TOTALS(MXSIGY)  
COMMON/CMISC/ IMISC(20), FMISC(20)  
DATA KALL/OI., NSIGY/3!, TOTALS/O.0, U5, U.3/, ALPHA/O.2, 0.6, 0.4/  

IF(KALL.EQ.0) THEN  

THE FIRST CALL TO PHASEF IS USED FOR INITIALIZATION ONLY  

PI = FMISC(1)  
IMISC(5) = NSIGY  
WRITE(6, 100)  
DO 50 IV=1, NSIGY  
   ALBESS(IV) = TOTALS(IV)/ALPHA(IV)  
50 WRITE(6,102) IV, YSIG(IV), TOTALS(IV), ALPHA(IV), ALBESS(IV)  
P = 0.25/PI  
KALL = 1  
PHASEF = 0.  

ELSE  
PHASEF = P  
ENDIF  
RETURN  

FORMAT(///' A DEPTH DEPENDENT, SPHERICAL VOLUME SCATTERING FUNCTION IS USED;'//, SIGMA(Y, COS(Psi)) = S(Y)/(4*PI) WHERE; //  
2* Y S(Y) / ALPHA //)  
102 FORMAT(1H, 14, F8.1, F8.3, F10.3, F11.3)  
END
SUBROUTINE PNTAMP(Y,AMP.A,AMPY,IROW)
C
ON NHM4/PNTAMP
C
THIS ROUTINE PRINTS THE AMPLITUDES AT Y = A, X,..., Z.
A TITLE GIVING THE APPROPRIATE COLUMN HEADINGS SHOULD BE WRITTEN
BEFORE CALLING PNTAMP.
C
IF ONLY AMPA IS TO BE PRINTED (THE CASE OF AMPA = AO(A,+)), GIVE
AMPY(1,1) A VALUE .GT. 1.0E200
C
DIMENSION Y(1),AMP(1),AMPY(IROW,1)
COMMON/CMISC/ IMISC(20)
C
NMU = IMISC(1)
NY = IMISC(4)
IDBUG = IMISC(9)
NRHAT = IMISC(10)
IL = 0
C
IAOP = 1 IF PRINTOUT IS FOR AMPA ONLY
IAOP = 0 OTHERWISE
IAOP = 0
IF(AMPY(1,1).GT.1.E200) IAOP = 1
C
IF(IAOP.EQ.1) WRITE(6,1599)
IF(IAOP.EQ.0) WRITE(6,1600) (V(IY),IY=1,NY)
UO 1602 I=1,2*NMAT
IF(1.EQ.NRHAT+1) THEN
WRITE(6,1610)
IL = 0
ENDIF
IMOD = MOD(I,NMU)
IF(IMOD.EQ.1) THEN
WRITE(6,1606) IL
ENDIF
IMU = IMU + 1
C
SELECT FULL OR PARTIAL PRINTOUT
IF(IDBUG.EQ.1 .AND. IL.GT.2) GO TO 1602
IF(IAOP.NE.0) THEN
IF(IMOD.EQ.1) WRITE(6,1612) IMU,AMP(1),(AMPY(I,J),J=1,NY)
IF(IMOD.NE.1) WRITE(6,1614) IMU,AMP(1),(AMPY(I,J),J=1,NY)
ELSE
IF(IMOD.EQ.1) WRITE(6,1612) IMU,AMP(1)
IF(IMOD.NE.1) WRITE(6,1614) IMU,AMP(1)
ENDIF
1602 CONTINUE
RETURN
C
FORMATS
C
1599 FORMAT(1H0,2X,'COSINES')
1600 FORMAT(1H0,2X,'COSINES',23X,5('Y = ',F7.3,5X)/33X,5('Y = ',F7.3,5X))
1606 FORMAT(1H0,’ L = ’,I3)
1610 FORMAT(1H0,2X,'SINES')
1612 FORMAT(1H0,10X,12.1P6E15.4/28X,5E15.4)
1614 FORMAT(1H0,10X,12.1P6E15.4/28X,5E15.4)
END
SUBROUTINE QAPHAS(NUQB,NVQB,INCBAS)
C
ON NHM4/QAPHAS
C
THIS ROUTINE COMPUTES THE QUAD-AVERAGED GEOMETRIC PHASE
FUNCTIONS GEOPP = P+(Y;R,U,V) AND GEOPM = P-(Y;R,U,V) USING 11.3.
VALUES ARE COMPUTED AT EACH Y LEVEL WHERE THE POINT GEOMETRIC
PHASE FUNCTION IS GIVEN (BY FUNCTION PHASEF)
C
NUQB AND NVQB ARE THE BASE NUMBER OF QUAD SUBDIVISIONS IN THE MU AND PHI
DIRECTIONS, USED FOR NUMERICAL INTEGRATION OF THE CONTINUOUS
PHASE FUNCTION: THE NUMBER OF QUAD SUBDIVISIONS IS INCREASED
BY A FACTOR OF INCBAS IN THE FORWARD SCATTERING QUADS AND IN
THE ADJACENT QUADS.
C
THE ARRAY PHASE(I.IV) CONTAINS THE TABULATED VALUES OF
PHASEF(Y(IY),COS(PSI(I))
C
PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGV=3)
PARAMETER(MXGEOP=MXMU*(MXPHI/2 + 1))
C
COMMON/CRTSIG/ RHOMAT(MXMU,MXV,MXSIGV),TAUMAT(MXMU,MXV,MXSIGV),
1 GEOPP(MXMU,MXGEOP,MXSIGV),GEOPM(MXMU,MXGEOP,MXSIGV),
COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),YOUT(MXY),BNDMU(MXMU),
1 BNDPHI(MXPHI),OMEGA(MXMU),DELMU(MXMU)
COMMON/CMISC/ IMISC(20),FMISC(20)
COMMON/CWORK/ RADSKY(MXMU,MXPHI),PHASE(2701,MXSIGV),
1 CKSUM(MXMU,MXV)
C
NMU = IMISC(1)
NPHI = IMISC(2)
NL = IMISC(3)
NSIGV = IMISC(5)
IDBUG = IMISC(9)
TWOPI = 2.0*FMISC(1)
RADEG = FMISC(3)
C
DELPHI = TWOPI/FLOAT(NPHI)
NOPI = NPHI/2
NOPII = NOPI + 1
C
DO 50 IV=1,MXSIGV
DO 50 J=1,MXGEOP
DO 50 I=1,MXMU
GEOPP(I,J,IV) = 0.
50 GEOPM(I,J,IV) = 0.
C
LOOP OVER THE DEPTHS (Y INDEX) WHERE THE OPTICAL PROPERTIES OF
THE WATER ARE DEFINED.
C
DO 100 IV=1,NSIGV
C
LOOP OVER THE MU, U, AND V QUAD INDICES
C
DO 100 IU=1,NMU
DO 100 IR=1,NMU
NCUMPV = NOPII
IF(IU EQ.NMU .OR. IR.EQ.NMU) NCUMPV = 1
DO 100 IV=1,NCUMPV
C
SELECT THE SUBQUAD RESOLUTION: IDENTICAL OR ADJACENT QUADS INVOLVE
FORWARD SCATTERING, AND NEED HIGHER RESOLUTION TO RESOLVE THE
PHASE FUNCTION ACCURATELY.
C
IF(IV.LE.2) THEN
  IF(IR.EQ.IU .OR. IR.EQ.IU+1 .OR. IR.EQ.IU-1) THEN
    NUQ = INCBAS*NUQB
    NVQ = INCBAS*NVOB
  ELSE
    NUQ = NUQB
    NVQ = NVOB
  ENDIF
ELSE
§5. PROGRAM 4

ELSE
   NUQ = NUQB
   NVQ = NVQB
ENDIF

C BOUNDARIES OF THE MU (= IU) QUAD
   UMUMIN = 0.
   IF(IU.GT.1) UMUMIN = BNDMU(IU-1)
   DMU = DELTMU(IU)/FLOAT(NUQ)
   UU = UMUMIN + 0.5*DMU
C SIZE OF THE PHI-J SUBQUADS
   IF(IU.EQ.NMU) THEN
      DPHI = TWOPI/FLOAT(NVQ)
   ELSE
      DPHI = DELPHI/FLOAT(NVQ)
   ENDIF

C BOUNDARIES OF THE MU PRIME (= IR) QUAD
   RMUMIN = 0.
   IF(IR.GT.1) RMUMIN = BNDMU(IR-1)
   DMUP = DELTMU(IR)/FLOAT(NUQ)
   UOP = RMUMIN + 0.5*DMUP
C SIZE OF THE PHI PRIME-L SUBQUADS
   IF(IR.EQ.NMU) THEN
      DPHIP = TWOPI/FLOAT(NVQ)
   ELSE
      DPHIP = DELPHI/FLOAT(NVQ)
   ENDIF

C FACT = DMU*DPHI*DMII*DPHIP/OMEGA(IU)

C BOUNDARIES OF THE PHI (= IV) QUAD
   PHIMIN = BNDPHI(NPHI)
   IF(IV.GT.1) PHIMIN = BNDPHI(IV-1)
   PHIO = PHIMIN + 0.5*DPHI
C COMPUTE THE STORAGE INDEX BY(12.7)
   IF(IR.EQ.NMU) THEN
      K(COL = IU
   ELSE
      IF(IU.EQ.NMU) THEN
         KCOL = NMU
      ELSE
         IF(LT.EQ.NL+1) THEN
            KCOL = IU + NMU*(IV-1)
         ELSE
            KCOL = IU + NMU*(NL - MOD(IV-1,NL))
         ENDIF
      ENDIF
   ENDIF
   ENDIF
C INTEGRATE OVER PHI PRIME ONLY FOR THE PHI PRIME = U QUADS (IS = 1)
   PHIOP = BNDPHI(NPHI) + 0.5*DPHIP
C COMPUTE THE QUADRUPLE INTEGRAL (11.3) OVER THE SELECTED QUADS
   SUMP = 0.
   SUMM = 0.
   DO 110 JU=1,NUQ
C DEFINE A MU VALUE
      UMU = UO + FLOAT(JU-1)*DMU
      ROOTJU = SQRT(1.0 - UMU*UMU)
C DO 110 JR=1,NUQ
C DEFINE A MU PRIME VALUE
      RMUP = UOP + FLOAT(JR-1)*DMUP
      ROOTJR = SQRT(1.0 - RMUP*RMUP)
      A1 = UMU*RMUP
      A2 = ROOTJU*ROOTJR
C DO 110 JV=1,NVQ
C DEFINE A PHI VALUE
      VPHI = PHIO + FLOAT(JV-1)*DPHI
C DO 110 JU=1,NUQ
§5. PROGRAM 4

DO 110 JS=1,NVQ
C DEFINE A PHI PRIME VALUE
SPHII = PHIII + FLOAT(JS-1)*PHII
C COMPUTE CONTRIBUTIONS TO INTEGRALS
COSPPP = COS(VPHI - PHII)
C COSPPSI = A1 + A2*COSPPP
IF(ABS(COSPSI).GT.1.0) COSPSI = SIGN(1.0,COSPSI)
C GET PSI IN DEGREES AND DO A TABLE LOOKUP FOR PHASEF(Y,COSPSI)
C
PSI = RADEG*ACOS(COSPSI)
IF(PSSI.LE.10.0) THEN
  IPSI = IFIX(PSSI*1.0. + 1.0)
ELSE
  IPSI = IFIX(PSSI*1.0. + 90.0)
ENDIF
SUMM = SUMM + PHASE(IPSI,IV)
C
COSPSI = A1 + A2*COSPPP
IF(ABS(COSPSI).GT.1.0) COSPSI = SLU(u,COSPSI)
PSI = RADEG*ACOS(COSPSI)
IF(PSI.LE.10.0) THEN
  IPSI = IFIX(PSI*10.0. + 1.5)
ELSE
  IPSI = IFIX(PSI*10.0. + 90.1.5)
ENDIF
SUMM = SUMM + PHASE(IPSI,IV)
110 CONTINUE
C
GEOPP(IR,KCOL,IV) = GEOPP(IR,KCOL,IV) + SUMM*FACTR
100 GEOPM(IR,KCOL,IV) = GEOPM(IR,KCOL,IV) + SUMM*FACTR
C
C COMPUTE THE CHECK ON THE TOTAL SCATTERING, (11.5)
C
WRITE(6,208)
DO 200 IV=1,NSIGY
WRITE(6,212)
DO 200 IR=1,NMU
  POLAR CAP QUAD
  SUMM = (GEOPP(IR,NMU,IV) + GEOPM(IR,NMU,IV))*OMEGA(NMU)/OMEGA(IR)
  DO 202 IU=1,NMU-1
    FACTR = OMEGA(IU)/OMEGA(IR)
    SUMM = SUMM + (GEOPP(IR,KCOL,IV) + GEOPM(IR,KCOL,IV))*FACTR
    PHI = PI VALUES
    KCOL = NMU*NOPI + IU
    IF(IR.EQ.NMU) KCOL = IU
    SUMM = SUMM + (GEOPP(IR,KCOL,IV) + GEOPM(IR,KCOL,IV))*FACTR
    U,LT, PHI,LT, PI VALUES
    DO 202 IV=2,NOMI
      KCOL = NMU*(IV-1) + IU
      IF(IR.EQ.NMU) KCOL = IU
      SUMM = SUMM + 2.0*(GEOPP(IR,KCOL,IV) + GEOPM(IR,KCOL,IV))*FACTR
      CKSUM(IR,IV) = SUMM
  200 WRITE(6,210) IV,IR,SUMP,GEOPP(IR,IR,IV)
C
C USE THE CHECKSUMS TO REDEFINE THE FORWARD SCATTERING QUADS BY 11.7
C
DO 300 IV=1,NSIGY
DO 300 IR=1,NMU
  GEUPP(IR,IR,IV) = 1.0 - CKSUM(IR,IV) + GEUPP(IR,IR,IV)
C
IF(DBG,GE,1) THEN
  CALL PARRAY(GEOPP,NMU,4*NMI,NSIGY,MAXGEOP,2.
  1 QUAD-AVERAGED P+ (Y,R,1/U,V)')
  CALL PARRAY(GEOPM,NMU,4*NMI,NSIGY,MAXGEOP,2.
  1 QUAD-AVERAGED P- (Y,R,1/U,V)')
ENDIF
C
208 FORMAT(1H,11H CHECKSUMS ON QUAD-AVERAGED EU GEOMETRIC P+ AND P- FUNCTION
INS//H='2X','Y' R SUM (1))
210 FORMAT(1H ,13,I5,11.5,F15.5,1PE20.3)
212 FORMAT(1H )
C
RETURN
END
SUBROUTINE QASKY(RSKY, CARD, SHTOTL, THETAS, PHIS)

ON NHM4/QASKV

THIS ROUTINE COMPUTES THE INPUT SKY QUAD-AVERAGED RADIANCE
DISTRIBUTION, USING 3.3 EVALUATED AS IN APPENDIX B OF THIS TECH
MEMO. SEE ALSO STEP 7A4 ON PAGE 130.

RSKY IS THE RATIO OF SKY TO TOTAL SCALAR IRRADIANCE.  
RSKY = 0.0 FOR A BLACK SKY (SUN ONLY). RSKY = 1.0 FOR 
NO SUN

CARD IS THE CARDIOIDAL PARAMETER. CARD = 0. FOR A UNIFORM SKY.  
CARD = 2. FOR A CARDIOIDAL SKY

SHTOTL IS THE TOTAL (SKY + SUN) SCALAR IRRADIANCE ON THE 
WATER SURFACE FROM ABOVE

THETAS, PHIS ARE THE SUN SOURCE ANGLES (IN DEGREES, RELATIVE TO 
PHI = 0. IN THE DOWNWIND DIRECTION)

UPON RETURN, RSKY IN CWORK/ HOLDS THE QUAD-AVERAGED SKY 
RADIANCES FOR USE IN AMPAO IN MAIN.

PARAMETER(MXMU=10, MXPHI=24, MXY=30)

COMMON/GRID/ FMU(MXMU), PHI(MXPHI), Y(MXY), BNGMU(MXMU),  
1 BNDPHI(MXPHI), OMEGA(MXMU)

COMMON/MISC/ IMISC(20), FMISC(20)

COMMON/CWORK/ RSKY(MXMU,MXPHI), THETAB(MXMU), PHIB(MXPHI)

NMU = IMISC(1)
NPHI = IMISC(2)
10BUG = IMISC(9)
PI = FMISC(1)
RADEG = FMISC(3)

SET UP THE BACKGROUND SKY QUAD-AVERAGED RADIANCES USING B.7

WRITE(6,500) SHTOTL, RSKY, CARD
FNO = RSKY*SHTOTL/(2.0*PI*(1.0 + 0.5*CARD))
DO 100 I=1,NMU-1
RAD = FNO*(1.0 + CARD*FMU(I))
DO 100 J=1,NPHI
100 RADSXY(I,J) = RAD

PARLAR CAP
RADSXY(NMU,1) = FNO*(1.0 + CARD*FMU(NMU))

ADD IN THE SUN TO THE APPROPRIATE QUAD USING B.8

WRITE(6,502) THETAS, PHIS

CONVERT THE BOUNDARY MU AND PHI VALUES TO DEGREES
DO 101 I=1,NMU
101 THETAB(I) = RADEG*ACOS(BNGMU(I))
DO 103 J=1,NPHI
103 PHIJB(J) = RADEG*BNDPHI(J)

DETERMINE THE (MU, PHI) INDICES OF THE QUAD CONTAINING THE SUN

PH = MOD(PHIS + 360., 360.)
DO 201 I=1,NMU, 1
IF(THETAS_LT.THETAB(I) AND .THETAS_GE.THETAB(I+1)) IMUS = I + 1
201 CONTINUE
IF(.THETAS_GT.THETAB(I)) IMUS = 1

DO 202 J=1,NPHI
202 CONTINUE
IF (.PHI_LT.PHIB(J)) GO TO 206
206 JPHIS = J
200 CONTINUE

THE = RADEG*ACOS(FMU(IMUS))
WRITE(6,510) IMUS, JPHIS, THE, RADEG*PHI(JPHIS)

100
§5. PROGRAM 4

CHANGE PHI INDEX FROM SOURCE LOCATION TO BEAM DIRECTION

JPHIS = MOD(JPHIS + NPHI/2, NPHI)
IF(JPHIS.EQ.0) JPHIS = NPHI
IF(IMUS.EQ.NMU) JPHIS = 1
RADSKY(IMUS,JPHIS) = RADSKY(IMUS,JPHIS) +
1 (1.0 - RSKY)*SHTOTL/OMEGA(IMUS)

IF(IBUG.NE.0) CALL PZARAV(RAUSKY,NMU,NPHI,MU,Z,
1 'QUAD-AVERAGED SKY RADIANCES')
RETURN

FORMATS

500 FORMAT(1HI, 'THE INPUT RADIANCE DISTRIBUTION HAS:/'
15X, 'TOTAL SCALAR IRRADIANCE (SUN + SKY) = ',PE10.3,
2 'WATTS PER SQUARE METER /';
35X, 'RATIO OF SKY TO TOTAL SCALAR IRRADIANCE, R = ',FP6.3/
45X, 'CARDIOIDAL PARAMETER, C = ',FP4.1)
502 FORMAT(1H1, 'THE SUN IS REQUESTED AT SKY LOCATION (THETA, PHI) = (''
1 F4.1, ',F5.1, ',)
510 FORMAT(1H1, 'THE SUN IS PLACED IN QUADRANT (S) = Q(', '12, ', '12,
1') CENTERED AT (THETA, PHI) = (',FP6.3, ',F4.3, ',')
END

SUBROUTINE RFMPA(X,Y,Z,IRVW,NMU,L)

ON Nh4a/RFMPA

THIS ROUTINE FORMS THE MATRIX PRODUCT X * Y = Z, WHERE X AND Y ARE
ROW VECTORS AND Z IS A BLOCK MATRIX STORED ON THE PACKED FORMAT
OF 12.4.

DIMENSION X(1),Y(1),Z(1)

LP1 = L + 1
DO 100 JZB=1,LP1
JZB2 = L/2 + 1
IF(MOD(JZB,2).EQ.0) JZB2 = (L+1)/2
JZ = (JZB-1)*NMU
JY = (JZB+1)/2 - 1)*NMU

DO 200 J=1,NM
SUM = 0
DO 301 K=1,NM
KY = K*K - 1
IF(MOD(KY,2).EQ.0) KV = K*K
I2 = (K-1)*NM
DO 301 K=1,NM
301 SUM = SUM + A(K*K)*B(12*K,J2+1)
200 Z(I,J2+1) = SUM
100 CONTINUE

RETURN
END

101
§5. PROGRAM 4

SUBROUTINE RHOTAU(L)
C
C ON NHM4/RHOTAU
C
C THIS ROUTINE COMPUTES THE DISCRETIZED SPECTRAL PHASE FUNCTIONS RHOHAT AND TAUHAT FROM THE QUAD-AVERAGED GEOMETRIC SCATTERING FUNCTIONS. THE GOVERNING EQUATIONS ARE 5.6 AND 5.20B TO 5.20E.
C
PARAMETER(MXMu=10, MXPHI=24, MXY=30, MXSIGV=3)
PARAMETER(MXGDOP=MXMu*(MXPHI/2 + 1))
COMMON/CSIGV/ YSIG(MXSIGV), ALBESS(MXSIGV)
COMMON/CGRID/ FMu*MXMu), PHI(MXPHI)
COMMON/CRSIG/ RHOMAT(MXMu, MXMu, MXSIGV), TAUHAT(MXMu, MXMu, MXSIGV).
1. GEOFP(MXMu, MXGDOP, MXSIGV).
COMMON/CMISC/ IMISC(20)
COMMON/CWORK/ COSLPV(MXPHI)
C
NMu = IMISC(1)
NPHI = IMISC(2)
NL = IMISC(3)
NSIGV = IMISC(5)
C
DO 100 IV=1,NPHI
100 COSLPV(IV) = COS(FLOAT(L)*PHI(IV))
C
IF(L.EQ.U .OR. L.EQ.NL) THEN
EPSL = FLOAT(NPIIC
ELSE
EPSL = FLOAT(NL)
END IF
C
LOOP OVER THE DEPTHS WHERE THE INHERENT OPTICAL PROPERTIES ARE GIVEN
DO 200 IV=1,NSIGV
ALBEDO = ALBESS(IV)
C
POLAR CAP OUTPUT. IU = NMU
C
IF(L.EQ.U THEN
FMU1 = 1.0/FMU(NMu)
C QUAD INPUT: USE 5.20D. PHAT IS GIVEN BY 5.6C
DO 300 IR=1,NMu
RHOMAT(IR,NMu,IV) = ALBEDO*EPSL*GEOFP(IR,NMu,IV)*FMU1
300 TAUHAT(IR,NMu,IV) = ALBEDO*EPSL*GEOFP(IR,NMu,IV)*FMU1
C
POLAR CAP INPUT: USE 5.20E. PHAT IS GIVE BY 5.6D
RHOMAT(NMu,NMu,IV) = (ALBEDO*GEOFP(NMu,NMu,IV) - 1.0)*FMU1
C
ELSE
DO 302 IR=1,NMu
RHOMAT(IR,NMu,IV) = 0.
302 TAUHAT(IR,NMu,IV) = 0.
END IF
C
QUAD (NON-PULAR CAP) OUTPUT
DO 400 IU=1,NMu:
FMU1 = 1.0/FMU(IU)
C
POLAR CAP INPUT. IU=NNU. USE 5.20K. PHAT IS GIVEN BY 5.6B
C
IF(L.EQ.U THEN
RHOMAT(NMu, IU,IV) = ALBEDO*EPSL*GEOFP(NMu, IU, IV)*FMU1
TAUHAT(NMu, IU, IV) = ALBEDO*EPSL*GEOFP(NMu, IU, IV)*FMU1
ELSE
RHOMAT(NMu, IU, IV) = 0.
TAUHAT(NMu, IU, IV) = 0.
END IF
C
102
§5. PROGRAM 4

QUAD (NON-POLAR CAP) INPUT; USE 5.2UB. PHAT MUST NOW BE COMPUTED BY 5.6A

DO 310 IR=1,NMU-1
    SUMP = 0.
    SUMM = 0.
    DO 400 IV=1,NPHI
        COMPUTE STORAGE INDICES BY (12.7)
        IF(IV.LE.NL+1) THEN
            J = IU + NMU*(IV-1)
        ELSE
            J = IU + NMU*(NL - MOD(IV-1,NL))
        ENDIF
        SUMP = SUMP + GEOPP(IR,J,IV)*COSLPV(IV)
        SUMM = SUMM + GEOPM(IR,J,IV)*COSLPV(IV)
    400 IF(IR.EQ.IU) THEN 
        DELT = 1.
    ELSE
        DELT = 0.
    ENDIF
    310 TAUHAT(IR,IU,IV) = (ALBEDO*SUMP - DELT)*FMU1
    200 CONTINUE
    RETURN
END

SUBROUTINE RICAT(I,L)

ON NMMA/RICAT

THIS ROUTINE SOLVES FOR THE ARRAYS RXA = R(Y,X) AND TXY = T(X,Y)
BY INTEGRATING 6.43 AND 6.44 IN A UPWARD SWEEP
WITH INITIAL VALUES OF R(X,X) = 0 AND T(X,X) = 1. BY 6.47.
RXB = R(Y,B) IS FOUND BY INTEGRATING 6.48 IN AN UPWARD SWEEP
WITH INITIAL CONDITION R(2,B) = R(BATZ,2), BY 6.58.

THE ARRAYS RXA AND TXY ARE STORED IN THE VECTOR RT AS FOLLOWS
FOR A GIVEN Y-VALUE:
RXA(I,J) IS RT(I)*((J-1)*NMU+I)
TXY(I,J) IS RT(I)*((J-1)*NMU+I*MU*NPHI)

PARAMETER(MXMU=10, MXPHI=24, MXY=30)
PARAMETER(MXEU=2*MXMU*MXMU)
DIMENSION COVER*(24)

103
§5. PROGRAM 4

COMMON/CRTR, RXY(MXMU, MXMUMXV), TXY(MXMU, MXMUMXV),
1 RYB(MXMU, MXMUMXV), R2YB(MXMU, MXMUMXV)
COMMON/CBOTBC/ RHATZB(MXMU, MXMUMXV)
COMMON/CORID/ FMU(MXMU, MXPHI, YMXV)
COMMON/CMISC/ IMISC(20), FMISC(20)
COMMON/CWORK/ WERA(MXEQN, 9), RT(MXEQN)

C C SUBROUTINE ODTAB EVALUATES THE RHS OF 6.43, 6.44 AND 6.48
EXTERNAL ODTAB
C
NMU = IMISC(1)
NY = IMISC(4)
IDBUG = IMISC(9)
IBOTM = IMISC(12)
NMU2 = NMU*NMU
NEQNS = 2*NMU
TOL = FMISC(7)
IF(IDBUG.GE.1) WRITE(6,3002) TOL
C
C INITIALIZE THE ARRAYS AT Y = X USING 6.47
DO 500 I=1,NMU
DO 500 J=1,NMU
RXY(I,J,1) = 0.
RT(I+(J-1)*NMU) = 0.
DELT = 0.
IF( I.EQ.J) DELT = 1.
TXY(I,J,1) = DELT
500 RT(I+(J-1)*NMU+NMU2) = DELT
C
YSTART = Y(1)
CPREV = 0.
IMISC(13) = 1
IND = .
C
INTEGRATE 6.43 AND 6.44 TO FIND R(Y, X) AND T(X,Y) AT EACH Y LEVEL
DO 520 IV=2,NV
VEND = YSTART + Y(IV) - Y(IV - 1)
IF(IDBUG.GE.1) WRITE(6,3000) YSTART,YEND
C
CALL ODTAB(NEQNS, ODTAB, YSTART, HT, YEND, TOL, IND, ODTAB, MXEQN, 1
WERA, IER)
C
IDEV = (DVEK(1:4) - CPREV)
CPREV = DVEK(1:4)
IF(IDBUG.GE.1) WRITE(6,3001) IDEV
IF(IND.LE.0 OR. IER.GT.0) THEN
WRITE(6,1060) IMISC(13), IND, IER
STOP
ENDIF
C
SAVE THE SOLUTION AT Y = YEND
DO 520 J=1,NMU
DO 520 I=1,NMU
RXY(I,J,1) = RT(I+(J-1)*NMU)
520 TXY(I,J,1) = RT(I+(J-1)*NMU + NMU2)
C
INTEGRATE 6.48 FROM Z TO X TO FIND R1(Z, B) AT EACH Y LEVEL
C
INITIALIZE AT Y = Z WITH R1(Z, B) = RHATZB(1, B). USING 6.58
DO 550 J=1,NM
DO 550 I=1,NM
RXY(I,J,NV) = RHATZB(I,J)
550 RT(I+(J-1)*NMU) = RHATZB(I,J)
C
YSTART = V(NY)
NEQNS = NMU
CPREV = 0.
IND = 1
IMISC(13) = 2
C
INTEGRATE
DO 570 IV=1,NV+1
CPREV = NV : IV
VEND = YSTART - Y(IVREV+1) + Y(IVREV)
IF(IDBUG.GE.1) WRITE(6,3000) YSTART,YEND
C
104
§5. PROGRAM 4

CALL DVERK(NEQNS,DRTAB,YSTART,RT,YEND,TOL,IND,CDVERK,MAEQRN, 1 WERK,IER)
C
IDEV = CDVERK(24) - CDPREV
CDPREV = CDVERK(24)
IF(IDBUG.GE.1) WRITE(6,3001) IDEV
IF(IND.LE.0 .OR. IER.GT.0) THEN
WRITE(6,1060) IMISC(13),IND,IER
STOP
ENDIF
C
SAVE THE SOLUTION AT YEND
DO 570 J=1,NMU
DO 570 I=1,NMU
570 R1YB(I,J,1YREV) = RT(I*(J-1)*NMU)
C
USE R2(Y,B) = R1(Y,B) OR INTEGRATE 6.48 AGAIN, ACCORDING TO THE
C BOTTOM TYPE
C
IF(IDBUG.GE.1) WRITE(6,3001)
C
MATTE BOTTOM WITH L.GE.0. INTEGRATE AGAIN WITH INITIAL CONDITION
C R2(Z,B) = 0
C NOTE ADDED IN PROOFING: I DO NOT THINK THIS INTEGRATION IS
C NECESSARY: JUST SET R2(Z,B) = 0. SINCE AMP2 = 0. HOWEVER, THIS
C HAS NOT BEEN CHECKED BY COMPARING EACH COMPUTATION, SO I MAY BE
C MISSING SOMETHING. CM, 2 JUNE 88.
C
DO 595 J=1,NMU
DO 595 I=1,NMU
R2YB(I,J,NY) = 0.
595 RT(I*(J-1)*NMU) = 0.
C
YSTART = YNY;
NEQNS = NMU2
CDPREV = 0.
IND = 1
IMISC(13) = 2
INTEGRATE
DO 597 1Y=1,NV 1
1YREV = NV - 1Y
YEND = YSTART - Y(YREV+1) * Y(YREV)
IF(IDBUG.GE.1) WRITE(6,3000) YSTART,YEND
C
CALL DVERK(NEQNS,DRTAB,YSTART,RT,YEND,TOL,IND,CDVERK,MAEQRN, 1 WERK,IER)
C
IDEV = CDVERK(24) - CDPREV
CDPREV = CDVERK(24)
IF(IDBUG.GE.1) WRITE(6,3001) IDEV
IF(IND.LE.0 .OR. IER.GT.0) THEN
WRITE(6,1060) IMISC(13),IND,IER
STOP
ENDIF
C
SAVE THE SOLUTION AT YEND
DO 597 J=1,NMU
DO 597 I=1,NMU
597 R2YB(I,J,1YREV) = RT(I*(J-1)*NMU)
C
ELSE
C MATTE BOTTOM WITH L.GT.0 OR INFINITELY DEEP, HOMOGENEOUS LAYER.
C USE R2(Y,B) = R1(Y,B)
C
DO 600 1Y=1,NV
DO 600 J=1,NMU
DO 600 I=1,NMU
600 R2YB(I,J,1Y) = R1YB(I,J,1Y)
C
ENDIF
C
RETURN
C
1060 FORMAT( /// SUB RILATI: ERROR IN CALL TO DVERK: IDE = .15,5X, 1 IND = .15,5X, IER = .15)
3000 FORMAT( /// YSTART = .FB 4.5X, YEND = .FB 4)
3001 FORMAT( /// TOL = .FB 4.5X, DERIVATIVE EVALUATIONS )
3002 FORMAT(/// OUTPUT FROM INTEGRATION ROUTINE DVERK (TOL = .)
END
SUBROUTINE V2ZGEO

ON NHM4/V2ZGEO

THIS ROUTINE COMPUTES THE GEOMETRICAL DEPTHS ZETA (IN METERS) WHICH
CORRESPOND TO THE OPTICAL DEPTHS Y (NONDIMENSIONAL) WHERE
OUTPUT IS REQUESTED.
EQUATION 7.1 IS INTEGRATED, WHEREIN ALPHA IS A FUNCTION OF OPTICAL
DEPTH Y

PARAMETER (MXMU=10, MXPHI=24, MXY=30, MXSIGV=3)

COMMON/CGRID/ FMU(MXMU), PHI(MXPHI), YOUT(MXY), BNDMU(MXMU),
1 BNDPHI(MXPHI), OMEGA(MXPHU), DELTMU(MXMU), ZGEO(MXY)
COMMON/CSIGY/ YSIG(MXSIGV), ALBESS(MXSIGV), TOTALS(MXSIGV)
COMMON/CMISC/ IMISC(20)

EXTERNAL FALPHA

DATA AERR,RERR/R.E-8/

NY = IMISC(4)
NSIGV = IMISC(5)
ALPHA1 = ALBESS(1)/TOTALS(1)

IF(NSIGV.EQ.1) THEN

DO 100 IV=1,NY
100 ZGEO(IV) = ALPHA1*YOUT(IV)

ELSE

WATER COLUMN HAS VARIABLE OPTICAL PROPERTIES WITH DEPTH: INTEGRATE

DZETA = Y/ALPHA1

ZGEO(1) = ALPHA1*YOUT(1)
DO 200 IV=2,NY
200 ZGEO(IV) = ZGEO(IV-1) + DZETA(FALPHA,YOUT(IV-1),YOUT(IV),
1 AERR,RERR,ERRR,IERR)

ENDIF
RETURN
END

§5. PROGRAM 4
§6. PROGRAM 5

A. Program Description

This program first synthesizes the radiances from the amplitudes found in Program 4. Then the results are analyzed and derived quantities are computed, as detailed in §75/§8. Multiple runs of Program 5 can be made for a given set of output from Program 4. For example, one run can be made to check the balance of the radiative transfer equation, another run to compute the irradiances and other derived quantities, etc.

We note again, as discussed in §75/§7a, that the expensive computations for the quad-averaged upper boundary r and t arrays need be done only once for a given wind speed and quad resolution. Likewise, the expensive discretization of the phase function is a one-time computation for a given phase function. The actual solution of the radiative transfer equation in Programs 4 and 5 is relatively inexpensive. Therefore, holding the wind speed and phase function fixed, it is possible to make many runs of Programs 4 and 5 in order to study the effects of varying the incident radiance distribution, the scattering-to-absorption ratio $s/a = \omega/(1-\omega)$, the bottom boundary type, etc. It is often convenient to make a run of Programs 4 and 5 with radiance output (see record 5 of Program 4 and records 2, 4 and 5, below) at some standard set of depths, say at y values of 0.0, 0.5, 1.0, 2.0, 5.0, 10.0, and 20.0 optical depths (here YOUT(1) = x = 0.0 and YOUT(NY) = z = 20.0, with NY = 7). If inspection of this run indicates a "region of interest" (e.g. large changes in the radiance field with depth, or "kinks" in the K-function curves) between $y = 2.0$ and $y = 5.0$, say, then another run of Programs 4 and 5 can be made to give greater resolution in the region of interest. The second run could save the output at y values of 0.0, 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 10.0 and 20.0 (now NY = 11).

In addition to the specific analyses selected by the input records below, a basic "skeleton" of radiance values is always printed (e.g. upward, downward and horizontal radiances in the alongwind and crosswind directions at selected depths, cf. subroutine RADY).

Other useful quantities automatically computed and printed are the contrast transmittance (cf. §75/§8k and subroutine CONTRM), and the backward and forward scattering functions (cf. §75/§8d and subroutine BFSCAT). If desired, this output can be removed by deleting the calls to the appropriate subroutines.

Additional output is included where convenient in many of the subroutines. For example, path functions (cf. §75/§8g) are computed along with the radiance K-functions (subroutine KRAD). Distribution functions (§75/8.11) and reflectance functions (§75/8.14) are computed along with the irradiances (subroutine IRRAD). Eccentricities (§75/8.16b) are included with the backward and forward scattering functions.

B. Input

From five to nine free-format records are read to specify the type of analysis desired.
§6. PROGRAM 5

Record 1: ITITLE

This is an alphanumeric title for the run, used to identify the printout. Up to 80 characters are allowed.

Record 2: IPRAD, IPRAD1, IPRAD2, IPRAD3, JPRAD1, JPRAD2, JPRAD3

This record (along with record 2a, if required) specifies the extent of printout of the radiance distribution by subroutine PRINT.

IPRAD < 0 if a printout of the radiance distribution is desired at every y level where the radiance was computed: \( y = a, \ YOUT(1) = x, \ldots, YOUT(NY) = z \)

IPRAD = 0 if no printout of the radiances is desired

IPRAD > 0 if printout is desired only at certain y levels, IPRAD in number, to be specified in record 2a

IPRAD1, IPRAD2, IPRAD3 are DO-LOOP indices of the form

\[ \text{DO 302 I=IPRAD1, IPRAD2, IPRAD3} \]

which select the \( \mu \)-bands of quads for which printout is desired. For example, consider the \( m = 10 \) by \( 2n = 24 \) quad partition of 75/FIG. 4a. There are \( m = 10 \) \( \mu \)-bands in each hemisphere (\( \mu, \ u = 1, \ldots, m \)). If (IPRAD1, IPRAD2, IPRAD3) = (1, 10, 1) then all \( \mu \)-bands will be printed. If (IPRAD1, IPRAD2, IPRAD3) = (1, 10, 3) then only bands \( u = 1, 4, 7 \) and 10 are printed out (the polar cap values, \( u = m \), are always printed). See DO-LOOPs 302 and 103 in subrouting PRINT (where index I is u).

JPRAD1, JPRAD2, JPRAD3 are DO-LOOP indices which select the \( \phi \)-bands to be printed, \( \phi, \ v = 1, \ldots, 2n \). Referring again to 75/FIG. 4a, if (JPRAD1, JPRAD2, JPRAD3) = (1, 24, 1) then all \( \phi \)-bands would be printed. If (JPRAD1, JPRAD2, JPRAD3) = (1, 24, 6) then only the \( \phi \)-bands at \( v = 1, 7, 13, 19 \) (corresponding to \( \phi = 0^\circ, 90^\circ, 180^\circ, 270^\circ \) in 75/FIG. 4a) are printed. See DO-LOOPS 302 and 103 in PRINT (where index J is v).

Record 2a: IYPRAD(1),...,IYPRAD(IPRAD)

This record is read only if IPRAD > 0. The values of IYPRAD are the j indices of \( y_j \), \( j = 1, \ldots, YOUT \), at which printout is desired. (See 75/FIG. 6 and input record 5 of Program 4, where \( y_j \) is YOUT(j).)

Record 3: IRTECK, NIC, NJC

This record specifies whether or not the balance of the radiative transfer equation (RTE) is to be checked; see 75/§8a and pay special attention to 75/8.3 and the requirement of closely spaced \( y_j \) values.
§6. PROGRAM 5

IRTECK < 0 if the balance of the RTE is to be computed at all possible interior y-levels, YOUT(2),...,YOUT(NY-1)

IRTECK = 0 if no RTE balance check is to be made

IRTECK > 0 if record 3a gives the indices of the y-levels where the RTE check is to be made. Normally, the RTE is checked only at the center y-level of three closely spaced y-levels (see 75/8.3). Thus if the user plans to check the RTE, foresight must be shown in specifying the y-levels in record 5 of Program 4. "Closely spaced" y-levels are separated by, say, 0.01 optical depths. Thus a choice of y-levels in record 5 of Program 4 might be 0.0, 0.99, 1.00, 1.01, 4.99, 5.00, 5.01,.... The balance of the RTE could then be checked at levels 1.00 and 5.00.

NIC, NJC are DO-LOOP increments used to select particular μ and θ values where the RTE balance is to be checked. See DO-LOOPs 300 in subroutine RTECK, which are of the form

DO 300 J = 1, NPHI, NJC
DO 300 I = 1, 2*NMU, NIC
where μ(I) is in Ξ if I ≤ NMU
and μ(I) is in Ξ+ if NMU < 1 ≤ 2*NMU

Record 3a: IYRTE(1),...,IYRTE(IRTECK)
This record is read only if IRTECK > 0. IYRTE(J) is the index j in 75/8.3. It is assumed that y_{j-1}, y_j and y_{j+1} are closely spaced. Note that IYRTE(1) ≥ 2 and IYRTE(IRTECK) ≤ NY-1.

Record 4: IPIRAD
This record (and record 4a if required) specifies the y levels at which irradiances, distribution functions, and reflectances are printed out. (Irradiances, etc. are automatically computed at all y-levels, e.g. for use in computing K-functions, but are printed out only at desired levels.)

IPIRAD < 0 if the irradiances, etc. are to be printed out for all y-levels

= 0 if irradiances are to be printed out only at levels y_j, j = 1, 2, 4, 6, 8,.... This is convenient when YOUT (Record 5 in Program 4) has specified closely spaced pairs of depths, as is convenient for computing K-functions (see subroutines KFCN and KRAD)

> 0 if the irradiances are to be printed out only at selected y-levels, IPIRAD in number, to be specified in record 4a

Record 4a: IYIRAD(1),...,IYIRAD(IPIRAD)
This record is read only if IPIRAD > 0. IYIRAD(j), j = 1, ...,IPIRAD, are the indices of the y_j-levels whose irradiance data is to be printed out; 1 ≤ IYIRAD(j) ≤ NY.
Record 5: IPKFCN, ISTART, ISTOP, ISTEP, JSTART, JSTOP, JSTEP

This record (and record 5a if required) specifies the computation and printout of K-functions for irradiance and radiance, using 75/8.12 and 75/8.26. The y-derivatives are estimated by using consecutive pairs of depths, i.e. \( dy = y_{j+1} - y_j = YOUT(j+1) - YOUT(j) \) if the K-function is requested at level \( j \). These derivative estimates will be inaccurate if \( y_{j+1} \) and \( y_j \) are not closely spaced — say, 0.01 optical depths apart. Thus foresight must be shown when specifying output depths in record 5 of Program 4 if K-functions are to be computed.

- **IPKFCN < 0** if irradiance (and optionally radiance) K-functions are to be computed at all possible depths \( YOUT(1), \ldots, YOUT(NY-1) \)
- **IPKFCN = 0** if the K-functions are to be computed at levels \( y_j, j = 1, 3, 5, 7, \ldots \). This is convenient if record 5 of Program 4 has selected closely spaced pairs of output depths, i.e. \( y_1 \) and \( y_2 \) are closely spaced, \( y_3 \) and \( y_4 \) are closely spaced, etc. An example of record 5 of Program 4 following this scheme is
  
  \[
  0.0, 0.01, 0.50, 0.51, 1.00, 1.01, 2.00, 2.01, \ldots 
  \]

  One could then accurately compute K-functions at levels
  
  \[
  0.005, 0.505, 1.005, 2.005, \ldots 
  \]

  by using IPKFCN = 0

- **IPKFCN > 0** if K-functions are to be computed only at selected y-levels, IPKFCN in number, to be specified in record 5a

- **ISTART, ISTOP, ISTEP** are DO-LOOP indices which select the \( \mu \)-bands of quads for which radiance K-functions are to be computed, if ISTART > 0. (ISTART, ISTOP, ISTEP) are identical in form to (IPRAD1, IPRAD2, IPRAD3) in record 2. See DO-LOOP 200 in subroutine KRAD.

- **ISTART < 0** if radiance K-functions are not to be computed

- **JSTART, JSTOP, JSTEP** are DO-LOOP indices which select \( \phi \)-bands of quads for which radiance K-functions are to be computed. See (JPRAD1, JPRAD2, JPRAD3) in record 2 and DO-LOOP 200 in subroutine KRAD.

Record 5a: IYKFCN(1),...,IYKFCN(IPKFCN)

This record is read only if IPKFCN > 0. IYKFCN(j), \( j = 1, \ldots, IPKFCN \), are the indices of the \( y_j \)-levels where the K-functions are to be computed; \( 1 \leq IYKFCN(j) \leq NY-1 \).

C. File Management

Three files are read by Program 5, and one is written.
## §6. PROGRAM 5

<table>
<thead>
<tr>
<th>symbol name</th>
<th>external name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NURAX</td>
<td>TAPE22</td>
<td>the quad-averaged geometric $\tau(a,x)$ array from Program 2</td>
</tr>
<tr>
<td>NUTXA</td>
<td>TAPE25</td>
<td>the quad-averaged geometrix $\tau(x,a)$ array from Program 2</td>
</tr>
<tr>
<td>NUIN</td>
<td>TAPE40</td>
<td>the radiance amplitudes and other information, generated by Program 4</td>
</tr>
<tr>
<td>NUOUT</td>
<td>TAPE50</td>
<td>a file containing discretized phase functions, radiances, and other information, for use by the graphics routines</td>
</tr>
</tbody>
</table>

Files NURAX and NUTXA are used only by subroutine CONTRM, which computes the contrast transmittance.
§6. PROGRAM 5

D. Code Listing

```plaintext
PROGRAM MAIN(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE4=TAPE5, TAPE22, TAPE25)

C ON NHM5/MAIN5
C
C THIS IS PROGRAM 5 OF THE NATURAL HYDROSOL MODEL
C
C THIS PROGRAM TAKES THE SPECTRAL AMPLITUDES GENERATED BY PROGRAM 4 AND SYNTHESIZES THE GEOMETRIC, QUAD-AVERAGED RADIANCE FIELDS. VARIOUS DERIVED QUANTITIES ARE ALSO COMPUTED, IF DESIRED.
C
PARAMETER(MX11=10, MXP1=24, MXS=30, MXSIGY=3)
PARAMETER(MX1=MXP1/2, MGEOP=MX1*(MX1+1), MAMP=2*M1*(M1+1))
PARAMETER(MXRTH=M1*(MX1+1), MXC=R=M1*(M1+2)/2)
PARAMETER(MXR=W=M1*(MX1+1)*(1+2*M1+5*M1))

DIMENSION IYRTE(MXS), IYPRAD(MX1), IYRAD(MX1), IYKFCN(MXS)
COMMON/C5IGY/ YSIG(MXS), TSIG(MXS), TOTALS(MXS)
COMMON/CAMP/ AAM(MX1), AYM(MXAMP, MX1), ADP(MXAMP, MX1)
COMMON/CUEOP/ GEOPP(MX1, MXGEO, MXSIGY), GEOPM(MX1, MGEOP, MXSIGY)
COMMON/CAMP/ AAM(MX1), AYM(MXAMP, MX1), AYP(MXAMP, MX1)
COMMON/CARAD/ FMU(MX1), PH1(MXPHI), VMX1)
COMMON/CRAD/ RAOAP(MX1, M1), RADP(MX1, M1, M1), RADM(MX1, MX1, MX1)
COMMON/CRADIR/ RAOAD(MX1, M1, M1, M1), RADAAD(MX1, M1, M1, M1)
COMMON/CRADIR/ RAOAP(MX1, M1), RADP(MX1, M1), RADM(MX1, M1, M1, M1),
COMMON/CRADIR/ RAOAD(MX1, M1, M1, M1)
COMMON/CIPRAD/ S1P(0:MX1), S1M(0:MX1), S2P(0:MX1), S2M(0:MX1),
COMMON/CWOR/ IM1SC(20), FM1SC(20)
COMMON/CRAD/ WERK(MXWERK)

DATA NOUT/50/

C INITIALIZE THE PROGRAM
CALL INISHL(IRTECK, IYRTE, IRAD, IYPRAD, IRAD, IYRAD, IPKFCN, IVKFCN)
C AAM AND AYM NOW CONTAIN DIFFUSE AMPLITUDES (SAME FOR AAP AND AYP)
C COMPUTE IRRADIANCE QUANTITIES FROM THE L = 0 AMPLITUDES
CALL IRRAD(IPRAD, IYRAD)
C
NY = IM1SC(4)
C
C COMPUTE THE DIFFUSE RADIANCE AT Y = A, X, ..., Z
CALL SYNRAD(AAM, RADM, MX1)
DO 100 K=1, NY
100 CALL SYNRAD(AYM(1, K), RADM(1, K), M1)
C
CALL SYNRAD(AAM, RADM, MX1)
DO 110 K=1, NY
110 CALL SYNRAD(AYP(1, K), RADM(1, K), M1)
C
C COMPUTE THE DOWNWARD DIRECT RADIANCE AT Y = A, X, ..., Z
```

112
§ 6. PROGRAM 5

CALL SYNRAD(AOAM, RADOAM, XMU)
DO 120 K = 1, NV
120 CALL SYNRAD(AOVM(I, K), RADM(I, 1, K), XMU)

C COMPUTE THE UPWARD DIRECT RADIANCE AT Y = A. THE UPWARD DIRECT
C RADIANCE IS ZERO FOR Y = X, ...., Z.
C
CALL SYNRAD(AOAP, RADOAP, XMU)
PRINT SELECTED RADIANCES AND COMPUTE THE RADIANCE-IRRADIANCE RATIOS
CALL RADV(IPIRAD, IVIRAD)
C COMPUTE THE CONTRAST TRANSMITTANCE
CALL CONTM
C COMPUTE THE K FUNCTIONS FOR IRRADIANCE
CALL KFCN(IPKFCN, IVKFCN)
C COMPUTE THE K-FUNCTIONS FOR RADIANCE
IF (ISTART.GT.0) CALL KRAD(IPKFCN, IVKFCN)
C COMPUTE THE BACKSCATTER AND FORWARD SCATTER FUNCTIONS
CALL BFSCAT(IPIRAD, IVIRAD)
C PRINT OUT THE RADIANCES
IF (IPRAD.GT.0) (CALL PRINT(IPRAD, IVPRAD)
C CHECK THE BALANCE OF THE RTE AT INTERIOR Y VALUES
IF (IRTECK.NE.0) CALL RTECK(IRTECK, IVRTE)
C SAVE THE RADIANCE INFORMATION FOR ANALYSIS BY THE PLOTTING PROGRAM

NMU = IMISC(1)
NPHI = IMISC(2)
NY = IMISC(4)
NSIGV = IMISC(5)
KCOL = IMISC(10)
C
REWIND NUOUT
WRITE(NUOUT) IM, IMUL, FMUL, FMU, PHI, Y, BMU, BNU, PHI, OMEGA, DELTMU,
1 VSIG, ALBESS, TOTALS, ZGEO
WRITE(NUOUT) ((GEOPM(I, J, K), I = 1, NMU), J = 1, KCOL), K = 1, NSIGV)
WRITE(NUOUT) ((RAAP(I, J, K), I = 1, NMU), J = 1, NPRI)
WRITE(NUOUT) ((RADP(I, J, K), I = 1, NMU), J = 1, NPRI), K = 1, NY)
WRITE(NUOUT) ((RADM(I, J, K), I = 1, NMU), J = 1, NPRI), K = 1, NY)
WRITE(NUOUT) ((RADOAP(I, J, K), I = 1, NMU), J = 1, NPRI)
WRITE(NUOUT) ((RADOAP(I, J, K), I = 1, NMU), J = 1, NPRI), K = 1, NY)
WRITE(NUOUT) ((RAAP(I, J, K), I = 1, NMU), J = 1, KCOL), K = 1, NSIGV)
C ZERO-MODE AMPLITUDE
WRITE(NUOUT) (AAAM(I), I = 1, NMU), (AAMD(I), I = 1, NMU)
1 ((AAVM(I, K), I = 1, NMU), K = 1, NV), ((ANAM(I), I = 1, NMU), K = 1, NV)
WRITE(NUOUT) (AAAM(I), I = 1, NMU), (AAMD(I), I = 1, NMU)
1 ((AAVM(I, K), I = 1, NMU), K = 1, NV), ((ANAM(I), I = 1, NMU), K = 1, NV)
ENDFILE NUOUT
WRITE(*, 200) NUOUT
200 FORMAT(1HO, ' TAPE', I2, ' WRITTEN.')
END
SUBROUTINE INISHL(IRTEK, IVRTE, IPRAD, I'PRAD, IPRAD, IVIRAD,
1 IPKFCN, IVKFCN)
C
ON NHM5/INISHL5
C
THIS ROUTINE INITIALIZES PROGRAM 5 OF THE NHM
C
PARAMETER(MXMU=10, MXPHI=24, MXV=30, MXSIGY=3)
PARAMETER(MXL=MXPHI/2, MXGEOP=MXU*(M XL+1), MXAMP=2*MXM*(MX+1))
PARAMETER(MXRRTH=MXU*(MXL+1), MACRTH=MXU*((MXL+2)/2))
C
DIMENSION IVRTE(MXV), IVPRAD(MXV), IYIRAD(MXV), IVKFCN(MXV)
DIMENSION ITITLE(IO)
COMMON/CGEOP/ GEOPP(MXMU,MXGEOP,MXSIGY) .GEOPM(MXMU,MXGEOP,MXSIGY)
COMMON/CAMP/ AAM(MXAMP), ADAM(MXAMP, MXV), ADAP(MXAMP)
COMMON/CGRID/FMU(MXMU), PHI(MXPHI), Y(MXV), BNDMU(MXMU),
1. BNDPHI(MXPHI), OMEGA(MXMU), DELTMU(MXMU), ZGEO(MXV)
COMMON/CSIGV/ YSIG(MXSIGV), ALBESS(MXSIGV), TOTALS(MXSIGV)
COMMON/CKRAD/ IPRAD1, IPRAD2, IPRAD3, JPRAD1, JPRAD2, JPRAD3
COMMON/CPRAD/ IPRAD1, IPRAD2, IPRAD3, JPRAD1, JPRAD2, JPRAD3
COMMON/CPRAD/ IPRAD1, IPRAD2, IPRAD3, JPRAD1, JPRAD2, JPRAD3
COMMON/CPRAD/ IPRAD1, IPRAD2, IPRAD3, JPRAD1, JPRAD2, JPRAD3
C
DATA NUIN/40/, IDBUG/0/
C
READ HEADER RECORDS FROM THE AMPLITUDE FILE
C
READ(NUIN) IMISC, FMISC, FMU, PHI, Y, BNDMU, BNDPHI, OMEGA, DELTMU,
1. YSIG, ALBESS, TOTALS, ZGEO
C
NMU = IMISC(1)
NPHI = IMISC(2)
NY = IMISC(4)
NSIGY = IMISC(5)
NRHAT = IMISC(6)
NRAMP = 2*NRHAT
KCOL = NMU*(NPHI/2 + 1)
C
READ(NUIN) ((GEOPP(I,J,K),I=1,NMU), J=1,KCOL,K=1,NSIGY)
READ(NUIN) ((GEOPM(I,J,K),I=1,NMU), J=1,KCOL,K=1,NSIGY)
C
READ IN PARAMETERS FOR RADIANCE ANALYSIS
C
READ(5,50) ITITLE
WRITE(6, 1000) ITITLE
C
READ SPECIFICATIONS FOR RADIANCE PRINTOUT
C
READ(5,*) IPRAD, IPRAD1, IPRAD2, IPRAD3, JPRAD1, JPRAD2, JPRAD3
IF(IPRAD.LT.0) THEN
DE 106
106 IPRAD(IY) = IY
IPRAD = NY
ELSE IF(IPRAD.GT.0) THEN
READ(5,*) (IPRAD(IY), IY=1,IPRAD)
ENDIF
C
READ SPECIFICATIONS FOR RTE FILE
C
READ(5,*) IRTEK, NIC, NJC
IF(IRTEK.LT.0) THEN
DE 108
108 IVRT E(IY) = IY
ELSE IF(IRTEK.GT.0) THEN
READ(5,*) (IVRT E(IY), IY=1,IRTEK)
ENDIF
C
§6. PROGRAM 5

C READ SPECIFICATIONS FOR IRRADIANCE OUTPUT
C READ(5,*), IPIRAD
C IF (IPIRAD.LT.0) THEN
110 IPIRAD = IV
C IPIRAD = NY
ELSEIF (IPIRAD.EQ.0) THEN
111 IPIRAD = 1
DO 112 IV = 1, NY
IPISRAD = IV + 1
ELSEIF (IPIRAD.EQ.0) THEN
112 IPIRAD = IV
ELSE
ENDIF
C READ SPECIFICATIONS FOR K-FUNCTION OUTPUT AND RADIANCE K-FUNCTION DIRECTIONS
C READ(5,*), IPKFCN, ISTART, ISTOP, JSTART, JSTOP, JSTEP
C IF (IPKFCN.LT.0) THEN
113 IPKFCN = IV
IFIPKFCN = NY / 2
ELSEIF (IPKFCN.EQ.0) THEN
IPKFCN = 0
DO 114 IV = 1, NY - 2
IPKFCN = IPKFCN + 1
114 IPKFCN = IV
ELSE
READ(5,4) (IPKFCN(IV), IV = 1, IPKFCN)
ENDIF
C IMISC(9) = IDBUG
IMISC(15) = NIC
IMISC(16) = NJC
C RECORDS WRITTEN BY MAIN4 (DIRECT BEAM)
READ(NUIN), (AAM(I), I = 1, NRAMP)
DO 15 J = 1, NY
15 READ(NUIN), (AVM(I,J), I = 1, NRAMP)
C (TOTAL BEAM)
READ(NUIN), (AAP(I), I = 1, NRAMP)
READ(NUIN), (ADV(I), I = 1, NRAMP)
C PRINTOUT
C IF (IDBUG.GT.0) THEN
WRITE(6, 1018)
WRITE(6, 1022)
(N) 116 IV = 1, NY:
THETA = RADEG*ALOS(TMU(I))
1160 WRITE(6, 1024) THETA, TMU(I)
WRITE(6, 1026)
DO 129 J = 1, NY
129 WRITE(6, 1026) J, RADEG*PHI(I)
C WRITE(6, 1030)
130 WRITE(6, 1030) J, RADEG*PHI(I)
131 WRITE(6, 1034)
132 WRITE(6, 1034) J, RADEG*PHI(I)
ENDIF
§6. PROGRAM 5

IF(IDBUG.GE.2) THEN
WRITE(6,1038)
CALL PNTAMP(Y,ADAM,ADOAM,AMXAMP)
WRITE(6,1039)
CALL PNTAMP(Y,ADOAP,1.E20,MXAMP)
WRITE(6,1040)
CALL PNTAMP(Y,AAM,AYM,MXAMP)
WRITE(6,1042)
CALL PNTAMP(Y,AAP,AVP,MXAMP)
ENDIF

CONVERT THE DOWNWARD TOTAL AMPLITUDES TO DIFFUSE AMPLITUDES
AT Y = A, X, ..., Z BY 8.23

CONVERT THE UPWARD TOTAL AMPLITUDES TO DIFFUSE AMPLITUDES AT Y = A.
THE UPWARD TOTAL = THE UPWARD DIFFUSE FOR Y = X, ..., Z.

DO 600 I=1,NRAMP
AAM(I) = AAM(I) - ADAM(I)
AAP(I) = AAP(I) - AOAP(I)
DO 600 K=1,NY
600 AYM(I,k) = AYM(I,k) - AOYM(I,k)
IF(IDBUG.GE.2) THEN
WRITE(6,1044)
CALL PNTAMP(YAAM,AYM,MXAMP)
WRITE(6,1046)
CALL PNTAMP(Y,AAP,AVP,MXAMP)
ENDIF
RETURN

FORMATS

50 FORMAT(10A8)
1000 FORMAT(1H, ' PROGRAM 5 OF THE NATURAL HYDROSOL MODEL'//
' 1 ' SYNTHESIS AND ANALYSIS OF THE RADIANCE FIELDS'//
' 2 ' RUN TITLE: ',10A8)
1018 FORMAT(1H, ' THE RADIANCE FIELDS ARE COMPUTED AT THE FOLLOWING GRID
1D VALUES:')
1022 FOMAT(1H, ' THE THETA VALUES ARE'//
' 1 ' THETA , bx , MU'//)
1024 FOMAT(1H, '.15,F10.3,F10.4)
1026 FOMAT(1H, ' THE PHI VALUES ARE'//
' J ' PHI'//)
1030 FOMAT(1H, ' OPT DEPTH'//)
1032 FOMAT(1H, ' THE Y VALUES ARE'//
' K ' Y'//)
1036 FOMAT(1H, ' THE DOWNWARD DIRECT BEAM RADIANCE AMPLITUDES ARE'//
' 1 '1X,' MU',7X,' A(MU)'//
' 1 I X , ' A*(MU)'//
' 1 I X , ' A(Y)'//
1039 FOMAT(1H, ' THE UPWARD DIRECT BEAM RADIANCE AMPLITUDES ARE'//
' 1 '1X,' MU',7X,' A(DA)'//
' 1 I X , ' A*(MU)'//
1040 FOMAT(1H, ' THE DOWNWARD TOTAL RADIANCE AMPLITUDES ARE'//
' 1 '1X,' MU',7X,' A(A)'//
' 1 I X , ' A*(MU)'//
1042 FOMAT(1H, ' THE UPWARD TOTAL RADIANCE AMPLITUDES ARE'//
' 1 '1X,' MU',7X,' A(Y)'//
' 1 I X , ' A*(MU)'//
1044 FOMAT(1H, ' THE DOWNWARD DIFFUSE RADIANCE AMPLITUDES ARE'//
' 1 '1X,' MU',7X,' A*(A)'//
' 1 I X , ' A*(MU)'//
1046 FOMAT(1H, ' THE UPWARD DIFFUSE RADIANCE AMPLITUDES ARE'//
' 1 '1X,' MU',7X,' A*(A)'//
END
SUBROUTINE BFSCAT(IPIRAD, IVIRAD)

ON NHM5/BFSCAT

THIS ROUTINE COMPUTES THE BACKSCATTER FUNCTIONS B(Z,+≈) AND B(Z,−)

USING 8.15. FORWARD SCATTER

FUNCTIONS F(Z,+≈) AND F(Z,−) ARE COMPUTED FROM EQ. 8.16.

COMPUTED VALUES ARE CHECKED USING EQ. 8.17.

THE ECCENTRICITIES ARE ALSO COMPUTED.

PARAMETER(MXMU=10, MXP+1=24, MXY=30, MXSIGY=3)
PARAMETER(MXL=MXP+1/2, MXGEOP=MXMU*(MXL+1))

DIMENSION IVIRAD(MXY)

COMMON/CGRID/ FMU(MXMU), PHI(MXPHI), Y(MXY), BNMU(MXMU),
1 BNDPHI(MXPHI), OMEGA(MXMU), DELTMU(MXMU), ZGEO(MXY)
COMMON/CRADIF/ RADAP(MXMU, MXP+1), RADP(MXMU, MXP+1),
1 RADAM(MXMU, MXP+1), RADM(MXMU, MXP+1), MXY)
COMMON/CRADIR/ RADOAP(MXMU, MXP+1), RADOAM(MXMU, MXP+1),
1 RADOAM(MXMU, MXP+1), RADOAM(MXMU, MXP+1),
COMMON/CGEOP/ GEEP(MXMU, MXGEOP, MXSIGY), GEOPM(MXMU, MXGEOP, MXSIGY)
COMMON/CINRAD/ H(P0,MXY), H(M0,MXY), CARH(P0,MXY), CARH(M0,MXY),
1 DPL(0,MXY), DMV(0,MXY)
COMMON/CSIGY/ YSIG(MXSIGY), ALBESS(MXSIGY), TOTALS(MXSIGY)
COMMON/CMISC/ IMISC(20), FMISC(20)
COMMON/CWORK/ GEOPPY(MXMU, MXGEOP), GEOMV(MXMU, MXGEOP), BZPMV(MXY),
1 BZPMV(MXY), FZPMV(MXY), FZPMV(MXY), SY(MXY)
DATA EPS/1.E-12/

NNU = IMISC(1)
NPHI = IMISC(2)
NSIGY = IMISC(4)
NPHAT = IMISC(5)
NPHI = NPHI/2
WRITE(6,300)

DO 99 IV=1,IVIRAD
IV = IVIRAD(IV)
SUMBP = 0.
SUMBM = 0.
SUMFP = 0.
SUMFM = 0.
BZP = -1.
BZM = -1.
FZP = -1.
FZM = -1.
$MFBP = -1.
BBARP = -1.
BBARM = -1.
$MBFBM = -1.

YNGW = Y(IV)

YNGW = Y(IV)

YNGW = Y(IV)

COMPUTE THE QUAD-AVERAGED GEOMETRIC PHASE FUNCTION AT THE NEEDED
Y VALUE BY LINEAR INTERPOLATION OF THE KNOWN VALUES

IF(NSIGY.EQ.1) THEN
S = TOTALS(1)
DO 56 J=1,NPHAT
55 IF(J.GT.NMUL) THEN
56 GEOPV(1,J) = GEOPP(1,J,1)
57 GEOMV(1,J) = GEOMM(1,J,1)
58 ELSE
59 IF(YNGW.GT.YSIG(NSIGY)) THEN
60 S = TOTALS(NSIGY)
61 DO 52 J=1,NPHAT
59 GEOPV(1,J) = GEOPP(1,J,NSIGY)
63 GEOMV(1,J) = GEOMM(1,J,NSIGY)
64 ELSE
65 GEOPV(1,J) = GEOPP(1,J,NSIGY)
66 GEOMV(1,J) = GEOMM(1,J,NSIGY)

117
§ 6. PROGRAM 5

ELSE
DO 55 JY = 2, NSIGY
IF(YNOW.LT.VSIG(JY)) GO TO 56
55 CONTINUE
56 DV = (YNOW - VSIG(JY))/VSIG(JY) - VSIG(JY-1)

C
S = (1.0 - DV)*TOTALS(JY-1) + DV*TOTALS(JY)
DO 58 J = 1, NRHAT
DO 58 J = 1,NHAT
GEOPPP(1,J) = (1.0 - DV)*GEOPP(I,J,JY-1) + DV*GEOPP(I,J,JY)
58 GEOPPM(1,J) = (1.0 - DV)*GEOPPM(I,J,JY-1) + DV*GEOPPM(I,J,JY)

C ENDIF
C
SV(IY) = S
DO 100 IU = 1, NMU
OM = OMEGA(IU)
IVM = NPHI
IF(IU.EQ.NMU) IVM = 1
DO 100 IV = 1, IVM
SUMBP2 = 0.
SUMBM2 = 0.
SUMFP2 = 0.
SUMFM2 = 0.

C
DO 200 IR = 1, NMU
ISMA = NPHI
IF(IR.EQ.NMU) ISMA = 1
DO 200 IS = 1, ISMA

C
IVS = IABS(IV-IS)
IF(IR.EQ.NMU) THEN
KCOL = IU
ELSE
IF(IU.EQ.NMU) THEN
KCOL = NMU
ELSE
IF(IVS.LE.NOP1) THEN
KCOL = IU + NMU*IVS
ELSE
KCOL = IU + NMU*(NOP1 - MOD(IVS,NOP1))
ENDIF ENDIF

C PP = GEOPPP(IR,KCOL)
PM = GEOPPM(IR,KCOL)

C RPTOTL = RADP(IR,IS,IY):
RMTOTL = RADM(IR,IS,IY) + RADM(IR,IS,IY)

C
SUMBP2 = SUMBP2 + RPTOTL*PM
SUMBM2 = SUMBM2 + RMTOTL*PM
SUMFP2 = SUMFP2 + RPTOTL*PM
200 SUMFM2 = SUMFM2 + RMTOTL*PM

C
SUMBP = SUMBP + QUV*SUMBP2
SUMBM = SUMBM + QUV*SUMBM2
SUMFP = SUMFP + QUV*SUMFP2
100 SUMFM = SUMFM + QUV*SUMFM2

C IF(CAPHPI(IY).LE.EPS*S*SUMBP) THEN
CAP = S/CAPHPI(IY)
FZP = SUMFP*CAP
BZP = SUMBP*CAP
SDP = HPS(IY)*CAP
SMFBP = SDP - FZP - BZP
BBARP = BZP/DPV(IY)
ENDIF
IF (CAPHM(IY) .GE. EPS*S*SUMBM) THEN
CAP = S/CAPHM(IY)
FZM = SUMFM*CAP
BZM = SUMBM*CAP
SDM = HM(IY)*CAP
SMFMBM = SDM - FZM - BZM
BBARM = BZM/DMY(IY)
ENDIF

C
BZPY(IY) = BZP
BZMY(IY) = BZM
FZPY(IY) = FZP
FZMY(IY) = FZM

99 WRITE(6,302) IV, Y(IY), ZGEO(IY), BZP, BZM, FZP, FZM, SMFMBP, SMFMBM,
1 BBARP, BBARM
C
C ECCENTRICITIES
C
WRITE(6,400)
DO 402 IV = 1, IPHAD
IV = IVIRAD(IY)
DPS1 = 1.0/(DPY(IY)*SY(IY))
DMS1 = 1.0/(DMY(IY)*SY(IY))
402 WRITE(6,410) IV, Y(IY), ZGEO(IY), BZPY(IY)*DPS1, BZMY(IY)*DMS1,
1 FZPY(IY)*DPS1, FZMY(IY)*DMS1
C
RETURN
C
FORMATS
C
300 FORMAT(1H1, '// BACKWARD AND FORWARD SCATTERING FUNCTIONS'',
1 ' (DIMENSIONS OF 1/METER)'//
2 ' IV Y ZGEO' .6X, 'B(Y,+)' .8X, 'B(Y,-)' .8X, 'F(Y,+)' .8X,
3 'F(Y,-)' .8X, '(5-F-B)(+)' .8X, '(5-F-B)(-)'.8X, 'BBAR(+)' .7X,
4 'BBAR(-)''//)
302 FORMAT(15,2F7.2,1P8.14,3)
400 FORMAT//( ' ECCENTRICITIES''// IV Y ZGEO'',
1 ' EPSB(Y,+)' EPSB(Y,-) EPSF(Y,+) EPSF(Y,-)''//)
410 FORMAT(15,2F7.2,1F13.4,3F15.4)
END

SUBROUTINE CONTAM
C
ON NHMS/CONTAM
C
THIS ROUTINE COMPUTES THE QUAD- AVERAGED GEOMETRICAL TRANSMITTANCE VIA E.42.
THE QUAD-AVERAGED GEOMETRICAL TRANSMITTANCE VIA E.42 AND TXAGEO ARE REQUIRED.
C
PARAMETER(MXMU=24, MKPHI=24, MMU=30)
PARAMETER(MXROW=MXMU*MKPHI)
C
COMMON/CRADIF/ RADAP(MXMU, MKPHI), RADAM(MXMU, MKPHI)
COMMON/CWXR( MAXROW) TXAGEO(MXMU)
C
DATA IOBUG/0/, NURAX/22/, NUTAX/35/
NMU = IMISC(1)
NPHI = IMISC(2)

READ THE FIRST NMU COLUMNS OF THE STORED TOP HALF OF RAXGEO,
BUT SAVE ONLY COLUMN NMU (THE POLAR CAP OUTPUT DIRECTION)

NROW = NMU*NPHI
NROW2 = NROW/2
REWIND NURAX
READ(NURAX) NUNIT
IF(NUNIT.NE.NURAX) THEN
WRITE(6,102) NUNIT,'NURAX',NURAX
STOP
ENDIF
DO 710 J=1,NMU
710 READ(NURAX) (RAXGEO(I),I=1,NROW2)

DEFINE THE BOTTOM HALF OF THE POLAR CAP OUTPUT DIRECTION FROM
THE TOP HALF
DO 712 I=NROW2+1,NROW
712 RAXGEO(I) = RAXGEO(I-NROW2)

READ THE FIRST NMU ROWS OF THE FIRST NMU COLUMNS, TO GET TXA(M./M.,)

REWIND NUTXA
READ(NUTXA) NUNIT
IF(NUNIT.NE.NUTXA) THEN
WRITE(6,102) NUNIT,'NUTXA',NUTXA
STOP
ENDIF
DO 720 J=1,NMU
720 READ(NUTXA) (TXAGEO(I),I=1,NMU)

EQUATION 8.33
RADOT = RADP(NMU,1,1)*TXAGEO(NMU)

RADOR = (RADOAM(NMU,1) + RADAM(NMU,1))*RAXGEO(NMU)
DO 800 JS=1,NPHI
DO 800 IR=1,NMU-1
800 RADOR = RADOR + (RADOAM(IR,JS) + RADAM(IR,JS))*
1 RAXGEO(IR+NNMU*(JS-1))

IF(IDBUG.NE.0) THEN
WRITE(6,400) (RAXGEO(I),I=1,NROW)
WRITE(6,402) (TXAGEO(I),I=1,NMU)
WRITE(6,404) RADOT,RADOR
ENDIF

TRANS = RADOT/(RADOT + RADOR)

WRITE(6,100) (TRANS
RETURN

100 FORMAT(///: THE CONTRAST TRANSMITTANCE IS T = '',F6.3///)
102 FORMAT(///: ERROR IN SUB CONTRM: NUNIT = ',',13,' AND ',A6,' = ',',13)
400 FORMAT(///: SUB CONTRM: R(R,A,X,R,5/M.,) VALUES''/(2X,1P10E12.4))
402 FORMAT(///: T(X,A,R,1/M.,) VALUES''/(2X,1P10E12.4))
404 FORMAT(///: RADOT = ',',1PE12.4,5X, RAXGEO = '1PE12.4)
SUBROUTINE IRRAD(IPIRAD, IVIRAD)

ON NHM5/IRRAD

THIS ROUTINE COMPUTES VARIOUS IRRADIANCE QUANTITIES FROM THE L = 0 TOTAL RADIANCE AMPLITUDES, USING B.5 AND B.6. IRRADIANCES ARE COMPUTED AT ALL Y LEVELS, FOR POSSIBLE USE IN COMPUTING K-FUNCTIONS, ETC., BUT PRINTOUT IS ONLY AT SELECTED Y LEVELS.

THE ZERO ELEMENT OF IRRADIANCE ARRAYS HOLDS THE VALUES FOR Y = A

PARAMETER(MXMU=10, MXPHI=24, MXY=30)
PARAMETER(MXAMP=2*MXMU*(MXPHI/2 + 1))

DIMENSION IVIRAD(MXY)

COMMON/CAMPO/ACAM(MXAMP), AOYM(MXAMP, MXY), AOAP(MXAMP)
COMMON/CAMP/ AAM(MXAMP), AAP(MXAMP), AYM(MXAMP, MXY), AYP(MXAMP, MXY)
COMMON/CGRID/ FMU(MXMU), PHI(MXPHI), Y(MXY), BNDMU(MXMU),
BNDPHI(MXPHI), OMEGA(MXPHI), DELMU(MXMU), ZGEO(MXY)
COMMON/CIRRAD/ SHP(0:MX), SHM(0:MXY), SCAPHP(0:MXY), SCAPHM(0:MXY),
DPV(0:MX), DMV(0:MXY)
COMMON/CMISC/ IMISC(20), FMISC(20)
DATA EPS/1.E-12/.

1 NMI = IMISC(1)
1 NV = IMISC(4)
1 TWOPI = 2.*FMISC(1)

COMPUTE QUANTITIES AT Y = A

HP = 0.
HM = 0.
CAPHP = 0.
CAPHM = 0.
DO 140 I=1,NMI
AMPP = ADAP(I) - AAP(I)
AMPM = AOAM(I)
DMU = DELMU(I)
HP = HP + AMPP*DMU
HM = HM + AMPM*DMU
140 CAPHP = CAPHP + AMPP*FMU(I)*DMU
140 CAPHM = CAPHM + AMPM*FMU(I)*DMU

SHP(0) = TWOPI*HP
SHM(0) = TWOPI*HM
SCAPHP(0) = TWOPI*CAPHP
SCAPHM(0) = TWOPI*CAPHM

TOTSH = SHP(0) * SHM(0)
DP = -1.0E202
DM = -1.0E202
RM = -1.0E202
IF(SCAPHP(0),GT,EPS*SHP(0)) DP = SHP(0)/SCAPHP(0)
IF(SCAPHM(0),GT,EPS*SHM(0)) DM = SHM(0)/SCAPHM(0)
IF(SCAPHP(0),GT,EPS*SCAPHP(0)) RM = SCAPHP(0)/SCAPHM(0)
WRITE(6,201)
WRITE(6,201) SHP(0), SHM(0), TOTSH, SCAPHP(0), SCAPHM(0), DP, DM, RM
IF(FTOTAL.NE.1.) WRITE(6,201)

1 DO 100 IV=1,NV
1 HP = 0.
1 HM = 0.
1 CAPHP = 0.
1 CAPHM = 0.
§6. PROGRAM 5

COMPUTE IRRADIANCES FROM AMPLITUDES

DO 150 I=1, NMU

DEFINE THE TOTAL AMPLITUDES (DIFFUSE + DIRECT) FOR L = 0

AMPP = AYP(I, IV)
AMPM = AYM(I, IV) + FTOTAL*ADYM(I, IV)
DMU = DELTMU(I)

HP = HP + AMPP*DMU
HM = HM + AMPP*DMU
CAPHP = CAPHP + AMPP*FMU(I)*DMU
CAPHM = CAPHM + AMPM*FMU(I)*DMU

150

SH-P(IY) = TWOPI*HP
SHM(IY) = TWOPI*HM
SCAPHP(IY) = TWOPI*CAPHP
SCAPHM(IY) = TWOPI*CAPHM

TOTH = SHP(IY) * SHM(IY)
DP = -1.0E202
DM = -1.0E202
RM = -1.0E202

IF(SCAPHP(IY), GT, EPS*SHP(IY)) DP = SHP(IY)/SCAPHP(IY)
IF(SCAPHM(IY), GT, EPS*SHM(IY)) DM = SHM(IY)/SCAPHM(IY)
IF(SCAPHP(IY), GT, EPS*SCAPHM(IY)) RM = SCAPHP(IY)/SCAPHM(IY)

OPY(IY) = DP
OXY(IY) = DM

CHECK FOR PRINTOUT

IPRINT = 0
DO 300 IVY=1, IPRAD
IF(IV.Y.EQ.IVIRAD(IVY)) IPRINT = 1
300 CONTINUE
IF(IPRINT.NE.0) WRITE(6,202) IVV(IVY), ZGEO(IVY), SHP(IVY), SHM(IVY), TOTH, SCAPHP(IVY), SCAPHM(IVY), DP, DM, RM
100 CONTINUE

RETURN

200 FORMAT(1H1, // ' IRRADIANCE QUANTITIES COMPUTED FROM THE L = 0 AMPLITUDES' // IV Y ZGEO', 4X , 'SCALAR H(+)', 4X, 'SCALAR H(-)', 2 6X, 'SCALAR H+', 7X, 'CAP H(+)', 7X, 'CAP H(-)', 5X, 'D(+)', 5X, 'D(-)', 8X, 1 'R(-)', /)
201 FORMAT(' ONLY THE DIFFUSE AMPLITUDES ARE USED FOR X.LE.Y.LE.Z')
202 FORMAT(15, 2F7.2, 1P5E15.4, 0P2F9.4, 1PE15.4)
203 FORMAT(10X, 'A A', 1X, 1P5E15.4, 0P2F9.4, 1PE15.4/)
§6. PROGRAM 5

SUBROUTINE KFCN(IPKFCN, IYKFCN)
C
ON NHM5/KFCN
C
THIS ROUTINE COMPUTES THE K-FUNCTIONS ASSOCIATED WITH THE SCALAR
C AND PLANE IRRADIANCES. THE FUNCTIONS ARE COMPUTED AS RATES
C OF CHANGE WITH RESPECT TO BOTH OPTICAL AND GEOMETRICAL DEPTH.
C
WARNING: EACH PAIR OF DEPTHS Y(IY) AND Y(IY+1) IS USED TO ESTIMATE
C THE K'S AT THE MIDPOINT, BUT THESE ESTIMATES MAY BE QUITE
C INACCURATE IF THE Y'S ARE NOT CLOSELY SPACED.
C
PARAMETER(MXMU=10, MXPHI=24, MXY=30)
C
DIMENSION IYKFCN(MXV)
C
COMMON/CURID/ FMU(MXMU), PHI1(MXPHI), Y(MXY).BNDMU(MXMU),
1 BNOMU(MXPHI), OMEGA(MXMU), DELMU(MXMU), ZGEO(MXY)
 COMMIN/CIRRAD/ HP(0:MXY), HM(0:MXY), CAPHP(0:MXY), CAPHM(0:MXY)
C
WRITE(6,300)
C
DO 100 IY=1,IPKFCN
IV = IYKFCN(IY)
C = -2./(Y(IY+1) - Y(IY))
YMID = 0.5*(Y(IY+1) + Y(IY))
AKP = C*(HP(IY+1) - HP(IY))/(HP(IY+1) + HP(IY))
AKM = C*(HM(IY+1) - HM(IY))/(HM(IY+1) + HM(IY))
CAPKP = C*(CAPHP(IY+1) - CAPHP(IY))/((CAPHP(IY+1) + CAPHP(IY))
CAPKM = C*(CAPHM(IY+1) - CAPHM(IY))/((CAPHM(IY+1) + CAPHM(IY))
100 WRITE(6,302) (IV,Y(IY),Y(IY+1),YMID,AKP,AKM,CAPKP,CAPKM
C
WRITE(6,400)
C
DO 500 IY=1,IPKFCN
IV = IYKFCN(IY)
C = -2./(ZGEO(IY+1) - ZGEO(IY))
ZMID = 0.5*(ZGEO(IY+1) + ZGEO(IY))
AKP = C*(HP(IY+1) - HP(IY))/(HP(IY+1) + HP(IY))
AKM = C*(HM(IY+1) - HM(IY))/(HM(IY+1) + HM(IY))
CAPKP = C*(CAPHP(IY+1) - CAPHP(IY))/((CAPHP(IY+1) + CAPHP(IY))
CAPKM = C*(CAPHM(IY+1) - CAPHM(IY))/((CAPHM(IY+1) + CAPHM(IY))
500 WRITE(6,302) ZGEO(IV),ZGEO(IV+1),ZMID,AKP,AKM,CAPKP,CAPKM
RETURN
C
300 FORMAT(1I1,' OPTICAL DEPTH K-FUNCTIONS (NONDIMENSIONAL) FOR IRRADIANCES (VALID ONLY WHEN UPPER AND LOWER ARE CLOSELY SPACED)')
2 ' UPPER (YLOWER) ',2X,'7x.
3 'K(+)' K(-) (CAP K(+)) (CAP K(-))'
302 FORMAT(9F10.3,4F10.5)
400 FORMAT(' GEOMETRIC DEPTH K-FUNCTIONS (UNITS OF 1/METER) FOR IRRADIANCES (VALID ONLY WHEN UPPER AND LOWER ARE CLOSELY SPACED)')
2 ' ZUPPER ZLOWER ',6X,'5x.
3 'K(+)' K(-) (CAP K(+)) (CAP K(-))'
END
SUBROUTINE KRAD(IPKFCN, IYKFCN)
C
C THIS ROUTINE COMPUTES THE K-FUNCTIONS FOR RADIANCES, USING 8.26B.
C FOR A SELECTED SET OF DIRECTIONS. THE FUNCTIONS ARE COMPUTED AS
C RATES OF CHANGE WITH RESPECT TO BOTH OPTICAL AND GEOMETRIC DEPTHS.
C THE PATH FUNCTION IS ALSO COMPUTED, USING 2.2 AND THE SAME DEPTH
C DERIVATIVES.
C
C WARNING: A SELECTED PAIR OF DEPTHS Y(IY) AND Y(IY+1) IS USED TO
C ESTIMATE DERIVATIVES OF THE RADIANCE AT THE MIDPOINT, BUT
C THESE ESTIMATES MAY BE QUITE INACCURATE IF THE Y LEVELS ARE
C NOT CLOSELY SPACED (E.G. 0.01 OPTICAL DEPTHS APART)
C
PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
C
DIMENSION IYKFCN(MXY)
C
COMMON/CGRID/ FMU(MXMU), PHI(MXPHI), Y(MXY), BNDMU(MXMU),
1 BNDPHI(MXPHI), OMEGA(MXPHI), DELTMLJ(MXMU), ZGEO(MXY)
COMMON/CSIGY/YSIG(MXSIGY), ALBESS(MXSIGY), TOTALS(MXSIGY)
COMMON/CRADIF/RADAP(MXMU,MXPHI), RADP(MXMU,MXPHI), RADAM(MXMU,MXPHI),
1 RADOM(MXMU,MXPHI), RADOM(MXMU,MXPHI, MXY)
COMMON/CRADIR/RADOAP(MXMU,MXPHI), RADOAM(MXMU,MXPHI),
1 RADOAM(MXMU, MXPHI, MXY)
COMMON/CKRAD/ ISTART,ISTOPISTEP, JSTART, JSTOP, JSTEP
COMMON/CMISC/ IMISC(20), FMISC(20)
C
NMU = IMISC(1)
NSIGY = IMISC(5)
RADEG = FMISC(3)
WRITE(6,300)
LINE = 5
C
DO 200 J=JSTART, JSTOP, JSTEP
PHIDEG = RADEG*PHI(J)
C
NON-POLAR QUADS
DO 200 I=ISTART,ISTOP,ISTEP
THEDEG = RADEG*ACOS(FMUC(I))
C
WRITE(6,301)
LINE = LINE + 1
DO 200 IIV=1,IPKFCN
IV = IYKFCN(IIV)
C = 1.0/(Y(IIV+1) - Y(IIV))
YMID = 0.5*(Y(IIV+1) + Y(IIV))
D = (Y(IIV+1) - Y(IIV))/(ZGEO(IIV+1) - ZGEO(IIV))
ZMID = 0.5*(ZGEO(IIV+1) + ZGEO(IIV))

GET RADIANCES, RADIANCE DERIVATIVES, AND ATTENUATION FUNCTION
AT YMID
C
RPMDV = 0.5*(RADP(I,J,IIV+1) + RADP(I,J,IIV))
RMMDV = 0.5*(RADM(I,J,IIV+1) + RADM(I,J,IIV)) +
1 RADOM(I,J,IIV)
C
DNPDV = C*(RADP(I,J,IIV+1) - RADP(I,J,IIV))
DNMDV = C*(RADM(I,J,IIV+1) - RADM(I,J,IIV)) +
1 RADOM(I,J,IIV)
C
IF(NSIGY.EQ.1 .OR. YMID.LE.YSIG(1)) THEN
ALPHA = TOTALS(1)/ALBESS(1)
ELSEIF(YMID.GE.YSIG(NSIGY)) THEN
ALPHA = TOTALS(NSIGY)/ALBESS(NSIGY)
ELSE
DO 55 JY=2,NSIGY
IF(YMID.LT.YSIG(JY)) GO TO 56
55 CONTINUE
56 DY = (YMID - YSIG(JY-1))/YSIG(JY) - YSIG(JY-1))
ALPHA = (1.0 - DY)*TOTALS(JY-1)/ALBESS(JY-1) +
1 DY*TOTALS(JY)/ALBESS(JY)
ENDIF
C
END
§6. PROGRAM 5

THE PATH FUNCTION AT YMID, USING 2.2
PATHFP = -FMU(I)*ALPHA*DNPDY + ALPHA*RPMID
PATHFM = -FMU(I)*ALPHA*DNMDY + ALPHA*RMMID

THE K-FUNCTIONS AT YMID, USING 8.2bB
FKP = -DNPDY/RPMID
FKM = -DNMDY/RMMID

WRITE(6,302) I,J,THEDEG,PHIDEG,Y(IY),Y(IY+1),YMID,RPMID,RMMID,
1 PATHFP,PATHFM,FKP,FKM,ZMID,D*FKP,D*FKM
LINE = LINE + 1
IF(LINE.GT.58) THEN
WRITE(6,300)
LINE = 5
ENDIF
200 CONTINUE

WRITE(6,301)
LINE = LINE + 1
DO 100 IY=1,IPKFCN
IV = IKFCN(IY)
C = 1.0/(Y(IY)+1) - Y(IY))
YMID = 0.5*(Y(IY+1) + Y(IY))
D = (Y(IY+1) - Y(IY))/(ZGEO(IY+1) - ZGEO(IY))
ZMID = 0.5*(ZGEO(IY+1) + ZGEO(IY))

GET RADIANCES, RADIANCE DERIVATIVES, AND ATTENUATION FUNCTION AT YMID
RPMID = 0.5*(RADP(NMU,1,IY+1) + RAUP(NMU,1,IY))
RMMID = 0.5*(RAUM(NMU,1,1Y+1) + RADUM(NMU,1,1Y))
D + RADOM(NMU,1,1Y)

DNPDY = C*(RADP(NMU,1,IV+1) - RAU(NMU,1,IV)))
DNMDY = C*(RAUM(NMU,1,IV+1) - RADUM(NMU,1,1Y+1) - I RADOM(NMU,1,1Y))

IF(NSIGV.EQ.1) THEN YMID.LE.YSIG(1) THEN
ALPHA = TOTALS(JY)/ALBESS(JY)
ELSEIF(YMID.GE.YSIG(NSIGV)) THEN
ALPHA = TOTALS(JY)/ALBESS(JY)
ELSE
DO 57 JY=2,NSIGV
IF(YMID.LT.YSIG(JY)) GO TO 58
57 CONTINUE
58 DV = (YMID - YSIG(JY-1))/(YSIG(JY) - YSIG(JY-1))
ALPHA = (1.0 - DV)*TOTALS(JY-1)/ALBESS(JY-1) + 1 DV*TOTALS(JY)/ALBESS(JY)
ENDIF

THE PATH FUNCTION AT YMID, USING 2.2
PATHFP = -FMU(I)*ALPHA*DNPDY + ALPHA*RPMID
PATHFM = -FMU(I)*ALPHA*DNMDY + ALPHA*RMMID

THE K-FUNCTIONS AT YMID, USING 8.2bB
FKP = -DNPDY/RPMID
FKM = -DNMDY/RMMID

RETURN
§6. PROGRAM 5

300 FORMAT(1H1, /* RADIANCES, PATH FUNCTIONS, AND RADIANCE K-FUNCTIONS */
1 FOR SELECTED DIRECTIONS (VALID ONLY WHEN YUPPER AND YLOWER ARE CLOSELY SPACED) /* T90, NONDIMENSIONAL, I/1X, DIMENSIONAL (I/M) */
3 /* I J THETA PHI YUPPER YLOWER Y RAD+(Y) */
4 /* I J THETA PHI YUPPER YLOWER Y RAD-(Y) */
5 /* ZGEO K(+) K(-) */
301 FORMAT(1H1)
END

SUBROUTINE PRINT(IPRAD,IYPRAD)
ON NHM5/PRINT

THIS ROUTINE PRINTS OUT THE FINAL RADIANCE FIELDS AT SELECTED Y LEVELS
PARAMETER(MXMU=10, MXP=24, MXY=30)

DIMENSION IPRAD(MXY)
COMMON/CRADIF/RADAP(MXMU,MXP),RADP(MXMU,MXP,MXY),
1 RADAM(MXMU,MXP),RADM(MXMU,MXP,MXY)
COMMON/CRADIF/RADOAP(MXMU,MXP),RADOAM(MXMU,MXP)
1 RADOAM(MXMU,MXP,MXY)
COMMON/CGRID/ FMU(MXMU),PHI(MXP),Y(MXY),BNDMU(MXMU),
1 BNDPHI(MXP),OMEGA(MXMU),DELMU(MXMU),ZGEO(MXY)
COMMON/CPRAD/IPRAD1,IPRAD2,IPRAD3,IPRAD1,IPRAD2,IPRAD3
COMMON/CMISC/ IMISC(20),FMISC(20)
COMMON/CWORK/ THEDEG(MXMU),PHIDEG(MXP)

C NMU = IMISC(I)
C NPHI = IMISC(J)
C RADEG = FMISC(J)

C CONVERT MU AND PHI TO DEGREES
DO 50 I=1,NMU
50 THEDEG(I) = RADEG*ACOS(FMU(I))
DO 51 J=1,NPHI
51 PHIDEG(J) = RADEG*PHI(J)

C WRITE RADIANCE AT Y = A
WRITE(6,300)
DO 302 I=IPRAD1,IPRAD2,IPRAD3
WRITE(6,102)
302 WRITE(6,304) I,J,THEDEG(I),PHIDEG(J),RADAP(I,J),RADOAP(I,J),
1 RADAM(I,J)
WRITE(6,102)
I=NMU
J=1
WRITE(6,304) I,J,THEDEG(I),PHIDEG(J),RADAP(I,J),RADOAP(I,J),
1 RADAM(I,J)

126
WRITE RADIENCES AT $y = x, \ldots, z$

WRITE(6, 100)
DO 101 IYV=1,IPRAD
IY = (IPRAD(IYV))
WRITE(6, 110)
DO 103 I=IPRAD1,IPRAD2,IPRAD3
WRITE(6, 102)
DO 103 J=JPRAD1,JPRAD2,JPRAD3
103 WRITE(6, 104) I, J, IV, THEDEG(I), PHIDEG(J), Y(IV), ZGEO(IV),
RADP(I, J, IV), RADM(I, J, IV), RADOM(I, J, IV)
I = NMU
J = 1
WRITE(6, 104) I, J, IV, THEDEG(I), PHIDEG(J), Y(IV), ZGEO(IV),
RADP(I, J, IV), RADM(I, J, IV), RADOM(I, J, IV)
101 CONTINUE
C
RETURN
C
100 FORMAT(1HL, ' THE FINAL DIFFUSE AND DIRECT RADIENCES AT INTERIOR Y
VALUES ARE'/)
102 FORMAT(1HL )
104 FORMAT(314,2F9.3,2F8.3,1P3E15.5)
110 FORMAT(1HL, ' I, J, K, THETA', 4X, 'PHI', 7X, 'Y, ZGEO', 6X,
7HRAD(+), 8X, 7HRAD(-))
300 FORMAT(1HL, ' THE FINAL DIFFUSE AND DIRECT RADIENCES AT Y = A ARE'/
1/ ' I, J, K, THETA', 4X, 'PHI', 8X, 7HRAD(+), 8X, 7HRAD(-))
304 FORMAT(214,2F9.3,1P3E15.5)
C
END
SUBROUTINE RADOV(IPIRAD, IVIRAD)

C
ON NHM5/RADV

C
THIS ROUTINE PRINTS SELECTED RADIANCES (UP, DOWN, AND HORIZONTAL
ALONG-WIND AND CROSS-WIND). RADIANCE-IRRADIANCE RATIOS ARE ALSO
COMPUTED.

PARAMETER(MXMU=10, MXPHI=24, MXY=30)

DIMENSION IVIRAD(MXY)

COMMON/CRADIF/RADAP(MXMU, MXPHI), RADP(MXMU, MXPHI, MXY),
1 RADOM(MXMU, MXPHI, MXY)
COMMON/CRADIR/RADOAP(MXMU, MXPHI), RADOAM(MXMU, MXPHI),
1 RADOM(MXMU, MXPHI, MXY)
COMMON/CGRID/FMU(MXMU), PH1(MXPHI), Y(MXY), BNDMU(MXMU),
1 BNDPHI(MXPHI), OMEGA(MXMU), DELMU(MXMU), ZGEO(MXY)
COMMON/CIRRAD/ SHP(0, MXY), SHM(0, MXY), SCAPHM(0, MXY), SCAPHP(0, MXY)
COMMON/CMISC/ IMISC(20)

NMU = IMISC(1)
NPHI = IMISC(2)

Y = A

RADUP = RADOAP(NMU,1) + RADAP(NMU,1)
RADON = RADOAM(NMU,1) + RADAM(NMU,1)
RHO = 0.5*(RADOAP(1,1) + RADAP(1,1) + RADAM(1,1) + RADOM(1,1))
J90 = NPHI/4 + 1
RH90 = 0.5*(RADOAP(1,J90) + RADAP(1,J90) + RADAM(1,J90)
1 RADOM(1,J90))
J180 = NPHI/2 + 1
RH180 = 0.5*(RADOAP(1,J180) + RADAP(1,J180) + RADAM(1,J180)
1 RADOM(1,J180))
Rho = RADUP/RSCAPHM(0)
Qm = RSCAPHP(0)/RADUP
WRITE(6,100)
WRITE(6,102) RADUP,RADON, RHO, RH90, RH180, RN, QM

DO 200 IV=1, IVIRAD
IV = IVIRAD(IV)
RADUP = RADOAP(NMU,1,IV) + RADAP(NMU,1,IV)
RADON = RADOM(NMU,1,IV) + RADAM(NMU,1,IV)
RHO = 0.5*(RADOAP(1,1,IV) + RADAP(1,1,IV) + RADAM(1,1,IV)
1 RADOM(1,1,IV))
RH90 = 0.5*(RADOAP(1,J90,IV) + RADAP(1,J90,IV) + RADAM(1,J90,IV)
1 RADOM(1,J90,IV))
RH180 = 0.5*(RADOAP(1,J180,IV) + RADAP(1,J180,IV) + RADAM(1,J180,IV)
1 RADOM(1,J180,IV))
Rho = RADUP/RSCAPHM(IV)
Qh = RSCAPHP(0)/RADUP
WRITE(6,104) IV, Y(IV), ZGEO(IV), RADUP, RADON, RHO, RH90, RH180, RN, QM

RETURN

100 FORMAT(10X*, "SELECTED RADIANCES AND RADIANCE-IRRADIANCE RATIOS'/'
1: Y(1) = ZGEO(N(Y,M,)), NH(NH(Y,0),2), NH(NH(Y,180),2)
102 FORMAT(10X*, "A", 1P7E15.4)
104 FORMAT(15,2F7.2,1P7E15.4)
END
SUBROUTINE RTECK(INTELC, Y(NY))

ON NHMS/RTECK

THIS ROUTINE CHECKS THE FINAL TOTAL RADIANCES BY SEEING IF THEY
SATISFY THE QUAD-AVERAGED RADIATIVE TRANSFER EQUATION 3.12 AT
INTERIOR Y VALUES, X.LT.Y.LT.Z

IRTECK.LT.0 IF THE BALANCE OF THE RTE IS TO BE COMPUTED AT ALL
POSSIBLE INTERIOR Y LEVELS, Y(2), ..., Y(NY-1)

IRTECK.EQ.0 IF NC RTE BALANCE CHECK IS TO BE MADE

IRTECK.GT.0 IF THE RTE BALANCE IS COMPUTED AT THE Y LEVELS GIVEN BY
Y(IVRTE(1)), ..., Y(IVRTE(IRTECK))

NIC, NJC... ARE USED TO SELECT PARTICULAR MU AND PHI VALUES

WHERE THE RTE BALANCE CHECK IS TO BE MADE, IF IRTECK.NE.0
MU(I) AND PHI(J) ARE CHECKED, WHERE

DO J=1,NPHI,NJC
DO I=1,2*NMU,NIC

AND Y(J) IS IN X(-) IF I.LE.NMU
AND Y(I) IS IN X(*) IF I.GT.NMU, AND, I.LE.2*NMU

C WARNING: DV/DY IS COMPUTED USING A CENTERED DIFFERENCE. IF THE Y
VALUES ARE NOT EVENLY SPACED OR IF THEY ARE FAR APART, THIS
ESTIMATE OF THE DERIVATIVE MAY BE QUITE INACCURATE, CAUSING A POOR
BALANCE OF THE RTE EVEN THOUGH THE COMPUTED RADIANCES ARE CORRECT.

PARAMETER(MXMU=10, MXPFI=24, MXY=30, MXSIGY=3)
PARAMETER(MXL=MAX(AU/2, MXGEO+MAX(MXU)(1)'))

COMMON/CRADIF/RADAP(MXMU,MXPFI),RADP(MXMU,MXPFI,MY),
RADAM(MXMU,MXPFI,MY),RADM(MXMU,MXPFI,MY)

COMMON/CRADIR/RADAP(MXMU,MXPFI),RADAP(MXMU,MXPFI),RADUAM(MXMU,MXPFI),
RADUAM(MXMU,MXPFI,MY)

COMMON/CGMG/MUG(MXMU,MXPFI),PHI(MXMU,MXPFI,MY)

COMMON/CGMEO/MGEO(MXMU,MXPFI,MY),MGEO(MXMU,MXPFI,MY)

COMMON/CSIG/MSIGY(MAXSIGY),ALBES(MXSIGY)

COMMON/CMISC/IMISC(20),FMISC(20)

COMMON/CWAVE/CMWAPY(MAXMU,MXPFI,UTOPVY(MXMU,MXPFI),MXGEO)

DIMENSION(YK(YEY))

NMU = IMISC(1)
NPHI = IMISC(2)
NY = IMISC(4)
NSIG = IMISC(5)
NPHAT = IMISC(6)
NL = IMISC(15)
NJC = IMISC(16)
NMJ2 = NMU/2
VOPJ = NPHI/2

IF(IRTECK.LT.0 THEN
IVO = 2
IVMAX = NY-1
ELSE IF(IRTECK.GT.0 THEN
IVO = 1
IVMAX = 1
ELSE
RETURN
ENDIF
WHILE Y<.001

C DO 100 I=IVO,IVMAX,1

IC = I+1, AND IC LABELS THE MU, PHI, AND Y VALUES FOR WHICH THE RTE
IS EVALUATED
IL=1 FOR STANDARD RADIANCE
IL=8 FOR PLANAR RADIANCE

DO 100 IVY=1,IVMAX
IVH = Y(ICY)

100 CONTINUE
C DEFINE THE ALBEDO AND PHASE FUNCTION AT THE NEEDED Y VALUE BY
C LINEAR INTERPOLATION OF THE KNOWN VALUES

C IF(NSIGY.EQ.1 .OR. YNOW.LE.YSIG(1)) THEN
ALBEDO = ALBESS(1)
DO 50 J=1,NRHAT
DO 50 I=1,NMU
GEOPPY(I,J) = GEOPP(I,J,1)
50 GEOPMY(I,J) = GEOPM(I,J,1)
C ELSEIF(YNOW.GE.YSIG(NSIGY)) THEN
ALBEDO = ALBESS(NSIGY)
DO 52 J=1,NRHAT
DO 52 I=1,NMU
GEOPPY(I,J) = GEOPP(I,J,NSIGY)
52 GEOPMY(I,J) = GEOPM(I,J,NSIGY)
C ELSE
DO 55 JY=2,NSIGY
IF(YNOW.GT.YSIG(JY)) GO TO 56
55 CONTINUE
56 DY = (YNOW - YSIG(JY-1))/(YSIG(JY) - YSIG(JY-1))
C ALBEDO = (1.0 - DY)*ALBESS(JY-1) + DY*ALBESS(JY)
DO 58 J=1,NRHAT
DO 58 I=1,NMU
GEOPPY(I,J) = (1.0 - DY)*GEOPP(I,J,JY-1) + DY*GEOPP(I,J,JY)
58 GEOPMY(I,J) = (1.0 - DY)*GEOPM(I,J,JY-1) + DY*GEOPM(I,J,JY)
ENDIF
C DO 300 JC=1,NPHI,NJC
C IC2 = I/J,YC
IF(IC2.LE.NMU) THEN
IC = -IC2
ELSEIF(IC2.GT.NMU AND. 1C2.LE.NMU2) THEN
IC = IC2 - NMU
ELSE
IC = 0 NOT VALID
GO TO 300
ENDIF
IICA = IABS(IC)
IF(IICA.EQ.NMU .AND. JC.NE.1) GO TO 300
IF(IC2.GT.0) THEN
FMUIC = FMU(IICA)
ELSE
FMUIC = -FMU(IICA)
ENDIF
C EVALUATE THE TERMS OF THE RTE
C TERM1 = MU*DN/DY
C IF(IC.GT.0) THEN
TERM1 = RADM(IICA,JC,1YC) - RADM(IICA,JC,1YC+1)
ELSE
TERM1 = RADM(IICA,JC,1YC) - RADM(IICA,JC,1YC+1) + 1
RADDM(IICA,JC,1YC+1) - RADDM(IICA,JC,1YC+1)
ENDIF
TERM2 = FMUIC*TERM1/(Y(1YC) - Y(1YC+1))
C TERM2 = -N
C IF(IC.GT.0) THEN
TERM2 = -RADM(IICA,JC,1YC)
ELSE
TERM2 = -RADM(IICA,JC,1YC) - RADDM(IICA,JC,1YC)
ENDIF
§ 6. PROGRAM 5

\[ \text{TERM3} = \text{(ALBEDO OF SINGLE SCATTERING)} \times \text{INTEGRAL OF (RADIANCE \times PHASE FUNCTION)} \]

\[ = \text{ALBEDO} \times \text{PATH FUNCTION} \]

\[
\begin{align*}
\text{TERM3} &= 0. \\
\text{DO 700 IR} &= 1, \text{NMU} \\
\text{ISMAX} &= \text{NPHI} \\
\text{IF}(\text{IR}, \text{EQ}, \text{NMU}) &\Rightarrow \text{ISMAX} = 1 \\
\text{DO 700 IS} &= 1, \text{ISMAX} \\
\text{RP} &= \text{RDPD} \text{IR}, \text{IS}, \text{IVC} \\
\text{RM} &= \text{RADM} \text{IR}, \text{IS}, \text{IVC} + \text{RAUM} \text{IR}, \text{IS}, \text{IVC} \\
\text{COMPUTE STORAGE INDEX FOR GEOPP AND GEOPM} \\
\text{IVS} &= \text{IABS} \text{JC} - \text{IS} \\
\text{IVINDX} &= \text{IVS} + 1 \\
\text{IF}(\text{IVS, GT}, \text{NPPI}) &\Rightarrow \text{IVINDX} = \text{NPPI} + 1 - \text{MOD} \text{IVS, NPPI} \\
\text{KCOL} &= \text{NMU} \times (\text{IVINDX} - 1) + \text{ICA} \\
\text{IF}(\text{ICA, EQ}, \text{NMU}) &\Rightarrow \text{KCOL} = \text{NMU} \\
\text{PP} &= \text{GEOPPY} \text{IR}, \text{KCOL} \\
\text{PM} &= \text{GEOPM} \text{IR}, \text{KCOL} \\
\text{IF}(\text{IC}, \text{GT}, \text{U}) \Rightarrow \text{TERM3} &= \text{TERM3} + \text{KM} \times \text{PM} + \text{RP} \times \text{PP} \\
\text{ELSE} &\Rightarrow \text{TERM3} &= \text{TERM3} + \text{KM} \times \text{PP} + \text{RP} \times \text{PM} \\
\text{ENDIF} \\
\text{700 CONTINUE} \\
\text{TERM3} &= \text{ALBEDO} \times \text{TERM3} \\
\text{OUTPUT} \\
\text{SUM} &= \text{TERM1} + \text{TERM2} + \text{TERM3} \\
\text{WRITE(6,201) IC,JC,IVC,TERM1,TERM2,TERM3,SUM} \\
\text{300 CONTINUE} \\
\text{FORMATS} \\
\text{200 FORMAT(1X,'CHECK ON CONVERGENCE TOTAL RADIANCES:')} \\
\text{16X,'EVALUATION OF THE RADIATIVE TRANSFER EQUATION FOR SELECTED MU,} \\
\text{2 MU, PM1 AND Y VALUES:')} \\
\text{3 X, MU, PM1, Y,5X,'MU*ON/DY',9X, = N,5X, + INT(N*SIGMA)/A ,} \\
\text{4 5X, = ZERO')} \\
\text{201-FORM1(315,IP4*15.8)} \\
\text{RETURN} \\
\text{END} \]
SUBROUTINE SYNRA(DAMP,RAD,IR)O

ON NHM5/SYNRA(D

THIS ROUTINE SYNTHESIZES THE RADIANCE FIELD R(MU,PHI) (FOR A
GIVEN Y VALUE) USING 5.3 AND 5.4.

PARAMETER(MXMU=10, MXPHI=24)
PARAMETER(MXL=MXPHI/2)

DIMENSION AMP(1),RAD(IR,1)
DIMENSION COSLOP(0:MXL, MXPHI),SINLP(0:MXL, MXPHI)
COMMON/CORID/ FMU(MXMU),PHI(MXPHI)
COMMON/CMISC/ IMISC(20)

DATA KALL/0/

IF(KALL.EQ.0) THEN

THE FIRST CALL DOES INITIALIZATION

NMU = IMISC(1)
NPHI = IMISC(2)
NL = IMISC(3)
NRHAT = IMISC(10)

DO 50 L=0,NL
DO 50 J=1,NPHI
COSLP(L,J) = COS(FLOAT(L)*PHI(J))
50  SINLP(L,J) = SIN(FLOAT(L)*PHI(J))
KALL = 1
ENDIF

LOOP OVER ALL MU AND PHI VALUES

DO 100 I=1,NMU-1
DO 100 J=1,NPHI-1

SUM OVER L VALUES, EQ 5.4

SUM = 0.
DO 200 L=0,NL
200 SUM = SUM + AMP(NMU*L+1)*COSLP(L,J)
1 + AMP(NRHAT+NMU*L+11)*SINLP(L,J)
100 RAD(I,J) = SUM

POLAR CAP TERM BY 5.4
RAD(NMU,1) = AMP(NMU)
DO 102 J=2,NPHI
102 RAD(NMU,J) = 0.

RETURN
END
§7A. PLOTTING RADIANCES

7. GRAPHICS PROGRAMS

The running of the Natural Hydrosol Model is completed with the computations of Program 5. TAPE50, written by Program 5, contains the computed radiances and other information. However, the most convenient form for the output is often graphical. We therefore include in this report a few programs for plotting radiance distributions, chromaticity diagrams, and the like.

Each of the listed programs uses standard CalComp Basic Software, as implemented on the author's CDC Cyber 855 computer. This implementation uses both TAPE98 and TAPE99 in order to generate output files for both videoterminal and hardcopy plot devices. This is non-standard, but only minor rewriting will be required to use the programs on other computer systems.

A. Plotting Radiance Distributions

Program MPRAD reads the radiance data from TAPE50 and plots radiance distributions, as a function of depth and direction, on a variety of formats.

1. Input

Each plot is generated by two to four free-format records.

Record 1: ITYPE, NTIT, NYPLT

- **ITYPE** specifies the type of plot to be made, as described in record 2, below.
- **NTIT**
  - \( \leq 0 \) if no title is desired at the top of the plot
  - \( > 0 \) if an alphanumerical title for the top of the plot is to be read in record 1a
- **NYPLT**
  - \( \leq 0 \) if all y-levels are to be plotted
  - \( > 0 \) if only selected y-levels, NYPLT in number, are to be plotted, as specified in record 1b

Record 1a: ITITLE

This record is read only if ITIT \( > 0 \). ITITLE is an alphanumerical title for the top of the plot. Up to 80 characters are allowed.

Record 1b: IYPLT(1),...,IYPLT(NYPLT)

This record is read only if NYPLT \( > 0 \). The values of IYPLT are the J indices of YOUT(J), \( J = 1,\cdots,NY \) at which plots are to be made (cf. record 5 of Program 4).
§7A. PLOTTING RADIANCES

Record 2: depends on ITYPE

This record is the specification record. It gives the values of the parameters needed to specify the details of the plot, as follows:

If ITYPE = 1, make polar plots of the logarithm of the diffuse radiance as a function of $\theta$. The specification record gives JPHI and JPI, which are the $\phi$-indices of two half-planes. Normally $\phi(JPI) = \phi(JPHI) + \pi$, so that a planar cross section of the radiance is plotted. A separate plot is made for each depth.

If ITYPE = 2, make polar plots of the logarithm of the total radiance as a function of $\theta$. Otherwise as for ITYPE = 1.

If ITYPE = 3, plot the logarithm of the diffuse radiance as a function of $\theta$. The specification record gives JPHI and JPI as for ITYPE = 1. All depths are on the same plot.

If ITYPE = 4, plot the logarithm of the total radiance as a function of $\theta$, otherwise as for ITYPE = 3. All depths are on the same plot.

If ITYPE = 5, make a polar plot of the diffuse radiances as a function of $\phi$. The specification record gives ITHETA, the index defining a particular $\theta$-cone:
   - If ITHETA > 0, upward radiances are plotted
   - If ITHETA < 0, downward radiances are plotted
   A separate plot is made for each depth.

If ITYPE = 6, make a polar plot of the total radiances as a function of $\phi$, otherwise as for ITYPE = 5.

If ITYPE = 7, plot the logarithm of the diffuse radiances as a function of $\phi$. The specification record gives ITHETA as for ITYPE = 5. All depths are on the same plot.

If ITYPE = 8, plot the logarithm of the total radiances as a function of $\phi$, otherwise as for ITYPE = 7.

If ITYPE = 9, plot the logarithm of the total path function as a function of $\theta$. The specification record gives JPHI and JPI as for ITYPE = 1. All depths are on the same plot.

Note: ITYPE = 5, 6, or 9 cannot be used in the listed code, since the required subroutines PPHIPLR and PPATH have not been written as of the date of compilation of this report.
§7A. PLOTTING RADIANCES

2. Code Listing

```fortran
PROGRAM MPRAD( INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE50, 
              1 TAPE98, TAPE99)

ON NHM6/MPRAD

THIS PROGRAM CONTROLS THE PLOTTING OF THE RADIANCES, USING THE 
FILE OF RADIANCE DATA WRITTEN BY PROGRAM 5 (TAPE50).

ALL PLOTTING IS DONE USING STANDARD CALCOMP CALLS 
(TAPE98 AND TAPE99 ARE USED BY THE CALCOMP ROUTINES, AS 
implemented on the Author's CDC CYBER 655 COMPUTER.)

EACH PLOT IS GENERATED BY TWO TO FOUR FREE-FORMAT DATA RECORDS.

THE FIRST RECORD GIVES ITYPE, NTIT, NVPLT WHERE 
ITYPE SPECIFIES THE TYPE OF PLOT TO BE MADE, AS DESCRIBED BELOW. 
NTIT .LE. 0 IF NO TITLE FOR THE TOP OF THE PLOT IS DESIRED 
.GT. 0 IF A TITLE FOR THE TOP OF THE PLOT IS TO BE READ IN 
NVPLT .LE. 0 IF ALL Y LEVELS ARE TO BE PLOTTED 
.GT. 0 IF NVPLT Y LEVELS ARE TO BE PLOTTED 

IF NTIT.GT.0, THE NEXT RECORD GIVES THE DESIRED TITLE

IF NVPLT.GT.0, THE NEXT RECORD GIVES THE INDICES OF THE Y 
LEVELS FOR THE PLOTTING

THE LAST RECORD, THE SPECIFICATION RECORD, GIVES THE VALUES OF 
THE PARAMETERS NEEDED TO SPECIFY THE DETAILS OF THE PLOT

IF ITYPE.EQ.1, MAKE POLAR PLots OF THE LOGARITHM OF THE DIFFUSE 
RADIANCE AS A FUNCTION OF THETA. THE SPECIFICATION 
RECORD GIVES JPHI AND JPI, WHICH ARE THE 
PHI INDICES OF THE HALF-PLANES. NORMALLY PHI(JPI) = 
PHI(JPHI) + PI, SO THAT A PLANAR CROSS SECTION OF THE 
RADIANCE IS PLOTTED.
A SEPARATE PLOT IS MADE FOR EACH DEPTH.

IF ITYPE.EQ.2, MAKE POLAR PLots OF THE LOGARITHM OF THE TOTAL 
RADIANCE AS A FUNCTION OF THETA, OTHERWISE AS FOR 
ITYPE = 1.

IF ITYPE.EQ.3, PLOT THE LOGARITHM OF THE DIFFUSE RADIANCE AS A FUNCTION OF 
THETA. THE SPECIFICATION RECORD GIVES JPHI AND JPI 
AS FOR ITYPE = 1.
ALL DEPTHS ARE ON THE SAME PLOT.

IF ITYPE.EQ.4, PLOT THE LOGARITHM OF THE TOTAL RADIANCE AS A FUNCTION OF 
THETA, OTHERWISE AS FOR ITYPE = 3.
ALL DEPTHS ARE ON THE SAME PLOT.

IF ITYPE.EQ.5, MAKE A POLAR PLOT OF THE DIFFUSE RADIANCES AS A 
FUNCTION OF THETA. THE SPECIFICATION RECORD GIVES 
ITHETA, THE INDEX DEFINING A PARTICULAR THETA CONE:
IF ITHETA.GT.0, UPWARD RADIANCES ARE PLOTTED
IF ITHETA.LT.0, DOWNWARD RADIANCES ARE PLOTTED
A SEPARATE PLOT IS MADE FOR EACH DEPTH.

IF ITYPE.EQ.6, MAKE A POLAR PLOT OF THE TOTAL RADIANCES AS A 
FUNCTION OF PHII, OTHERWISE AS FOR ITYPE = 5.

IF ITYPE.EQ.7, PLOT THE LOGARITHM OF THE DIFFUSE RADIANCES AS A 
FUNCTION OF PHII. THE SPEC RECORD GIVES ITHETA AS 
FOR ITYPE = 5.
ALL DEPTHS ARE ON THE SAME PLOT.

IF ITYPE.EQ.8, PLOT THE LOGARITHM OF THE TOTAL RADIANCES AS A
```
FUNCTION OF PHI, OTHERWISE AS FOR ITYPE = 7.

C IF ITYPE.EQ.9, PLOT THE LOGARITHM OF THE TOTAL PATH FUNCTION AS A
FUNCTION OF THETA. THE SPEC REC GIVES JPHI AND
JPI AS FOR ITYPE = 1.
ALL DEPTHS ARE ON THE SAME PLOT.

C**** WARNING: ITYPE = 5, 6 OR 9 CANNOT BE USED, SINCE THE REQUIRED
SUBROUTINES PPPIPLR AND PPATH HAVE NOT YET BEEN WRITTEN (CM, 3 JUNE 88).

C PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
C PARAMETER(MXL=MXPHI/2, MXGEOP=MXMU*(MYL+1))
C COMMON/CSIGY/ YSIG(MXSIGY), ALBESS(MASIGy), TOTALS(MXSIGY)
C COMMON/CGEOP/ GEOP(MXMU, MXGEOP, MXSIGY), GEOPM(MXMU, MXGEOP, MXSIGY)
C COMMON/CGGRID/ THEA(MXMU), PHI(MXPHI), Y(MXY), BNDMU(MXMU).
1 BNDPHI(MXPHI), OMEGA(MXMU), DELTMU(MXMU), ZGEO(MXY)
C COMMON/CRADIF/ RADAP(MXMU, MXPHI), RADP(MXMU, MXPHI, MXY).
1 RADAM(MXMU, MXPHI), RADAO(MXMU, MXPHI, MXY)
C COMMON/CRADIR/ RADAP(MXMU, MXPHI), RADAP1(MXMU, MXPHI).
1 RADAM(MXMU, MXPHI, MXY)
C COMMON/CMISC/ IMISC(20), FMISC(20), NTIT, ITITLE(8)
C COMMON/CWORK/ WORK(5000)
C DIMENSION FMU(MXMU), IVPLT(MXY)
C DATA NUIN/50/, FPS/I.OE-10/
C INITIALIZE THE LALCOMP PLOTTING ROUTINES
CALL PLOTT
C C READ THE RADIANCE DATA WRITTEN BY PROGRAM 5

C REWIND NUIN
READ(NUIN) IMISC, FMISC, FMU, PHI, Y, BNDMU, BNDPHI, OMEGA, DELTMU,
1 YSIG, ALBESS, TOTALS, ZGEO
C NMU = IMISC(1)
NPHI = IMISC(2)
NY = IMISC(4)
NSIGY = IMISC(5)
KCOL = IMISC(10)
C READ(NUIN) (((GEOMP(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
READ(NUIN) (((GEOMP(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
READ(NUIN) (((RADP(I,J),J=1,NMU),J=1,NPHI)
READ(NUIN) (((RADP(I,J,K),J=1,NMU),J=1,NPHI)
READ(NUIN) (((RADP1(I,J,K),J=1,NMU),J=1,NPHI)
READ(NUIN) (((RADP1(I,J,K),J=1,NMU),J=1,NPHI)
C DIRECT RADIANCES WHICH SHOULD BE ZERO ARE SOMETIMES NEGATIVE
DUE TO ROUNDOFF ERROR: RESET TO ZERO
DO 60 K=1,NV
DO 53 I=1,NMU
IF(RADOM(I,J,K).LT.EPS) RADOM(I,J,K) = 0.
60 CONTINUE
C CONVER FMU TO THETA
DO 53 I=1,NMU
53 THEA(I) = ACOS(FMU(I))
C READ RECORDS DESCRIBING THE PLOTS
C THE FIRST RECORD:
100 READ(5,*,END=200) ITYPE, NYPLT, NTIT
C THE TITLE RECORD, IF REQUESTED
IF(NTIT.GT.0) THEN
READ(5,70) ITITLE
NTIT = NCHAR(ITITLE,8)
ENDIF
§7A. PLOTTING RADIANCES

IF(NVPLT.LE.0) THEN
  DO 300 IY=1,NV
  300 IVPLT(IY) = IY
ELSE
  THE Y-INDEX RECORD, IF REQUESTED
  READ(5.0) (IVPLT(IY),IY=1,NVPLT)
ENDIF

THE SPECIFICATION RECORD

IF(ITYPE.GE.1 .AND. ITYPE.LE.4) THEN
  READ(5.0) JPHI, JPI
ELSEIF(ITYPE.GE.5 .AND. ITYPE.LE.8) THEN
  READ(5.0) ITHETA
ELSEIF(ITYPE.EQ.9) THEN
  READ(5.0) JPHI, JPI
ENDIF

CALL THE APPROPRIATE PLOT SUBROUTINE

IF(ITYPE.EQ.1 .OR. ITYPE.EQ.2) THEN
  CALL PTHEPLR(ITYPE,NVPLT,IVPLT,JPHI, JPI)
ELSEIF(ITYPE.EQ.3 .OR. ITYPE.EQ.4) THEN
  CALL PTHELOG(ITYPE,NVPLT,IVPLT,JPHI, JPI)
ELSEIF(ITYPE.EQ.5 .OR. ITYPE.EQ.6) THEN
  CALL PPHIPLR
ELSEIF(ITYPE.EQ.7 .OR. ITYPE.EQ.8) THEN
  CALL PPHILOG(ITYPE,NVPLT,IVPLT,ITHETA)
ELSEIF(ITYPE.EQ.9) THEN
  CALL PPATH
ENDIF
GO TO 100

200 CALL PLOT(U.,U.,-98)

70 FORMAT(BA10)
END

FUNCTION NCHAR(TITLE,NWURDS)

GIVEN AN ALPHANUMERIC TITLE, TITLE, OF NWURDS (MAX 12). THIS
FUNCTION RETURNS THE NUMBER OF NON-BLANK CHARACTERS. (FOR USE IN
PLOTTING CENTERED TITLES)

DIMENSION TITLE(NWURDS),ICHAR(120)
DATA IBLANK/I0H /

MAXCHR = 10*NWURDS
ENCODER(7,200,IFMT) MAXCHR

DO 110 I=1,MAXCHR
  NCCHR = NCCHR + 1
  IF(ICHAR(NCHR).NE.IBLANK) RETURN
  CONTINUE
  NCCHR = 0
  RETURN

200 FORMAT(1H(,'I3.3A1'))
END
SUBROUTINE PPHILOG(ITYPE,NYPLT,IYPLT,ITHETA)

ON NHM6/PPHILOG

THIS ROUTINE PLOTS THE LOGARITHM OF THE DIFFUSE OR TOTAL RADIANCE AS A FUNCTION OF PHI FOR A GIVEN THETA VALUE. ALL Y VALUES ARE DISPLAYED ON THE SAME GRAPH.

IF ITYPE.EQ.7, THE DIFFUSE RADIANCES ARE PLOTTED
IF ITYPE.EQ.8, THE TOTAL RADIANCES ARE PLOTTED

IF ITHETA .GT. 0, PLOT UPWARD RADIANCES N(+THETA PHI,Y) = RADP
IF ITHETA .LT. 0, PLOT DOWNWARD RADIANCES N(-THETA PHI,Y) = RADM

PARAMETER(MXMU=10, MXPHI=24, MXY=30)
PARAMETER(MXPTS=MXPHI+3)
COMMON/CGRID/ THETA(MXMU), PHI(MXPHI), Y(MXY)
 COMMON/CRADIF/ RADAP(MXMU, MXPHI), RADP(MXMU, MXPHI, MXY),
           1 RADAM(MXMU, MXPHI), RADM(MXMU, MXPHI, MXY)
 COMMON/CRADIR/ RADAPC(MXMU, MXPHI), RADP(MXMU, MXPHI, MXY),
           1 RADAM(MXMU, MXPHI, MXY)
 COMMON/CMISC/ IMISC(20), FMISC(20), NTIT, ITITLE(8)
 COMMON/CWORK/ XPLT(MXPTS, MXY), YPLT(MXPTS, MXY), BCD(5)

DIMENSION IYPLT(MAY)

DATA XINCH, YINCH/4.0, 5.0/, H, BOX/0.15, 1.0/, EPS/1.0.E-12/
DATA ISYM Bl/0/

IT = IABS(ITHETA)
NPHI = IMISC(2)
NY = IMISC(4)
NPHI1 = NPHI + 1
PI = FMISC(1)
RADEG = FMISC(3)
TWOP1 = 2.*PI
PI2 = 0.5*PI
HBOX = H*BOX

IYMAX = NY
IF(NYPLT.GT.0) IYMAX = NYPLT

IF(ITYPE.EQ.7) THEN
FACT = 0.
ELSEIF(ITYPE.EQ.8) THEN
FACT = 1.
ELSE
WRITE(6,800) ITYPE
RETURN
ENDIF
CALL PLOT(1,2,-3)

DETERMINE THE ALLOWED RANGE OF Y VALUES

IYT = 1
IYB = NY
IF(ITHETA.EQ.0) GO TO 99

CHECK FOR ZERO UPWARD RADIANCE AT THE BOTTOM (NAKED SLAB CASE)
IF(ITHETA.LT.0) GO TO 98
IYB = NY - 1
DO 90 J=1,NPHI
IF(RADP(1,J, NY), GT, EPS*RADM(1, J, NY)) IYB = NY
90 CONTINUE
GO TO 99

CHECK FOR ZERO DOWNWARD RADIANCE AT THE TOP (NAKED SLAB CASE)
98 IYT = 2
DO 91 J=1,NPHI
IF(RADM(1,J,1), GT, EPS) IYT = 1
91 CONTINUE
99 CONTINUE

138
§7A. PLOTTING RADIANCES

DEFINE ARRAYS FOR PLOTTING
C DO 100 K=1.IMAX
   IV = IVPLT(K)
   IF(IV.EQ.1 .AND. IYT.NE.1) GO TO 100
   IF(IV.EQ.NY .AND. IYB.NE.NV) GO TO 100
   DO 101 J=1.NPHI
      XPLT(J,IV) = PHI1(J)
      IF(THETA.GT.U) THEN
         YPLT(J,IV) = ALOG10(RADP(IT,J,IV))
      ELSE
         YPLT(J,IV) = ALOG10(RADM(IT,J,IV) + FACT*RADOM(IT,J,IV))
      ENDIF
   101 CONTINUE
   XPLT(NPHI,IV) = TWOPI
C 100 YPLT(NPHI,IV) = YPLT(1,IV)
C
C FIND THE MAXIMUM AND MINIMUM VALUES TO BE PLOTTED
C
C RADMAX = -1.E30
C RADMIN = 1.E30
DO 110 K=1.IMAX
   IV = IVPLT(K)
   IF(IV.EQ.1 .AND. IYT.NE.1) GO TO 110
   IF(IV.EQ.NY .AND. IYB.NE.NV) GO TO 110
   DO 111 J=1.NPHI
      RAD = YPLT(J,IV)
      IF(RAD.GT.RADMAX) RADMAX = RAD
      IF(RAD.LT.RADMIN) RADMIN = RAD
   111 CONTINUE
  110 CONTINUE
C
C LABEL THE VERTICAL AXIS FOR A LOG PLOT
C
C MINH = IFIX(RADMIN)
C MAXH = ABS(MRANGE)
C IDIV = ABS(MRANGE)
C 302 IF(S.LE.IDIV .AND. IDIV.LE.10) GO TO 300
C 301 IF(IDIV.GT.10) GO TO 303
C 300 DLABL = FLOAT(ABS(MRANGE))/FLOAT(IDIV)
C 301 DLABL = (IDIV+1)/2
C 302 DLABL = FLOAT(ABS(MRANGE))/FLOAT(IDIV)
C 303 DLABL = DLABL.IDIV = DLABL
C
C CALL PL(T(0.,VINCH.2)
XX = -7.6*HBOX
DO 310 I=1.IDIV
   YY = VINCH - 0.45*H - FLOAT(I-1)*DINCH
   FLABL = FLOAT(MAXH) - FLOAT(I-1)*DINCH
   ENCODE(8,311,BULBL)
   CALL SYMBOL(XX,YY,.0,.0)
   FLABL = FLOAT(MINH) - FLOAT(I-1)*DINCH
   ENCODE(8,311,BULBL)
   XX = -1.2
   YY = 0.5*VINCH - 6.5*HBOX
   CALL SYMBOL(XX,YY,.13,.0,13)
C
C DEFINE SCALE FACTORS CONSISTENT WITH THE LABELS
C DO 200 K=1.IMAX
   IV = IVPLT(K)
   IF(IV.EQ.1 .AND. IYT.NE.1) GO TO 200
   IF(IV.EQ.NY .AND. IYB.NE.NV) GO TO 200
   XPLT(NPHI+1,IV) = 0.
   XPLT(NPHI+2,IV) = 12
   YPLT(NPHI+1,IV) = FLOAT(MAXH - MINH)/VINCH
   200 YPLT(NPHI+2,IV) = FLOAT(MAXH - MINH)/VINCH
§7A. PLOTTING RADIANCES

CALL PLOT(0.,0.,.3)
CALL PLOT(XINCH,J.,2)
V1 = -0.45*M
V2 = -2.35*M
CALL SYMBOL(0.,V1,H,13.,-1)
CALL SYMBOL(0.5*HBOX,Y2,H,1H6.,.1)
CALL SYMBOL(0.25*XINCH-V1,H,2H90.,.2)
CALL SYMBOL(0.5*V1,H,1H6.,-1)
CALL SYMBOL(0.75*XINCH-V1,H,1H6.,-.1)
CALL SYMBOL(0.5*XINCH-1.5*HBOX,Y2,H,3H,270.,.3)
CALL SYMBOL(XINCH,Y1,H,13.,-1)
CALL SYMBOL(XINCH-1.5*HBOX,Y2,H,3H,270.,.3)
CALL SYMBOL(0.5*XINCH-1.*HBOX,-4.*H,14*PHI IN DEGREES,0.,.14)

IF(ITYPE.EQ.7) THEN
  IF(I1THETA.GT.0) THEN
    ENCODE(41,210,BCD) THETA(1)*RADEG
    NCHAR = 41
  ELSE
    ENCODE(43,212,BCD) THETA(1)*RADEG
    NCHAR = 43
  ENDIF
ELSE
  IF(I1THETA.GT.0) THEN
    ENCODE(39,214,BCD) THETA(1)*RADEG
    NCHAR = 39
  ELSE
    ENCODE(41,216,BCD) THETA(1)*RADEG
    NCHAR = 41
  ENDIF
ENDIF

XX = 0.5*XINCH - 0.5*FLOAT(NCHAR)*HBOX
CALL SYMBOL(XX,-0.9,H,BCD,0.0,NCHAR)

IF(NTIT.GT.0) THEN
  XX = 0.5*XINCH - 0.5*FLOAT(NTIT)*HBOX
  CALL SYMBOL(XX,VINCH+3.0*H,1TITLE,NCHAR,NTIT)
ENDIF

DO 400 K=1,IYMAX
  IV = IV(X)
  IF(IV.EQ.1 AND. IVT.NE.1) GO TO 400
  IF(IV.EQ.NY .AND. IVB.NE.NY) GO TO 400
  ENCODE(10,401,BCD) Y(IV)
  CALL LINE(IVPLT(1,IV),PLT(1,IV),NPhl1=1,ISYMBL,1)
  YY = (VPLT(NPHI+1,1 IV)) - VPLT(NPHI+2,1 IV) - 0.5*H
  CALL SYMBOL(XINCH,YY,H,BCD,0.0,10)
400

CALL PLOT(-1.,-2.,-3)
CALL PLOT(10.,0.0,0.)
WRITE(6,802)
RETURN

FORMATS
210 FORMAT(35HDIFFUSE UPWARD RADIANCE FOR THETA =.F6.2)
212 FORMAT(37HDIFFUSE DOWNWARD RADIANCE FOR THETA =.F6.2)
214 FORMAT(33HTOTAL UPWARD RADIANCE FOR THETA =.F6.2)
216 FORMAT(33HTOTAL DOWNWARD RADIANCE FOR THETA =.F6.2)
311 FORMAT(F6.2,2H ~)
401 FORMAT(4H Y =.F6.2)
800 FORMAT(: ERROR: SUB PPHILOG CALLED WITH ITYPE =.,13)
802 FORMAT(1H , END OF PPHILOG"
END
§7A. PLOTTING RADIANCES

SUBROUTINE PTHELOG(ITYPE,NYPLT,IYPLT,JPHI,JPI)
C
ON NHM6/PTHELOG
C
THIS ROUTINE PLOTS THE LOGARITHM OF THE DIFFUSE OR TOTAL RADIANCE
AS A FUNCTION OF THETA FOR HALF PLANES DEFINED BY JPHI AND JPI.
ALL Y VALUES ARE DISPLAYED ON THE SAME PLOT.
C
IF(ITYPE.EQ.3, THE DIFFUSE RADIANCE IS PLOTTED
IF(ITYPE.EQ.4, THE TOTAL RADIANCE IS PLOTTED
PARAMETER(MXMU=10, MXPHI=24, MXV=30)
PARAMETER(MXPTS=4*MXMU+1, MXY=1+1)
C
COMMON/CGRID/ THETA(MXMU),PHI(MXPHI),Y(MXY)
COMMON/CRADIF/ RADAP(MXMU,MXPHI),RADP(MXMU,MXPHI,MXY),
1 RADAM(MXMU,MXPHI),RADM(MXMU,MXPHI,MXY)
COMMON/CRADIR/ RADOAP(MXMU,MXPHI),RADOAM(MXMU,MXPHI),
1 RADOM(MXMU,MXPHI,MXY)
COMMON/CMISC/ IMISC(20),FMISC(20),NTIT,ITITLE(8)
COMMON/CWORK/ XPLT(MXPTS,MXV1),YPLT(MXPTS,MXY1),BCD(4),NPLT(MXY1)
C
DIMENSION IYPLT(MXV)
C
DATA XINCH,YINCH/4.0,5.0/, H,BOX/0.15,1.0/, EPS/1.E-12/
DATA ISYMBL/0/
C
NMU = IMISC(1)
NY = IMISC(4)
PI = FMISC(1)
RADM = FMISC(3)
PI2 = 0.5*PI
IMAX = NY
IF(NYPLT.GT.0) IMAX = NYPLT
HBOX = H*BOX
C
IF(ITYPE.EQ.3) THEN
FACT = 0.
ELSEIF(ITYPE.EQ.4) THEN
FACT = 1.
ELSE
WRITE(6,800) ITYPE
RETURN
ENDIF
END IF
C
CALL PLOT(1.,2.,-3)
C
C
Determine the range of THETA at the first and last IV values.
C
Do not plot zero radiances. Check for zero downward radiance at the
C
top and for zero upward radiance at the bottom (naked slab cases).
C
C
IVTP = IVBP = 0
EPSREL = EPS*RADM(1,JPHI,IYPLT(IMAX))
DO 700 I=1,NMU
IF(RADM(I,JPHI,IYPLT(1)),GT,EPS) IVTP = 1
IF(RADM(I,JPHI,IYPLT(1)),GT,EPS) IVBP = 1
IF(RADM(I,JPHI,IYPLT(IMAX)),GT,EPSREL) IVTP = 1
IF(RADM(I,JPHI,IYPLT(IMAX)),GT,EPSREL) IVBP = 1
700 CONTINUE
C
C
Define the arrays to be plotted
C
Polar caps always have a PHI index of 1
C
LY = 0
IF(IYTP.EQ.1) GO TO 200
IF(IYPLT(1).NE.1) GO TO 200
C
SPECIAL CASE: The top boundary requires two plots for a naked upper boundary
C
LY = LY + 1
L = U
L = L + 1
XPLT(L,LY) = THETA(NMU) - PI
YPLT(L,LY) = ALOG10(RADP(NMU,1,1))
§7A. PLOTTING RADIANCES

DO 100 I=2,NMU
L = L + 1
II = NMU - I + 1
XPLT(L,LY) = THETA(II) - PI
100 YPLT(L,LY) = ALOG10(RADP(II,JPHI,1))

C
NPLT(LY) = L
C
LY = LY + 1
L = 1
DO 110 I=1,NMU-1
L = L + 1
XPLT(L,LY) = PI - THETA(I)
110 YPLT(L,LY) = ALOG10(RADP(I,JPHI,1))
L = L + 1
XPLT(L,LY) = PI - THETA(NMU)
YPLT(L,LY) = ALOG10(RADP(NMU,1,1))
NPLT(LY) = L

C
DO 150 K=1,IVMAX
IV = IVPLT(K)
IF(IV.EQ.1 .AND. LY.GT.0) GO TO 150
LY = LY + 1
L = 0
IF(IV.EQ.NV .AND. IVBP.EQ.0) GO TO 169

C
L = L + 1
XPLT(L,LY) = THETA(NMU) - PI
YPLT(L,LY) = ALOG10(RADP(NMU,1,IV))
DO 160 I=2,NMU
L = L + 1
II = NMU - I + 1
XPLT(L,LY) = THETA(II) - PI
160 YPLT(L,LY) = ALOG10(RADP(II,JPHI,1))

C
DO 170 I=1,NMU-1
L = L + 1
XPLT(L,LY) = -THETA(I)
170 YPLT(L,LY) = ALOG10(RADM(I,JPHI,1) + FACT*RADOM(I,JPHI,1))

C
L = L + 1
XPLT(L,LY) = THETA(NMU)
YPLT(L,LY) = ALOG10(RADM(NMU,1,IV) + FACT*RADOM(NMU,1,IV))
DO 180 I=1,NMU-1
L = L + 1
II = NMU - I
XPLT(L,LY) = THETA(II)
180 YPLT(L,LY) = ALOG10(RADM(II,JPHI,1) + FACT*RADOM(II,JPHI,1))

C
IF(IV.EQ.NY .AND. IVBP.EQ.0) GO TO 149

C
DO 190 I=1,NMU-1
L = L + 1
XPLT(L,LY) = PI - THETA(I)
190 YPLT(L,LY) = ALOG10(RADP(I,JPHI,1))
L = L + 1
XPLT(L,LY) = PI - THETA(NMU)
YPLT(L,LY) = ALOG10(RADP(NMU,1,IV))

C
IF(IV.EQ.NY .AND. IVBP.EQ.0) GO TO 149

C
DO 149 I=1,NMU-1
L = L + 1
XPLT(L,LY) = PI - THETA(I)
149 CONTINUE
NUMPLT = LY
C
FIND THE MAXIMUM AND MINIMUM VALUES TO BE PLOTTED
C
RADMAX = -1.E30
RADMIN = 1.E30
DO 500 LY=1,NUMPLT
NPTS = NPLT(LY)
DO 500 J=1,NPTS
RAD = YPLT(J,LY)
IF(RAD.GT.RADMAX) RADMAX = RAD
IF(RAD.LT.RADMIN) RADMIN = RAD
500 CONTINUE
§7A. PLOTTING RADIANCES

C

C LABEL THE VERTICAL AXIS FOR A LOG PLOT
C

MINH = IFIX(RADMIN)
IF(RADMIN.LT.0.) MINH = MINH - 1
MAXH = IFIX(RADMAX)
IF(RADMAX.GT.0) MAXH = MAXH + 1
MRANGE = MINH - MAXH
IDIV = IABS(MRANGE)

302 IF(S.LE.IDIV .AND. IDIV.LE.10) GO TO 300
IF(IDIV.GT.10) GO TO 301
IDIV = IDIV*2
GO TO 302

300 DLABL = FLOAT(IABS(MRANGE))/FLOAT(IDIV)
IF(DLABL.LE.1.) GO TO 303
IF(FLOAT(IFIX(DLABL)).EQ.OLABL) GO TO 303
MRANGE = IDIV*IFIX(DLABL + 1.)
GO TO 300

303 DINC1 = YINC/FLOAT(IDIV)
IDIV1 = IDIV + 1/
C

CALL PLOT(0.,0.,3)
CALL PLOT(0.,YINC,2)
XX = -7.6*BOX*H
GO 310 I=1.IDIV
YY = YINC - 0.45*H - FLOAT(I-1)*DINC1
FLABL = FLOAT(MAXH) - FLOAT(I-1)*DLABL
ENCODE(8,311,BCD) FLABL

310 CALL SYMBOL(XX,YY,H,BCD,0,0.8)
XX = -1.2
YY = 0.5*YINC - 6.5*BOX*H
CALL SYMBOL(XX,YY,H,13HLOG(RADIUS),90.0,13)
C

CALL PLOT(0.,0.,3)
CALL PLOT(XINC,H,0,2)
C

V1 = -0.45*H,
V2 = -2.75*H
CALL SYMBOL(0.,V1,H,13,0.,-1)
CALL SYMBOL(0.,-2.,*HBOX,V2,H,4H-180,0.,4)
CALL SYMBOL(0.,25*XINC,Y1,H,13,0.,-1)
CALL SYMBOL(0.,25*XINC-1.5*HBOX,V2,H,3H-90,0.,3)
CALL SYMBOL(0.,5*XINC,Y1,H,13,0.,-1)
CALL SYMBOL(0.,5*XINC-0.5*HBOX,V2,H,1H0,0.,1)
CALL SYMBOL(0.,75*XINC,Y1,H,13,0.,-1)
CALL SYMBOL(0.,75*XINC-0.5*HBOX,V2,H,2H90,0.,2)
CALL SYMBOL(0.,15*XINC-1.5*HBOX,V2,H,3H180,0.,3)
CALL SYMBOL(0.,5*XINC-15.*HBOX,-4.*H,H)
1 30H VIEWING ANGLE THETA IN DEGREES,0.,30)
C

H1 = 0.7*H
HBOX = 0.7*HBO
ENCODE(11,210,BCD) RADEG*PHI(JPHI)
CALL SYMBOL(0.,2.5*XINC-5.5*HBOX,-0.95,H1,BCD,0.,11)
ENCODE(11,210,BCG) RADEG*PHI(JPHI)
CALL SYMBOL(0.,75*XINC-5.5*HBOX,-0.95,H1,BCD,0.,11)
C

IF(ITYPE.EQ.3) THEN
CALL SYMBOL(0.5*XINC-22.5*HBOX,-1.4*H,
1 45H DIFFUSE FIELD RADIANCE AS A FUNCTION OF THETA,0.,45)
ELSE IF(ITYPE.EQ.4) THEN
CALL SYMBOL(0.5*XINC-21.5*HBOX,-1.4*H,
1 43HTOTAL FIELD RADIANCE AS A FUNCTION OF THETA,0.,43)
ENDIF
C

PLOT THE REFERENCE LINE AT THETA = 0 AND THE TITLE
CALL PLOT(0.5*XINC,H,0.,3)
CALL DASHPT(0.5*XINC,YINC,0.1)
IF(NTIT.GT.0) THEN
XX = 0.5*XINC - 0.5*FLOAT(NTIT)*HBOX
CALL SYMBOL(XX,YINC+3.0*H,H,ITITLE,0.,NTIT)
ENDIF
C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C
§7A. PLOTTING RADIANCES

PLT THE RADIANCES

IV = 0
DO 400 LY=1,NUMPLT
NPTS = NPLT(LY)
XPLT(NPTS+1,LY) = -PI
XPLT(NPTS+2,LY) = PI2
YPLT(NPTS+1,LY) = FLOAT(MINH)
YPLT(NPTS+2,LY) = FLOAT(MAXH - MINH)/VINCH
CALL LINE(XPLT(1,LY),YPLT(1,LY),NPTS,1,ISYM,1)
IF(LY.EQ.1 .AND. NUMPLT.GT.NYPLT) GO TO 400
LY = IV + 1
ENCODE(10,401,BCD) Y(1VPLT(IV))
YY = (YPLT(NPTS,LY) - YPLT(NPTS+1,LY))/YPLT(NPTS+2,LY) - 0.5*H
CALL SYMBOL(XINCH,YY,H,BCD,0.0,10)
400 CONTINUE
CALL PLOT(-1.,-2.,-3)
CALL PLOT(10.0,0.0,-3)
WRITE (6,802)
RETURN

FORMATS

210 FORMAT(S5PHI =,F6.1)
311 FORMAT(F6.2,2H-)
401 FORMAT(4HV =,F6.2)
800 FORMAT(' ERROR: SUB PTHELOG CALLED WITH ITYPE =',I3)
802 FORMAT(' END OF PTHELOG')

SUBROUTINE PTHEPLR(ITYPE,NYPLT,LYPLT,JPHI,JPI)

ON NHMR/PTHEPLR

THIS ROUTINE MAKES A POLAR PLOT OF THE LOGARITHM OF THE DIFFUSE OR TOTAL RADIANCE AS A FUNCTION OF PHII FOR HALF-PLANES DEFINED BY JPHI AND JPI. A SEPARATE PLOT IS MADE FOR EACH DEPTH.

IF ITYPE.EQ.1, THE DIFFUSE RADIANCE IS PLOTTED
IF ITYPE.EQ.2, THE TOTAL RADIANCE IS PLOTTED

PARAMETER(MXU=10, MXPHI=24, MXY=1U)
PARAMETER(MXPTS=4*MXU+1)

COMMON/CGRUHN,THETA(MXU),PHI(MXPHI),YU(MXY)
COMMON/CRADIF/RADAP(MXU,MXPHI),RADAP(MXU,MXPHI,MXY),
1 (RADAP+1,LY,MXPHI),RADM(MXU,MXPHI,MXY)
COMMON/CRADIF/RADAP(MXU,MXPHI),RADAP(MXU,MXPHI,MXY),
1 (RADAP+1,LY,MXPHI),RADM(MXU,MXPHI,MXY)
COMMON/CMDISP/IMISC(U),FMDISP(V),NT,ITITLE(B)
COMMON/CWORK/X(MXPTS),Y(MXPTS),MC(2U)

144
DIMENSION TITLE(6), IVPLT(MXY)
DATA ISCMBL/1/, RINCH/3.0/, DICIRC/0.5/, H.BOX/0.15,1.0/
CALL PLOT(4.,5.,-3)

NMU = IMISC(1)
NY = IMISC(4)
PI = FMISC(1)
DEGRAD = FMISC(2)
RADEG = FMISC(3)
P12 = 0.5*PI
PI32 = 1.5*PI
HBOX = H*BOX

IF(ITYPE.EQ.1) THEN
  FACT = 0
ELSEIF(ITYPE.EQ.2) THEN
  FACT = 1
ELSE
  WRITE(6,300) ITYPE
  RETURN
ENDIF

IVMAX = NY
IF(NYPLT.GT.0) IVMAX = NYPLT

DO 888 K = 1, IVMAX
  IV = IVPLT(K)

  CONVERT RADIANCES AND NHM THETA VALUES (MEASURED FROM THE Y AXIS) TO X = LOG(RADIANCE) AND Y = CALCOMP THETA (MEASURED FROM THE X AXIS)

  POLAR CAPS ALWAYS HAVE A PHI INDEX OF 1

  L = 0
  DO 200 I = 1, NMU - 1
     L = L + 1
     X(L) = ALOG10(RADP(I, JPHI, IV))
     Y(L) = P12 - THETA(I)
  200  L = L + 1
     X(L) = ALOG10(RADP(NMU, 1, IV))
     Y(L) = P12

  DO 201 I = 1, NMU - 1
     L = L + 1
     X(L) = ALOG10(RADP(I, JPHI, IV))
     Y(L) = P12 + THETA(I)
  201  L = L + 1

  DO 202 I = 1, NMU - 1
     L = L + 1
     X(L) = ALOG10(RADM(I, JPHI, IV) + FACT*RADM(I, JPHI, IV))
     Y(L) = PI32 - THETA(I)
  202  L = L + 1
     X(L) = ALOG10(RADM(NMU, 1, IV) + FACT*RADM(NMU, 1, IV))
     Y(L) = PI32

  DO 203 I = 1, NMU - 1
     L = L + 1
     X(L) = ALOG10(RADM(I, JPHI, IV) + FACT*RADM(I, JPHI, IV))
     Y(L) = PI32 + THETA(I)
  203  L = L + 1

  FIND THE MAXIMUM AND MINIMUM LOG VALUES TO BE PLOTTED

  RADMAX = 1.0E30
  RADMIN = 1.0E30
  DO 250 L = 1, NPTS
     RAD = X(L)
     IF(RAD.GT.RADMAX) RADMAX = RAD
     IF(RAD.LT.RADMIN) RADMIN = RAD
  250 CONTINUE
§7A. PLOTTING RADIANCES

C

LABEL THE RADIAL (VERTICAL) AXIS FOR A LOG PLOT

C

MINH = IFLX(RADMIN)
IF(RADMIN.LT.0.) MINH = MINH - 1
MAXH = IFLX(RADMAX)
IF(RADMAX.GT.0.) MAXH = MAXH + 1
MRANGE = -MINH - MAXH
IDIV = IABS(MRANGE)
302 IF(3.LE.IDIV.AND. IDIV.LE.6) GO TO 300
IF(IDIV.GT.6) GO TO 301
IDIV = IDIV*2
GO TO 302
301 IDIV = (IDIV + 1)/2
GO TO 302

300 DLABL = FLOAT(IABS(MRANGE))/FLOAT(IDIV)
IF(DLABL.LE.1.) GO TO 303
IF(FLOAT(IDC)(DLABL)).EQ.DLABL) GO TO 303
MRANGE = IDIV*1.1*(DLABL + 1.)
GO TO 300

303 DINCCH = RINCH/FLOAT(IDIV)

C

CALL PLOT(0.,-RINCH,3)
CALL PLOT(0.,-RINCH,2)
CALL PLOT(-1.,-RINCH,3)
CALL PLOT(1.,-RINCH,2)
CALL PLOT(-1.,RINCH,2)
CALL PLOT(1.,RINCH,2)
XX = -7.5*BOX*X

DO 310 I=1,101
YY = RINCH - 0.45*M - FLOAT(I-1.)*DINCCH
R(I) = RINCH - FLOAT(I-1.)*DINCCH
FLABL = FLOAT(MAXH) - FLOAT(I-1.)*DLABL
ENGINDE(R,11),B,L,FLABL)
310 CALL SYMBOL((XX,YY,11,BL,0.0,0.6)

XX = -1.1
YY = 0.5*RINCH - 0.5*BOX*Y
CALL SYMBOL((XX,YY,21,BL,0.0,0.6)

C

DRAW MAGNITUDE CIRCLES

C

DO 260 I=1,101
R = R(I)
THQ = 90.0 + RADS*ASIN(1.0/R)
ELSEIF(R.GT.1.35) AND. R.GT.0.1) THEN
THQ = 180.0 + RADS*ASIN(0.1/R)
ELSE
THQ = 180.0
ENDIF
XS = R*COS(THQ)
YS = R*SIN(THQ)
CIRCLE((XS,YS),100.,R,L,THQ)
260 CALL CIRCLE((XX,YY,11,BL,0.0,0.6,THQ)

C

CONVERT LOGICAL RADIANCE AND CALL SUMP THE ETA TO X AND Y IN INCHES

C

C1 = RINCH/FLOAT(MAXH) - MINH
C2 = -FLOAT(MAXH) + 1.
DO 500 I=1,101
R(I) = C1*(I-1. + 2
X(I) = MINH*(I-1.) + C2
Y(I) = MINH*(I-1.) + 1.
500
C

SCALE THE RADIANCE

C

XX(NPTS+1) = 0.
YY(NPTS+1) = 0.
XX(NPTS+2) = 1.
YY(NPTS+2) = 1.
C

PLOT RADIANCES

C

CALL LINE(XX,YY,NPTS,1.15,SYMBOL,1)
C

LABEL THE PLOT

C

146
IF (ITYPE.EQ.1) THEN
    ENCODE(52, 400, TITLE) YOD(IY)
    CALL SYMBOL(-26.0*HBOX, -4..H, TITLE, 0..52)
ELSE
    ENCODE(50, 402, TITLE) YOD(IY)
    CALL SYMBOL(-25.0*HBOX, -4..H, TITLE, 0..50)
ENDIF
CALL PLOT(0..-3.6.3)
CALL PLOT(0..-3.2.2)
ENCODE(26, 404, TITLE) RADEG*PHI(JPI), RADEG*PHI(JPHI)
CALL SYMBOL(-13..H8OX.-3.5, TITLE, 0..26)
IF(NTIT.GT.0) THEN
    XX = -0.5*FLOAT(NTIT)*HBOX
    CALL SYMBOL(XX, RINCH*3.0*H, ITITLE, 0..NTIT)
ENDIF
C
888 CALL PLOT(10.0, 0.0, -3)
C
882 WRITE(6, 802)
RETURN
C
400 FORMAT(46HDIFFUSE RADIANCE AS A FUNCTION OF THETA AT V =, F6.2)
402 FORMAT(44HTOTAL RADIANCE AS A FUNCTION OF THETA AT V =, F6.2)
404 FORMAT(5HPHI =, F6.1, 5X, 5HPHI =, F6.1)
411 FORMAT(F6.2, 2H -)
800 FORMAT(1H , ' ERROR: SUB PTHPLR CALLED WITH ITYPE =', I3)
802 FORMAT(1H , ' END OF PTHPLR')
B. Plotting Chromaticity Diagrams

The Natural Hydrosol Model computes monochromatic radiances. However, independent runs of the NHM can be made at various wavelengths, using wavelength-dependent input radiances and inherent optical properties, and the results can be combined to generate wavelength-dependent output.

It is often of interest, e.g. in remote sensing, to plot the ocean color on a standard C.I.E. chromaticity diagram. Program MPCHRO reads 13 data values corresponding to 13 wavelengths (400 nm, 425 nm, ..., 675 nm, and 700 nm). Each datum is obtained from a run of Programs 4 and 5, 13 runs in all. The data are processed using standard tristimulus functions, as described in Appendix C, and plots of the resultant color point are made on a 1931 C.I.E. chromaticity diagram.

In order to compute a correct color (e.g. for the upward radiance), the incident radiance on the water surface must have the correct color, i.e. wavelength dependence (corresponding, say, to the solar spectrum). The proper wavelength dependence of the incident lighting can be achieved by adjusting the value of SHTOTL (input record 6 in Program 4) in each of the 13 NHM wavelength runs. However, it is generally more convenient to make all NHM runs with SHTOTL equal to some nominal value, say 1.0 W m\(^{-2}\) nm\(^{-1}\). In this case, the output values of the 13 NHM runs must be adjusted before computing the chromaticity. Subroutine ATMOS uses a simple model atmosphere and solar spectrum (described in Appendix D) to weight the 13 data values according to wavelength and solar zenith angle before proceeding with the chromaticity calculations.

1. Input

Two records are read to specify the details of the plot, and then repeated pairs of records are read to specify the wavelength data to be plotted.

Record 1: LABPNT, IPLBLU, IATMOS

<table>
<thead>
<tr>
<th>LABPNT</th>
<th>IPLBLU</th>
<th>IATMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

148
§7B. PLOTTING CHROMATICITIES

Record 2: ITOP
This record gives a title for the top of the plot. Up to 80 alphanumerical characters are allowed.

Record 3: ITITLE
This record gives a label for the plotted point. Up to 80 characters are allowed.

Record 4: THETPS, P(1), P(2),...P(13)

THETPS is the zenith angle, in degrees, of the sun in the run of Program 4 which generated the data. If IATMOS = 1, this value is used in the atmospheric model of Appendix D to correct the raw data values P(I), I = 1,...,13.

P(1) are the 13 data values to be used in computing the color. P(1) corresponds to wavelength 400 nm, P(2) to 425 nm,..., P(13) to 700 nm.

Typically, P(I) is the radiance in a given direction at a given depth, or an irradiance at a given depth.

Records 5 and 6, 7 and 8,...
Pairs of records of the same form as 3 and 4 are repeated for each point to be plotted. Up to 50 points are allowed by the dimensions in the listed code (see parameter MXPTS in program MPCHRO).
2. Code Listing

```
PROGRAM MPCHRO(INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT,TAPE98.
    ! TAPE99)

ON NHM6:MPCHRO
C
THIS PROGRAM COMPUTES AND PLOTS CHROMATICITIES ON A STANDARD CIE
1931 CHROMATICITY DIAGRAM, GIVEN RADIANCES OR IRRADIANCES AT 13
WAVELENGTHS: 400. NM, 425. NM, 450. NM, ..., 675. NM, 700. NM
C
ALL PLOTTING IS DONE USING STANDARD CALCOMP CALLS
C(TAPE98 AND TAPE99 ARE USED BY THE CALCOMP ROUTINES, AS
IMPLEMENTED ON THE AUTHOR'S CDC CYBER 855 COMPUTER.)
C
PARAMETER(NWAVE=13, MXPTS=50)
C
DIMENSION P(NWAVE),IWAVE(NWAVE),PP(5)
DIMENSION XCHP(MXPTS),CHR(MXPTS),IUP(8),ITITLE(MWAVE)
C
DATA IWAVE((4CP425,450,475,500,525,550,575,600,625,650,675,700/
C

READ THE OVERALL PLOT SPECS. AND A TITLE FOR THE TOP OF THE PLOT
C
LABPNT = 0, IF POINTS ARE NOT TO BE LABELLED
  1. IF EACH POINT IS NUMBERED AND LABELLED
1PLBLU = 0, IF ONLY THE FULL CHROMATICITY DIAGRAM IS TO BE DRAWN
  1. IF ONLY THE BLUE CORNER IS TO BE DRAWN
  2. IF BOTH FULL AND BLUE CORNER ARE TO BE DRAWN
1ATMOS = 0, IF THE RAW P(LAMDA) ARE TO BE USED
  1. IF THE RAW P(LAMDA) ARE TO BE SCALLED BY THE ATMOSPHERIC MODEL
1TOP..., A TITLE FOR THE TOP OF THE PLOT (80 CHAR MAX)
C
NPTS = 0
READ(5.*)& LABPNT,1PLBLU,1ATMOS
READ(5.100) ITOP
WRITE(6,90)
C
99 NPTS = NPTS * 1:
C READ A LABEL AND A SET OF VALUES TO BE PROCESSED
C
READ(5,100,END=500) (ITITLE(I),NP(I),I=1,M)
READ(5,*,END=500) THETPS,ITOP,ITITLE,NWAVE
DO 150 5=1,NWAVE
150 RAMP(I) = 1.25
C SCALE THE (I) ACCORDING TO THE ATMOSPHERIC MODEL
C IF(1ATMOS.NE.0) CALL ATMOS(THETPS)
C COMPUTE THE CHROMATICITY COORDINATES
C CALL CHRMXY(P,CHR,NP,ITOP,ITITLE,NWAVE)
C WRITE(6,200) N1,N2,ITITLE,CHR(1),ITOP,NWAVE
IF(1ATMOS.NE.0) WRITE(6,201) ITOP
WRITE(6,201) ITOP
DO 201 1 = NWAVE
201 WRITE(6,202) ITITLE,CHR(1),ITOP
WRITE(6,201) ITOP,201
C
GO TO 99
C DRAW A CHROMATICITY DIAGRAM AND PLOT POINTS ON IT
C YOU NPTS = NPTS * 1
C```

150
§7B. PLOTTING CHROMATICITIES

CALL PLOTS
CALL PLOT(2., 2., -3)
IF(IPLBLU.NE.0) CALL PLTBLU(XCHR, YCHR, NPTS, LABPNT, ITOP, TITLE)
CALL PLOT(10., 0., -3)
IF(IPLBLU.NE.1) CALL PLTCHR(XCHR, YCHR, NPTS, LABPNT, ITOP, TITLE)
CALL PLOT(0., 0., -98)

C
C FORMATS
C
90 FORMAT(1H1)
100 FORMAT(BA10)
200 FORMAT(///' POINT NUMBER'.13.',BA10/)
201 FORMAT(///' THE ATMOSPHERIC MODEL WITH THE TPS = ',F6.2,' IS USED'//)
202 FORMAT(' THE FUNCTION OF WAVELENGTH GIVEN BY'//
1.'LAMBDA P(LAMBDA) (RAW P)'//)
203 FORMAT(16,1P2E16.4)
204 FORMAT(' HAS CHROMATICITY COORDINATES (X,Y) = ('F5.4,','
1.F6.4,) OR (DOMINANT WAVELENGTH, PURITY) = ('F5.1,','F6.4,')')
END

SUBROUTINE ATMOS(THETPS, P)

PARAMETER(NWAVEL=13)
DIMENSION P(NWAVEL), ALPHAL(NWAVEL), SOLARC(NWAVEL)
DATA SOLARC/1.54, 1.89, 2.20, 2.20, 1.98, 1.92, 1.95, 1.87, 1.81, 1.72,
1.62, 1.53, 1.44/
DATA ALPHAL/0.566, 0.428, 0.364, 0.293, 0.217, 0.190, 0.220, 0.206,
1.192, 1.165, 1.134, 1.114, 1.103/
DATA DEGRAD/0.*17453293/

SECTH = 1.0/COS(DEGRAD*THETPS)
DO 100 I=1, NWAVEL
100 P(I) = P(I)*SOLARC(I)*EXP(-ALPHAL(I)*SECTH)
RETURN
END

151
SUBROUTINE CHRMXY(P.A),DOMWL,PURITY)

ON NHMB/CHRMXY

GIVEN A SET OF RADIANCES OR IRRADIANCES, P, AT THE NHM WAVELENGTHS,
THIS ROUTINE COMPUTES THE CHROMATICITY COORDINATES (X,Y) FOR
PLOTTING ON A CHROMATICITY DIAGRAM. THE CORRESPONDING DOMINANT
WAVELENGTH AND PURITY ARE ALSO COMPUTED. SEE APPENDIX C OF THIS
REPORT FOR DETAILS.

PARAMETER (NWAVEL=13, MXPURE=37)
DIMENSION P(NWAVEL),XBAR(NWAVEL),YBAR(NWAVEL),ZBAR(NWAVEL)
DIMENSION WAVEL(MXPURE),XPURE(MXPURE),YPURE(MXPURE),SLOPUR(MXPURE)

THE 13 TRISTIMULUS FUNCTION VALUES
DATA XBAR/0.0143, 0.2148, 0.3362, 0.4214, 0.0049, 0.1096, 0.4334, 0.8425,
1.0462, 0.7514, 0.2835, 0.0636, 0.0114/
DATA YBAR/0.0004, 0.0073, 0.0380, 0.1126, 0.3230, 0.7932, 0.9950, 0.9154,
1.0310, 0.3210, 0.1070, 0.0232, 0.0041/
DATA ZBAR/0.0679, 0.0911, 0.7711, 1.0391, 1.7721, 1.0421, 0.0049, 0.1096,
0.4334, 0.8425, 1.1062, 2.1421, 0.0049, 0.1096/

PARAMETER DATA WAVEL/400., 450., 460., 470., 475., 480., 485., 490.,
500., 510., 515., 520., 525., 530., 535., 540., 545.,
605., 610., 620., 626., 640., 650., 700.,
1.0454, 0.4525, 0.6922, 0.8458, 0.9680, 1.096, 0.913, 0.687,
1.1929, 2.296, 3.278, 3.018, 4.373, 3.737, 4.087, 4.441, 4.788,
3.5125, 5.448, 5.752, 5.629, 6.029, 6.482, 6.658, 6.915, 7.079,
4.7190, 7.260, 7.347,
DATA YPURE/0.0045, 0.2177, 0.429, 0.6756, 0.868, 1.0027, 1.207,
1.2950, 1.4217, 1.544, 1.6548, 1.7502, 1.8120, 1.8338, 1.8626, 1.8059,
2.7816, 2.725, 2.524, 2.6921, 2.6589, 2.6245, 2.5896, 2.5437, 2.5020,
3.4866, 4.544, 5.42, 3.965, 3.725, 3.514, 3.340, 3.083, 2.920,
4.2809, 2.747, 2.653/
DATA FALLO/0/

THE 37 SPECTRUM LOCUS VALUES
DO 100 I=2,NWAVEL-1,2
CAPX = P(I)*XBAR(I) + P(NWAVEL)*XBAR(NWAVEL)
CAPY = P(I)*YBAR(I) + P(NWAVEL)*YBAR(NWAVEL)
CAPZ = P(I)*ZBAR(I) + P(NWAVEL)*ZBAR(NWAVEL)
100 CONTINUE

DO 110 I=3,NWAVEL-2,2
CAPX = CAPX + 4.0*P(I)*XBAR(I)
CAPY = CAPY + 4.0*P(I)*YBAR(I)
CAPZ = CAPZ + 4.0*P(I)*ZBAR(I)
110 CONTINUE

F = 66B00./FLO17.*NWAVEL-1,
CAPX = F*CAPX
CAPY = F*CAPY
CAPZ = F*CAPZ

NORMALIZE THE INTEGRALS TO GET THE CHROMATICITY COORDINATES, BY C.3

X = CAPX/(CAPX + CAPY + CAPZ)
Y = CAPY/(CAPX + CAPY + CAPZ)

IF(KALL.EQ.1) THEN
ENDDO
END

152
§7B. PLOTTING CHROMATICITIES

COMPUTE THE DOMINANT WAVELENGTH AND PURITY OF THE CHROMATICITY COORDINATES (X,Y)

DY = YW - Y
DX = XW - X
SLOPE = DY/DX

99 IF(DY.GE.0. .AND. DX.GE.0.) THEN
C
SEARCH LOWER LEFT OF SPECTRUM LOCUS, POINTS 2 TO 9
DO 200 I=2,9
IF(SLOPE.LT.SLOPUR(I) .AND. SLOPE.GE.SLOPUR(I)) GO TO 250
200 CONTINUE
C
ELSEIF(DY.LE.0 .AND. DX.GE.0) THEN
C
SEARCH UPPER LEFT OF SPECTRUM LOCUS, POINTS 9-22
DO 202 I=9,22
IF(SLOPE.GE.SLOPUR(I)) GO TO 250
202 CONTINUE
C
ELSEIF(DY.LE.0 .AND. DX.LE.0.) THEN
C
SEARCH UPPER RIGHT OF SPECTRUM LOCUS, POINTS 22-33
DO 204 I=22,33
IF(SLOPE.GE.SLOPUR(I)) GO TO 250
204 CONTINUE
C
ELSEIF(DY.GE.0 .AND. DX.LE.0.) THEN
C
SEARCH LOWER RIGHT OF SPECTRUM LOCUS, POINTS 33-37
DO 206 I=33,37
IF(SLOPE.GE.SLOPUR(I)) GO TO 250
206 CONTINUE
C
ENDIF
C
POINT IS IN PURPLE REGION, REVERSE (X,Y) AND THE WHITE POINT AND SEARCH AGAIN
DX = - DX
DY = - DY
GO TO 99
C
COMPUTE INTERSECTION POINT OF CHROMATICITY LINE AND SPECTRUM LOCUS

250 XPI = XPURE(I)
YPI = YPURE(I)
XPIM1 = XPURE(I-1)
YPIM1 = YPURE(I-1)
S1 = (X - XW)/(Y - YW)
S2 = (XPI - XPIM1)/(YPI - YPIM1)

XI = (S2*XW - S1*XPIM1 - S1*S2*(YW - YPIM1))/(S2 - S1)
VI = (XW - XPIM1 - S1*YW + S2*YPIM1)/(S2 - S1)

GET DOMINANT WAVELENGTH BY INTERPOLATION
DI = SQRT((XI - XPIM1)**2 + (YI - YPIM1)**2)
DI = SQRT((XI - XPIM1)**2 + (YI - YPIM1)**2)
F = DI/DI

DOMWVL = (1.0 - F)*WAVEL(I-1) + F*WAVEL(I)
EQ = SQRT((XW - X)**2 + (YW - Y)**2)
EW = SQRT((XW - XI)**2 + (YW - YI)**2)
PURITY = EQ/EW
RETURN
END
§7B. PLOTTING CHROMATICITIES

SUBROUTINE PLTBLU(XPLT,YPLT,NPTS,LABPNT,ITOP,ITITLE)

ON NMM6/PLTBLU

THIS ROUTINE DRAWS THE ‘BLUE CORNER’ OF A CHROMATICITY DIAGRAM AND PLOTS POINTS ON IT

STANDARD CALCOMP PLOTTING Routines ARE USED

PARAMETER (MXPURE=9)

DIMENSION IWAVEL(MXPURE),XPURE(MXPURE),YPURE(MXPURE)
DIMENSION XPLT(NPTS),YPLT(NPTS),ITOP(8),ITITLE(8,NPTS)

DATA IWAVEL/400,450,460,470,480,485,490,494/
DATA XPURE/,1733.,1566.,1440.,1241.,1096.,913.,687.,544.,259/
DATA YPURE/0.048,0.0177,0.0297,0.0578,0.0868,0.1327,0.2007,0.2950,0.4/

DATA XINCH,YINCH/-4.0,4.0,0.0,4.0,0.0,0.0,0.0,1/
DATA RADEG/57.295779513/

INITIALIZE AND DRAW AXES

CALL AXIS(0.0,0.0,1,0.0,0.0,0.0,0.0,1)
CALL AXIS(0.0,0.0,1,0.0,0.0,0.0,0.0,1)
CALL PLOT(0.0,0.0,1,0.0,0.0,1)
CALL PLOT(XINCH,YINCH,2)
CALL PLOT(XINCH,0.0,2)
NC = NCHAR(ITOP,8)
IF(NC.NE.0) THEN
CALL SYMBOL(0.5*XINCH-0.5*XPLT,0.5,YPLT,0.5,HTIC,YPLT,0.5*XPLT,0.5,HTIC)
ENDIF

DRAW THE SPECTRUM LOCUS

XSCALE = XINCH/0.4
YSCALE = YINCH/0.4
CALL PLOT(XSCALE*XPURE(1),YSCALE*YPURE(1),3)
DO 100 I=2,MXPURE-1
CALL PLOT(XSCALE*XPURE(I),YSCALE*YPURE(I),2)
C
ADD TIC MARKS
SLOPE = -(XPURE(I+1)-XPURE(I))/(YPURE(I+1)-YPURE(I-1))
THETA = RADEG*ATAN(SLOPE) - 90.
CALL SYMBOL(XSCALE*XPURE(I),YSCALE*YPURE(I),HTIC,13,THETA,-1)
100 CONTINUE
CALL PLOT(XSCALE*XPURE(MXPURE),YSCALE*YPURE(MXPURE),2)
CALL PLOT(XSCALE*0.4,YSCALE*0.1100,1)
CALL PLOT(XSCALE*XPURE(1),YSCALE*YPURE(1),2)

ADD WAVELENGTH LABELS TO SELECTED TICS

DO 110 I=1,8
 ENCODE(4.120,BCD) IWAVEL(I)
110 CALL SYMBOL(XSCALE*XPURE(I)-5.0*HTIC,YSCALE*YPURE(I)-0.5*HTIC,
1 HTIC,BCD,0.0,4)

PLOT THE WHITE POINT

CALL SYMBOL(XSCALE*3.0,YSCALE*0.0,0.0,0.0,0.0,1)

PLOT CHROMATICITY COORDINATES (-11T,0.61) ON THE DIAGRAM

IF(NPTS.GT.0) THEN
DO 200 I=1,NPTS
 X = XSCALE*XP(T(I))
 Y = YSCALE*YP(T(I))
 CALL SYMBOL(X+0.5*HTIC,Y+0.5*HTIC,1,0.0,1)
 IF(LABPNT.NE.0) THEN
 FPN = FLAT(I)
 CALL NUMBER(X+0.5*HTIC,Y+0.5*HTIC,FPN,0.0,0.0,-1)
200 ENDF
§7B. PLOTTING CHROMATICITIES

\[
X = X_{\text{INCH}} + 1.0 \\
Y = Y_{\text{INCH}} - \text{FPN} \times 2.0 \times \text{HTIC} \\
\text{CALL NUMBER}(X, Y, \text{HTIC}, \text{FPN}, 0.0, -1) \\
\text{NS} = \text{NCHAR}(\text{ITITLE}(1,1), 8) \\
\text{CALL SYMBOL}(X \times 2.5 \times \text{HTIC}, Y, \text{HTIC}, \text{ITITLE}(1,1), 0.0, \text{NS}) \\
\text{ENDIF} \\
\text{200 CONTINUE} \\
\text{ENDIF} \\
\text{CALL PLOT}(0.0, 0.0, -3) \\
\text{RETURN} \\
\text{C} \\
\text{120 FORMAT(1H13)} \\
\text{END}
\]

SUBROUTINE PLTCHR(XPLT, YPLT, NPTS, LABPNT, ITOP, ITITLE)

This routine draws a chromaticity diagram and plots points on it.

Standard CALCOMP plotting routines are used.

PARAMETER (MXPURE=37)
DIMENSION IWAVEL(MXPURE), XPURE(MXPURE), YPURE(MXPURE)
DIMENSION XPLT(NPTS), YPLT(NPTS), ITOP(8), ITITLE(8, NPTS)

DATA IWAVEL/400, 450, 460, 470, 475, 480, 485, 490, 1495, 500, 505, 510, 515, 520, 525, 530, 535, 540, 545, 2550, 550, 555, 560, 565, 570, 575, 580, 585, 590, 595, 600, 3605, 610, 620, 630, 640, 650, 700/
DATA XPURE/.1733, .1566, .1440, .1241, .1096, .0913, .0687, 1 .0454, .0235, .0139, .0389, .0743, .1142, .1547, 2 .1929, .2296, .2658, .3016, .3373, .3731, .4087, .4441, .4788, 3 .5125, .5448, .5752, .6029, .6270, .6482, .6689, .6915, .7079, 4 .7190, .7260, .7347/
DATA XINCH,VINCH/4.0, 4.5/, HTIC/0.1/
DATA RADEG/57.295779513/

Initialize and draw axes

CALL AXIS(O, 0, 0, 1HX, -1, XINCH, 0.0, 0.0, 0.2) 
CALL AXIS(O, 0, 0, IHY, 1, VINCH, 90.0, 0.0, 0.2) 
CALL PLOT(O, YINCH, 3) 
CALL PLOT(XINCH, VINCH, 2) 
CALL PLOT(XINCH, 0.2) 
NC = NCHAR(ITOP, 8) 
IF (NC .NE. 0) THEN 
CALL SYMBOL(0.5*XINCH - 0.5*FLOAT(NC)*HTIC, VINCH+2.0*HTIC, HTIC, 1 ITOP, 0.0, NC) 
ENDIF

155
§7B. PLOTTING CHROMATICITIES

DRAW THE SPECTRUM LOCUS

XSCALE = XINCH/0.8
YSCALE = YINCH/0.9
CALL PLOT(XSCALE*XPURE(1),YSCALE*YPURE(1),3)
DO 100 I=2,MXPURE-1
CALL PLOT(XSCALE*XPURE(I),YSCALE*YPURE(I),2)
100 CONTINUE

ADD TIC MARKS:
SLOPE = -(XPURE(I+1) - XPURE(I-1))/YPURE(I+1) - YPURE(I-1)
THETA = RADIUS*ATAN(SLOPE) - 90.
CALL SYMBOL(XSCALE*XPURE(I),YSCALE*YPURE(I),HTIC,13,THETA,-1)

ADD WAVELENGTH LABELS TO SELECTED TICS:
ENCODE(4,120,B,4) IWAIVEL(1)
CALL SYMBOL(XSCALE*XPURE(I),YSCALE*YPURE(I)-0.25*HTIC,HTIC,BCD,1 20.0,4)
ENCODE(4,120,B,4) IWAIVEL(3)
CALL SYMBOL(XSCALE*XPURE(3)-4.0*HTIC, YSCALE*YPURE(3)-2.0*HTIC,1 HTIC,BCD,20.0,4)
ENCODE(4,120,B,4) IWAIVEL(4)
CALL SYMBOL(XSCALE*XPURE(4)-4.0*HTIC, YSCALE*YPURE(4)-2.0*HTIC,1 HTIC,BCD,20.0,4)

DO 110 J=8,34
ENCODE(4,120,B,4) IWAIVEL(1)
CALL SYMBOL(XSCALE*XPURE(I),YSCALE*YPURE(I)-0.25*HTIC,HTIC,BCD,1 20.0,4)
ENCODE(4,120,B,4) IWAIVEL(37)
CALL SYMBOL(XSCALE*XPURE(37),YSCALE*YPURE(37)-0.25*HTIC,HTIC,BCD,1 20.0,4)

PLOT THE WHITE POINT:

CALL SYMBOL(XSCALE/3.0,YSCALE/3.0,4.0,3.0,1)

PLOT CHROMATICITY COORDINATES (XPLT,YPLT) ON THE DIAGRAM:

IF(NPTS.GT.0) THEN
DO 200 I=1,NPTS
X = XSCALE*XPURE(I)
Y = YSCALE*YPURE(I)
CALL SYMBOL(X,Y,5.0*HTIC,1.0,0.0,-1)
IF(LABPNT.NE.0) THEN
FIN = F.OAT(I)
CALL NUMBER(X,Y,5.0*HTIC,Y-0.5*HTIC,HTIC,FIN,0.0,1)
X = XINCH
Y = YINCH - FIN*2.0*HTIC
CALL NUMBER(X,Y,HTIC,FIN,0.0,1)
NS = NCHAR(I,E(1,1),9)
CALL SYMBOL(X,Y,5.0*HTIC,Y,HTIC,11/11,11/I,1.1,0.0,NS)
ENDIF
200 CONTINUE
ENDIF

CALL PLOT(XPLT,YPLT)
RETURN

FORMAT(1H13)
END
§7C. PLOTTING WAVELENGTH DATA

C. Plotting Data as a Function of Wavelength

If data have been generated for each of the 13 wavelengths described in paragraph B above, it is often desirable to plot the data as a function of wavelength. Program MPWAVE generates such plots.

1. Input

Three or four records are read to specify the details of the plot, and the pairs of records containing the wavelength-dependent data are read.

Record 1: ITITLE

This is a title for the top of the plot. Up to 80 alphanumeric characters are allowed.

Record 2: LABYAX

This is a label for the y-axis (the ordinate) of the plot. Up to 80 characters are allowed.

Record 3: NTRACE, ILOG, IAUTOY, IPLABL, IATMOS

NTRACE gives the number of data curves (traces) to be drawn on a given set of axes (i.e. on the same plot). (Up to 20 traces are allowed in the listed code; see parameter MXTRAC in program MPWAVE.)

ILOG = 0 if the actual data values are to be plotted
      = 1 if the logarithm (base 10) of the data is to be plotted

IAUTOY = 0 if Record 3a specifies the y-axis scaling
         = 1 if the plot program examines the data to determine convenient y-axis scaling

IPLABL = 0 if the plotted curves are not numbered or labeled
         = 1 if the plotted curves are to be numbered and labeled

IATMOS = 0 if the raw data values, P(I), are to be used
         = 1 if the raw data values are to be transformed by the atmospheric model

Record 3a: YINCH, YMAX, YMIN, IDIV, NCODE

This record is read only if IYAUTO = 0. If IYAUTO = 1, the default values shown below are used.

YINCH (default value: 6.0). The length y of the y-axis in inches. The x-axis is always 6.0 inches long and is labeled with wavelength values.

YMAX (default: internally generated). The maximum y value, used to label the y-axis and scale the data.
§7C. PLOTTING WAVELENGTH DATA

YMIN  (default: internally generated). The minimum y value, used to label the y-axis and scale the data.

IDIV  (default: 5 to 10, internally generated). The number of divisions in the y-axis labeling.

NCODE (default: 8, see record 3b). The number of characters to be transferred in the FORTRAN ENCODE statement. See record 3b.

Record 3b: YFMT
This record is read only if IYAUTO = 0.

YFMT is an execution-time format used to generate data values for labeling the tic marks on the y-axis. It should end with ",2HA-)", which draws tic marks with the "-" in the 2HA-. The "A" symbol indicates a blank. The default is (F6.2, 2HA-) This generates tic mark labels of the form

```
123.45 --- y-axis vertical line
```

where the "-" from the "2HA-" in the format is the plotted tic mark. The default value of NCODE = 8 is the total of 6 (from the F6.2) plus 2 (from the 2HA-).

The format

(F3.0, 2HA-)

would generate tic mark labels of the form

```
12.
```

Now, NCODE = 5

Record 4: LABTRC
This is a label for the plotted trace. Up to 80 alphanumeric characters are allowed.

Record 5: THETPS, P(1), P(2),...P(13)
This record has the same form as record 4 of Program MPCHRO: THETPS is the solar zenith angle and P(I), I = 1,...,13 is the data value for wavelength I.

Records 6 and 7, 8 and 9, ...
Pairs of records of the same form as 4 and 5 are repeated for each trace to be plotted. Up to 20 traces are allowed on the same plot.
§7C. PLOTTING WAVELENGTH DATA

2. Code Listing

PROGRAM MPWAVE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE98,TAPE99)
 
ON NHM6/MPWAVE
 
THIS PROGRAM PLOTS DATA AS A FUNCTION OF WAVELENGTH.
 
ALL PLOTTING IS DONE USING STANDARD CALCOMP CALLS
(TAPE98 AND TAPE99 ARE USED BY THE CALCOMP ROUTINES, AS
IMPLEMENTED ON THE AUTHOR'S CDC CYBER 855 COMPUTER.)
 
PARAMETER(MXWAVE=13, MXTRAC=20)
DIMENSION DATA(MXWAVE,MXTRAC),LABTRC(8,MXTRAC)
DIMENSION LABVAX(8),ITITLE(B)
COMMON/CSCALE/yINCH,VMAXYMIN, IDIVNCODE,VFMT(2)
COMMON/CWORK/X(MXPLT),V(MXPLT),TRACE(MXPLT,MXTRAC)
 
DATA NWAVEL/13/
 
CALL PLOTS
CALL PLOT(I.O,2.0,-3)
 
READ THE TITLE FOR THE TOP OF THE PLOT
998 READ(5,100,END=1000) ITITLE
 
TITLE FOR THE V AXIS
READ(5,100) LABVAX
 
READ PLOT SPECIFIERS
 
NTRACE = THE NUMBER OF TRACES (CURVES) TO BE DRAWN
ILOG = 1, IF LOG(BASE 10) OF THE DATA IS TO BE PLOTTED
0, IF ACTUAL DATA VALUES ARE TO BE PLOTTED
IAUTOY = 1, IF THE PLOT PROGRAM EXAMINES THE DATA TO DETERMINE
CONVENIENT Y AXIS (ORDINATE) SCALING
IPLABL = 1, IF THE PLOTTED CURVES ARE TO BE NUMBERED AND LABELLED
0, IF THE CURVES ARE NOT NUMBERED AND LABELLED
IATMOS = 1, IF THE RAW DATA VALUES ARE TO BE SCALLED BY THE
WAVELENGTH-SOLAR ANGLE DEPENDENT ATMOSPHERIC MODEL
0, IF THE RAW DATA ARE NOT SCALLED
 
READ(5,*) NTRACE,ILOG,IAUTOY,IPLABL,IATMOS
 
READ THE SPECIFICATIONS FOR SCALING THE Y (VERTICAL) AXIS, IF DESIRED
 
VINCH,...THE LENGTH OF THE Y AXIS, IN INCHES
VMAX, VMIN,...THE MAXIMUM AND MINIMUM Y VALUES TO BE USED FOR
THE Y AXIS LABELS
IDIV,...THE NUMBER OF DIVISIONS OF THE Y AXIS, FOR LABELLING TIC MARKS
NCODE,...THE NUMBER OF CHARACTERS IN THE TOTAL WIDTH OF THE Y AXIS
TIC MARK LABELS (USED IN ENCLOSE STATEMENTS). E.G. IF THE
NEXT DATA RECORD HAS (F6.2,2H1) AS THE FORMAT, THEN
NCODE = 6 + 2 = 8
VFMT,...A FORMAT FOR LABELLING THE Y AXIS TICS. IT SHOULD END
WITH .2H2 WHICH DRAWS THE TIC MARKS WITH THE MINUS SIGN
 
IF(IAUTOY.EQ.0) THEN
READ(5,*) VINCH,VMAX,VMIN,IDIV,NCODE
READ(5,100) VFMT
ELSE IF (IATMOS.EQ.0) THEN
READ(5,*) VINCH,VMAX,VMIN,IDIV,NCODE
READ(5,100) VFMT
ENDIF
 
READ THE TRACE LABELS AND THE DATA TO BE PLOTTED, ON STANDARD
WAVELENGTH FORMAT
 
 ON 200 NTH=1,NTHALF
READ(5,100) (LABTRC(I,NTR),I=1,8)
§7C. PLOTTING WAVELENGTH DATA

READ(5,*) THETPS, (DATA(I,NTR), I=1,NWAVEL)

IF(IATMOS.NE.0) THEN
  SCALE THE RAW DATA ACCORDING TO THE ATMOSPHERIC MODEL
  CALL ATMOS(THETPS,DATA(1,NTR))
END IF

200 CONTINUE

CALL PLOTVWL(DATA,NTRACE,ILOG,IAUTOY,LABTRC,IPLABL,LABYAX,ITITLE)
CALL PLOT(18.0,0.0,-3)
GO TO 998

100 CALL PLOT(J,0.0,-98)
WRITE(6,99)

C 100 FORMAT(8A10)
999 FORMAT(IH 'WAVELENGTH PLOTS COMPLETED'
END

SUBROUTINE PLOTVWL(DATA,NTRACE,ILOG,IAUTOY,LABTRC,IPLABL,LABYAX, ITITLE)

ON NIM6/PLT/WVL

THIS ROUTINE PLOTS DATA AS A FUNCTION OF THE 13 NHM WAVELENGTHS.

THE INPUT IS

DATA(13,NTRACE)...THE ARRAY OF DATA VALUES TO BE PLOTTED. EACH COLUMN HOLDS ONE FUNCTION OF WAVELENGTH TO BE PLOTTED AS ONE TRACE ON THE GRAPH.

ILOG...= 0, IF THE ACTUAL DATA VALUES ARE TO BE PLOTTED
       = 1, IF LOG(BASE 10) OF THE DATA IS TO BE PLOTTED

IAUTOY...= 0, IF THE Y (VERTICAL) SCALING IS PREDETERMINED IN THE MAIN PROGRAM COMMON BLOCK (SCALE)
       = 1, IF THE DATA VALUES SHOULD BE EXAMINED TO DETERMINE APPROPRIATE Y SCALING VALUES

LABTRC(1,NTRACE)...LABELS USED TO IDENTIFY THE TRACES, UP TO 80 CHARACTERS EACH. THE LABELS ARE PLOTTED IF IPLABL.NE.0

LABYAX(I)...A LABEL FOR THE VERTICAL (DATA) AXIS, UP TO 80 CHAR

ITITLE(I)...A TITLE FOR THE TOP OF THE PLOT, UP TO 80 CHAR

160
§7C. PLOTTING WAVELENGTH DATA

PARAMETER(MXWAVE=13, MXPLT=MXWAVE+2, MXTRAC=20)

DIMENSION DATA(MXWAVE,NTRACE), LABTRC(B,NTRACE), LABYAX(B)
DIMENSION TITLE(B), VMAX, YMIN, IDIV, NCODE, VFMT(2)
COMMON/CSCALE/, VINCH, VMAX, YMIN, IDIV, NCODE, VFMT(2)
COMMON/CWORX/, XPLT(MXPLT), YPLT(MXPLT), TRACE(MXPLT, MXTRAC)

DATA JSYMB/1/, LSYMB/1/, NWAVE/L, VCH/13/
DATA XINCH/B/, YINCH/B/, H/0.15/
DATA FMT/10(F6.2, 2H-.10H)

C SELECT ACTUAL OR LOG VALUES
IF(ILOG.EQ.0) THEN
  DO 100 J=1,NTRACE
  DO 100 I=1,NWAVE
  TRACE(I,J) = DATA(IJ)
  100 CONTINUE
ELSEIF(ILOG.EQ.1) THEN
  DO 102 J=1,NTRACE
  DO 102 I=1,NWAVE
  TRACE(I,J) = ALOG10(DATA(IJ))
  102 CONTINUE
IF(IAUTOY.EQ.0) THEN
  USE PREDETERMINED SCALING VALUES
  DINCH = YINCH/FLOAT(IDIV)
  DLABL = (VMAX - YMIN)/FLOAT(IDIV)
  FLABLO = YMAX
  YZERO = YMIN
  YPINCH = (VMAX - YMIN)/YIN
ELSE
  EXAMINE THE VALUES TO BE PLOTTED TO DETERMINE THE VERTICAL SCALING
  YMIN = 1.0E200
  VMAX = -1.0E200
  DO 110 J=1,NTRACE
  DO 110 I=1,NWAVE
  YMIN = AMIN1(YMIN,TRACE(I,J))
  YMAX = AMAX1(YMAX,TRACE(I,J))
  110 CONTINUE
  MINV = IFIX(YMIN)
  MAXV = IFIX(YMAX)
  MRANGE = MAXV - MINV
  IDIV = IABS(MRANGE)
  IF(5.LE.IDIV .AND. IDIV.LE.10) GO TO 300
  IF(IDIV.GT.10) GO TO 301
  IDIV = IDIV*2
  GO TO 302
  301 IDIV = (IDIV + 1)/2
  GO TO 302
  300 DLABL = FLOAT(IABS(MRANGE))/FLOAT(IDIV)
  IF(DLABL.LE.1.) GO TO 303
  IF(FLOAT(IFIX(DLABL)).EQ.DLABL) GO TO 303
  MRANGE = IDIV*IFIX(DLABL + 1.)
  GO TO 300
  303 DINCH = YINCH/FLOAT(IDIV)
  DLABL = FLOAT(MAXV)
  YZERO = FLOAT(MINV)
  YPINCH = FLOAT(MAXV - MINV)/YIN
  NCODE = B
  VFMT(1) = FMT(1)
  VFMT(2) = FMT(2)
END IF
§7C. PLOTTING WAVELENGTH DATA

C DRAW BORDER AND LABEL Y AXIS
C
CALL PLOT(0.,0.,3)
CALL PLOT(0.,VINCH,2)
CALL PLOT(XINCH,VINCH,2)
CALL PLOT(0.,0.,2)
C
XX = -(FLOAT(NCODE) - .4)*H
DO 310 I = 1,1000,1
YY = VINCH - 0.45*M - FLOAT(I-1)*VINCH
FLABL = FLABLD - FLOAT(I-1)*FLABL
ENCODE(NCODE,YFMT,BCD) FLABL
310 CALL SYMBOL(XX,YY,BCD,0.,0.,NCODE)
XX = 1.2
NC = NCHAR(LABYAX,8)
YY = 0.5*VINCH - 0.5*FLOAT(NC)*H
CALL SYMBOL(XX,YY,LABYAX,90.,0.,NC)
C
C DRAW HORIZONTAL AXIS
C
V1 = -0.45*M
V2 = -2.35*M
DO 120 I = 1,1000,1
XX = XINCH+FLOAT(I-1)/100.0*WAVEL-1:
CALL SYMBOL(XX,V1,LABL,1.,1.,-1):
IF(MOD(I,2),NE,0) THEN
LAMBDA = 400. + 25.*I-1:
ENCODE(3,122,BCD) LAMBDA:
CALL SYMBOL(XX-1.5*M,V2,LABYAX,90.,0.,NC)
ENDIF
120 CONTINUE
CALL SYMBOL(0.5*XINCH - 0.5*H,0.,0.,WAVELNGTH IN NM,0.,16)
C
C DRAW TITLE AT TOP
C
NC = NCHAR(ITLE,8)
XX = 0.5*XINCH - 0.5*FLOAT(NC)*H
YY = VINCH + 2.0*M
CALL SYMBOL(XX,YY,ITLE,0.,NC)
C
C PLOT THE TRACES
C
SET UP THE X COORDINATES, WITH SCALING FACTORS
DO 200 I = 1,WAVEL
200 XPLT(I) = 400.0 + 25.0*FLOAT(I-1)
XPLT(NWAVEL+1) = 400.0
XPLT(NWAVEL+2) = 300.
C
C SET UP Y COORDINATES AND PLOT
C
VPLT(NWAVEL+1) = YERU
VPLT(NWAVEL+2) = VINC -
GO TO 210, I = 1, NWAVE
DO 212 I = 1, NWAVE.
212 VPLT(I) = TRABL(I,J)
C
CALL LINE(XPLT,YPLT,ITPLT,1,0.5SYMB,0.0SYMB)
C
NUMBER THE TRACES
YY = WAVEL+1 - WAVEL+11*XPLT+WAVEL+11 - 0.5*M
ENCODE(4,214,BCD,0.)
CALL SYMBOL(XX-1.5*M,YY,BCD,0.,NC)
C
C PLOT FULL RANGE 1, 5, 10, 25
IF(IPLABL,LE,0.0) GO TO 210
FFN = FLOAT(J)
XX = XINCH + 1.5*
YY = VINCH - FFN*2.0*M
CALL NUMBER(XX,YY,FFN,INC,0.,NC)
NL = NCHAR(LABR(J),I,J,8)
CALL SYMBOL(XX+2.5*M,YY,LABR(J),0.,NC)
210 CONTINUE

162
§7C. PLOTTING WAVELENGTH DATA

RETURN
C
122 FORMAT(I3)
214 FORMAT(I2)
END
APPENDIX A. IMSL Routines Used by the NHM

The following IMSL subroutines are used by the NHM.

<table>
<thead>
<tr>
<th>IMSL routine</th>
<th>where called, program/subroutine</th>
<th>description of IMSL routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCADRE</td>
<td>4/Y2ZGEO</td>
<td>numerically integrates a function of one variable</td>
</tr>
<tr>
<td>DVERK</td>
<td>4/RICATI</td>
<td>solves systems of ordinary differential equations using a high-order Runge-Kutta scheme</td>
</tr>
<tr>
<td>EIGRF</td>
<td>4/EIGENR</td>
<td>finds eigenvalues and eigenvectors of a real-valued matrix</td>
</tr>
<tr>
<td>GGNML</td>
<td>1/INISHL</td>
<td>generates pseudo-random numbers from a Gaussian distribution</td>
</tr>
<tr>
<td>GGUBFS</td>
<td>1/MAIN</td>
<td>generates pseudo-random numbers from a uniform distribution</td>
</tr>
<tr>
<td>LINV1F</td>
<td>4/AMPINT</td>
<td>inverts a matrix</td>
</tr>
<tr>
<td>LINV2F</td>
<td>4/EIGENR</td>
<td>inverts a matrix (high-accuracy version)</td>
</tr>
<tr>
<td>VMULFF</td>
<td>4/EIGENR</td>
<td>multiplies two matrices</td>
</tr>
<tr>
<td>VSRTA</td>
<td>4/EIGENR</td>
<td>sorts an array by algebraic value</td>
</tr>
<tr>
<td>VSRTR</td>
<td>1/TIP</td>
<td>sorts an array by algebraic value and returns the permutations</td>
</tr>
</tbody>
</table>

APPENDIX B. A Simple Model for Incident Radiance Distributions

For some purposes, the input radiance distribution on the water surface can be approximated by a continuous sky radiance distribution plus a point sun.

For the continuous sky distribution we use a cardioidal radiance distribution given by

\[ N(\theta, \phi) = N_0(1 + C \cos \theta) \]

or

\[ N(\mu, \phi) = N_0(1 + C \mu) \]

where \( N_0 \) and \( C \) are constants to be chosen. Note that this sky radiance distribution is independent of azimuthal angle \( \phi \) or wavelength \( \lambda \). The form (B.1) yields the quad-averaged radiances

\[ N(u, v) = N_0(1 + C \mu_u) \]

where, as always, \( \mu_u \) is the average \( \mu \)-value of quad \( Q_{uv} \).

The scalar irradiance \( h(\text{sky}) \) of the radiance distribution (2) is given by 75/8.4:
APPENDICES

\[
m-1 \quad 2n
h(\text{sky}) = \sum_{u=1}^{m-1} \sum_{v=1}^{2n} N(u,v) \Omega_{uv} + N(m,\cdot) \Omega_m
= \sum_{u=1}^{m-1} \sum_{v=1}^{2n} N_0 (1 + C \mu_u) \Delta \mu_u \left( \frac{2\pi}{2n} \right) + N_0 (1 + C \mu_m) 2\pi \Delta \mu_m
\]

\[
= 2\pi N_0 \sum_{u=1}^{m} (1 + C \mu_u) \Delta \mu_u
= 2\pi N_0 \left[ \sum_{u=1}^{m} \Delta \mu_u + C \left( \sum_{u=1}^{m} \mu_u \Delta \mu_u \right) \right]
= 2\pi N_0 \left( 1 + \frac{C}{2} \right)
\]

For a uniform sky, \( C = 0 \), and we get \( h(\text{sky}) = 2\pi N_0 \), as expected.

The plane irradiance \( H(\text{sky}) \) is given by \( 75/8.7 \), which reduces to

\[
H(\text{sky}) = 2\pi N_0 \left( \frac{1}{2} + \frac{C}{3} \right)
\]

\[ (B.4) \]

after using \( \sum_{u=1}^{m} \mu_u^2 \Delta \mu_u = \frac{1}{3} \). For a uniform sky, \( C = 0 \), we get \( H(\text{sky}) = \pi N_0 \), as expected.

The well-known cardioidal radiance distribution, which approximates a heavy overcast with no discernible sun, corresponds to \( C = 2 \).

Subroutine QASKY uses (B.2) as background for a point sun. Using this model, we can study the effects of going from all direct beam (the sun in a black sky) to all diffuse light (heavy overcast), while keeping the total scalar irradiance constant.

Let \( h(\text{sun}) + h(\text{sky}) = h(\text{total}) \), and define the ratio of sky scalar irradiance to total scalar irradiance as

\[
R \equiv \frac{h(\text{sky})}{h(\text{sun}) + h(\text{sky})} = \frac{h(\text{sky})}{h(\text{total})}
\]

Using (B.3) and (B.5) and solving for \( N_0 \) gives

165
\[ N_0 = \frac{R \cdot h(\text{total})}{2\pi \left( 1 + \frac{C}{2} \right)} \]  
(B.6)

and \( h(\text{sun}) = h(\text{total}) \cdot (1-R) \).

Subroutine QASKY computes the quad-averaged input radiances by

\[ N(u,v) = \frac{R(1 + C\mu_0)}{2\pi \left( 1 + \frac{C}{2} \right)} h(\text{total}) \]  
for "sky only" quads, and

\[ N(u,v) = \left[ \frac{R(1 + C\mu_0)}{2\pi \left( 1 + \frac{C}{2} \right)} + \frac{1-R}{\Omega_{uv}} \right] h(\text{total}) \]  
(B.8)

for the "sky + sun" quad. Note that for \( R = 1 \) (no sun) and \( C = 0 \) (uniform sky), each quad gets a quad-averaged radiance of magnitude \( N(u,v) = h(\text{total})/2\pi \).

**APPENDIX C. Computation of Chromaticity Coordinates**

The standard way of displaying water color is the *chromaticity diagram*\(^*\). The chromaticity coordinates \( X, Y, Z \) are given by

\[ X = 680 \int_{400}^{700} P(\lambda) \bar{x}(\lambda) \, d\lambda \]  
(C.1)

with corresponding equations for \( Y \) and \( Z \). Here \( \lambda \) is wavelength in nanometers, \( P(\lambda) \) is a radiance or irradiance, and \( \bar{x}(\lambda) \) is the tristimulus (color matching) function for red.

This integral can be approximated by Simpson's rule if the 400-700 nm interval is divided into an even number of subintervals. For runs with the NHM we choose 12 subintervals of width \( \Delta \lambda = 25 \) nm, and run the monochromatic NHM at the 13 wavelengths of \( \lambda_1 = 400 \) nm, \( \lambda_2 = 425 \) nm, ..., \( \lambda_{13} = 700 \) nm. Then \( X \) is computed by

\[ X = 680 \frac{\Delta \lambda}{3} \left[ P(400) \bar{x}(400) + 4P(425) \bar{x}(425) + 2P(450) \bar{x}(450) + \cdots + 2P(650) \bar{x}(650) + 4P(675) \bar{x}(675) + P(700) \bar{x}(700) \right] \]  
(C.2)

---

\(^*\) See, for example, *Hydrologic Optics, Vol. 1, Introduction*, by R.W. Preisendorfer, Pacific Marine Environmental Laboratory/NOAA, Honolulu, HI, pages 142-151. Available from NTIS as document no. PB-259793/8ST.
The normalized chromaticity coordinates are given by
\[ x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}. \] (C.3)

The \((x,y)\) normalized coordinates can be used to plot a point on a 1931 C.I.E. chromaticity diagram.

The table below gives the values of \(x(\lambda), y(\lambda), z(\lambda)\) for the 13 \(\lambda\)-values used in the NHM.

<table>
<thead>
<tr>
<th>(\lambda)</th>
<th>(x(\lambda))</th>
<th>(y(\lambda))</th>
<th>(z(\lambda))</th>
<th>weight for Simpson’s rule integrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>.0143</td>
<td>.0004</td>
<td>.0679</td>
<td>1</td>
</tr>
<tr>
<td>425</td>
<td>.2148</td>
<td>.0073</td>
<td>1.0391</td>
<td>4</td>
</tr>
<tr>
<td>450</td>
<td>.3362</td>
<td>.0380</td>
<td>1.7721</td>
<td>2</td>
</tr>
<tr>
<td>475</td>
<td>.1421</td>
<td>.1126</td>
<td>1.0419</td>
<td>4</td>
</tr>
<tr>
<td>500</td>
<td>.0049</td>
<td>.3230</td>
<td>.2720</td>
<td>2</td>
</tr>
<tr>
<td>525</td>
<td>.1096</td>
<td>.7932</td>
<td>.0573</td>
<td>4</td>
</tr>
<tr>
<td>550</td>
<td>.4334</td>
<td>.9950</td>
<td>.0987</td>
<td>2</td>
</tr>
<tr>
<td>575</td>
<td>.8425</td>
<td>.9154</td>
<td>.0018</td>
<td>4</td>
</tr>
<tr>
<td>600</td>
<td>1.0622</td>
<td>.6310</td>
<td>.0008</td>
<td>2</td>
</tr>
<tr>
<td>625</td>
<td>.7514</td>
<td>.3210</td>
<td>.0001</td>
<td>4</td>
</tr>
<tr>
<td>650</td>
<td>.2835</td>
<td>.1070</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>675</td>
<td>.0636</td>
<td>.0232</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>700</td>
<td>.0114</td>
<td>.0041</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

| sums        | 4.2699         | 4.2712         | 4.2617         |

Integrals for \(P(\lambda) = 1\)
\[ 72319, \quad 73005, \quad 72170. \]

Note that the integrals \( \int_{400}^{700} x(\lambda) \, d\lambda = 72319 \), etc., agree to within 1%, which is the same order of accuracy as the output of the NHM.

Converting \((x,y)\) into (dominant wavelength, purity)

Subroutine CHRMXY draws the spectrum locus of the chromaticity diagram by connecting 37 tabulated pure-color coordinates \([x_p(I), y_p(I)], I = 1, \ldots, 37\) to make a closed curve. The

computed \((x, y)\) point is then plotted on this diagram. For the plotted point \((x, y)\) we can compute a dominant wavelength (or dominant color), \(\lambda\), and a purity, \(p\). For a point \((x, y)\) on the diagram drawn by CHRMXY this is a simple exercise in analytic geometry, and proceeds as follows.

1) First compute the slope of the line between the white point \((x_w, y_w)\) and each of the 37 plotted spectrum locus points \([x_p(I), y_p(I)]\), \(I = 1, \ldots, 37\).

2) Then compute the slope of the line between the white point \((x_w, y_w)\) and the plotted point \((x, y)\).

3) Then search through the set of "spectrum locus slopes" from 1) until the slope from 2) is located between the \(I^{th}\) and \((I+1)^{st}\) spectrum locus slopes. We now know that the dominant wavelength \(\lambda\) will be somewhere between \(\lambda_I\) and \(\lambda_{I+1}\), where \(\lambda_I\) is the wavelength of the \(I^{th}\) plotted spectrum locus point.

Since different pairs of points \((x_1, y_1), (x_w, y_w)\) and \((x_2, y_2), (x_w, y_w)\) can have the same slope, it is necessary to note if \(x \leq x_w\) or \(x > x_w\) and if \(y \leq y_w\) or \(y > y_w\). The slopes in the corresponding quadrant of the chromaticity diagram (lower left, etc.) can then be searched.

4) Compute the intersection point \((x_i, y_i)\) between the line connecting the \(I^{th}\) and \((I+1)^{st}\) spectrum locus points and the line determined by the white point and the plotted point. The point \((x_i, y_i)\) is computed from the solution of

\[
\begin{align*}
\frac{x_i - x_w}{y_i - y_w} &= \frac{x - x_w}{y - y_w} \equiv s_1 & \text{the line determined by } (x, y) \text{ and } (x_w, y_w) \\
\frac{x_i - x_{p(I-1)}}{y_i - y_{p(I-1)}} &= \frac{x_{p(I)} - x_{p(I-1)}}{y_{p(I)} - y_{p(I-1)}} \equiv s_2 & \text{the line determined by } [x_p(I), y_p(I)] \text{ and } [x_p(I+1), y_p(I+1)]
\end{align*}
\]

which gives

\[
x_i = \frac{s_2 \cdot x_w - s_1 \cdot x_{p(I-1)} - s_1 \cdot s_2 \cdot y_w - y_{p(I-1)}}{s_2 - s_1} \quad \text{(C.4)}
\]

\[
y_i = \frac{s_2 \cdot x_w - s_1 \cdot x_{p(I-1)} - s_1 \cdot y_w + s_2 \cdot y_{p(I-1)}}{s_2 - s_1}
\]

5) Given the intersection point \((x_i, y_i)\), compute the distance \(d_1\) from \([x_p(I), y_p(I)]\) to \((x_i, y_i)\) and the distance \(d_2\) from \((x_i, y_i)\) to \([x_p(I+1), y_p(I+1)]\). Then the dominant wavelength is

\[
\lambda = \left(1 - \frac{d_1}{d_7}\right) \lambda_I + \left(\frac{d_1}{d_2}\right) \lambda_{I+1} \quad \text{(C.5)}
\]

6) Compute the distance \(d_3\) from \((x, y)\) to \((x_w, y_w)\) and the distance \(d_4\) from \((x_i, y_i)\) to \((x_w, y_w)\). Then the purity is

\[
p = \frac{d_1}{d_4} \quad \text{(C.6)}
\]
APPENDIX D. A Simple Model Atmosphere and Solar Spectrum

The input required by the NHM is the incident radiance at sea level. If the NHM is being used at only one wavelength, then the input spectral scalar irradiance can be set to some convenient value, say 1.0 W m\(^{-2}\) nm\(^{-1}\). However, if runs are being made at various wavelengths and the results are being combined, e.g. to compute colors, then the radiance on the water surface should account for atmospheric effects and for the wavelength dependence of the solar spectrum. It is usually most convenient to make all NHM runs with the same input, and then to correct the output when computing colors, etc. This is allowed by the linearity of the radiative transfer equation.

Subroutine ATMOS uses a crude model atmosphere which depends only on the solar zenith angle, \(\theta_s\), to incorporate atmospheric path length effects on the sun’s direct beam. This routine also uses tabulated solar spectrum values to incorporate the wavelength dependence of the solar spectrum. The model is based on tabulated values of the scalar irradiance at sea level* for atmospheric conditions of

- pressure = 760 mm Hg
- 2.0 cm of precipitable water vapor per unit of optical air mass
- 300 dust particles per cm\(^3\) in the air
- 0.28 cm of ozone per unit of optical air mass

The optical air mass is 1 when \(\theta_s = 0^\circ\) (the sun is at the zenith); it is 2 = sec 60\(^\circ\) when \(\theta_s = 60^\circ\), and so on. The scalar irradiance at sea level, \(h_{SL}(\lambda, \theta_s)\), is given by

\[
h_{SL}(\lambda, \theta_s) = h_s(\lambda) e^{-\alpha_s \sec \theta_s} \tag{D.1}
\]

where \(h_s(\lambda)\) is the solar scalar irradiance at wavelength \(\lambda\), outside the atmosphere, and \(\alpha_s\) includes all scattering and absorption effects of the model atmosphere. The table below gives the values of \(h_s(\lambda)^\dagger\) and \(\alpha_s^\ddagger\).


† These values are taken from *Hydrologic Optics, Vol. 1*, page 23. The associated solar constant is 1396 W m\(^{-2}\), which is somewhat too large.

‡ From the *Handbook of Geophysics and Space Environments*. The associated solar constant is 1322 W m\(^{-2}\), which is somewhat too low. The \(\alpha_s\) are rescaled to be consistent with 1396 W m\(^{-2}\).
APPENDICES

<table>
<thead>
<tr>
<th>$\lambda$ (nm)</th>
<th>$h_\lambda(\lambda)$ (W m$^{-2}$ nm$^{-1}$)</th>
<th>$\alpha_\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>1.54</td>
<td>.566</td>
</tr>
<tr>
<td>425</td>
<td>1.89</td>
<td>.428</td>
</tr>
<tr>
<td>450</td>
<td>2.20</td>
<td>.364</td>
</tr>
<tr>
<td>475</td>
<td>2.20</td>
<td>.293</td>
</tr>
<tr>
<td>500</td>
<td>1.98</td>
<td>.217</td>
</tr>
<tr>
<td>525</td>
<td>1.92</td>
<td>.210</td>
</tr>
<tr>
<td>550</td>
<td>1.95</td>
<td>.220</td>
</tr>
<tr>
<td>575</td>
<td>1.87</td>
<td>.206</td>
</tr>
<tr>
<td>600</td>
<td>1.81</td>
<td>.192</td>
</tr>
<tr>
<td>625</td>
<td>1.72</td>
<td>.165</td>
</tr>
<tr>
<td>650</td>
<td>1.62</td>
<td>.134</td>
</tr>
<tr>
<td>675</td>
<td>1.53</td>
<td>.114</td>
</tr>
<tr>
<td>700</td>
<td>1.44</td>
<td>.104</td>
</tr>
</tbody>
</table>