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

UNITED STATES DEPARTMENT OF COMMERCE

C.William Verity
Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

Environmental Research Laboratories

Vernon E. Derr,
Director

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

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.

(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, Sprinfield 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

> Unpolarized Irradiance Reflectances and Glitter Patterns of Random Capillary Waves on Lakes and Seas, by Monte Carlo Simulation, by Rudolph W. Preisendorfer and Curtis D. Mobley, NOAA Tech. Memo. ERL PMEL-63, Pacific Marine Environmental Laboratory, Seattle, WA 98115, Sept. 1985, 141 pages. (Available from NTIS as report number PB86-123577.)

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 refering 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 (9^{th} 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 ParkWest Tower One
2500 CityWest Boulevard
Houston, TX 77042-3020

[2] Licensed and distributed by
Numerical Algorithms Group, Inc.
1101 31^{st} 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 1

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 $\phi = 180°$ 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 $\phi = 180°$. 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 1 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

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 $\underline{r}(a,x)$ and $\underline{t}(a,x)$ is computed; if the rays are water-incident, one row of $\underline{r}(x,a)$ and $\underline{t}(x,a)$ 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 $\underline{r}$ and $\underline{t}$ arrays, for a given quad resolution and wind speed. This determination must be empirically made, and the individual elements of the f(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_{rs}, the output quads Q_{uv} which are near the specular (still water) reflection or refraction directions of the parent rays in Q_{rs} 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_{rs} 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 $\underline{r}$ and $\underline{t}$ 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_{rs}. The computed values of r(a,x; r,slu,v) and t(a,x: r,slu,v), u = 1,···,m and v = 1,···,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_{rs}, 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 (see the first PARAMETER statement in the MAIN program)

parameter	value in listed code	definition
MXMU	10	the maximum number of quads in the μ-direction in a hemisphere, including the polar cap
MXPHI	24	the maximum number of quads in the ϕ-direction, $0 \le \phi \le 2\pi$
MXSTAK	10	the maximum number of rays in the push-pull stack at once
MXNHEX	7	the maximum order of the hexagonal grid used for ray tracing

Refering to 75/§3a, MXMU gives the maximum allowed value of m (≡ NMU, below), and MXPHI gives the maximum value of N (≡ NPHI, 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 MXMU ≥ 23 and MXPHI ≥ 60. For efficient use of computer storage, one should pick MXMU and MXPHI to be the same as the actual number of μ and ϕ cells in the quad partitioning, NMU and NPHI, respectively, to be specified in record 2, below. The value of MXSTAK = 10 should be sufficient for any problem (see 63/page 11, i.e. page 11 in the ray-tracing report). MXNHEX = 7 is sufficient for simulation of capillary wave surfaces. 75/Fig. 8 and 63/Fig. 5 each show hexagonal grids of order two (NHEX = 2). Proper resolution of mixed gravity-capillary waves requires high-order hexagonal grids (NHEX of 100 or more), and so 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, DSEED

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 (NHEX = 7 is adequate for capillary waves).

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 $\Delta\theta$ 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, $1 \le IR \le NMU$, the rays are air-incident. If IR is negative, $-NMU \le IR \le -1$, the rays are water-incident. JS must be in the first quadrant, i.e. $1 \le JS \le NPHI/4+1$.

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:

symbolic filename	external filename	description
NUSFC	TAPE15	the file of random surface realizations; created in the initial run of Program 1, and read in the ray-tracing run of Program 1
NUDU	TAPE16	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,s|u,v)
NUDD	TAPE17	ray data file for initial rays downward, final rays downward; used to compute t(a,x; r,s|u,v)
NUUD	TAPE18	ray data file for initial rays upward, final rays downward; used to compute r(x,a; r,s|u,v)
NUUU	TAPE19	ray data file for initial rays upward, final rays upward; used to compute t(x,a; r,s|u,v)

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.

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE15,
     1             TAPE16,TAPE17,TAPE18,TAPE19)
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                  +
C     +  THIS IS PROGRAM 1 OF THE NATURAL HYDROSOL MODEL +
C     +                                                  +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     ON NHM1/M1ALL
C
C     THIS PROGRAM BEGINS COMPUTATION OF THE GEOMETRIC REFLECTANCE AND
C     TRANSMITTANCE ARRAYS WHICH DESCRIBE THE AIR-WATER INTERFACE FOR
C     A GIVEN WIND SPEED.
C
C     THIS STANDARD VERSION OF MAIN1 DOES ALL INPUT QUADS IN THE FIRST
C     QUADRANT
C
C     NOTE:  THIS VERSION OF THE CODE STRIVES TO MINIMIZE THE EXECUTION
C     TIME, AT THE EXPENSE OF MODULARITY AND READABILITY OF THE CODE.
C     SOME SECTIONS OF FREQUENTLY EXECUTED CODE ARE WRITTEN AS STRAIGHT
C     LINE CODE WITH SIMPLE VARIABLES, RATHER THAN BEING GROUPED IN
C     SUBROUTINES OR DO LOOPS WITH ARRAYS, IN ORDER TO AVOID CALLING
C     AND INDEXING OVERHEAD.  ALMOST ALL ERROR CHECKING AND INTERMEDIATE
C     OUTPUT HAS BEEN REMOVED.
C
C     THIS PROGRAM USES THE MONTE CARLO RAY TRACING TECHNIQUE
C     DESCRIBED IN NOAA TECH MEMO ERL-PMEL-63.  COMMENTS REFERRING TO
C     THIS REPORT ARE PREFACED BY 63/.  THUS 63/2.12 REFERS TO
C     EQUATION 2.12 IN TECH MEMO 63.
C     REFERENCES WITHOUT THE 63/ REFER TO NOAA TECH MEMO ERL-PMEL-75.
C
C     NUSFC = TAPE15...CONTAINS THE RANDOM SURFACE REALIZATIONS
C
C     RESULTS OF COMPLETED RAY PATHS ARE WRITTEN TO FILES AS FOLLOWS:
C
C     NUDU = TAPE16...INITIAL RAY DOWNWARD, FINAL RAY   UPWARD: R- = R(A,X)
C     NUDD = TAPE17...INITIAL RAY DOWNWARD, FINAL RAY DOWNWARD: T- = T(A,X)
C     NUUD = TAPE18...INITIAL RAY   UPWARD, FINAL RAY DOWNWARD: R+ = R(X,A)
C     NUUU = TAPE19...INITIAL RAY   UPWARD, FINAL RAY   UPWARD: T+ = T(X,A)
C
C     PROGRAM 2 READS THESE FILES AND TALLIES THE RESULTS TO GENERATE
C     THE ACTUAL R AND T ARRAYS.
C
      PARAMETER(MXMU=10, MXPHI=24, MXSTAK=10, MXNHEX=7)
      PARAMETER(MXNODE=3*MXNHEX*(MXNHEX+1)+1, MXNTIP=4*MXNHEX+1)
C
      COMMON/CMUPHI/ BNDMU(MXMU),BNDPHI(MXPHI)
      COMMON/CNODES/ NNODE,FNODE(2,MXNODE),ZNODE(MXNODE)
      COMMON/CHEXGR/ NHEX,R1(2),R2(2),R1HAT(2),R2HAT(2),R1RAT,TARGET(2)
      COMMON/CTIP/ NTIP,SMIN,YTIP(2,MXNTIP),S(MXNTIP),KS(MXNTIP),ZMIN,
     1             ZMAX
      COMMON/CSTACK/ NSTACK,STACK(MXSTAK,7)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DIMENSION PIN(3),XIIN(3),P(3),XIREFL(3),XIREFR(3)
      DIMENSION NRAYQD(MXMU),NBRNCH(10)
      DOUBLE PRECISION DSEED
C
      DATA RADEPS/1.0E-10/, NUSFC,NUDU,NUDD,NUUD,NUUU/15,16,17,18,19/
      DATA NSTACK/0/, KTRACE/0/, NBRNCH/10*0/, NREFLO,NREFRO,NTIR/3*0/
C
C     INITIALIZE THE PROGRAM
C
      CALL INISHL(NREADO,NRAYQD,DSEED)
C
      NMU = IMISC(1)
```

```
      NPHI = IMISC(2)
      PI = FMISC(1)
      TWOPI = 2.0*PI
      JPI2 = NPHI/4 + 1
      DPHINP = BNDPHI(2) - BNDPHI(1)
      NUMDU = 0
      NUMDD = 0
      NUMUD = 0
      NUMUU = 0
      NUMTP1 = 0
C
C*****  BEGIN COMPUTATIONS  *****
C
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
C     LOOP OVER MU PRIME CELLS (THETA = -PI/2 TO +PI/2)
C
      DO 1001 I=-NMU,NMU
      IF(I.EQ.0) GO TO 1001
C
C     GET MU PRIME BOUNDARIES OF THE INCOMING QUAD
      FMUMIN = 0.
      IA = IABS(I)
      IF(IA.GT.1) FMUMIN = BNDMU(IA-1)
      DMU = BNDMU(IA) - FMUMIN
      NUMRAY = NRAYQD(IA)
C
      IF(IA.EQ.NMU) THEN
C     IF I IS A POLAR CAP, DO ONLY J = 1 INDEX
      JCOMP = 1
      PHIMIN = 0.
      DPHI = TWOPI
      ELSE
C     FOR NON-POLAR QUADS, DO ONLY FIRST QUADRANT (0 .LE. PHI PRIME .LE. PI/2)
      JCOMP = JPI2
      PHIMIN = BNDPHI(NPHI)
      DPHI = DPHINP
      ENDIF
C
C     LOOP OVER PHI PRIME CELLS (0 .LE. PHI PRIME .LE. PI/2, OR 2*PI IF POLAR CAP)
C
      DO 1000 J=1,JCOMP
C
      IF(J.GT.1) PHIMIN = BNDPHI(J-1)
C
C     LOOP OVER THE RANDOM STARTING POINTS WITHIN THE QUAD
C     NOTE THAT DIFFERENT QUADS MAY HAVE DIFFERENT NUMBERS OF RAYS TRACED
C
      REWIND NUSFC
      READ(NUSFC) HEADER
      READ(NUSFC) HEADER
      NREAD = NREAD0
      DO 1000 NRAY=1,NUMRAY
C
C     SELECT A SURFACE REALIZATION
C
   55 CONTINUE
      IF(NREAD.EQ.1) THEN
C     READ A SURFACE REALIZATION AS GENERATED
      READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(IZ),IZ=1,NNODE)
C
      ELSEIF(NREAD.EQ.2) THEN
C     READ THE SURFACE AS ROTATED BY 180 DEGREES
      READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(IZ),IZ=NNODE,1,-1)
C
      ELSEIF(NREAD.EQ.3) THEN
C     READ THE SURFACE AND INVERT
      READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(IZ),IZ=1,NNODE)
      DO 502 IZ=1,NNODE
  502 ZNODE(IZ) = -ZNODE(IZ)
C
      ELSEIF(NREAD.EQ.0) THEN
C     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(IZ) = -ZNODE(IZ)
      ENDIF
```

```
C
      GO TO 506
C     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
C
C     CHOOSE A RANDOM MU PRIME VALUE
C     I POSITIVE (NEGATIVE) GIVES UPWARD (DOWNWARD) RAYS WITH MU PRIME =
C     XI PRIME(3) = XIIN(3) POSITIVE (NEGATIVE)

  777 RMU = (FMUMIN + DMU*GGUBFS(DSEED))*SIGN(1.0,FLOAT(I))
C     NO RAYS FROM THE POLE ITSELF
      IF(ABS(RMU).GT.1.0-RADEPS) GO TO 777
      ROOT = SQRT(1.0 - RMU*RMU)
C
C     CHOOSE A RANDOM PHI VALUE
C
      SPHI = AMOD(PHIMIN + GGUBFS(DSEED)*DPHI,TWOPI)
C
C     DEFINE A TEMPORARY RAY AS -XI PRIME, AND FOLLOW THIS RAY TO
C     THE BOUNDARY TO GET SMIN
      XIIN(1) = -ROOT*COS(SPHI)
      XIIN(2) = -ROOT*SIN(SPHI)
      XIIN(3) = -RMU
C
      CALL TIP(TARGET,XIIN,0)
C     DEFINE THE INITIAL POINT ON THE HEXAGON BOUNDARY
      TEMP = SMIN/FMISC(20)
      PIN(1) = TARGET(1) + TEMP*XIIN(1)
      PIN(2) = TARGET(2) + TEMP*XIIN(2)
      PIN(3) = TEMP*XIIN(3)
C
C     RESET XIIN TO THE DESIRED INCIDENT RAY DIRECTION, XI PRIME
C     (THE DIRECTION OF PHOTON TRAVEL)
      XIIN(1) = -XIIN(1)
      XIIN(2) = -XIIN(2)
      XIIN(3) = -XIIN(3)
      RAD = 1.0
      INRAY = 1
C
C     PERFORM RAY TRACING COMPUTATIONS
C+++++  THIS IS THE RECURSIVE TREE FOR A GIVEN INITIAL RAY  +++++
C
      KBRNCH = 0
  999 CALL TRACE(INRAY,RAD,PIN,XIIN, IOUT,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
C     RAY HAD NO FACET INTERCEPTS.
C
C     GET THE QUAD INDICES OF THE FINAL RAY DIRECTION
      PHIFIN = AMOD(ATAN2(XIIN(2),XIIN(1))+TWOPI,TWOPI)
      AMUFIN = XIIN(3)
      CALL MPINDX(AMUFIN,PHIFIN,K,L)
C
C     RECORD THE RESULT FOR THE APPROPRIATE R OR T CONTRIBUTION
C
      IF(I.LT.0) THEN
C
C     DOWNWARD INITIAL RAY
      IF(AMUFIN.GT.0.0) THEN
C     UPWARD FINAL RAY
      NUMDU = NUMDU + 1
      WRITE(NUDU) -I,J,K,L,RAD
      ELSE
C     DOWNWARD FINAL RAY
      NUMDD = NUMDD + 1
      WRITE(NUDD) -I,J,K,L,RAD
      ENDIF
```

```
C
      ELSE
C
C     UPWARD INTIAL RAY
      IF(AMUFIN.GT.0.0) THEN
C     UPWARD FINAL RAY
      IF(RAD.EQ.1.0) THEN
C     ERROR RAY, DUE TO FINITE HEXAGON
      NUMTP1 = NUMTP1 + 1
      ELSE
      NUMUU = NUMUU + 1
      WRITE(NUUU) I,J,K,L,RAD
      ENDIF
C
      ELSE
C     DOWNWARD FINAL RAY
      NUMUD = NUMUD + 1
      WRITE(NUUD) I,J,K,L,RAD
      ENDIF
      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
      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
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
      CALL PULL(RAD,PIN,XIIN)
      GO TO 999
      ENDIF
C
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
C
 1000 CONTINUE
 1001 CONTINUE
C
C*****  END OF COMPUTATIONS  *****
C
      ENDFILE NUDU
      ENDFILE NUDD
      ENDFILE NUUD
      ENDFILE NUUU
C
      NRAYTL = IMISC(17)
      WRITE(6,600) NRAYTL,KTRACE
      WRITE(6,601) NREFLO,RADEPS,NREFRO,RADEPS,NTIR
      WRITE(6,602) NUMDU,NUMDD,NUMUD,NUMUU,NUMTP1
      WRITE(6,604) (K,K=2,10),(NBRNCH(K),K=2,10)
      WRITE(6,605)
```

```
C
C     FORMATS
C
  514 FORMAT(1H ,' NREAD =',I2,3X,
     1'FILE OF SURFACE REALIZATIONS EXHAUSTED.  UNIT',I3,' REWOUND.')
  600 FORMAT(1H0,' END OF RAY TRACING COMPUTATIONS'/
     21H ,I10,' TOTAL RAYS WERE STARTED IN THIS RUN'/
     31H ,I10,' TOTAL RAYS WERE TRACED TO COMPLETION')
  601 FORMAT(1H ,I5,' REFLECTED RAYS WITH RADIANCE .LT.',1PE9.1,
     1' WERE DISCARDED'/I6,' REFRACTED RAYS WITH RADIANCE .LT.',
     2E9.1,' WERE DISCARDED'/1H ,I10,
     3' TOTAL INTERNAL REFLECTIONS OCCURRED')
  602 FORMAT(1H ,I10,' RAYS STARTED DOWNWARD AND FINISHED   UPWARD'/
     11H ,I10,' RAYS STARTED DOWNWARD AND FINISHED DOWNWARD'/
     21H ,I10,' RAYS STARTED   UPWARD AND FINISHED DOWNWARD'/
     31H ,I10,' RAYS STARTED   UPWARD AND FINISHED   UPWARD'/
     41H ,I10,' RAYS STARTED   UPWARD AND FINISHED   UPWARD WITH RAD = 1
     5.0 (DISCARDED)')
  604 FORMAT(1H0,' BRANCH OCCURRENCE TALLY'//'  NUM BRANCHES:    ',
     18I10,I7,' OR MORE'/'  NUM OCCURRENCES:',9I10)
  605 FORMAT(1H0,' NORMAL EXIT FROM NHM, PROGRAM I.')
      END
```

```
      SUBROUTINE INISHL(NREAD0,NRAYQD,DSEED)
C
C     ON NHM1/IN1ALL
C
C     THIS ROUTINE INITIALIZES NHM1/M1ALL
C
C     TWO INPUT RECORDS ARE READ:
C     RECORD 1 (DEFINES THE HEXAGON GRID AND THE WATER SURFACE):
C
C     IDBUG = 0 FOR MINIMAL OUTPUT
C           = 1 FOR GREATER OUTPUT
C           = 2 FOR FULL DEBUGGING OUTPUT
C     IGENSF = 0 IF A FILE OF RANDOM SURFACES ALREADY EXISTS (USUAL CASE)
C           .GT.0 IF THIS IS A SPECIAL RUN FOR GENERATING AND SAVING A
C                 FILE OF RANDOM SURFACES.  IGENSF SURFACES WILL BE GENERATED.
C     NHEX = THE ORDER OF THE HEXAGONAL SURFACE GRID (= MXNHEX FOR EFFICIENCY)
C     WNDSPD = THE WIND SPEED IN M/SEC AT 12.5 M ELEVATION
C     DSEED = THE SEED FOR RANDOM NUMBER GENERATION
C
C     RECORD 2 (DEFINES THE QUAD GRID AND SELECTS RAY PARAMETERS):
C
C     NMU = THE NUMBER OF MU CELLS IN ONE HEMISPHERE (0 TO PI/2)
C     NPHI = THE NUMBER OF PHI CELLS (0 TO 2*PI).  MUST BE A MULTIPLE OF 4
C     MUPART = 1 IF ALL QUADS ARE TO HAVE EQUAL SOLID ANGLES
C              2 IF ALL QUADS ARE TO HAVE EQUAL DELTA THETA VALUES
C     NREAD0 = 1, IF THE SURFACE REALIZATION FILE (NUSFC) IS TO BE READ
C                 FROM THE BEGINNING
C            = 2, 3, OR 0, IF NUSFC IS TO BE READ STARTING WITH A ROTATION
C                 OR INVERSION (SEE LOOP 55 IN MAIN)
C     NUMRAY:  IF NUMRAY.GT.0, NUMRAY IS THE NUMBER OF RAYS
C                 TO BE SENT FROM EACH INPUT QUAD (NRAYQD(IR) = NUMRAY)
C              IF NUMRAY.LT.0, THE NEXT RECORD GIVES
C                 NRAYQD(IR),IR=1,2,...,NMU
C
```

```
      PARAMETER(MXMU=10, MXPHI=24)
      PARAMETER(MXNHEX=7, MXNODE=3*MXNHEX*(MXNHEX+1)+1)
      COMMON/CMUPHI/ BNDMU(MXMU),BNDPHI(MXPHI)
      COMMON/CHEXGR/ NHEX,R1(2),R2(2),R1HAT(2),R2HAT(2),R1RAT,TARGET(2)
      COMMON/CNODES/ NNODE,FNODE(2,MXNODE),ZNODE(MXNODE)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DIMENSION FMU(MXMU),PHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),
     1          NRAYQD(MXMU)
C
      DATA PI,DEGRAD,RADEG/3.141592654, 0.01745329252, 57.2957795/
      DATA REFR/1.333333333/
      DATA DELTA, EPS/1.0, 1.111/, TARGET/0.5, 0.370333333/
      DATA NUSFC,NUDU,NUDD,NUUD,NUUU/15,16,17,18,19/
C
C     READ THE INPUT RECORDS
C
      READ(5,*) IDBUG,IGENSF,NHEX,WNDSPD,DSEED
      WRITE(6,300) NHEX,WNDSPD,DSEED
C
      IF(IGENSF.EQ.0) THEN
      READ(5,*) NMU,NPHI,MUPART,NREAD0,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
C     GET THE TOTAL NUMBER OF RAYS TO BE TRACED
      NUMRAY = 0
      DO 1100 I=1,NMU-1
 1100 NUMRAY = NUMRAY + NRAYQD(I)
      NUMRAY = 2*(NUMRAY*(NPHI/4 + 1) + NRAYQD(NMU))
C
      WRITE(6,301) NMU,NPHI,NREAD0,NUMRAY
      ENDIF
C
C     STORE THE NEEDED PARAMETERS
C
      IMISC(1) = NMU
      IMISC(2) = NPHI
      IMISC(9) = IDBUG
      IMISC(17) = NUMRAY
      FMISC(1) = PI
      FMISC(2) = DEGRAD
      FMISC(3) = RADEG
      FMISC(15) = WNDSPD
      FMISC(16) = DELTA
      FMISC(17) = EPS
      FMISC(18) = REFR
C     RAD48 IS THE CRITICAL ANGLE FOR TOTAL INTERNAL REFLECTION
      RAD48 = ASIN(1.0/REFR)
      FMISC(19) = RAD48
C
      IF(IGENSF.GT.0) THEN
C
C*****THIS IS AN INITIAL RUN FOR GENERATION OF A FILE OF RANDOM SURFACES
C
      WRITE(6,304)
      REWIND NUSFC
C
C     CHECK TO SEE IF NUSFC IS EMPTY
      READ(NUSFC,END=50) DUMMY
      STOP 'SURFACE FILE ALREADY EXISTS'
   50 REWIND NUSFC
C
C     DEFINE GRID VECTORS AS IN 63/PAGES 24-26
C
      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(2)
      R1HAT(2) = R1(1)
```

```
      R2HAT(1) = -R2(2)
      R2HAT(2) = R2(1)
      R1RAT = -2.0*EPS/DELTA
C
C     DEFINE THE HEXAGONAL SURFACE GRID NODE LOCATIONS
C
      FMISC(16) = DELTA
      FMISC(17) = EPS
      CALL TRIADS(NHEX)
C
C     WRITE THE HEADER RECORDS
C
      WRITE(NUSFC) IGENSF,NHEX,NNODE,WNDSPD,DSEED
      WRITE(NUSFC) R1,R2,R1HAT,R2HAT,R1RAT,FNODE
C
C     DEFINE THE STANDARD DEVIATION FOR SURFACE HEIGHTS BY 63/2.12
C
      SIGSFC = 0.0397*SQRT(WNDSPD)
      WRITE(6,302) DELTA,EPS,SIGSFC
C
C     GENERATE AND SAVE THE CAPLIIARY WAVE SURFACE REALIZATIONS,
C     63/SECTION 2C
C
      DO 55 NSFC=1,IGENSF
C
C     DRAW N(0,1) RANDOM NUMBERS
      CALL GGNML(DSEED,NNODE,ZNODE)
C
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,(ZNODE(I),I=1,NNODE)
C
      ENDFILE NUSFC
      WRITE(6,60) IGENSF
      STOP
      ENDIF
C
C*****THIS IS A PRODUCTION RUN FOR RAY TRACING
C
C     READ THE EXISTING FILE OF SURFACE REALIZATIONS AND TEST FOR
C     COMPATABILITY WITH REQUESTED PARAMETERS
C
      WRITE(6,308)
      REWIND NUSFC
      READ(NUSFC) NSF1,NHEX1,NNODE,WIND1
      READ(NUSFC) R1,R2,R1HAT,R2HAT,R1RAT,FNODE
C
      IF(NHEX1.NE.NHEX .OR. WIND1.NE.WNDSPD) THEN
      WRITE(6,70) NHEX1,WIND1
      STOP
      ENDIF
C
C     DEFINE THE MU AND PHI VALUES WHICH FORM THE QUAD BOUNDARIES FOR
C     GEOMETRIC DISCRETIZATION (SECTION 3).
C
      IF(MUPART.EQ.1) THEN
C
C     PARTITION THE UNIT SPHERE SO THAT ALL QUADS HAVE EQUAL SOLID ANGLES
C
      CALL EQSANG(NMU,NPHI,DELTMU)
C
      ELSEIF(MUPART.EQ.2) THEN
C
C     PARTITION THE UNIT SPHERE INTO EQUALLY SPACED THETA VALUES
C
      CALL EQTHET(NMU,DELTMU)
C
      ENDIF
C
```

```
C     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.
C
C     DEFINE THE MU VALUES AT THE QUAD CENTERS
      FMU(1) = 0.5*DELTMU(1)
      DO 104 I=2,NMU
  104 FMU(I) = 0.5*(BNDMU(I-1) + BNDMU(I))
C
C     DEFINE THE BOUNDARY PHIS BY PHI - DPHI/2 TO PHI + DPHI/2
C
      DELPHI = 2.0*PI/FLOAT(NPHI)
      BNDPHI(1) = 0.5*DELPHI
      DO 102 J=2,NPHI
  102 BNDPHI(J) = BNDPHI(J-1) + DELPHI
C
C     DEFINE THE PHI VALUES AT THE QUAD CENTERS
C
      DO 103 J=1,NPHI
  103 PHI(J) = DELPHI*FLOAT(J-1)
C
C     DETERMINE THE SOLID ANGLE OF THE QUADS
C
      DO 400 I=1,NMU-1
  400 OMEGA(I) = DELPHI*DELTMU(I)
      OMEGA(NMU) = 2.0*PI*DELTMU(NMU)
C
      WRITE(6,310)
      DO 312 I=1,NMU
      THETAC = ACOS(FMU(I))*RADEG
      THETAB = ACOS(BNDMU(I))*RADEG
  312 WRITE(6,314) I,FMU(I),THETAC,BNDMU(I),THETAB,DELTMU(I),
     1 OMEGA(I),NRAYQD(I)
      WRITE(6,316) DELPHI*RADEG
C
C     WRITE HEADER RECORDS FOR OUTPUT FILES
C
      REWIND NUDU
      REWIND NUDD
      REWIND NUUD
      REWIND NUUU
      WRITE(NUDU) NUDU,'DOWN UP   ',NRAYQD
      WRITE(NUDU) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
      WRITE(NUDD) NUDD,'DOWN DOWN',NRAYQD
      WRITE(NUDD) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
      WRITE(NUUD) NUUD,'UP   DOWN',NRAYQD
      WRITE(NUUD) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
      WRITE(NUUU) NUUU,'UP   UP   ',NRAYQD
      WRITE(NUUU) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
C
      RETURN
C
C     FORMATS
C
   60 FORMAT(1H0,I10,' SURFACE REALIZATIONS GENERATED')
   70 FORMAT(1H0,' SURFACE REALIZATION FILE NOT COMPATABLE WITH REQUESTE
     1D PARAMETERS'//'   NHEX1 =',I3,5X,'WIND1 =',F7.3)
  300 FORMAT(1H1,' NATURAL HYDROSOL MODEL, PROGRAM 1'//
     1'  MONTE CARLO AIR-WATER SURFACE RAY TRACING'//
     2'  THE HEXAGON GRID PARAMETERS FOR THIS RUN ARE'//
     35X,'NHEX   =',I2,' = ORDER OF THE WAVE FACET HEXAGON'//
     45X,'WNDSPD =',F7.3,' = THE WIND SPEED IN M/SEC AT 12.5 M'//
     55X,'DSEED  =',1PD20.10,' = THE SEED FOR RANDOM NUMBER GENERATION')
  301 FORMAT(1H0,' 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)'//
     35X,'NREADO =',I2,' = PARAMETER FOR READING THE SFC. REAL. FILE'//
     45X,'NUMRAY =',I10,' = THE TOTAL NUMBER OF INPUT RAYS TO BE TRACED'
     5)
  302 FORMAT(1H0,'  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 CAPIL
     1LARY WAVE SURFACE REALIZATIONS')
  308 FORMAT(1HO,' THIS IS A PRODUCTION RUN FOR RAY TRACING')
  310 FORMAT(1HO,' THE MU AND THETA VALUES DEFINING THE QUADS ARE'//
     15X,'I   CNT MU   THETA',8X,'BND MU   THETA',7X,
     2'DELTA MU  SOLID ANGLE   NRAYQD'/)
  314 FORMAT(1H ,I5,2(F9.4,F9.3,4X),F9.4,F12.4,I10)
  316 FORMAT(1HO,' THE QUADS HAVE A WIDTH OF DELTA PHI =',F7.3,
     1' DEGREES')
      END
```

```
      SUBROUTINE EQSANG(NMU,NPHI,DELTMU)
C
C     ON NHM1/EQSANG
C
C     THIS ROUTINE PARTITIONS THE UNIT SPHERE INTO MU BANDS WHICH HAVE
C     EQUAL SOLID ANGLES FOR ALL QUADS, INCLUDING THE POLAR CAP, AS
C     ON PAGE 22.
C
      DIMENSION DELTMU(NMU)
C
      WRITE(6,200)
C
      DMU = FLOAT(NPHI)/FLOAT(NMU*NPHI - NPHI + 1)
      DO 100 I=1,NMU-1
  100 DELTMU(I) = DMU
      DELTMU(NMU) = DMU/FLOAT(NPHI)
      RETURN
C
  200 FORMAT(1HO,' THE UNIT SPHERE IS PARTITIONED SO THAT ALL QUADS HAVE
     1 EQUAL SOLID ANGLES')
      END
```

```
      SUBROUTINE EQTHET(NMU,DELTMU)
C
C     ON NHM1/EQTHET
C
C     THIS ROUTINE PARTITIONS THE UNIT SPHERE INTO MU BANDS WHICH HAVE
C     EQUAL DELTA THETA SPACINGS, PLUS A POLAR CAP OF HALF-ANGLE DTHETA/2,
C     AS ON PAGE 24.
C
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DIMENSION DELTMU(NMU)
C
      WRITE(6,200)
      PI2 = 0.5*FMISC(1)
C
      DTHETA = PI2/(FLOAT(NMU) - 0.5)
      DO 100 I=1,NMU-1
  100 DELTMU(I) = COS(PI2-FLOAT(I)*DTHETA) - COS(PI2-FLOAT(I-1)*DTHETA)
      DELTMU(NMU) = 1.0 - COS(PI2 - FLOAT(NMU-1)*DTHETA)
      RETURN
C
  200 FORMAT(1HO,' THE UNIT SPHERE IS PARTITIONED INTO MU BANDS WHICH HA
     1VE EQUAL DELTA THETA SPACING')
      END
```

```
      SUBROUTINE FINTCP(INRAY,A,B,C,PIN,XIIN,SIM1,SI, INTCP,P,UON)
C
C     ON NHM1/FINTCP
C
C     THIS ROUTINE DETERMINES IF THE TRACK INTERCEPTS A PARTICULAR FACET.
C
C     INPUT IS
C     INRAY = 1 FOR AN INITIAL RAY, = 0 FOR A DAUGHTER RAY
C     A, B, C...THE 2-D TRIAD NODE LOCATIONS
C     PIN...THE INITIAL POINT OF THE CURRENT TRACK
C     XIIN...THE DIRECTION OF THE CURRENT TRACK
C     SIM1 AND SI...THE DISTANCES S(I-1) AND S(I) ALONG THE TRACK,
C                   MEASURED FROM PIN. SIM1.LT.SI BY CONSTRUCTION.
C
C     OUTPUT IS
C     INTCP = 0 IF THERE IS NO INTERCEPT
C             1 IF THE TRACK DOES INTERCEPT THE FACET
C     UON = THE UNIT OUTWARD NORMAL TO THE FACET
C     P = THE 3-D FACET INTERCEPT POINT, IF INTCP = 1
C
      PARAMETER(MXNHEX=7, MXNODE=3*MXNHEX*(MXNHEX+1)+1)
      DIMENSION A(2),B(2),C(2),PIN(3),XIIN(3),P(3),UON(3)
      COMMON/CNODES/ NNODE,FNODE(2,MXNODE),ZNODE(MXNODE)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      XIDXIH = FMISC(20)
C
C     GET THE NODES ASSOCIATED WITH A, B AND C
C
      CALL GETNOD(A, NA)
      CALL GETNOD(B, NB)
      CALL GETNOD(C, NC)
C
      IF((NA.EQ.NB) .OR. (NA.EQ.NC) .OR. (NB.EQ.NC)) THEN
      WRITE(6,300) NA,A,NB,B,NC,C
      STOP
      ENDIF
C
C     DEFINE THE FACET VERTICES BY 63/3.15
C
      V11 = A(1)
      V12 = A(2)
      V13 = ZNODE(NA)
      V21 = B(1)
      V22 = B(2)
      V23 = ZNODE(NB)
      V31 = C(1)
      V32 = C(2)
      V33 = ZNODE(NC)
C
C     GET THE UNIT OUTWARD NORMAL, 63/PAGE 42
C
      UON1 = (V32 - V12)*(V23 - V13) - (V33 - V13)*(V22 - V12)
      UON2 = (V33 - V13)*(V21 - V11) - (V31 - V11)*(V23 - V13)
      UON3 = (V31 - V11)*(V22 - V12) - (V32 - V12)*(V21 - V11)
      SGN = SIGN(1.0,UON3)/SQRT(UON1*UON1 + UON2*UON2 + UON3*UON3)
      UON1 = SGN*UON1
      UON2 = SGN*UON2
      UON3 = SGN*UON3
      UON(1) = UON1
      UON(2) = UON2
      UON(3) = UON3
C
C     GET S(Q), 63/PAGE 43
C
      SQ = ((V11-PIN(1))*UON1 + (V12-PIN(2))*UON2 + (V13-PIN(3))*UON3)/
     1 (XIIN(1)*UON1 + XIIN(2)*UON2 + XIIN(3)*UON3)
C
C     CHECK FOR FACET INTERCEPT BY 63/3.16
C
      IF(SIM1.LT.SQ*XIDXIH .AND. SQ*XIDXIH.LE.SI) THEN
C
C     HAVE A FACET INTERCEPT
C
C     CHECK INITIAL RAYS TO SEE IF THE RAY IS COMING IN 'UNDER THE GRID'
C
```

```
      IOK = 1
      IF(INRAY.EQ.1) THEN
      XPDOTN = XIIN(1)*UON(1) + XIIN(2)*UON(2) + XIIN(3)*UON(3)
      IF(XIIN(3).LE.0.0 .AND. XPDOTN.GT.0.0) IOK = 0
      IF(XIIN(3).GT.0.0 .AND. XPDOTN.LT.0.0) IOK = 0
      ENDIF
C
      IF(IOK.EQ.1) THEN
C     FACET 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
C     RAY IS UNDER THE GRID, LET IT PASS THROUGH THE SURFACE UNDETECTED
      INTCP = 0
      ENDIF
C
      ELSE
      INTCP = 0
      ENDIF
C
      RETURN
C
  300 FORMAT(1HO,' SUB FINTCP: ILL-DEFINED FACET'//
     110X,'NODE, A =',I5,1P2E12.3//10X,'NODE, B =',I5,2E12.3//
     210X,'NODE, C =',I5,2E12.3)
      END
```

```
      SUBROUTINE GETABC(YTIP1,YTIP2,KS1,KS2, A,B,C)
C
C     ON NHM1/GETABC
C
C     GIVEN TWO TRIAD INTERCEPT POINTS, YTIP1 AND YTIP2, AND THEIR K
C     VALUES, KS1 AND KS2, THIS ROUTINE RETURNS THE TRIAD VERTICES A, B AND C
C
C     NOTATION USED:  YJK = YJ(K)
C
      DIMENSION YTIP1(2),YTIP2(2),A(2),B(2),C(2)
      COMMON/CHEXGR/ NHEX,R11,R12,R21,R22,RKHAT(4),R1RAT
      COMMON/CMISC/ IMISC(20),FMISC(20)
C
      DELTA = FMISC(16)
      EPS = FMISC(17)
C
      IF(KS1+KS2.EQ.3) THEN
C
C     HAVE CASE R1-R2 (SEE 63/FIGURE 7), USE 63/3.8-3.12
C
      IF(KS1.EQ.1) THEN
C
C     Y1 IS YTIP1, Y2 IS YTIP2
      Y11 = YTIP1(1)
      Y12 = YTIP1(2)
      Y21 = YTIP2(1)
      Y22 = YTIP2(2)
      ELSE
C
C     Y1 IS YTIP2, Y2 IS YTIP1
      Y11 = YTIP2(1)
      Y12 = YTIP2(2)
      Y21 = YTIP1(1)
      Y22 = YTIP1(2)
      ENDIF
```

```
C
      D1 = Y11*R1RAT + Y12
      D2 = -Y21*R1RAT + Y22
C
      A1 = 0.25*(D2-D1)*DELTA/EPS
      A2 = 0.5*(D1+D2)
      A(1) = A1
      A(2) = A2
C
      SGN1 = SIGN(1.0, (Y11-A1)*R11 + (Y12-A2)*R12)
      SGN2 = SIGN(1.0, (Y21-A1)*R21 + (Y22-A2)*R22)
C
      B(1) = A1 + SGN1*DELTA*0.5
      B(2) = A2 + SGN1*EPS
      C(1) = A1 - SGN2*DELTA*0.5
      C(2) = A2 + SGN2*EPS
C
      ELSEIF(KS1+KS2.EQ.1) THEN
C
C     HAVE CASE I-R1 (SEE 63/FIGURE 8), USE 63/3.13
C
      IF(KS1.EQ.0) THEN
C
C     Y0 IS YTIP1, Y1 IS YTIP2
      Y01 = YTIP1(1)
      Y02 = YTIP1(2)
      Y11 = YTIP2(1)
      Y12 = YTIP2(2)
      ELSE
C
C     Y0 IS YTIP2, Y1 IS YTIP1
      Y01 = YTIP2(1)
      Y02 = YTIP2(2)
      Y11 = YTIP1(1)
      Y12 = YTIP1(2)
      ENDIF
C
      A1 = 0.5*(Y02 - Y11*R1RAT - Y12)*DELTA/EPS
      A(1) = A1
      A(2) = Y02
C
      SGN1 = SIGN(1.0, (Y11-A1)*R11 + (Y12-Y02)*R12)
C
      B(1) = A1 + SGN1*DELTA*0.5
      B(2) = Y02 + SGN1*EPS
      C(1) = A1 + SIGN(1.0, Y01 - A1)*DELTA
      C(2) = Y02
C
      ELSEIF(KS1+KS2.EQ.2) THEN
C
C     HAVE CASE I-R2 (SEE 63/FIGURE 8), USE 63/3.14
C
      IF(KS1.EQ.0) THEN
C
C     Y0 IS YTIP1, Y2 IS YTIP2
      Y01 = YTIP1(1)
      Y02 = YTIP1(2)
      Y21 = YTIP2(1)
      Y22 = YTIP2(2)
      ELSE
C
C     Y0 IS YTIP2, Y2 IF YTIP1
      Y01 = YTIP2(1)
      Y02 = YTIP2(2)
      Y21 = YTIP1(1)
      Y22 = YTIP1(2)
      ENDIF
C
      A1 = 0.5*(-Y21*R1RAT + Y22 - Y02)*DELTA/EPS
      A(1) = A1
      A(2) = Y02
C
      SGN2 = SIGN(1.0, (Y21-A1)*R21 + (Y22-Y02)*R22)
C
```

```
      B(1) = A1 - SGN2*DELTA*0.5
      B(2) = Y02 + SGN2*EPS
      C(1) = A1 + SIGN(1.0, Y01 - A1)*DELTA
      C(2) = Y02
C
      ELSE
C
C     ERROR IN INPUT
      WRITE(6,100) YTIP1,YTIP2,KS1,KS2
      STOP
      ENDIF
      RETURN
C
  100 FORMAT(1H0,' ERROR IN SUB GETABC'//1H ,' YTIP1 =',1P2E12.3,4X,
     1'YTIP2 =',2E12.3,4X,'K(1), K(2) =',2I3)
      END
```

```
      SUBROUTINE GETNOD(A, NODE)
C
C     ON NHM1/GETNOD
C
C     GIVEN A VECTOR A, WHICH LOCATES ANY POINT IN THE HEXAGON, THIS
C     ROUTINE RETURNS THE INDEX, NODE, OF THE NEAREST TRIAD NODE.
C
      PARAMETER(MXNHEX=7, MXNODE=3*MXNHEX*(MXNHEX+1)+1)
      COMMON/CHEXGR/ NHEX
      COMMON/CNODES/ NNODES,FNODE(2,MXNODE)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DIMENSION A(2)
      DELTA = FMISC(16)
      EPS = FMISC(17)
C
C     CHECK Y VALUES OF THE LEFT HEXAGON BOUNDARY POINTS
C
      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).GT.AY) GO TO 100
  200 CONTINUE
      DO 202 J=NHEX,2,-1
      K = K + NHEX + J
      IF(FNODE(2,K).GT.AY) GO TO 100
  202 CONTINUE
C
C     NOW CHECK X VALUES ALONG CONSTANT Y ROW
C
  100 AX = A(1) - 0.5*DELTA
      DO 204 J=K,NNODES
      IF(FNODE(1,J).GT.AX) GO TO 102
  204 CONTINUE
C
      WRITE(6,206) A
      STOP
C
  102 NODE = J
      RETURN
C
  206 FORMAT(1H0,' SUB GETNOD: POINT A = (',1PE12.3,',',E12.3,
     1') NOT WITHIN HEXAGON')
      END
```

```
      SUBROUTINE MPINDX(FMU,PHI,I,J)
C
C     ON NHM1/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.BNDMU(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
      JB = 1
  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
C     IDFMT = 1 FOR 10F12.4
C             2 FOR 1P10E12.4
C             3 FOR 10I12
C             4 FOR 12A10
C             5 FOR 1P6E20.8
C             6 FOR 20F5.1
C             7 FOR 1P5E25.15
C
C     THE ARRAY IS PARTITIONED BY COLUMNS INTO BLOCKS, AND ROWS AND
C     COLUMNS ARE NUMBERED.  THE CHARACTER STRING TITLE CONTAINS ANY
C     DESIRED TITLE (UP TO 130 CHARACTERS FOR PRINTER OUTPUT).
C
      DIMENSION A(IDIM,NC)
      CHARACTER TITLE*(*)
```

```
C
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
C     PARTITION ARRAY
C
      KMANY = (((NC-1)/KSIZE) + 1)*KSIZE
      IBLOCK = 0
      NBLOCK = 1
      IF(NR.LE.25) NBLOCK = 60/(NR+4)
C
C     PRINT ARRAY
C
      DO 210 L=KSIZE,KMANY,KSIZE
      L1 = L - (KSIZE - 1)
      L2 = L
      IF(L.GE.KMANY) L2 = NC
C
      DO 210 I=1,NR
      IF(MOD(I-1,50).NE.0) GO TO 210
C
      IF(IBLOCK.EQ.0 .OR.IBLOCK.GE.NBLOCK) THEN
C
C     PRINT TITLE AND COLUMN HEADINGS IF NEW PAGE
      IBLOCK = 1
      WRITE(6,110) TITLE
      WRITE(6,IFMT1) (K,K=L1,L2)
      ELSE
C
C     PRINT COLUMN HEADINGS FOR A NEW BLOCK
      IBLOCK = IBLOCK + 1
      WRITE(6,IFMT1) (K,K=L1,L2)
      ENDIF
C
C     WRITE A LINE OF DATA
  210 WRITE(6,IFMT2) I,(A(I,J),J=L1,L2)
C
      RETURN
C
C     FORMATS
C
C     IFMT1 FOR COLUMN LABELS
  905 FORMAT(1H //10X,5I25)
  906 FORMAT(1H //10X,6I20)
  910 FORMAT(1H //10X,10I12)
  912 FORMAT(1H //10X,12I10)
  920 FORMAT(1H //10X,20I5)
```

```
C
C       IFMT2 FOR DATA
   11 FORMAT(1H ,I9,10F12.4)
   21 FORMAT(1H ,I9,1P10E12.3)
   31 FORMAT(1H ,I9,10I12)
   41 FORMAT(1H ,I9,12A10)
   51 FORMAT(1H ,I9,1P6E20.8)
   61 FORMAT(1H ,I9,20F5.1)
   71 FORMAT(1H ,I9,1P5E25.15)
C
  100 FORMAT(1H0,'INVALID FORMAT OPTION IN P2ARAY, IDFMT =',I5)
  110 FORMAT(1H1,A)
      END
```

```
      SUBROUTINE P3ARAY(A,NR,NC,NP,IDIM,JDIM,IDFMT,TITLE)
C
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
C     IDFMT = 1 FOR 10F12.4
C             2 FOR 1P10E12.4
C             3 FOR 10I12
C             4 FOR 12A10
C             5 FOR 1P6E20.8
C             6 FOR 20F5.1
C             7 FOR 1P5E25.15
C
C     THE ARRAY IS PRINTED BY PLANES.  FOR EACH PLANE
C     THE ARRAY IS PARTITIONED BY COLUMNS INTO BLOCKS, AND ROWS AND
C     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
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
C     PARTITION ARRAY
C
      KMANY = (((NC-1)/KSIZE) + 1)*KSIZE
      IBLOCK = 0
      NBLOCK = 1
      IF(NR.LE.25) NBLOCK = 60/(NR+4)
C
C     PRINT ARRAY
C
      DO 210 IP=1,NP
C
      DO 210 L=KSIZE,KMANY,KSIZE
      L1 = L - (KSIZE - 1)
      L2 = L
      IF(L.GE.KMANY) L2 = NC
C
      DO 210 I=1,NR
      IF(MOD(I-1,50).NE.0) GO TO 210
C
      IF(IBLOCK.EQ.0 .OR.IBLOCK.GE.NBLOCK) THEN
C
C     PRINT TITLE AND COLUMN HEADINGS IF NEW PAGE
      IBLOCK = 1
      WRITE(6,110) TITLE,IP
      WRITE(6,IFMT1) (K,K=L1,L2)
      ELSE
C
C     PRINT COLUMN HEADINGS FOR A NEW BLOCK
      IBLOCK = IBLOCK + 1
      WRITE(6,IFMT1) (K,K=L1,L2)
      ENDIF
C
C     WRITE A LINE OF DATA
  210 WRITE(6,IFMT2) I,(A(I,J,IP),J=L1,L2)
C
      RETURN
C
C     FORMATS
C
C     IFMT1 FOR COLUMN LABELS
  905 FORMAT(1H //10X,5I25)
  906 FORMAT(1H //10X,6I20)
  910 FORMAT(1H //10X,10I12)
  912 FORMAT(1H //10X,12I10)
  920 FORMAT(1H //10X,20I5)
C
C     IFMT2 FOR DATA
   11 FORMAT(1H ,I9,10F12.4)
   21 FORMAT(1H ,I9,1P10E12.3)
   31 FORMAT(1H ,I9,10I12)
   41 FORMAT(1H ,I9,12A10)
   51 FORMAT(1H ,I9,1P6E20.8)
   61 FORMAT(1H ,I9,20F5.1)
   71 FORMAT(1H ,I9,1P5E25.15)
C
  100 FORMAT(1H0,'INVALID FORMAT OPTION IN P2ARAY, IDFMT =',I5)
  110 FORMAT(1H1,A//' THREE-DIMENSIONAL ARRAY, PLANE (THIRD INDEX)',I3)
      END
```

```
      SUBROUTINE PULL(R,P,XI)
C
C     ON NHM1/PULL
C
C     THE ROUTINE PULLS R, P AND XI OFF OF 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
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
C     ON NHM1/PUSH
C
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
C     TEST FOR OVERFLOW OF STACK
C
      IF(NSTACK.GE.MXSTAK) THEN
      WRITE(6,100) NSTACK
      RETURN
      ENDIF
C
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(1H0,' STACK FULL, NSTACK =',I6,' RAY DISCARDED')
      END
```

```
      FUNCTION REFLF(THETAP,THETA)
C
C     ON NHM1/REFLF
C
C     THIS FUNCTION RETURNS THE REFLECTANCE, GIVEN THE REFLECTED AND
C     REFRACTED ANGLES, THETA PRIME AND THETA
C
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DATA EPSO/1.0E-5/
C
      PI = FMISC(1)
      REFR = FMISC(18)
      TPMT = THETAP - THETA
      TPPT = THETAP + THETA
C
C     CHECK FOR NORMAL INCIDENCE
C
      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
C     63/3.20
      REFLF = 0.5*((SIN(TPMT)/SIN(TPPT))**2 + (TAN(TPMT)/TAN(TPPT))**2)
      ELSE
C
C     USE LIMITING CASE FOR NORMAL INCIDENCE
      REFLF = ((REFR - 1.0)/(REFR + 1.0))**2
      ENDIF
C
      RETURN
      END
```

```
      SUBROUTINE RSPLIT(RIN,XIIN,UON, RREFL,XIREFL,RREFR,XIREFR)
C
C     NHM1/RSPLIT
C
C     THIS ROUTINE DETERMINES THE REFLECTED AND REFRACTED DIRECTIONS
C     AND THE ASSOCIATED RADIANCES AT THE INTERCEPTED FACET.
C
C     FACTORS OF REFR**2 AND 1/REFR**2 ARE NOT INCLUDED IN THE
C     TRANSMITTED RADIANCES
C
C     INPUT
C         RIN...THE RADIANCE OF THE INCOMING RAY
C         XIIN...THE DIRECTION OF THE INCOMING RAY
C         UON...THE UNIT OUTWARD NORMAL OF THE INTERSECTED FACET
C
C     OUTPUT
C         RREFL....THE RADIANCE OF THE REFLECTED RAY
C         XIREFL...THE DIRECTION "   "       "       "
C         RREFR....THE RADIANCE  "   "  REFRACTED   "
C         XIREFR...THE DIRECTION "   "       "       "
C
      DIMENSION XIIN(3),UON(3),XIREFL(3),XIREFR(3)
      COMMON/CMISC/ IMISC(20),FMISC(20)
```

```
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
C     AIR-INCIDENT CASE
C
C     REFLECTED AND REFRACTED DIRECTION BY 63/3.18
C
      C = XPDOTN + SQRT(XPDOTN*XPDOTN + REFR*REFR-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
C     ANGLES BY 63/3.18
C
      THETAP = ACOS(ABS(XPDOTN))
      THETA = ASIN(SIN(THETAP)/REFR)
      R = REFLF(THETAP,THETA)
C
C     COMPUTE RADIANCES BY 63/3.30 AND 3.31A
      RREFL = RIN*R
      RREFR = RIN*(1.0-R)
C
      ELSE
C
C     WATER-INCIDENT CASE
C
C     REFLECTED AND REFRACTED DIRECTIONS BY 63/3.19
C
      ARG = (REFR*XPDOTN)**2 - REFR*REFR + 1.0
      IF(ARG.GE.0.0) THEN
      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*XIIN(J) - C*UON(J)
C
C     ANGLES BY 63/3.19
C
      THETAP = ACOS(ABS(XPDOTN))
C
C     COMPUTE THE REFLECTANCE
C
      IF(THETAP.GT.RAD48) THEN
C
C     HAVE TOTAL INTERNAL REFLECTION
      R = 1.0
      ELSE
C
C     REFLECTION AND REFRACTION
      THETA = ASIN(REFR*SIN(THETAP))
      R = REFLF(THETAP,THETA)
      ENDIF
C
C     RADIANCES BY 63/3.30 AND 3.31B
C
      RREFL =RIN*R
      RREFR = RIN*(1.0 - R)
C
      ENDIF
C
      RETURN
      END
```

```
      SUBROUTINE TIP(P,XI,IALL)
C
C     ON NHM1/TIP
C
C     GIVEN A POINT P AND A DIRECTION XI, THIS ROUTINE FIRST COMPUTES
C     THE TRACK OF THE RAY P + S*XI AS IN 63/SECTION 3B.
C     IF IALL = 0, THE COMPUTATIONS ARE CARRIED ONLY TO 63/3.6, AND
C                  SMIN IS RETURNED.
C     IF IALL = 1, THE TRIAD INTERCEPT POINTS YTIP ARE ALSO COMPUTED.
C
      PARAMETER(MXNHEX=7, MXNTIP=4*MXNHEX+1)
      DIMENSION P(3),XI(3)
      DIMENSION S0(-MXNHEX:MXNHEX),S1(-MXNHEX:MXNHEX),S2(-MXNHEX:MXNHEX)
      DIMENSION IR(MXNTIP),WORK(MXNTIP),KWORK(MXNTIP)
      COMMON/CHEXGR/ NHEX,R1(2),R2(2),R1HAT1,R1HAT2,R2HAT1,R2HAT2
      COMMON/CTIP/ NTIP,SMIN,YTIP(2,MXNTIP),S(MXNTIP),KS(MXNTIP)
      COMMON/CMISC/ IMISC(20),FMISC(20)
C
      DATA EPSDOT/1.0E-8/
C
      EPS = FMISC(17)
C
C     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
      S0(L) = 1.E30
      S1(L) = 1.E30
  100 S2(L) = 1.E30
C
C     COMPUTE S0 VALUES BY 63/3.5
C
      IF(ABS(XIH2).GT.EPSDOT) THEN
      SS = -P2/XIH2
      A = EPS/XIH2
      DO 110 L=-NHEX,NHEX
  110 S0(L) = SS + FLOAT(L)*A
      ENDIF
C
C     COMPUTE S1 BY 63/3.5
C
      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
C
C     COMPUTE S2 BY 63/3.5
C
      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
      ENDIF
C
C     FIND THE MINIMUM POSITIVE END POINT BY 63/3.6
C
      SMIN = AMIN1(AMAX1(S0(-NHEX),S0(NHEX)), AMAX1(S1(-NHEX),S1(NHEX)),
     1             AMAX1(S2(-NHEX),S2(NHEX)))
      IF(IALL.EQ.0) RETURN
C
C     SELECT THE NON-NEGATIVE S VALUES .LE. SMIN
C
```

```
      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
  210 IR(I) = I
      CALL VSRTR(S,NS,IR)
C
C     CORRESPONDINGLY PERMUTE THE ASSOCIATED K VALUES
C
      DO 212 I=1,NS
  212 KWORK(I) = KS(I)
      DO 214 I=1,NS
  214 KS(I) = KWORK(IR(I))
C
C     CHECK THE SORTED S VALUES FOR EQUAL ENTRIES.  DISCARD DEGENERATE
C     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(1).LT.0.0) S(1) = 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)
C
      ELSE
C     MULTIPLE S VALUES FOUND, RELABEL KS
      IF(ABS(XIH1).GT.EPSDOT) THEN
      KS(NTIP) = 0
      ELSE
      KS(NTIP) = 1
      ENDIF
      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)*XIH1
  300 YTIP(2,I) = P2 + S(I)*XIH2
C
      RETURN
C
      END
```

```
      SUBROUTINE TRACE(INRAY,RIN,PIN,XIIN, IOUT,P,RREFL,XIREFL,RREFR,
     1  XIREFR)
C
C     ON NHM1/TRACE
C
C     GIVEN AN INITIAL RADIANCE, RIN, STARTING POINT, PIN, AND
C     DIRECTION, XIIN, THIS ROUTINE TRACES THE RAY UNTIL IT EITHER
C     LEAVES THE HEXAGON REGION OR INTERCEPTS A FACET.
C
C     IF THE RAY LEAVES THE HEXAGON BEFORE INTERSECTING A FACET,
C         IOUT = 1 AND RETURN IS MADE.
C     IF THE RAY INTERCEPTS A FACET BEFORE LEAVING THE HEXAGON,
C         IOUT = 0 AND
C         P = THE INTERCEPT POINT
C         RREFL = THE REFLECTED RADIANCE
C         XIREFL = THE DIRECTION OF THE REFLECTED RAY
C         RREFR = THE REFRACTED RADIANCE
C         XIREFR = THE DIRECTION OF THE REFRACTED RAY
C     ARE RETURNED.
C
      PARAMETER(MXNHEX=7, MXNTIP=4*MXNHEX+1)
      COMMON/CTIP/ NTIP,SMIN,YTIP(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)
C
C     COMPUTE THE TRIAD INTERCEPT POINTS FOR THE RAY
C
      CALL TIP(PIN,XIIN,1)
C
C     LOCATE THE TRIAD INDICES FOR WHICH AN INTERCEPT IS POSSIBLE
C
      IF(XIIN(3).LE.0.0) THEN
C
C     DOWNWARD RAY
      TANTHP = TAN(ACOS(-XIIN(3)))
C
C     GET FIRST FACET TO BE CHECKED
      IF(PIN(3).GT.ZMAX) THEN
C
C     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
   55 I1 = I
C
      ELSE
C
C     INITIAL POINT BELOW THE MAXIMUM SURFACE (DAUGHTER RAY OR LOW-ANGLE
C     INITIAL RAY)
      I1 = 2
      ENDIF
C
C     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 = I
C
      ELSE
C
C     UPWARD RAY
      TANTHP = TAN(ACOS(XIIN(3)))
C     GET FIRST FACET
      IF(PIN(3).LT.ZMIN) THEN
C
C     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
   75 I1 = I
```

```
C
      ELSE
C
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=I1,NTIP
      IF(S(I).GE.D2) GO TO 85
   80 CONTINUE
   85 I2 = I
      ENDIF
      I2 = MIN0(I2,NTIP)
C
C     CHECK POSSIBLE PAIRS OF TRIAD INTERCEPT POINTS FOR A FACET INTERCEPT
C
      DO 100 I=I1,I2
C
C     GET THE TRIAD NODE VECTORS CORRESPONDING TO INTERCEPT POINTS I AND I-1
C
      CALL GETABC(YTIP(1,I-1),YTIP(1,I),KS(I-1),KS(I), A,B,C)
C
C     SEE IF THE RAY TRACK INTERCEPTS THIS FACET
C
      CALL FINTCP(INRAY,A,B,C,PIN,XIIN,S(I-1),S(I), INTCP,P,UON)
C
      IF(INTCP.EQ.1) GO TO 200
C
  100 CONTINUE
C
C     IF HERE, NO INTERCEPT WAS FOUND
      IOUT = 1
      RETURN
C
C     IF HERE, AN INTERCEPT WAS FOUND.  COMPUTE THE REFLECTED AND
C     REFRACTED RAYS
C
  200 IOUT = 0
      CALL RSPLIT(RIN,XIIN,UON, RREFL,XIREFL,RREFR,XIREFR)
C
      RETURN
C
      END
```

```
      SUBROUTINE TRIADS(NHEX)
C
C     ON NHM1/TRIADS
C
C     GIVEN THE ORDER OF THE HEXAGON, NHEX, THIS ROUTINE DEFINES THE
C     VECTOR NODES, FNODE, OF THE HEXAGON TRIADS IN UNITS OF DELTA AND
C     EPSILON, AS IN 63/PAGE 26.
C
      PARAMETER(MXNHEX=7, MXNODE=3*MXNHEX*(MXNHEX+1)+1)
      COMMON/CNODES/ NNODE,FNODE(2,MXNODE)
      COMMON/CMISC/ IMISC(20),FMISC(20)
C
      DELTA = 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     C IS EVEN
C
      MXB = NHEX - IABS(IC)/2
      DO 200 IB=-MXB,MXB
      NF = NF + 1
      FNODE(1,NF) = FLOAT(IB)*DELTA
  200 FNODE(2,NF) = CEPS
C
      ELSE
C
C     C IS ODD
C
      MXB = NHEX - (IABS(IC)+1)/2
      DO 210 IB=-MXB,0
      NF = NF + 1
      FNODE(1,NF) = (-0.5+FLOAT(IB))*DELTA
  210 FNODE(2,NF) = CEPS
      DO 212 IB=0,MXB
      NF = NF + 1
      FNODE(1,NF) = (0.5+FLOAT(IB))*DELTA
  212 FNODE(2,NF) = CEPS
      ENDIF
C
  100 CONTINUE
C
      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
C
      IF(IPRINT.EQ.1) THEN
      WRITE(6,304)
      DO 306 I=1,NNODE,5
  306 WRITE(6,308) I,I+4,(FNODE(1,I+J),FNODE(2,I+J),J=0,4)
      ENDIF
C
      RETURN
C
  300 FORMAT(1HO,'  SUB TRIADS:',I3,' NODES DEFINED FOR AN ORDER',
     1I2,' HEXAGONAL GRID')
  302 FORMAT(1HO,' ERROR IN SUB TRIADS'//1H ,' NHEX =',I2,4X,'NF =',
     1I4,4X,'NNODE =',I4)
  304 FORMAT(1HO,' THE HEXAGON GRID NODES ARE LOCATED AT'/)
  308 FORMAT(1H ,' NODES',I3,'-',I3,' AT',5(' (',F7.2,',',F7.2,')'))
      END
```

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE16,TAPE17,
     1             TAPE18,TAPE19,TAPE15)
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                +
C     +  THIS IS PROGRAM 1 OF THE NATURAL HYDROSOL MODEL +
C     +                                                +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     ON NHM1/M11QD    FTN5/FTN200
C
C     THIS PROGRAM BEGINS COMPUTATION OF THE QUAD-AVERAGED GEOMETRIC
C     REFLECTANCE AND TRANSMITTANCE ARRAYS WHICH DESCRIBE THE AIR-WATER
C     INTERFACE FOR A GIVEN WIND SPEED.
C
C     THIS SPECIAL VERSION OF MAIN1 DOES ONLY ONE INPUT QUAD (ONE
C     ROW OF R OR T).
C
C     NOTE:  THIS VERSION OF THE CODE STRIVES TO MINIMIZE THE EXECUTION
C     TIME, AT THE EXPENSE OF MODULARITY AND READABILITY OF THE CODE.
C     SOME SECTIONS OF FREQUENTLY EXECUTED CODE ARE WRITTEN AS STRAIGHT
C     LINE CODE WITH SIMPLE VARIABLES, RATHER THAN BEING GROUPED IN
C     SUBROUTINES OR DO LOOPS WITH ARRAYS, IN ORDER TO AVOID CALLING
C     AND INDEXING OVERHEAD.  ALMOST ALL ERROR CHECKING AND INTERMEDIATE
C     OUTPUT HAS BEEN REMOVED.
C
C     THIS PROGRAM USES THE MONTE CARLO RAY TRACING TECHNIQUE
C     DESCRIBED IN NOAA TECH MEMO ERL-PMEL-63.  COMMENTS REFERRING TO
C     THIS REPORT ARE PREFACED BY 63/.  THUS 63/2.12 REFERS TO
C     EQUATION 2.12 IN TECH MEMO 63.
C     REFERENCES WITHOUT THE 63/ REFER TO NOAA TECH MEMO ERL-PMEL-75.
C
C     NUSFC = TAPE15...CONTAINS THE RANDOM SURFACE REALIZATIONS
C
C     RESULTS OF COMPLETED RAY PATHS ARE WRITTEN TO FILES AS FOLLOWS:
C
C     NUDU = TAPE16...INITIAL RAY DOWNWARD, FINAL RAY   UPWARD: R- = R(A,X)
C     NUDD = TAPE17...INITIAL RAY DOWNWARD, FINAL RAY DOWNWARD: T- = T(A,X)
C     NUUD = TAPE18...INITIAL RAY   UPWARD, FINAL RAY DOWNWARD: R+ = R(X,A)
C     NUUU = TAPE19...INITIAL RAY   UPWARD, FINAL RAY   UPWARD: T+ = T(X,A)
C
C     PROGRAM 2 READS THESE FILES AND TALLIES THE RESULTS TO GENERATE
C     THE ACTUAL R AND T ARRAYS.
C
      PARAMETER(MXMU=10, MXPHI=24, MXSTAK=10, MXNHEX=7)
      PARAMETER(MXNODE=3*MXNHEX*(MXNHEX+1)+1, MXNTIP=4*MXNHEX+1)
C
      COMMON/CMUPHI/ BNDMU(MXMU),BNDPHI(MXPHI)
      COMMON/CNODES/ NNODE,FNODE(2,MXNODE),ZNODE(MXNODE)
      COMMON/CHEXGR/ NHEX,R1(2),R2(2),R1HAT(2),R2HAT(2),R1RAT,TARGET(2)
      COMMON/CTIP/ NTIP,SMIN,YTIP(2,MXNTIP),S(MXNTIP),KS(MXNTIP),ZMIN,
     1  ZMAX
      COMMON/CSTACK/ NSTACK,STACK(MXSTAK,7)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DIMENSION PIN(3),XIIN(3),P(3),XIREFL(3),XIREFR(3)
      DIMENSION NRAYQD(MXMU),NBRNCH(10)
      DOUBLE PRECISION DSEED
C
      DATA RADEPS/1.0E-10/, NUSFC,NUDU,NUDD,NUUD,NUUU/15,16,17,18,19/
      DATA NSTACK/0/, KTRACE/0/, NBRNCH/10*0/, NREFLO,NREFRO,NTIR/3*0/
C
C     INITIALIZE THE PROGRAM
C
      CALL INISHL(IR,JS,NRAYQD,DSEED)
C
      NMU = IMISC(1)
      NO2PI = IMISC(2)
      PI = FMISC(1)
      TWOPI = 2.0*PI
C
C     GET MU BOUNDARIES OF THE INCOMING QUAD
      FMUMIN = 0.
      IA = IABS(IR)
      IF(IA.GT.1) FMUMIN = BNDMU(IA-1)
      DMU = BNDMU(IA) - FMUMIN
C
```

```
      IF(IA.EQ.NMU) THEN
C     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
      NUMUD = 0
      NUMUU = 0
      NUMTP1 = 0
C
C*****  BEGIN COMPUTATIONS  *****
C
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
C     SELECT A SURFACE REALIZATION
C
   55 CONTINUE
      IF(NREAD.EQ.1) THEN
C     READ A SURFACE REALIZATION AS GENERATED
      READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(I),I=1,NNODE)
C
      ELSEIF(NREAD.EQ.2) THEN
C     READ THE SURFACE AS ROTATED BY 180 DEGREES
      READ(NUSFC,END=50) NSF,ZMIN,ZMAX,(ZNODE(I),I=NNODE,1,-1)
C
      ELSEIF(NREAD.EQ.3) THEN
C     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)
C
      ELSEIF(NREAD.EQ.0) THEN
C     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
C     END OF FILE PROCESSING FOR THE STORED FILE OF CAPILLARY SURFACES
   50 WRITE(6,514) NREAD
      NREAD = NREAD + 1
      NREAD = MOD(NREAD,4)
      REWIND NUSFC
      READ(NUSFC) HEADER
      READ(NUSFC) HEADER
      GO TO 55
  506 CONTINUE
C
C     SELECT A RANDOM RAY DIRECTION WITHIN THE INPUT QUAD
C
C     CHOOSE A RANDOM MU VALUE
  777 RMU = (FMUMIN + GGUBFS(DSEED)*DMU)*SIGN(1.0,FLOAT(IR))
C     NO RAYS FROM THE POLE ITSELF
      IF(ABS(RMU).GT.1.0-RADEPS) GO TO 777
      ROOT = SQRT(1.0 - RMU*RMU)
C
C     CHOOSE A RANDOM PHI VALUE
      SPHI = AMOD(PHIMIN + GGUBFS(DSEED)*DPHI,TWOPI)
C
C     LOCATE THE INITIAL STARTING POINT FOR THIS TARGET AND DIRECTION
C
C     FOLLOW THE TRACK BACKWARDS TO THE BOUNDARY TO GET SMIN
C
```

```
C     DEFINE THE INITIAL RAY DIRECTION TO BE -XI PRIME
      XIIN(1) = ROOT*COS(SPHI)
      XIIN(2) = ROOT*SIN(SPHI)
      XIIN(3) = RMU
      CALL TIP(TARGET,XIIN,0)
C     DEFINE THE INITIAL POINT ON THE HEXAGON BOUNDARY
      TEMP = SMIN/FMISC(20)
      PIN(1) = TARGET(1) + TEMP*XIIN(1)
      PIN(2) = TARGET(2) + TEMP*XIIN(2)
      PIN(3) = TEMP*XIIN(3)
C
C     RESET XIIN TO THE INCOMING DIRECTION, XI PRIME
      XIIN(1) = -XIIN(1)
      XIIN(2) = -XIIN(2)
      XIIN(3) = -XIIN(3)
      RAD = 1.0
      INRAY = 1
C
C     PERFORM RAY TRACING COMPUTATIONS
C+++++  THIS IS THE RECURSIVE TREE FOR A GIVEN INITIAL RAY  +++++
C
      KBRNCH = 0
  999 CALL TRACE(INRAY,RAD,PIN,XIIN, IOUT,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
C     RAY HAD NO FACET INTERCEPTS.
C
C     GET THE QUAD INDICES OF THE FINAL RAY DIRECTION
      PHIFIN = AMOD(ATAN2(XIIN(2),XIIN(1))+TWOPI,TWOPI)
      AMUFIN = XIIN(3)
      CALL MPINDX(AMUFIN,PHIFIN,KU,LV)
C
C     RECORD THE RESULT FOR THE APPROPRIATE R OR T CONTRIBUTION
C
      IF(IR.GT.0) THEN
C
C     DOWNWARD INITIAL RAY
      IF(AMUFIN.GT.0.0) THEN
C     UPWARD FINAL RAY
      NUMDU = NUMDU + 1
      WRITE(NUDU) IR,JS,KU,LV,RAD
      ELSE
C     DOWNWARD FINAL RAY
      NUMDD = NUMDD + 1
      WRITE(NUDD) IR,JS,KU,LV,RAD
      ENDIF
C
      ELSE
C
C     UPWARD INTIAL RAY
      IF(AMUFIN.GT.0.0) THEN
C     UPWARD FINAL RAY
      IF(RAD.EQ.1.0) THEN
C     ERROR RAY, DUE TO FINITE HEXAGON
      NUMTP1 = NUMTP1 + 1
      ELSE
      NUMUU = NUMUU + 1
      WRITE(NUUU) -IR,JS,KU,LV,RAD
      ENDIF
C
      ELSE
C     DOWNWARD FINAL RAY
      NUMUD = NUMUD + 1
      WRITE(NUUD) -IR,JS,KU,LV,RAD
      ENDIF
      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
       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
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
       CALL PULL(RAD,PIN,XIIN)
       GO TO 999
       ENDIF
C
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
C
 1000 CONTINUE
C
C*****  END OF COMPUTATIONS  *****
C
       ENDFILE NUDU
       ENDFILE NUDD
       ENDFILE NUUD
       ENDFILE NUUU
C
       WRITE(6,600) NRAYTL,KTRACE
       WRITE(6,601) NREFLO,RADEPS,NREFRO,RADEPS,NTIR
       WRITE(6,602) NUMDU,NUMDD,NUMUD,NUMUU,NUMTP1
       WRITE(6,604) (K,K=2,10),(NBRNCH(K),K=2,10)
C
       WRITE(6,1002)
C
C      FORMATS
C
  514 FORMAT(1HO,' NREAD =',I2,3X,
     1'FILE OF SURFACE REALIZATIONS EXHAUSTED. FILE REWOUND.')
  600 FORMAT(1HO,' END OF COMPUTATIONS'//
     11H ,I10,' TOTAL RAYS WERE STARTED FROM THE SELECTED QUAD'//
     21H ,I10,' TOTAL RAYS WERE TRACED TO COMPLETION')
  601 FORMAT(1HO,I5,' REFLECTED RAYS WITH RADIANCE .LT.',1PE9.1,
     1' WERE DISCARDED'//I6,' REFRACTED RAYS WITH RADIANCE .LT.',
     2E9.1,' WERE DISCARDED'//1H ,' THERE WERE',I6,
     3' TOTAL INTERNAL REFLECTIONS')
  602 FORMAT(1HO,I10,' RAYS STARTED DOWNWARD AND FINISHED   UPWARD'//
     11H ,I10,' RAYS STARTED DOWNWARD AND FINISHED DOWNWARD'//
     21H ,I10,' RAYS STARTED   UPWARD AND FINISHED DOWNWARD'//
     31H ,I10,' RAYS STARTED   UPWARD AND FINISHED   UPWARD'//
     41H ,I10,' RAYS STARTED   UPWARD AND FINISHED   UPWARD WITH RAD = 1
     5.0 (DISCARDED)')
  604 FORMAT(1HO,' BRANCH OCCURRENCE TALLY'//'  NUM BRANCHES:    ',
     18I10,I7,' OR MORE'/'  NUM OCCURRENCES:',9I10)
 1002 FORMAT(1HO,'  NORMAL EXIT FROM NHM1')
       END
```

```
      SUBROUTINE INISHL(IR,JS,NRAYQD,DSEED)
C
C     ON NHM1/IN11QD
C
C     THIS ROUTINE INITIALIZES NHM1/M11QD
C
C     TWO INPUT RECORDS ARE READ:
C     RECORD 1 (DEFINES THE HEXAGON GRID AND THE WATER SURFACE):
C
C     IDBUG = 0 FOR MINIMAL OUTPUT
C           = 1 FOR GREATER OUTPUT
C           = 2 FOR FULL DEBUGGING OUTPUT
C     IGENSF = 0 IF A FILE OF RANDOM SURFACES ALREADY EXISTS (USUAL CASE)
C           .GT.0 IF THIS IS A SPECIAL RUN FOR GENERATING AND SAVING A
C                 FILE OF RANDOM SURFACES.  IGENSF SURFACES WILL BE GENERATED.
C     NHEX = THE ORDER OF THE HEXAGONAL SURFACE GRID (= MXNHEX FOR EFFICIENCY)
C     WNDSPD = THE WIND SPEED IN M/SEC AT 12.5 M ELEVATION
C     DSEED = THE SEED FOR RANDOM NUMBER GENERATION
C
C     RECORD 2 (DEFINES THE QUAD GRID AND SELECTS THE INCOMING RAY QUAD):
C
C     NMU = THE NUMBER OF MU CELLS IN ONE HEMISPHERE (0 TO PI/2)
C     NPHI = THE NUMBER OF PHI CELLS (0 TO 2*PI).  MUST BE A MULTIPLE OF 4
C     MUPART = 1 IF ALL QUADS ARE TO HAVE EQUAL SOLID ANGLES
C              2 IS ALL QUADS ARE TO HAVE EQUAL DELTA THETA VALUES
C     IR = THE INDEX OF THE INPUT MU QUAD (-NMU,...,-1,1,...,NMU)
C     JS = THE INDEX OF THE INPUT PHI QUAD (1,...,NPHI/4 + 1)
C     NUMRAY = THE NUMBER OF RAYS TO BE TRACED FROM THE INPUT QUAD
C
      PARAMETER(MXMU=10, MXPHI=24)
      PARAMETER(MXNHEX=7, MXNODE=3*MXNHEX*(MXNHEX+1)+1)
      COMMON/CMUPHI/ BNDMU(MXMU),BNDPHI(MXPHI)
      COMMON/CNODES/ NNODE,FNODE(2,MXNODE),ZNODE(MXNODE)
      COMMON/CHEXGR/ NHEX,R1(2),R2(2),R1HAT(2),R2HAT(2),R1RAT,TARGET(2)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DIMENSION DELTMU(MXMU),FMU(MXMU),PHI(MXPHI),OMEGA(MXMU)
      DIMENSION NRAYQD(MXMU)
      DOUBLE PRECISION DSEED
C
      DATA PI,RADEG,REFR/3.141592654, 57.2957795, 1.333333333/
      DATA DELTA, EPS/1.0, 1.111/, TARGET/0.5, 0.370333333/
      DATA NUSFC,NUDU,NUDD,NUUD,NUUU/15,16,17,18,19/
C
C     READ THE INPUT RECORDS
C
      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
C
C     STORE THE NEEDED PARAMETERS
C
      IMISC(1) = NMU
      IMISC(2) = NPHI
      IMISC(9) = IDBUG
      IMISC(17) = NUMRAY
      FMISC(1) = PI
      FMISC(3) = RADEG
      FMISC(15) = WNDSPD
      FMISC(16) = DELTA
      FMISC(17) = EPS
      FMISC(18) = REFR
C     RAD48 IS THE CRITICAL ANGLE FOR TOTAL INTERNAL REFLECTION
      RAD48 = ASIN(1.0/REFR)
      FMISC(19) = RAD48
C
      IF(IGENSF.GT.0) THEN
C
C*****THIS IS AN INITIAL RUN FOR GENERATION OF A FILE OF RANDOM SURFACES
C
```

```
      WRITE(6,304)
      REWIND NUSFC
C
C     CHECK TO SEE IF NUSFC IS EMPTY
      READ(NUSFC,END=50) DUMMY
      STOP 'SURFACE FILE ALREADY EXISTS'
   50 REWIND NUSFC
C
C     DEFINE THE GRID VECTORS AS IN 63/PAGES 24-26
C
      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(2)
      R1HAT(2) = R1(1)
      R2HAT(1) = -R2(2)
      R2HAT(2) = R2(1)
      R1RAT = -2.0*EPS/DELTA
C
C     DEFINE THE HEXAGONAL SURFACE GRID NODE LOCATIONS
C
      FMISC(16) = DELTA
      FMISC(17) = EPS
      CALL TRIADS(NHEX)
C
C     WRITE THE HEADER RECORDS
C
      WRITE(NUSFC) IGENSF,NHEX,NNODE,WNDSPD,DSEED
      WRITE(NUSFC) R1,R2,R1HAT,R2HAT,R1RAT,FNODE
C
C     DEFINE THE STANDARD DEVIATION FOR SURFACE HEIGHTS BY 63/2.12
C
      SIGSFC = 0.0397*SQRT(WNDSPD)
      WRITE(6,302) DELTA,EPS,SIGSFC
C
C     GENERATE AND SAVE THE CAPILLARY WAVE SURFACE REALIZATIONS,
C     63/SECTION 2C
C
      DO 55 NSFC=1,IGENSF
C
C     DRAW N(0,1) RANDOM NUMBERS
      CALL GGNML(DSEED,NNODE,ZNODE)
C
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,(ZNODE(I),I=1,NNODE)
C
      ENDFILE NUSFC
      WRITE(6,60) IGENSF
      STOP
      ENDIF
C
C*****THIS IS A PRODUCTION RUN FOR RAY TRACING
C
C     READ THE EXISTING FILE OF SURFACE REALIZATIONS AND TEST FOR
C     COMPATABILITY WITH REQUESTED PARAMETERS
C
      WRITE(6,308)
      REWIND NUSFC
      READ(NUSFC) NSF1,NHEX1,NNODE,WIND1
      READ(NUSFC) R1,R2,R1HAT,R2HAT,R1RAT,FNODE
C
      IF(NHEX1.NE.NHEX .OR. WIND1.NE.WNDSPD) THEN
      WRITE(6,70) NHEX1,WIND1
      STOP
      ENDIF
```

```
C
C     DEFINE THE MU AND PHI VALUES WHICH FORM THE QUAD BOUNDARIES FOR
C     GEOMETRIC DISCRETIZATION, SECTION 3.
C
      IF(MUPART.EQ.1) THEN
C
C     PARTITION THE UNIT SPHERE SO THAT ALL QUADS, INCLUDING THE POLAR
C     CAP, HAVE EQUAL SOLID ANGLES
C
      CALL EQSANG(NMU,NPHI,DELTMU)
C
      ELSEIF(MUPART.EQ.2) THEN
C
C     PARTITION THE UNIT SPHERE INTO EQUALLY SPACED THETA VALUES
C
      CALL EQTHET(NMU,DELTMU)
C
      ENDIF
C
C     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.
C
C     DEFINE THE MU VALUES AT THE QUAD CENTERS
      FMU(1) = 0.5*DELTMU(1)
      DO 104 I=2,NMU
  104 FMU(I) = 0.5*(BNDMU(I-1) + BNDMU(I))
C
C     DEFINE THE PHI VALUES AT THE QUAD CENTERS, AND
C     DEFINE THE BOUNDARY PHIS BY PHI - DPHI/2 TO PHI + DPHI/2
C
      DELPHI = 2.0*PI/FLOAT(NPHI)
      PHI(1) = 0.
      BNDPHI(1) = 0.5*DELPHI
      DO 102 J=2,NPHI
      PHI(J) = PHI(J-1) + DELPHI
  102 BNDPHI(J) = BNDPHI(J-1) + DELPHI
C
C     DETERMINE THE SOLID ANGLE OF THE QUADS
C
      DO 400 I=1,NMU-1
  400 OMEGA(I) = DELPHI*DELTMU(I)
      OMEGA(NMU) = 2.0*PI*DELTMU(NMU)
      IA = IABS(IR)
      NRAYQD(IA) = NUMRAY
C
      WRITE(6,310)
      DO 312 I=1,NMU
      THETAC = ACOS(FMU(I))*RADEG
      THETAB = RADEG*ACOS(BNDMU(I))
  312 WRITE(6,314) I,FMU(I),THETAC,BNDMU(I),THETAB,DELTMU(I),
     1  OMEGA(I),NRAYQD(I)
      WRITE(6,316) DELPHI*RADEG
C
C     WRITE HEADER RECORDS FOR OUTPUT FILES
C
      REWIND NUDU
      REWIND NUDD
      REWIND NUUD
      REWIND NUUU
      WRITE(NUDU) NUDU,'DOWN UP  ',IR,JS,NRAYQD
      WRITE(NUDU) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
      WRITE(NUDD) NUDD,'DOWN DOWN',IR,JS,NRAYQD
      WRITE(NUDD) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
      WRITE(NUUD) NUUD,'UP   DOWN',IR,JS,NRAYQD
      WRITE(NUUD) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
      WRITE(NUUU) NUUU,'UP   UP  ',IR,JS,NRAYQD
      WRITE(NUUU) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
C
      RETURN
C
```

```
C      FORMATS
C
   60 FORMAT(1H0,I10,' SURFACE REALIZATIONS GENERATED')
   70 FORMAT(1H0,' SURFACE REALIZATION FILE NOT COMPATABLE WITH REQUESTE
     1D PARAMETERS'//'   NHEX1 =',I3,5X,'WIND1 =',F7.3)
  300 FORMAT(1H1,' NATURAL HYDROSOL MODEL, PROGRAM 1 (1-QUAD VERSION)'//
     1'  MONTE CARLO AIR-WATER SURFACE RAY TRACING PROGRAM'//
     2'  THE HEXAGON GRID PARAMETERS FOR THIS RUN ARE'//
     35X,'NHEX   =',I2,' = ORDER OF THE SURFACE GRID HEXAGON'//
     45X,'WNDSPD =',F7.3,' = THE WIND SPEED IN M/SEC AT 12.5 M'//
     55X,'DSEED  =',1PD20.10,' = THE SEED FOR RANDOM NUMBER GENERATION')
  301 FORMAT(1H0,' 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)'//
     35X,'IR, JS =',2I3,' = THE INPUT QUAD Q(R,S)'//
     45X,'NUMRAY =',I10,' = THE TOTAL NUMBER OF RAYS TO BE TRACED')
  302 FORMAT(1H0,' THE WAVE FACET PARAMETERS ARE'//
     15X,'DELTA  =',1PE10.3//5X,'EPS    =',E10.3//5X,'SIGSFC =',E10.3)
  304 FORMAT(1H0,' THIS IS AN INITIAL RUN FOR GENERATING A FILE OF CAPIL
     1LARY WAVE SURFACE REALIZATIONS')
  308 FORMAT(1H0,' THIS IS A PRODUCTION RUN FOR RAY TRACING (1 QUAD)')
  310 FORMAT(1H0,' 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 ,I5,2(F9.4,F9.3,4X),F9.4,F12.4,I10)
  316 FORMAT(1H0,' THE QUADS HAVE A WIDTH OF DELTA PHI =',F7.3,
     1' DEGREES')
      END
```

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

$$\left\{\begin{matrix}\text{TAPE22}\\ \text{TAPE23}\\ \text{TAPE24}\\ \text{TAPE25}\end{matrix}\right\} \text{ if TAPE20 is } \left\{\begin{matrix}\text{TAPE16}\\ \text{TAPE17}\\ \text{TAPE18}\\ \text{TAPE19}\end{matrix}\right\}.$$

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 *new* ray data from a second run of Program 1. TAPE21 is the corresponding output file from the *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,

$$\text{if TAPE20 is the new } \left\{\begin{matrix}\text{TAPE16}\\ \text{TAPE17}\\ \text{TAPE18}\\ \text{TAPE19}\end{matrix}\right\}, \text{ then TAPE21 is } \left\{\begin{matrix}\text{TAPE22}\\ \text{TAPE23}\\ \text{TAPE24}\\ \text{TAPE25}\end{matrix}\right\} \text{ from the previous run,}$$

$$\text{and NUOUT is the updated } \left\{\begin{matrix}\text{TAPE22}\\ \text{TAPE23}\\ \text{TAPE24}\\ \text{TAPE25}\end{matrix}\right\}.$$

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:

$$\left\{\begin{matrix}\text{TAPE22}\\ \text{TAPE23}\\ \text{TAPE24}\\ \text{TAPE25}\end{matrix}\right\} \text{ contains } \left\{\begin{matrix}\underline{r}(a,x)\\ \underline{t}(a,x)\\ \underline{r}(x,a)\\ \underline{t}(x,a)\end{matrix}\right\}.$$

All four of these files are read by Program 3. TAPE22 ($\underline{r}(a,x)$) and TAPE25 ($\underline{t}(x,a)$) also are read by Program 5, if the contrast transmittance is computed.

D. Code Listing

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE20,
     1  TAPE21,TAPE22,TAPE23,TAPE24,TAPE25)
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                +
C     +  THIS IS PROGRAM 2 OF THE NATURAL HYDROSOL MODEL +
C     +                                                +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     ON NHM2/M2ALL
C
C     THIS PROGRAM READS AN OUTPUT FILE WRITTEN BY NHM1/M1ALL AND
C     TALLEYS THE SCATTERED RAYS TO COMPUTE THE CORRESPONDING
C     GEOMETRIC REFLECTANCE OR TRANSMITTANCE ARRAY, AS DESCRIBED IN
C     SECTION 9.
C
C     THIS PROGRAM COMPUTES AND STORES THE "TOP HALF" OF RTGEO.
C     SEE SECTION 12B FOR THE BLOCK SYMMETRIES USED.
C
C     INPUT:
C     NEWRUN = 1, IF THIS RUN STARTS FROM SCRATCH
C              0, IF THIS IS A CONTINUATION RUN
C
C     TAPE20 = A FILE OF RAY DATA WRITTEN BY NHM1/M1ALL AS
C              TAPE16, 17, 18, OR 19
C
C     TAPE21, IF NEWRUN = 0, THE FILE 22, 23, 24, OR 25 WRITTEN BY THE
C             PREVIOUS RUN OF NHM2/M2ALL, CONTAINING THE RTGEO ARRAY
C
C     OUTPUT:
C     NUOUT = TAPE22 IF TAPE20 IS TAPE16 OF NHM1
C           = TAPE23 IF TAPE20 IS TAPE17 OF NHM1, ETC.
C
      PARAMETER (MXMU=10,MXPHI=24)
C     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(20)
      DIMENSION RTGEO(MXROW,MXCOL),NRAYQD(MXMU)
      CHARACTER RTLABL*6
C
C     INITIALIZE
C
      CALL INISHL(RTGEO,RTLABL,NUOUT,NEWRUN,NRAYQD)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      IDBUG = IMISC(9)
      RADEG = FMISC(3)
      NUMCOL = NMU*NPHI
      NUMROW = NUMCOL/2
      NREC = 0
C
C     READ AND ACCUMULATE RAY CONTRIBUTIONS.  THIS IS THE SUM OVER OMEGA
C     IN 9.1, BUT WITHOUT THE 1/S FACTOR.  THE SUM OVER J IN 9.1
C     WAS DONE AUTOMATICALLY AS THE RAY WAS TRACED TO COMPLETION.
C     THE INPUT QUAD Q(R,S) IS (I,J); THE OUTPUT QUAD Q(U,V) IS (K,L)
C
      IOLD = 0
      IF(NEWRUN.EQ.1) WRITE(6,102)
      NPNT = 0
C
  200 READ(20,END=250) I,J,K,L,RAD
      NREC = NREC + 1
C
```

```
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
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.1 .AND. I.NE.IOLD .AND. NPNT.LT.25) THEN
      IOLD = I
      NPNT = NPNT + 1
      WRITE(6,104) NREC,I,J,K,L,RAD,IROW,JCOL
      ENDIF
C
      RTGEO(IROW,JCOL) = RTGEO(IROW,JCOL) + RAD
      GO TO 200
C
  250 WRITE(6,110) NREC
C
C     RTGEO IS NOW PROPORTIONAL TO THE RADIANT FLUX TRANSFER FUNCTION
C
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 - 1
      IF(JS.EQ.1) MAXIR = NMU
      DO 252 IR=1,MAXIR
      IROW = IR + (JS - 1)*NMU
C     NRAYQD(IR) IS S OF 9.1
      FACT1 = FMU(IR)*OMEGA(IR)/FLOAT(NRAYQD(IR))
C     NON-POLAR QUADS
      DO 253 KU=1,NMU-1
C     FACT2 CONTAINS THE MU AND OMEGA FACTORS OF 9.7, AND 1/S OF 9.1
      FACT2 = FACT1/(FMU(KU)*OMEGA(KU))
      DO 253 LV=1,NPHI
      JCOL = KU + (LV-1)*NMU
  253 RTGEO(IROW,JCOL) = FACT2*RTGEO(IROW,JCOL)
C     POLAR CAPS:  KU = NMU
  252 RTGEO(IROW,NMU) = FACT1*RTGEO(IROW,NMU)/OMEGA(NMU)
C
C     RTGEO IS NOW THE QUAD-AVERAGED GEOMETRIC R OR T ARRAY
C
C     FILL OUT THE REMAINING ROWS (THE SECOND QUADRANT) OF THE "TOP HALF"
C     OF RTGEO BY SYMMETRY (SEE PAGE 190).
C
C     (IP,I) ARE THE (ROW,COLUMN) 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*(IBP - 1)
      DO 300 I=1,NPHI
      IB = NOPI + 2 - I
      IF(IB.LE.0) IB = IB + NPHI
      IRT = NMU*(I - 1)
      IRTB = NMU*(IB - 1)
C
C     COPY THE NMU BY NMU BLOCK
      DO 300 K=1,NMU
      DO 300 KP=1,NMU
  300 RTGEO(IRTBP+KP,IRTB+K) = RTGEO(IRTP+KP,IRT+K)
```

```
C
C     RESET THE INPUT SECOND QUADRANT, OUTPUT POLAR CAP DIRECTION
C     (COLUMN NMU), WHICH HAS PICKED UP ZERO VALUES FROM THE INPUT
C     FIRST QUADRANT, PHI = 180 BLOCKS
C
      DO 310 IP=2,N34
      IBP = NOPI + 2 - IP
      IRTP = NMU*(IP - 1)
      IRTBP = NMU*(IBP - 1)
      DO 310 K=1,NMU
  310 RTGEO(IRTBP+K,NMU) = RTGEO(IRTP+K,NMU)
C
C     RE-ZERO THE INPUT SECOND QUADRANT, OUTPUT PHI = 180 COLUMN, WHICH
C     HAS PICKED UP NON-ZERO VALUES FROM THE INPUT FIRST QUADRANT,
C     OUTPUT POLAR CAP (PHI = 0) COLUMN
C
      JCOL = NMU*(NOPI + 1)
      DO 312 I=1,NUMROW
  312 RTGEO(I,JCOL) = 0.
C
C     WRITE THE FINAL ARRAY TO THE OUTPUT FILE. ONLY THE "TOP HALF" IS
C     STORED (SEE PAGE 190).
C
      DO 270 JCOL=1,NUMCOL
  270 WRITE(NUOUT) (RTGEO(IR,JCOL),IR=1,NUMROW)
      ENDFILE NUOUT
      WRITE(6,271) NUOUT
C
C     PRINT SELECTED PARTS OF THE NEW RTGEO
C
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,I,THET,(RTGEO(I,J),J=1,NMU)
C
C     THE SPECULAR BLOCK FOR PHI PRIME = 90
      IS = NPHI/4 + 1
      IR1 = NMU*(IS - 1) + 1
      IR2 = IR1 + NMU -1
      WRITE(6,113)IS,RADEG*PHI(IS),IS,RADEG*PHI(IS),RTLABL,(J,J=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=IR1,IR2)
      IF(IDBUG.EQ.1) CALL P2ARAY(RTGEO,2*NMU,2*NMU,MXROW,2,
     1' THE UPPER LEFT BLOCKS OF THE NEW RTGEO ARRAY')
      IF(IDBUG.EQ.2) CALL P2ARAY(RTGEO,NUMROW,NUMCOL,MXROW,2,
     1' THE TOP HALF OF THE FULL RTGEO ARRAY')
      WRITE(6,605)
C
C     FORMATS
C
  102 FORMAT(1H0,' SELECTED RAY DATA:'//'        NREC    IR    JS    KU',
     1'    LV  FRESNEL RT    ROW    COL'/)
  104 FORMAT(1H ,I10,4I5,F10.5,I8,I6)
  110 FORMAT(1H ,I10,' DATA RECORDS READ FROM UNIT 20')
  113 FORMAT(1H0,' (SPECULAR) BLOCK FOR PHI PRIME(',I2,') =',F6.1,
     1' AND PHI(',I2,') =',F6.1,' OF THE NEW ',A6,' ARRAY'//
     217X,'COLUMN:    ',10I10/
     333X,'MU(1)      MU(2)      MU(3)      MU(4)      MU(5)      MU(6)      M
     4U(7)      MU(8)      MU(9)      MU(10)'/)
  115 FORMAT(' ROW',I4,'  MU(',I2,') =',F6.1,4X,10(2X,F8.5),(/24X,
     1 10(2X,F8.5)))
  271 FORMAT(1H0,' EOF WRITTEN ON FILE NUOUT = TAPE',I2)
  605 FORMAT(1H0,' NORMAL EXIT FROM NHM, PROGRAM II.')
      END
```

```
      SUBROUTINE INISHL(RTGEO,RTLABL,NUOUT,NEWRUN,NRAYQD)
C
C     ON NHM2/IN2ALL
C
C     THIS ROUTINE INITIALIZES PROGRAM NHM2/M2ALL
C
      PARAMETER (MXMU=10, MXPHI=24)
      PARAMETER(MXROW=MXMU*MXPHI/2, MXCOL=MXMU*MXPHI)
      DIMENSION RTGEO(MXROW,MXCOL)
      DIMENSION NRAYQD(MXMU),BNDMU(MXMU),BNDPHI(MXPHI),DELTMU(MXMU)
      DIMENSION IMISC2(20),FMISC2(20),NRAQD2(MXMU)
      COMMON/CMUPHI/ FMU(MXMU),PHI(MXPHI),OMEGA(MXMU)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      CHARACTER UPDOWN*9,RTLABL*6,UPDN2*9,RTLAB2*6
C
      READ(5,*) NEWRUN,IDBUG
C
C     READ HEADER RECORD OF RAY DATA FILE (TAPE16, 17, 18, OR 19)
      REWIND 20
      READ(20) NU20,UPDOWN,NRAYQD
      READ(20) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NUMRAY = IMISC(17)
      RADEG = FMISC(3)
      WNDSPD = FMISC(15)
      REFR = FMISC(18)
      IMISC(9) = IDBUG
C
      NUOUT = NU20 + 6
      IPI2 = NPHI/4 + 1
C
      IF(UPDOWN.EQ.'DOWN DOWN') THEN
      RTLABL = 'T(A,X)'
      ELSEIF(UPDOWN.EQ.'UP   UP  ') THEN
      RTLABL = 'T(X,A)'
      ELSEIF(UPDOWN.EQ.'DOWN UP  ') THEN
      RTLABL = 'R(A,X)'
      ELSEIF(UPDOWN.EQ.'UP   DOWN') THEN
      RTLABL = 'R(X,A)'
      ELSE
      WRITE(6,118) UPDOWN
      STOP
      ENDIF
C
      NUMCOL = NMU*NPHI
      NUMROW = NUMCOL/2
C
      WRITE(6,100) RTLABL,UPDOWN,NMU,NPHI,WNDSPD,REFR
      WRITE(6,110) NUMRAY
C
      IF(NEWRUN.EQ.1) THEN
C     THIS IS A NEW RUN, ZERO RTGEO
C     ZERO ONLY THOSE ARRAY ELEMENTS WHICH ARE ACTUALLY USED FOR STORAGE,
C     AS AN AID TO DEBUGGING ON THE 855
C     DO 98 JCOL=1,NUMCOL
C     DO 98 IROW=1,NUMROW
C  98 RTGEO(IROW,JCOL) = 0.
C     NON POLAR OUTPUT QUADS
      DO 98 IV=1,NPHI
      DO 98 IU=1,NMU-1
      JCOL = IU + (IV-1)*NMU
C     NON POLAR INPUT QUADS
      DO 99 IS=1,IPI2
      DO 99 IR=1,NMU-1
      IROW = IR + (IS-1)*NMU
   99 RTGEO(IROW,JCOL) = 0.
```

```
C       POLAR CAP INPUT QUAD
   98 RTGEO(NMU,JCOL) = 0.
C       NON POLAR INPUT QUADS, POLAR CAP OUTPUT QUAD
        DO 97 IS=1,IPI2
        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
C       THIS IS A CONTINUATION RUN, READ EXISTING RTGEO (TAPE21 = NUOUT OF PREVIOUS RUN)
        REWIND 21
        READ(21) NU21,NRAQD2,IMISC2,FMISC2
C
        NMU2 = IMISC2(1)
        NPHI2 = IMISC2(2)
        NUMRA2 = IMISC2(17)
        WNDSP2 = FMISC2(15)
C
C       CHECK FOR COMPATIBLE FILES
        IF(NUOUT.NE.NU21 .OR. NMU.NE.NMU2 .OR.
       1 NPHI.NE.NPHI2 .OR. WNDSPD.NE.WNDSP2) THEN
        WRITE(6,200)
        WRITE(6,202) NU20,NMU,NPHI,WNDSPD
        WRITE(6,202) NU21,NMU2,NPHI2,WNDSP2
        STOP
        ENDIF
C
C       THE "TOP HALF" OF RTGEO IS STORED
C
        DO 130 JCOL=1,NUMCOL
  130 READ(21) (RTGEO(IR,JCOL),IR=1,NUMROW)
        WRITE(6,112) RTLABL,NUMRA2
C
C       PRINT SELECTED PARTS OF THE EXISTING RTGEO
C
C       THE SPECULAR BLOCK FOR PHI 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*ACOS(FMU(I))
  114 WRITE(6,115) I,THET,(RTGEO(I,J),J=JC1,JC2)
C
C       THE SPECULAR BLOCK FOR PHI PRIME = 90
        IS = NPHI/4 + 1
        IV = IS + NPHI/2
        IR1 = NMU*(IS - 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=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=JC1,JC2)
        IF(IDBUG.GT.1) CALL P2ARAY(RTGEO,NMU,NUMCOL,MXROW,2,
       1' THE PHI PRIME = 0 BLOCKS OF THE EXISTING RTGEO ARRAY')
C
C       CONVERT THE GEOMETRIC R OR T ARRAY BACK INTO A RAY-TALLY ARRAY,
C       I.E. UNDO 9.7
C
```

```
      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-1
      F2 = F1*FMU(KU)*OMEGA(KU)
      DO 121 LV=1,NPHI
      JCOL = KU + (LV-1)*NMU
  121 RTGEO(IROW,JCOL) = F2*RTGEO(IROW,JCOL)
C     POLAR CAPS:  KU = NMU
  120 RTGEO(IROW,NMU) = F1*OMEGA(NMU)*RTGEO(IROW,NMU)
      NUMRAY = NUMRAY + NUMRA2
      IMISC(17) = NUMRAY
      DO 122 I=1,NMU
  122 NRAYQD(I) = NRAYQD(I) + NRAQD2(I)
      ENDIF
C
C     WRITE HEADER ON OUTPUT FILE
C
      REWIND NUOUT
      WRITE(NUOUT) NUOUT,NRAYQD,IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,
     1  DELTMU
C
      RETURN
C
C     FORMATS
C
  100 FORMAT(1H1,' NATURAL HYDROSOL MODEL, PROGRAM 2'//
     1'  RAY TALLY AND COMPUTATION OF ',A6,' FROM ',A9,//
     11H ,' FOR NMU =',I3,'   NPHI =',I3,'   WNDSPD =',F8.3,
     2' M/SEC'//1H ,'   REFR =',F7.4)
  110 FORMAT(1HO,' FOR THE CURRENT RUN, NUMRAY =',I10,
     1' TOTAL RAYS TRACED')
  112 FORMAT(1HO,' THE EXISTING GEOMETRIC ',A6,
     1' ARRAY WAS ACCUMULATED FROM'/I10,' RAYS')
  113 FORMAT(1HO,' (SPECULAR) BLOCK FOR PHI PRIME(',I2,') =',F6.1,
     1' AND PHI(',I2,') =',F6.1,' OF THE EXISTING ',A6,' ARRAY'//
     217X,'COLUMN:    ',10I10/
     333X,'MU(1)      MU(2)      MU(3)      MU(4)      MU(5)      MU(6)      M
     4U(7)      MU(8)      MU(9)     MU(10)'/)
  115 FORMAT(' ROW',I4,'  MU(',I2,') =',F6.1,4X,10(2X,F8.5),(/24X,
     1 10(2X,F8.5)))
  118 FORMAT(1HO,' UPDOWN =',A9,'  ERROR STOP')
  200 FORMAT(1HO,' FILES 20 AND 21 INCOMPATIBLE:'/)
  202 FORMAT(1HO,' FILE',I3,': NMU, NPHI, WNDSPD ='
     1//1H ,12X,2I4,F10.3)
      END
```

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE20,
     1             TAPE21,TAPE22,TAPE23,TAPE24,TAPE25)
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                  +
C     +  THIS IS PROGRAM 2 OF THE NATURAL HYDROSOL MODEL +
C     +                                                  +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     ON NHM2/M21QD
C
C     THIS PROGRAM READS AN OUTPUT FILE WRITTEN BY NHM1/M11QD AND
C     TALLEYS THE SCATTERED RAYS TO COMPUTE THE CORRESPONDING ROW
C     OF THE GEOMETRIC REFLECTANCE OR TRANSMITTANCE ARRAY, AS
C     DESCRIBED IN SECTION 9.
C
C     THIS SPECIAL VERSION OF MAIN2 DOES ONLY ONE INPUT QUAD (ONE ROW
C     OF R OR T).
C
C     INPUT:
C     NEWRUN = 1, IF THIS RUN STARTS FROM SCRATCH
C              0, IF THIS IS A CONTINUATION RUN
C
C     TAPE20 = A FILE OF RAY DATA WRITTEN BY NHM1/M11QD AS
C              TAPE16, 17, 18, OR 19
C
C     TAPE21, IF NEWRUN = 0, THE FILE 22, 23, 24, OR 25 WRITTEN BY THE
C             PREVIOUS RUN OF NHM2/M21QD, CONTAINING THE RTGEO ARRAY
C
C     OUTPUT:
C     NUOUT = THE FILE WITH THE COMPUTED RTGEO ARRAY
C             NUOUT = TAPE22 IF TAPE20 IS TAPE16 OF NHM1/M11QD
C                   = TAPE23 "     "   "  TAPE17 "     "    , ETC.
C
      PARAMETER(MXMU=10,MXPHI=24, MXCOL=MXMU*MXPHI)
      COMMON/CMUPHI/ FMU(MXMU),PHI(MXPHI),OMEGA(MXMU)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DIMENSION RTGEO(MXCOL),KNTRAY(MXCOL)
      CHARACTER RTLABL*6
C
C     INITIALIZE
C
      CALL INISHL(RTGEO,KNTRAY,IROW,RTLABL,NUOUT)
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NUMRAY = IMISC(17)
      RADEG = FMISC(3)
      NUMCOL = NMU*NPHI
      NREC = 0
C
C     READ AND ACCUMULATE RAY CONTRIBUTIONS.  THIS IS THE SUM OVER OMEGA
C     IN 9.1, BUT WITHOUT THE 1/S FACTOR.  THE SUM OVER J IN 9.1 WAS
C     DONE AUTOMATICALLY AS THE RAY WAS TRACED TO COMPLETION.
C
      WRITE(6,102)
  200 READ(20,END=250) I,J,K,L,RAD
      NREC = NREC + 1
C     ANY RAYS GOING INTO A POLAR CAP ARE STORED IN COLUMN NMU
      LL = L
      IF(K.EQ.NMU) LL = 1
      JCOL = K + (LL - 1)*NMU
      IF(NREC.LE.25) WRITE(6,104) NREC,I,J,K,L,RAD,JCOL
      KNTRAY(JCOL) = KNTRAY(JCOL) + 1
      RTGEO(JCOL) = RTGEO(JCOL) + RAD
      GO TO 200
C
  250 WRITE(6,110) NREC
C
C     RTGEO IS NOW PROPORTIONAL TO THE RADIANT FULX TRANSFER FUNCTION
C
C     CONVERT THE RAY-TALLY ARRAY INTO A GEOMETRIC R OR T ARRAY BY 9.7
C
```

```
      IR = MOD(IROW,NMU)
      IF(IR.EQ.0) IR = NMU
C     NUMRAY IS S OF 9.1
      FACT1 = FMU(IR)*OMEGA(IR)/FLOAT(NUMRAY)
      DO 252 KU=1,NMU-1
C     FACT2 CONTAINS THE MU AND OMEGA FACTOR OF 9.7, AND 1/S OF 9.1
      FACT2 = FACT1/(FMU(KU)*OMEGA(KU))
      DO 252 LV=1,NPHI
      JCOL = KU + (LV-1)*NMU
  252 RTGEO(JCOL) = FACT2*RTGEO(JCOL)
C     POLAR CAP:  KU = NMU
      RTGEO(NMU) = FACT1*RTGEO(NMU)/OMEGA(NMU)
C
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 = MAXO(2,IV-3)
      IV2 = MINO(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 = MAXO(2,IV-3)
      IV2 = MINO(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 IV=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 RT*MU(U)*OMEGA(U))/(MU(R)*OMEGA(R)) =',
     1' IRRAD REFL/TRANS =',F8.6)
C
C     WRITE FINAL ARRAY TO OUTPUT FILE
C
      WRITE(NUOUT) (RTGEO(JCOL),JCOL=1,NUMCOL)
      WRITE(NUOUT) (KNTRAY(JCOL),JCOL=1,NUMCOL)
      ENDFILE NUOUT
C
C     FORMATS
C
  102 FORMAT(1H0,' SELECTED RAY DATA:'//'        NREC   IR    JS    KU',
     1'    LV   FRESNEL RT    COL'/)
  104 FORMAT(1H ,I10,4I5,F10.5,I8)
  110 FORMAT(1H0,I8,' DATA RECORDS READ FROM UNIT 20')
  112 FORMAT(1H0,' SELECTED COLUMNS OF ROW',I4,' OF THE ',A6,
     1' ARRAY'//' (ACCUMULATED FROM',I8,' INITIAL RAYS:'//
     325X,'MU(1)      MU(2)      MU(3)      MU(4)      MU(5)      MU(6)      M
     4U(7)      MU(8)      MU(9)      MU(10)'/)
  262 FORMAT(' PHI(',I2,') =',F6.1,4X,10(2X,F8.5),(/24X,10(2X,F8.5)))
  272 FORMAT(' PHI(',I2,') =',F6.1,4X,10(2X,I8),(/24X,10(2X,I8)))
      END
```

```
      SUBROUTINE INISHL(RTGEO,KNTRAY,IROW,RTLABL,NUOUT)
C
C     ON NHM2/IN21QD
C
C     THIS ROUTINE INITIALIZES PROGRAM NHM2/M21QD
C
      PARAMETER(MXMU=10,MXPHI=24)
      COMMON/CMUPHI/ FMU(MXMU),PHI(MXPHI),OMEGA(MXMU)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DIMENSION RTGEO(1),KNTRAY(1)
      DIMENSION BNDMU(MXMU),BNDPHI(MXPHI),DELTMU(MXMU),NRAYQD(MXMU)
      DIMENSION IMISC2(20),FMISC2(20),NRAQD2(MXMU)
      CHARACTER UPDOWN*9,RTLABL*6,UPDN2*9,RTLAB2*6
C
      READ(5,*) NEWRUN
C
C     READ HEADER RECORDS OF RAY DATA FILE
      REWIND 20
      READ(20) NU20,UPDOWN,IR,JS,NRAYQD
      READ(20) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NUMRAY = IMISC(17)
      WNDSPD = FMISC(15)
      REFR = FMISC(18)
      NUOUT = NU20 + 6
      IA = IABS(IR)
      IF(IA.EQ.NMU) JS = 1
      IROW = IA + (JS-1)*NMU
C
C     DETERMINE THE TYPE OF ARRAY BEING PROCESSED
C
      IF(UPDOWN.EQ.'DOWN DOWN') THEN
      RTLABL = 'T(A,X)'
      ELSEIF(UPDOWN.EQ.'UP   UP  ') THEN
      RTLABL = 'T(X,A)'
      ELSEIF(UPDOWN.EQ.'DOWN UP  ') THEN
      RTLABL = 'R(A,X)'
      ELSEIF(UPDOWN.EQ.'UP   DOWN') THEN
      RTLABL = 'R(X,A)'
      ELSE
      WRITE(6,118) UPDOWN
      STOP
      ENDIF
C
      NUMCOL = NMU*NPHI
      WRITE(6,100) IROW,RTLABL,UPDOWN,NMU,NPHI,WNDSPD,REFR,IR,JS
      NRAQDT = NRAYQD(IA)
      WRITE(6,110) NRAQDT
C
      IF(NEWRUN.EQ.1) THEN
C
C     THIS IS A NEW RUN, ZERO RTGEO AND KNTRAY
C     ONLY THOSE ELEMENTS ACTUALLY USED FOR STORAGE ARE SET TO ZERO,
C     AS AN AID TO DEBUGGING
      DO 98 J=1,NPHI
      DO 98 I=1,NMU-1
      JCOL = I + (J-1)*NMU
      KNTRAY(JCOL) = 0
   98 RTGEO(JCOL) = 0.
C     POLAR CAP
      KNTRAY(NMU) = 0
      RTGEO(NMU) = 0.
C
      ELSE
C
C     THIS IS A CONTINUATION RUN, READ EXISTING RTGEO
      REWIND 21
      READ(21) NUOUT2,UPDN2,RTLAB2,IR2,JS2,NRAQD2
      READ(21) IMISC2,FMISC2
      NMU2 = IMISC2(1)
      NPHI2 = IMISC2(2)
      NUMRA2 = IMISC2(17)
      WNDSP2 = FMISC2(15)
```

```
C
C     CHECK FOR COMPATABLE FILES
      IF(UPDOWN.NE.UPDN2 .OR. RTLABL.NE.RTLAB2 .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)
      IFILE = 20
      WRITE(6,202) IFILE,UPDOWN,IR,JS,NMU,NPHI,WNDSPD,NUOUT
      IFILE = 21
      WRITE(6,202) IFILE,UPDN2,IR2,JS2,NMU2,NPHI2,WNDSP2,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,(RTGEO(JCOL+M),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
C     CONVERT THE GEOMETRIC R OR T ARRAY BACK INTO A RAY-TALLY ARRAY,
C     I.E. UNDO 9.7
C
      F1 = FLOAT(NUMRA2)/(FMU(IA)*OMEGA(IA))
      DO 120 KU=1,NMU-1
      F2 = F1*FMU(KU)*OMEGA(KU)
      DO 120 LV=1,NPHI
      JCOL = KU + (LV-1)*NMU
  120 RTGEO(JCOL) = F2*RTGEO(JCOL)
C     POLAR CAP:  KU = NMU
      RTGEO(NMU) = F1*OMEGA(NMU)*RTGEO(NMU)
      NRAQDT = NRAQDT + NUMRA2
      IMISC(17) = NRAQDT
      ENDIF
C
C     WRITE HEADER ON OUTPUT FILE
C
      REWIND NUOUT
      WRITE(NUOUT) NUOUT,UPDOWN,RTLABL,IR,JS,NRAQDT
      WRITE(NUOUT) IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
C
      RETURN
C
C     FORMATS
C
  100 FORMAT(1H1,' NATURAL HYDROSOL MODEL, PROGRAM 2 (1-QUAD VERSION)'//
     1' RAY TALLY FOR COMPUTATION OF'//
     11H ,' ROW',I4,' OF ',A6,' (FROM ',A9,') FOR'//1H ,' NMU =',I3,
     2'  NPHI  =',I3,'  WNDSPD =',F8.3,' M/SEC     REFR =',F7.4//
     31H ,' THE FIXED INPUT QUAD Q(R,S) WAS (',I2,',',I2,')')
  110 FORMAT(1H0,' FOR THE CURRENT RUN, NRAQDT =',I6)
  112 FORMAT(1H0,' ROW',I4,' OF THE EXISTING ',A6,' ARRAY '
     1//1H ,' (ACCUMULATED FROM',I8,' INITIAL RAYS)'/)
  116 FORMAT(1H ,' COLUMNS',I4,' TO',I4,10(2X,F8.5))
  119 FORMAT(1H ,' COLUMNS',I4,' TO',I4,10(2X,F8.5))
  118 FORMAT(1H0,' UPDOWN =',A9,'  ERROR STOP')
  200 FORMAT(1H0,' FILES 20 AND 21 INCOMPATABLE:'/)
  202 FORMAT(1H0,' FILE',I3,': UPDOWN, IR, J, NMU, NPHI, WNDSPD, NU
     1OUT ='//1H ,12X,A9,4I4,F10.3,I5)
      END
```

4. PROGRAM 3

A. Program Description

This program reads the four quad-averaged *geometric* reflectance and transmittance arrays computed by Program 2 ($\underline{r}(a,x)$ on TAPE22, etc.). The corresponding *spectral* arrays $\underline{\hat{r}}_1(a,x)$, $\underline{\hat{r}}_2(a,x)$, 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: IDBUG
where IDBUG = 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:

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

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

D. Code Listing

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1 TAPE22,TAPE23,TAPE24,TAPE25,TAPE30)
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                +
C     +  THIS IS PROGRAM 3 OF THE NATURAL HYDROSOL MODEL +
C     +                                                +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     ON NHM3/MAIN3
C
C     THIS PROGRAM COMPUTES THE UPPER BOUNDARY SPECTRAL REFLECTANCE AND
C     TRANSMITTANCE ARRAYS WHICH DESCRIBE THE AIR-WATER INTERFACE.
C     THE GOVERNING EQUATIONS ARE 5.31C TO 5.36.
C
C     THE ARRAYS ARE COMPUTED IN THE ORDER IN WHICH THE SPECTRAL ARRAYS
C     ARE NEEDED BY PROGRAM 4, NAMELY
C          THAT(A,X), RHAT(X,A), THAT(X,A), RHAT(A,X)
C
C     THE GEOMETRIC ARRAYS ARE READ FROM THE OUTPUT FILES WHICH
C     WERE WRITTEN BY PROGRAM 2 (TAPES 22, 23, 24, AND 25)
C
C     THE SPECTRAL ARRAYS ARE WRITTEN TO NUOUT (TAPE30)
C
C     RTHAT1 AND RTHAT2 ARE EACH NMU*(NL+1) BY NMU*INT((NL+2)/2) WORDS
C     THE STORED RT ARRAY IS NMU*NPHI/2 BY NMU*NPHI WORDS (THE TOP HALF)
C
      PARAMETER(MXMU=10, MXPHI=24)
      PARAMETER(IDRT=MXMU*MXPHI, MXNL=MXPHI/2)
      PARAMETER(ID1HAT=MXMU*(MXNL+1), ID2HAT=MXMU*((MXNL+2)/2))
      DIMENSION RT(IDRT,IDRT)
      DIMENSION RTHAT1(ID1HAT,ID2HAT),RTHAT2(ID1HAT,ID2HAT)
      COMMON/CPHI/ PHI(MXPHI)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DATA NURAX,NUTAX,NURXA,NUTXA/22,23,24,25/, NUOUT/30/
C
C     INITIALIZE THE PROGRAM
C
      CALL INISHL
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      IDBUG = IMISC(9)
C
      NROWRT = NMU*NPHI/2
      NCOLRT = NMU*NPHI
      NRHAT = IMISC(10)
      NCHAT = IMISC(11)
      IPRRT = MIN0(20,NROWRT)
      IPCRT = MIN0(20,NCOLRT)
      IPRHAT = MIN0(40,NRHAT)
      IPCHAT = MIN0(20,NCHAT)
      IF(IDBUG.GE.2) GO TO 888
C
C+++++DOWNWARD TRANSMITTANCE T(A,X)
C     RT CONTAINS THE GEOMETRIC T(A,X;R,S;U,V)
C     RTHAT1 CONTAINS THE SPECTRAL THAT1(A,X;R,L/U,K)
C     RTHAT2 CONTAINS THE SPECTRAL THAT2(A,X;R,L/U,K)
C
C     READ THE GEOMETRIC T(A,X)
C
      READ(NUTAX) NUNIT
      IF(NUNIT.EQ.NUTAX) THEN
      WRITE(6,700) 'THAT1(A,X)','THAT2(A,X)','T(A,X)'
      ELSE
      WRITE(6,702) NUNIT,'NUTAX',NUTAX
      STOP
      ENDIF
```

```
      DO 710 J=1,NCOLRT
  710 READ(NUTAX) (RT(I,J),I=1,NROWRT)
C
C     DEFINE A FULL RT ARRAY TO AVOID SPECIAL INDEXING IN RTSPEC
C
      CALL FULLRT(RT,NMU,NPHI,IDRT)
C
      CALL P2ARAY(RT,IPRRT,IPCRT,IDRT,2,'GEOMETRIC TAX(R,S/U,V)')
C
C     COMPUTE THAT(A,X)
      CALL RTSPEC(PHI,RT,IDRT, RTHAT1,RTHAT2,ID1HAT)
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)
      CALL P2ARAY(RTHAT1,IPRHAT,IPCHAT,ID1HAT,2,'AMP ARRAY THAT1(A,X)')
      CALL P2ARAY(RTHAT2,IPRHAT,IPCHAT,ID1HAT,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 RHAT1(X,A;R,L/U,K)
C     RTHAT2 CONTAINS THE SPECTRAL RHAT2(X,A;R,L/U,K)
C
C     READ THE GEOMETRIC R(X,A)
      READ(NURXA) NUNIT
      IF(NUNIT.EQ.NURXA) THEN
      WRITE(6,700) 'RHAT1(X,A)','RHAT2(X,A)','R(X,A)'
      ELSE
      WRITE(6,702) NUNIT,'NURXA',NURXA
      STOP
      ENDIF
      DO 720 J=1,NCOLRT
  720 READ(NURXA) (RT(I,J),I=1,NROWRT)
      CALL FULLRT(RT,NMU,NPHI,IDRT)
      CALL P2ARAY(RT,IPRRT,IPCRT,IDRT,2,'GEOMETRIC RXA(R,S/U,V)')
C
C     COMPUTE RHAT(X,A)
      CALL RTSPEC(PHI,RT,IDRT, RTHAT1,RTHAT2,ID1HAT)
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)
      CALL P2ARAY(RTHAT1,IPRHAT,IPCHAT,ID1HAT,2,'AMP ARRAY RHAT1(X,A)')
      CALL P2ARAY(RTHAT2,IPRHAT,IPCHAT,ID1HAT,2,'AMP ARRAY RHAT2(X,A)')
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(NUTXA) NUNIT
      IF(NUNIT.EQ.NUTXA) THEN
      WRITE(6,700) 'THAT1(X,A)','THAT2(X,A)','T(X,A)'
      ELSE
      WRITE(6,702) NUNIT,'NUTXA',NUTXA
      STOP
      ENDIF
      DO 730 J=1,NCOLRT
  730 READ(NUTXA) (RT(I,J),I=1,NROWRT)
C
      CALL FULLRT(RT,NMU,NPHI,IDRT)
      CALL P2ARAY(RT,IPRRT,IPCRT,IDRT,2,'GEOMETRIC TXA(R,S/U,V)')
C
C     COMPUTE THAT(X,A)
      CALL RTSPEC(PHI,RT,IDRT, RTHAT1,RTHAT2,ID1HAT)
C
```

```
      DO 231 J=1,NCHAT
  231 WRITE(NUOUT) (RTHAT1(I,J),I=1,NRHAT)
      DO 232 J=1,NCHAT
  232 WRITE(NUOUT) (RTHAT2(I,J),I=1,NRHAT)
      CALL P2ARAY(RTHAT1,IPRHAT,IPCHAT,ID1HAT,2,'AMP ARRAY THAT1(X,A)')
      CALL P2ARAY(RTHAT2,IPRHAT,IPCHAT,ID1HAT,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
  888 CONTINUE
C
C     READ THE GEOMETRIC R(A,X)
      READ(NURAX) NUNIT
      IF(NUNIT.EQ.NURAX) THEN
      WRITE(6,700) 'RHAT1(A,X)','RHAT2(A,X)','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,IDRT)
      CALL P2ARAY(RT,IPRRT,IPCRT,IDRT,2,'GEOMETRIC RAX(R,S/U,V)')
C
C     COMPUTE RHAT(A,X)
      CALL RTSPEC(PHI,RT,IDRT, RTHAT1,RTHAT2,ID1HAT)
C
      DO 201 J=1,NCHAT
  201 WRITE(NUOUT) (RTHAT1(I,J),I=1,NRHAT)
      DO 202 J=1,NCHAT
  202 WRITE(NUOUT) (RTHAT2(I,J),I=1,NRHAT)
      CALL P2ARAY(RTHAT1,IPRHAT,IPCHAT,ID1HAT,2,'AMP ARRAY RHAT1(A,X)')
      CALL P2ARAY(RTHAT2,IPRHAT,IPCHAT,ID1HAT,2,'AMP ARRAY RHAT2(A,X)')
C
      ENDFILE NUOUT
      WRITE(6,750) NUOUT
C
C     FORMATS
C
  700 FORMAT(1H1,' NOW COMPUTING ',A10,' AND ',A10,' FROM ',A6)
  702 FORMAT(1H0,' ERROR:  NUNIT =',I3,' AND ',A6,' =',I3)
  750 FORMAT(1H0,' NORMAL EXIT FROM PROGRAM 3.  TAPE',I3,' WRITTEN')
      END
```

```
      SUBROUTINE INISHL
C
C     ON NHM3/INISHL3
C
C     THIS ROUTINE INITIALIZES PROGRAM 3
C
      PARAMETER (MXMU=10, MXPHI=24)
      DIMENSION FMU(MXMU),BNDMU(MXMU),BNDPHI(MXPHI),
     1          OMEGA(MXMU),DELTMU(MXMU)
      COMMON/CPHI/ PHI(MXPHI)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DATA NURAX/22/, NUOUT/30/
C
      READ(5,*) IDBUG
C
C     READ HEADER RECORD OF ONE OF THE GEOMETRIC ARRAYS
C
      REWIND NURAX
      READ(NURAX) NUNIT,NRAYQD,IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,
     1  DELTMU
      REWIND NURAX
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NL = NPHI/2
      IMISC(3) = NL
      IMISC(9) = IDBUG
      NRHAT = NMU*(NL+1)
      IMISC(10) = NRHAT
      NCHAT = NMU*((NL+2)/2)
      IMISC(11) = NCHAT
      WNDSPD = FMISC(15)
      REFR = FMISC(18)
C
      WRITE(6,300) NMU,NPHI,NL,WNDSPD,REFR
C
C     WRITE HEADER RECORDS ON OUTPUT FILE
C
      REWIND NUOUT
      WRITE(NUOUT) NUOUT,IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
C
C     FORMATS
C
  300 FORMAT(1H1,' PROGRAM 3 OF THE NATURAL HYDROSOL MODEL'//
     11H0,' COMPUTATION OF UPPER BOUNDARY SPECTRAL REFLECTANCE AND TRANS
     2MITTANCE ARRAYS'///1H ,'     NMU =',I3//1H ,'    NPHI =',I3//1H ,
     3'      NL =',I3//1H ,' WNDSPD =',F7.3//1H ,'    REFR =',F6.3)
      END
```

```
      SUBROUTINE FULLRT(RT,NMU,NPHI,IDRT)
C
C     ON NHM3/FULLRT
C
C     THIS ROUTINE CREATES A FULL (SQUARE) GEOMETRIC R OR T ARRAY FROM
C     THE "TOP HALF" DESCRIBED IN SECTION 12B.  USE OF THE FULL ARRAY
C     MEANS THAT NO SPECIAL INDEXING CALCULATIONS (SEE PAGE 191) NEED
C     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
C     RESET THE POLAR CAP OUTPUT FOR THE BOTTOM HALF (ZERO VALUES
C     CAME FROM THE B-BLOCK, SEE PAGE 190)
C
      DO 110 IROW=NHALF+1,NFULL
  110 RT(IROW,NMU) = RT(IROW-NHALF,NMU)
C
C     RE-ZERO THE POLAR CAP OUTPUT COLUMN AT PHI = 180, WHICH HAS
C     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
C     ON NHM3/RTSPEC
C
C     THIS ROUTINE FIRST COMPUTES THE SPECTRAL AMPLITUDES FROM THE
C     VARIOUS SPECIAL CASES, 5.31C TO 5.36, GIVEN RT = R OR T IN
C     GEOMETRIC FORM.
C
C     THE AMPLITUDES, RTHAT1 = RHAT1 OR THAT1 AND RTHAT2 = RHAT2 OR THAT2,
C     ARE STORED ON THE COMPRESSED SPECTRAL ARRAY FORMAT OF (12.4).
C
C     THE SPECTRAL AMPLITUDES ARE THEN CHECKED USING RAYLEIGH'S EQUALITY 4.17
C
C     FINALLY, THE MATRIX ELEMENTS DEFINED BY (5.41) AND (5.43) ARE
C     COMPUTED FROM THE ARRAYS OF AMPLITUDES.
C
C     N.B.  IN THIS ROUTINE, K AND L ARE REVERSED FROM THE NOTATION
C     USED IN THE TECH REPORT ERL-PMEL-75.
C
      DIMENSION PHI(1), RT(IDRT,1), RTHAT1(ID1HAT,1),RTHAT2(ID1HAT,1)
      COMMON/CMISC/ IMISC(20)
C
```

```
      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
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 IR=1,NMU
C
      DO 102 IU=1,NMU
C
C     STORAGE INDICES FOR SPECTRAL ARRAYS
      IUS = NMU*L + IR
      IVS = NMU*K + IU - NMU*((K+L)/2 - L/2)
C
      IF(IR.LT.NMU .AND. IU.LT.NMU) THEN
C
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))
  204 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
C     SPECIAL CASES FOR THE POLAR CAPS
C
      ELSEIF(IR.EQ.NMU .AND. IU.LT.NMU) THEN
C
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
  200 SUM1 = SUM1 + RT(IRT,NMU*(IV-1)+IU)*COS(AK*PHI(IV))
      RTHAT1(IUS,IVS) = SUM1/EPSK
      RTHAT2(IUS,IVS) = 0.
      ELSE
      RTHAT1(IUS,IVS) = 0.
      RTHAT2(IUS,IVS) = 0.
      ENDIF
C
```

```
      ELSEIF(IR.LT.NMU .AND. IU.EQ.NMU) THEN
C
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
  202 SUM1 = SUM1 + RT(NMU*(IS-1)+IR,JRT)*COS(AL*PHI(IS))
      RTHAT1(IUS,IVS) = SUM1/EPSL
      RTHAT2(IUS,IVS) = 0.
      ELSE
      RTHAT1(IUS,IVS) = 0.
      RTHAT2(IUS,IVS) = 0.
      ENDIF
C
      ELSEIF(IR.EQ.NMU .AND. IU.EQ.NMU) THEN
C
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
      RTHAT1(IUS,IVS) = RT(NMU,NMU)
      RTHAT2(IUS,IVS) = 0.
      ELSE
      RTHAT1(IUS,IVS) = 0.
      RTHAT2(IUS,IVS) = 0.
      ENDIF
C
      ENDIF
C
  102 CONTINUE
  100 CONTINUE
C
C     CHECK THE COMPUTED SPECTRAL AMPLITUDES
C
      IF(IDBUG.NE.0) THEN
      IPRHAT = 40
      IPCHAT = 20
      CALL P2ARAY(RTHAT1,IPRHAT,IPCHAT,ID1HAT,2,
     1' THE SPECTRAL AMPLITUDES RTHAT1')
      CALL P2ARAY(RTHAT2,IPRHAT,IPCHAT,ID1HAT,2,
     1' THE SPECTRAL AMPLITUDES RTHAT2')
      CALL SPECHK(RT,IDRT,RTHAT1,RTHAT2,ID1HAT)
      ENDIF
C
C     CONVERT THE SPECTRAL AMPLITUDES TO THE SPECTRAL ARRAYS DEFINED BY
C     5.41 AND 5.43.  THE ARRAY ELEMENTS ARE THE AMPLITUDES
C     MULTIPLIED BY FACTORS OF 1, NL OR NPHI, AS SEEN IN TABLES 1 AND 2
C     ON PAGES 90 AND 91.
C
      EPSL = FLOAT(NPHI)
      DO 300 IROW=1,NMU-1
      DO 300 JCOL=1,NCHAT
  300 RTHAT1(IROW,JCOL) = EPSL*RTHAT1(IROW,JCOL)
C
      EPSL = FLOAT(NL)
      DO 302 IROW=NMU+1,NRHAT-NMU
      DO 302 JCOL=1,NCHAT
      RTHAT1(IROW,JCOL) = EPSL*RTHAT1(IROW,JCOL)
  302 RTHAT2(IROW,JCOL) = EPSL*RTHAT2(IROW,JCOL)
C
      EPSL = FLOAT(NPHI)
      DO 304 IROW=NRHAT-NMU+1,NRHAT-1
      DO 304 JCOL=1,NCHAT
  304 RTHAT1(IROW,JCOL) = EPSL*RTHAT1(IROW,JCOL)
C
      RETURN
      END
```

```
      SUBROUTINE SPECHK(RT,IDRT,RTHAT1,RTHAT2,ID1HAT)
C
C     ON NHM3/SPECHK
C
C     THIS ROUTINE CHECKS THE COMPUTED SPECTRAL R AND T AMPLITUDES
C     BY SEEING IF THE WEIGHTED SUM OF THE SPECTRAL AMPLITUDES SQUARED EQUALS
C     THE SUM OF THE GEOMETRIC ELEMENTS SQUARED (RAYLEIGH'S EQUALITY, 4.17)
C
C     THIS CHECK HOLDS ONLY FOR NON-POLAR QUADS.
C
      PARAMETER(MXMU=10)
      DIMENSION RT(IDRT,1),RTHAT1(ID1HAT,1),RTHAT2(ID1HAT,1)
      DIMENSION GEOSUM(MXMU,MXMU),SPCSUM(MXMU,MXMU)
      COMMON/CMISC/ IMISC(20)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NL = IMISC(3)
      NRTGEO = NMU*NPHI
C
      DO 100 I=1,NMU-1
      DO 100 J=1,NMU-1
C
C     COMPUTE THE SUM OF SQUARES OF THE GEOMETRIC ARRAY ELEMENTS
C
      SUM = 0.
      DO 110 IROW=I,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(1H ,' RT(NMU,NMU) =',F10.5)
C
C     COMPUTE THE WEIGHTED SUM OF SQUARES OF THE SPECTRAL AMPLITUDES.
C     THE AMPLITUDES ARE STORED ON THE ARRAY FORMAT OF (12.4).
C
      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
C
      DO 120 L=0,NL
C
      IF(MOD(K+L,2).NE.0) GO TO 120
C
      IF(L.EQ.0 .OR. L.EQ.NL) THEN
      EPSL = FLOAT(NPHI)
      GAML = 0.
      ELSE
      EPSL = FLOAT(NL)
      GAML = FLOAT(NL)
      ENDIF
C
C     COMPUTE ROW AND COLUMN INDICES OF THE COMPRESSED AMPLITUDE ARRAYS,
C     ARRAYS, BY (12.5)
C
      IROW = I + NMU*K
      ICOL = J + NMU*L - NMU*((K+L)/2 - K/2)
C
      SUM = SUM + EPSK*EPSL*RTHAT1(IROW,ICOL)**2 +
     1            GAMK*GAML*RTHAT2(IROW,ICOL)**2
  120 CONTINUE
C
      SPCSUM(I,J) = SUM
C
  100 CONTINUE
C
      CALL P2ARAY(GEOSUM,NMU-1,NMU-1,MXMU,2,
     1'SUMS OF SQUARES OF THE NON-POLAR GEOMETRIC R/T ARRAY ELEMENTS')
      CALL P2ARAY(SPCSUM,NMU-1,NMU-1,MXMU,2,
     1'RAYLEIGH SUMS OF SQUARES OF THE NON-POLAR SPECTRAL R/T AMPLITUDES
     2')
      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 $\neq 0$ 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).

parameter	value in listed code	definition
MXY	30	the maximum number of optical depths y_j, $j = 1,\cdots,$YOUT, at which the final output is desired (see 75/Fig. 6)
MXSIGY	3	the maximum number of optical depths y_i, $i = 1,\cdots,$YOP, at which the inherent optical properties are specified (see 75/Fig. 6)

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.

Record 2: IDBUG, 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 n_μ 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 n_ϕ in 75/11.3. Use values like those for NUQB.

INCBAS the factor for increasing the base numbers of subcells (n_μ and n_ϕ) 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, n_μ 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 r_- = RFLBOT (see 75/3.26)

	= 1	if the bottom is infinitely deep, and the water is homogeneous below depth z (see 75/§10).
RFLBOT		The bottom reflectance (used only if IBOTM = 0). $0.0 \leq$ RFLBOT ≤ 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 ≤ MXY).
YOUT(1) ⋮ YOUT(NY)		the depths where output is desired. Always set YOUT(1) = 0.0 ≡ 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, and 20.0.

Here YOUT(1) ≡ x = 0.0, YOUT(2) = 0.5,···,YOUT(NY) ≡ 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 ≤ 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 W m^{-2} nm^{-1}.
THETAS	The polar angle, θ_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.

PHIS	The azimuthal angle, ϕ_s, in degrees of the sun's location, measured counterclockwise from ϕ_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.

internal name	external name	description
NUSRT	TAPE30	The file of spectral $\hat{r}$ and $\hat{t}$ arrays for the air-water surface, from Program 3.
NUPHAS	TAPE39	The file containing the quad-averaged phase function. It is written if ICPHAS $\neq 0$ and read if ICPHAS = 0.
NUOUT	TAPE40	The file containing the radiance amplitudes (and other information) generated by Program 4.
NUSCR1 NUSCR2 NUSCR3	TAPE45 TAPE46 TAPE47	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 $R_2(y,b;\ell)$.

D. Code Listing

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE30,TAPE39,
     1       TAPE40,TAPE45,TAPE46,TAPE47)
C
C     ON NHM4/MAIN4
C
C     +++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                 +
C     +  THIS IS PROGRAM 4 OF THE NATURAL HYDROSOL MODEL +
C     +                                                 +
C     +++++++++++++++++++++++++++++++++++++++++++++++++++
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXL=MXPHI/2, MXALGP=MXMU*(MXL+1), MXAMP=2*MXMU*(MXL+1))
      PARAMETER(MXRRTH=MXMU*(MXL+1), MXCRTH=MXMU*((MXL+2)/2))
      PARAMETER(MXWERK=MXMU*MXMU*(1 + 3*(MXL+1)*(1 + (MXL+2)/2)))
C
      COMMON/CAMP/ AAM(MXAMP),AAP(MXAMP),AYM(MXAMP,MXY),AYP(MXAMP,MXY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CRADO/ JMUO(MXSRC),KPHIO(MXSRC),RADO(MXSRC)
      COMMON/CRTR/ RYX(MXMU,MXMU,MXY),TXY(MXMU,MXMU,MXY),
     1             R1YB(MXMU,MXMU,MXY),R2YB(MXMU,MXMU,MXY)
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1 BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CRTSIG/ RHOHAT(MXMU,MXMU,MXSIGY),TAUHAT(MXMU,MXMU,MXSIGY),
     1  ALGPP(MXMU,MXALGP,MXSIGY),ALGPM(MXMU,MXALGP,MXSIGY)
      COMMON/CBOTBC/ RHATZB(MXMU,MXMU)
      COMMON/CRTHAT/ THAT1(MXRRTH,MXCRTH),THAT2(MXRRTH,MXCRTH),
     1               RHAT1(MXRRTH,MXCRTH),RHAT2(MXRRTH,MXCRTH)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      COMMON/CWORK/ WERK(MXWERK)
C
      DATA NUSRT/30/, NUOUT/40/, NUSCR1,NUSCR2,NUSCR3/45,46,47/
C
C     **********  INITIALIZATION  **********
C
C     READ THE INPUT DATA
C
      CALL INISHL
C
      NSIGY = IMISC(5)
      IMISC(18) = NUSCR1
      IMISC(19) = NUSCR2
      IMISC(20) = NUSCR3
C
      NMU = IMISC(1)
      NL = IMISC(3)
      NY = IMISC(4)
      IDBUG2 = IMISC(9)
      NRHAT = IMISC(10)
      NCHAT = IMISC(11)
      NRAMP = 2*NRHAT
C
C     INITIAL LOAD OF THE SPECTRAL STORAGE ARRAYS
C        THAT1(A,X) INTO THAT1
C        THAT2(A,X) INTO THAT2
C        RHAT1(X,A) INTO RHAT1
C        RHAT2(X,A) INTO RHAT2
C
      DO 100 J=1,NCHAT
  100 READ(NUSRT) (THAT1(I,J),I=1,NRHAT)
      DO 101 J=1,NCHAT
  101 READ(NUSRT) (THAT2(I,J),I=1,NRHAT)
      DO 102 J=1,NCHAT
  102 READ(NUSRT) (RHAT1(I,J),I=1,NRHAT)
      DO 103 J=1,NCHAT
  103 READ(NUSRT) (RHAT2(I,J),I=1,NRHAT)
```

```
C
      IF(IDBUG2.GT.0) THEN
      CALL P2ARAY(THAT1,2*NMU,NMU,MXRRTH,2,'THAT1(A,X) AS LOADED')
      CALL P2ARAY(THAT2,2*NMU,NMU,MXRRTH,2,'THAT2(A,X) AS LOADED')
      CALL P2ARAY(RHAT1,2*NMU,NMU,MXRRTH,2,'RHAT1(X,A) AS LOADED')
      CALL P2ARAY(RHAT2,2*NMU,NMU,MXRRTH,2,'RHAT2(X,A) AS LOADED')
      ENDIF
C
C     ***********  BEGIN COMPUTATIONS  ***********
C
C     COMPUTE THE DIRECT BEAM AMPLITUDES A0(Y,-) AT ALL LEVELS Y = A, X, ..., Z
C     FROM THE QUAD-AVERAGED SKY RADIANCES (WHICH ARE STORED IN /CWORK/)
C
      CALL AMPA0
C     /CWORK/ IS NOW ENTIRELY FREE
C
C     SAVE A0(Y,-) (STORED IN AAM) AT ALL Y LEVELS
      WRITE(NUOUT) (AAM(I),I=1,NRAMP)
      DO 150 J=1,NY
  150 WRITE(NUOUT) (AYM(I,J),I=1,NRAMP)
      IF(IDBUG2.NE.0) THEN
      WRITE(6,1038)
      CALL PNTAMP(Y,AAM,AYM,MXAMP)
      ENDIF
C
C     COMPUTE THE INTERIOR TRANSFER FUNCTIONS BY INTEGRATION OF THE
C     RICATTI EQUATIONS
C     EACH L MODE IS INTEGRATED SEPARATELY
C
      DO 200 L=0,NL
C
C     SET DEBUGGING OUTPUT FOR SELECTED L VALUES
      IF(IDBUG2.GT.0) THEN
           IF(L.LE.1 .OR. L.GE.NL-1) THEN
           IDBUG = IDBUG2
           ELSE
           IDBUG = 0
           ENDIF
      ELSE
      IDBUG = IDBUG2
      ENDIF
      IMISC(9) = IDBUG
C
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
C     ARE GIVEN
C
      CALL RHOTAU(L)
      IF(IDBUG.GT.0) THEN
      WRITE(6,202) L
      CALL P3ARAY(RHOHAT,NMU,NMU,NSIGY,MXMU,MXMU,2,'RHOHAT(L)')
      CALL P3ARAY(TAUHAT,NMU,NMU,NSIGY,MXMU,MXMU,2,'TAUHAT(L)')
      ENDIF
C
C++++ SOLUTION STEP 2
C     COMPUTE RHAT1(Z,B) FOR THE DESIRED BOTTOM BOUNDARY CONDITION
C
      CALL BOTMBC(L)
      IF(IDBUG.GT.0) CALL P2ARAY(RHATZB,NMU,NMU,MXMU,2,'RHAT1(Z,B,L)')
C
C++++ SOLUTION STEPS 3 AND 4
C     INTEGRATE THE RICATTI EQUATIONS TO GET R(Y,X), T(X,Y), AND RP(Y,B)
C
      CALL RICATI(L)
C
C     WRITE R(Y,X), T(X,Y) AND RP(Y,B) FOR THIS L VALUE TO SCRATCH FILES
C
      DO 220 IY=1,NY
      WRITE(NUSCR1) ((RYX(I,J,IY),I=1,NMU),J=1,NMU)
      WRITE(NUSCR2) ((TXY(I,J,IY),I=1,NMU),J=1,NMU)
      WRITE(NUSCR3) ((R1YB(I,J,IY),I=1,NMU),J=1,NMU)
  220 WRITE(NUSCR3) ((R2YB(I,J,IY),I=1,NMU),J=1,NMU)
      IF(IDBUG.EQ.2) THEN
      CALL P3ARAY(RYX,NMU,NMU,NY,MXMU,MXMU,2,'R(Y,X,L)')
      CALL P3ARAY(TXY,NMU,NMU,NY,MXMU,MXMU,2,'T(X,Y,L)')
      CALL P3ARAY(R1YB,NMU,NMU,NY,MXMU,MXMU,2,'R1(Y,B,L)')
      CALL P3ARAY(R2YB,NMU,NMU,NY,MXMU,MXMU,2,'R2(Y,B,L)')
      ENDIF
```

```
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(Y,-) AND A(Y,+), X .LT. Y .LE. Z
C
      CALL AMPINT
C
C     FINAL LOAD OF SPECTRAL STORAGE ARRAYS
C        THAT1(X,A) INTO THAT1
C        THAT2(X,A) INTO THAT2
C        RHAT1(A,X) INTO RHAT1
C        RHAT2(A,X) INTO RHAT2
C
      DO 400 J=1,NCHAT
  400 READ(NUSRT) (THAT1(I,J),I=1,NRHAT)
      DO 401 J=1,NCHAT
  401 READ(NUSRT) (THAT2(I,J),I=1,NRHAT)
      DO 402 J=1,NCHAT
  402 READ(NUSRT) (RHAT1(I,J),I=1,NRHAT)
      DO 403 J=1,NCHAT
  403 READ(NUSRT) (RHAT2(I,J),I=1,NRHAT)
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(RHAT1,2*NMU,NMU,MXRRTH,2,'RHAT1(A,X) AS LOADED')
      CALL P2ARAY(RHAT2,2*NMU,NMU,MXRRTH,2,'RHAT2(A,X) AS LOADED')
      ENDIF
C
C++++ SOLUTION STEP 9
C     COMPUTE THE AMPLITUDE A(A,+)
C
      CALL AMPAP
C     WERK(1) NOW CONTAINS A0(A,+), THE REFLECTED DIRECT BEAM
C
C     **********  END OF COMPUTATIONS  **********
C
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,NY
  450 WRITE(NUOUT) (AYM(I,J),I=1,NRAMP)
      DO 451 J=1,NY
  451 WRITE(NUOUT) (AYP(I,J),I=1,NRAMP)
      ENDFILE NUOUT
C
      IF(IDBUG2.NE.0) THEN
      WRITE(6,1039)
      CALL PNTAMP(Y,WERK,1.E201,MXAMP)
      WRITE(6,1040)
      CALL PNTAMP(Y,AAM,AYM,MXAMP)
      WRITE(6,1042)
      CALL PNTAMP(Y,AAP,AYP,MXAMP)
      ENDIF
C
      WRITE(6,500) NUOUT
C
  202 FORMAT(1H1,' +++++ BEGINNING THE L =',I3,' LOOP +++++')
  500 FORMAT(1H0,' NORMAL EXIT FROM PROGRAM 4.  TAPE',I2,' WRITTEN.')
 1038 FORMAT(1H1,' THE DOWNWARD DIRECT BEAM RADIANCE AMPLITUDES ARE'//
     1 11X,'MU',7X,'A0(A,-)',8X,'A0(Y,-)')
 1039 FORMAT(1H1,' THE UPWARD DIRECT BEAM RADIANCE AMPLITUDES ARE'//
     1  11X,'MU',7X,'A0(A,+)')
 1040 FORMAT(1H1,' THE DOWNWARD TOTAL RADIANCE AMPLITUDES ARE'//
     1  11X,'MU',7X,'A(A,-)',9X,'A(Y,-)')
 1042 FORMAT(1H1,' THE UPWARD TOTAL RADIANCE AMPLITUDES ARE'
     1//11X,'MU',7X,'A(A,+)',9X,'A(Y,+)')
      END
```

```
      SUBROUTINE INISHL
C
C     ON NHM4/INISHL4
C
C     THIS ROUTINE INITIALIZES PROGRAM 4 OF THE NHM.
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXGEOP=MXMU*(MXPHI/2+1))
C
      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),
     1 BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CWORK/ RADSKY(MXMU,MXPHI),PHASE(2701,MXSIGY),PSITAB(2701)
C
      DIMENSION ITITLE(10)
C
      DATA NUSRT,NUPHAS,NUOUT/30,39,40/
C     KINV......THE NUMBER OF TERMS IN THE SUM (7.4)
C     TOL.......THE TOLERANCE FOR THE RICCATI EQUATION SOLVER (IMSL ROUTINE DVERK)
      DATA KINV/3/, TOL/1.0E-8/
C
C     READ INPUT RECORDS
C
C  RECORD 1:
C     ITITLE....A RUN TITLE, UP TO 80 CHARACTERS
C
C  RECORD 2:
C     IDBUG.....0 FOR NO INTERMEDIATE OUTPUT (PRODUCTION RUNS)
C               1 FOR MINIMAL OUTPUT FOR CHECKING (RECOMMENDED)
C               2 FOR FULL DEBUGGING OUTPUT
C     WAVENM....THE WAVELENGTH IN NANOMETERS, ONE OF THE 13 VALUES 400.,425.,....,
C               675.,700. (SEE PHASEF FOR THE PELAGOS SEA)
C     ABSORB....IF.GE.0., THEN THE ABSORBTION COEF A IS RESET TO THIS VALUE
C               (USED FOR LAKE LIMNE RUNS TO VARY A WITH WAVELENGTH)
C
C  RECORD 3:
C     ICPHAS....0 IF THE QUAD-AVERAGED PHASE FUNCTIONS ARE TO BE READ
C                 FROM UNIT NUPHAS
C           .NE.0 IF THE QUAD-AVERAGED PHASE FUNCTIONS ARE TO BE COMPUTED
C                 (BY SUBROUTINE QAPHAS) AND STORED ON UNIT NUPHAS
C                 IF ICPHAS.LT.0, THE RUN STOPS AFTER NUPHAS IS WRITTEN
C                 IF ICPHAS.GT.0, THE RUN CONTINUES, AND COMPUTES
C                                 AMPLITUDES
C     NUQB,NVQB.THE BASE NUMBERS OF SUBCELLS (IN THE MU AND PHI DIRECTIONS)
C               USED TO DISCRETIZE THE PHASE FUNCTION VIA EQ. 11.3
C               (USED ONLY IF ICPHAS.NE.0)
C     INCBAS....THE FACTOR FOR INCREASING THE BASE NUMBER OF SUBCELLS
C               FOR QUAD PAIRS WHICH INCLUDE FORWARD SCATTERING
C
C  RECORD 4:
C     IBOTM.....0 FOR A MATTE BOTTOM AT Y = Z, OF REFLECTANCE R- = RFLBOT
C               1 FOR AN INFINITELY DEEP BOTTOM, WITH HOMOGENEOUS WATER
C                 BELOW DEPTH Y = Z
C     RFLBOT....THE BOTTOM REFLECTANCE, (USED ONLY IF IBOTM = 0)
C               0.0 .LE. RFLBOT .LE. 1.0
C
C  RECORD 5:
C     IYOP......0 IF YOUT AS READ CONTAINS GEOMETRIC DEPTHS IN METERS
C                 (USE FOR UNIFORM WATER ONLY, AS OF 30 JUNE 86)
C               1 IF YOUT AS READ CONTAINS OPTICAL DEPTHS
C     NY........THE NUMBER OF Y LEVELS WHERE OUTPUT IS DESIRED
C     YOUT(1),...,YOUT(NY)...THE DEPTHS WHERE OUTPUT IS DESIRED
C
C  RECORD 6:
C     RSKY......THE RATIO OF SKY TO TOTAL INPUT SCALAR IRRADIANCE
C               RSKY = 0. FOR A BLACK SKY (SUN ONLY), RSKY = 1.0 FOR A
C               BACKGROUND SKY ONLY (NO SUN)
C     CARD......THE CARDIOIDAL PARAMETER FOR THE SKY RADIANCE
C               DISTRIBUTION.  CARD = 0. FOR A UNIFORM SKY, CARD = 2. FOR A
C               CARDIOIDAL SKY
```

```
C      SHTOTL....THE TOTAL (SKY + SUN) SCALAR IRRADIANCE ON THE WATER
C                SURFACE, WATTS PER SQUARE METER
C      THETAS, PHIS...THE SKY (SOURCE) LOCATION OF THE SUN, IN DEGREES.
C                THETAS IS 0. AT THE ZENITH, 90. AT THE HORIZON.  PHI IS
C                MEASURED COUNTERCLOCKWISE FROM PHI = 0. IN THE
C                DOWNWIND DIRECTION
C
       READ(5,1004) ITITLE
       READ(5,*) IDBUG,WAVENM,ABSORB
       READ(5,*) ICPHAS,NUQB,NVQB,INCBAS
       READ(5,*) IBOTM,RFLBOT
       READ(5,*) IYOP,NY,(YOUT(IY),IY=1,NY)
       READ(5,*) RSKY,CARD,SHTOTL,THETAS,PHIS
C
C      READ HEADER RECORDS FROM THE SPECTRAL DATA FILE, NUSRT
C
       REWIND NUSRT
       READ(NUSRT) NUNIT,IMISC,FMISC,FMU,PHI,BNDMU,BNDPHI,OMEGA,DELTMU
C
       NMU = IMISC(1)
       NPHI = IMISC(2)
       NL = IMISC(3)
       TWOPI = 2.0*FMISC(1)
       DEGRAD = FMISC(2)
       RADEG = FMISC(3)
       WNDSPD = FMISC(15)
       KCOL = NMU*(NL + 1)
C
       WRITE(6,1000)
       WRITE(6,1005) ITITLE
       WRITE(6,1010) NMU,NPHI,NY,NL,WNDSPD,WAVENM,KINV,TOL
       IF(ICPHAS.NE.0) WRITE(6,1014) NUQB,NVQB,INCBAS
       IF(IBOTM.EQ.0) WRITE(6,1030) RFLBOT
       IF(IBOTM.EQ.1) WRITE(6,1031)
C
       IMISC(4) = NY
       IMISC(8) = KINV
       IMISC(9) = IDBUG
       IMISC(12) = IBOTM
       FMISC(7) = TOL
       FMISC(13) = WAVENM
       FMISC(14) = RFLBOT
C
C      COMPUTE THE (INPUT) QUAD-AVERAGED RADIANCES FOR THE SKY
C
       CALL QASKY(RSKY,CARD,SHTOTL,THETAS,PHIS)
C      RADSKY IS IN /CWORK/ AND MUST BE SAVED UNTIL AMPAO IS CALLED IN MAIN
C
       IF(ICPHAS.NE.0) THEN
C
C      COMPUTE AND SAVE THE QUAD-AVERAGED PHASE FUNCTIONS
C
C      INITIALIZE THE POINT GEOMETRIC SCATTERING FUNCTION
C
C++++ NOTE:  MAKE SURE THE DESIRED VERSION OF PHASEF HAS BEEN LOADED
C      INTO THE EXECUTABLE ELEMENT (ABSOLUTE RUN FILE)
C
       XX = PHASEF(0.,0.)
       NSIGY = IMISC(5)
C
C      GENERATE A TABLE OF PHASE FUNCTION VALUES FOR LOOKUP IN QAPHAS
C      /CWORK/ IS USED TO HOLD THE TABLE OF PHASE VALUES
C
       DO 100 IY=1,NSIGY
       Y = YSIG(IY)
C
C      0 .LE. PSI .LE. 10 DEGREES, BY 0.01 DEGREE STEPS
       DPSI = DEGRAD*0.01
       DO 102 I=1,1001
       PSITAB(I) = FLOAT(I-1)*DPSI*RADEG
       COSPSI = COS(FLOAT(I-1)*DPSI)
  102  PHASE(I,IY) = PHASEF(Y,COSPSI)
C
```

```
C      10 .LT. PSI .LE. 180 DEGREES, BY 0.1 DEGREE STEPS
       DPSI = DEGRAD*0.1
       PSI0 = DEGRAD*10.0
       DO 100 I=1002,2701
       PSI0 = PSI0 + DPSI
       PSITAB(I) = PSI0*RADEG
       COSPSI = COS(PSI0)
  100  PHASE(I,IY) = PHASEF(Y,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) I,PSITAB(I),PHASE(I,1),I2,PSITAB(I2),PHASE(I2,1),
      1  I3,PSITAB(I3),PHASE(I3,1)
C
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
C      PSI = 0.01 TO 0.1
       IF(I.EQ.11) SUM01=SUM + PHASE(I,1)*SIN(PSITAB(I)*DEGRAD)*0.5*DPSI
C      PSI = 0.01 TO 1.0
       IF(I.EQ.101) SUM1=SUM + PHASE(I,1)*SIN(PSITAB(I)*DEGRAD)*0.5*DPSI
  120  SUM = SUM + PHASE(I,1)*SIN(PSITAB(I)*DEGRAD)*DPSI
       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
C      PSI = 0.01 TO 20.
       IF(I.EQ.1101) SUM20 = SUM + PHASE(1101,1)*0.5*DPSI
C      PSI = 0.01 TO 90.0
       IF(I.EQ.1801) SUM90 = SUM + PHASE(1801,1)*0.5*DPSI
  122  SUM = SUM + PHASE(I,1)*SIN(PSITAB(I)*DEGRAD)*DPSI
       SUM01 = TWOPI*SUM01
       SUM1 = TWOPI*SUM1
       SUM10 = TWOPI*SUM10
       SUM20 = TWOPI*SUM20
       SUM90 = TWOPI*SUM90
       SUM = TWOPI*SUM
       SUM980 = SUM - SUM90
       WRITE(6,1054) SUM01,SUM1,SUM10,SUM20,SUM90,SUM980,SUM
       ENDIF
C
C      COMPUTE THE QUAD-AVERAGED GEOMETRIC PHASE FUNCTION AS IN SECTION 11
C
       CALL QAPHAS(NUQB,NVQB,INCBAS)
C
C      STORE THE COMPUTED PHASE FUNCTIONS FOR LATER USE
C
       REWIND NUPHAS
       WRITE(NUPHAS) NSIGY,YSIG,ALBESS,TOTALS
       WRITE(NUPHAS) (((GEOPP(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
       WRITE(NUPHAS) (((GEOPM(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
       ENDFILE NUPHAS
       WRITE(6,1060) NUPHAS
       IF(ICPHAS.LT.0) STOP
C
       ELSE
C
C      READ EXISTING QUAD-AVERAGED PHASE FUNCTIONS FROM NUPHAS
C
       REWIND NUPHAS
       READ(NUPHAS) NSIGY,YSIG,ALBESS,TOTALS
       READ(NUPHAS) (((GEOPP(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
       READ(NUPHAS) (((GEOPM(I,J,K),I=1,NMU),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
       IF(ABSORB.GE.0.) THEN
       ALBESS(1) = TOTALS(1)/(TOTALS(1) + ABSORB)
       ENDIF
```

```
C
      ABSORB = TOTALS(1)*(1.0/ALBESS(1) - 1.0)
      WRITE(6,1070) ABSORB,TOTALS(1),ALBESS(1)
C
      IF(IDBUG.GT.0) THEN
      CALL P3ARAY(GEOPP,NMU,NMU,NSIGY,MXMU,MXGEOP,2,
     1'QUAD-AVERAGED P+(Y,R,1/U,V) AS LOADED')
      CALL P3ARAY(GEOPM,NMU,NMU,NSIGY,MXMU,MXGEOP,2,
     1'QUAD-AVERAGED P-(Y,R,1/U,V) AS LOADED')
      ENDIF
C
      ENDIF
C
      IF(IYOP.EQ.1) THEN
C
C     YOUT AS READ CONTAINS THE OPTICAL DEPTHS
C     COMPUTE THE GEOMETRIC DEPTHS CORRESPONDING TO THE OPTICAL DEPTHS
C     WHERE OUTPUT IS REQUESTED
C
      CALL Y2ZGEO
C
      ELSE
C
C     YOUT AS READ CONTAINS THE GEOMETRIC DEPTHS IN METERS.  COMPUTE
C     THE CORRESPONDING OPTICAL DEPTHS (UNIFORM WATER ONLY)
C
      ALFA = ABSORB + TOTALS(1)
      DO 200 IY=1,NY
      ZGEO(IY) = YOUT(IY)
  200 YOUT(IY) = ALFA*ZGEO(IY)
      ENDIF
C
      WRITE(6,1020) YOUT(1),YOUT(NY)
      WRITE(6,1025) (IY,YOUT(IY),ZGEO(IY),IY=1,NY)
C
C     WRITE HEADER RECORDS ONTO THE AMPLITUDE DATA FILE
C
      REWIND NUOUT
      WRITE(NUOUT) IMISC,FMISC,FMU,PHI,YOUT,BNDMU,BNDPHI,OMEGA,DELTMU,
     1  YSIG,ALBESS,TOTALS,ZGEO
      WRITE(NUOUT) (((GEOPP(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
      WRITE(NUOUT) (((GEOPM(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
C
      RETURN
C     NOW DONE WITH /CWORK/
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(1H1,' PROGRAM 4 OF THE NATURAL HYDROSOL MODEL'//
     1'  SOLUTION OF THE RADIATIVE TRANSFER EQUATION IN A PLANE-PARALLEL
     2 MEDIUM')
 1004 FORMAT(10A8)
 1005 FORMAT(//' RUN TITLE:  ',10A8)
 1010 FORMAT(//,' THE GRID PARAMETERS ARE'//T8,'NMU =',I3//
     1T7,'NPHI =',I3//T9,'NY =',I3//' OTHER PARAMETERS ARE'//
     2T9,'NL =',I3,' = HIGHEST WAVENUMBER L IN FOURIER ANALYSIS OF PHI'/
     3/T5,'WNDSPD =',F5.2,' M/SEC'//
     7T5,'LAMBDA =',F6.1,' NANOMETERS'//
     4T7,'KINV =',I3,' = HIGHEST POWER OF THE SERIES EXPANSION USED FOR
     5MATRIX INVERSION'//T3,'TOL =',1PE8.1,
     6' = ERROR TOLERANCE FOR RICCATI EQ. INTEGRATIONS')
 1014 FORMAT(//' THE PHASE FUNCTION IS QUAD-AVERAGED USING NUQB =',I3,
     1', NVQB =',I3,' AND INCBAS =',I3)
 1020 FORMAT(//'THE SLAB THICKNESS IS X =',F6.2,' .LE. Y .LE.',
     1 F6.2,' = Z OPTICAL DEPTHS')
 1025 FORMAT(//' OUTPUT IS AT THE FOLLOWING DEPTHS:'//
     1 ' Y INDEX  OPT DEPTH     GEO DEPTH (M)'//(I5,F12.3,F16.3))
 1030 FORMAT(//' THE BOTTOM BOUNDARY IS A MATTE SURFACE WITH REFLECTANCE
     1 R- =',F6.2)
 1031 FORMAT(//' THE BOTTOM BOUNDARY IS INFINITELY DEEP')
 1050 FORMAT(1H1,' SELECTED VALUES OF THE TABULATED PHASE FUNCTION'//
     14X,'INDEX      PHI      PHASE',7X,'INDEX      PHI      PHASE',7X,
     2'INDEX      PHI      PHASE'/)
 1052 FORMAT(3X,3(I5,0PF9.2,1PE13.5,3X))
 1054 FORMAT(1H0,' INTEGRALS OF 2*PI*PHASE(PSI)*SIN(PSI)*DPSI:'//
     1'  I(0.01-0.1) =',1PE13.6//'  I(0.01-1.0) =',E13.6//
     2'  I(0.01-10.) =',E13.6//'  I(0.01-20.) =',E13.6//
     3'  I(0.01-90.) =',E13.6//'  I(90.-180.) =',E13.6//
     4'  I(0.01-180) =',E13.6)
 1060 FORMAT(1H0,' TAPE',I2,' OF QUAD-AVERAGED PHASE FUNCTIONS WRITTEN')
 1070 FORMAT(//' FOR THIS RUN, A =',F6.3,3X,'S =',F6.3,3X,
     1'ALBEDO =',F5.3)
      END
```

```
      SUBROUTINE ADIPAK(X,Y,IROW,NMU,L)
C
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 = NMU*(L+1)
      MLC = NMU*((L+2)/2)
      DO 99 I=1,MLR
      DO 99 J=1,MLC
   99 Y(I,J) = X(I,J)
C
C     ADD 1.0 TO THE DIAGONAL ELEMENTS
C
      LP1 = L+1
      DO 100 IXB=1,LP1
      JXB = (IXB+1)/2
      I1 = (IXB-1)*NMU
      J1 = (JXB-1)*NMU
      DO 100 I=1,NMU
  100 Y(I1+I,J1+I) = X(I1+I,J1+I) + 1.
C
      RETURN
      END
```

```
      SUBROUTINE AMPA0
C
C     ON NHM4/AMPA0
C
C     THIS ROUTINE FOURIER ANALYZES THE QUAD-AVERAGED SKY RADIANCES
C     RADSKY = N0(A,-) TO GENERATE THE DIRECT BEAM SPECTRAL AMPLITUDE
C     A0(A,-).  A0(A,-) IS THEN TRANSMITTED
C     THROUGH THE UPPER BOUNDARY TO GET A0(X,-).  A0(X,-) IS THEN
C     ATTENUATED EXPONENTIALLY TO GET A0(Y,-) AT ALL DEPTHS.
C
C     COSINE AMPLITUDES ARE IN A0AM(I), I=1,2,...,NRHAT
C     SINE   AMPLITUDES ARE IN A0AM(I + NRHAT)
C
C     SPECTRAL STORAGE ARRAYS MUST BE LOADED WITH
C        THAT1(A,X) IN THAT1
C        THAT2(A,X) IN THAT2
C
C     IN THIS ROUTINE, /CAMP/ IS USED TO STORE A0(Y,-), A.LE.Y.LE.Z
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
      PARAMETER(MXL=MXPHI/2, MXRRTH=MXMU*(MXL+1),
     1  MXCRTH=MXMU*((MXL+2)/2), MXAMP=2*MXRRTH)
C
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY)
      COMMON/CAMP/ A0AM(MXAMP),DUMMY(MXAMP),A0YM(MXAMP,MXY)
      COMMON/CRTHAT/ THAT1(MXRRTH,MXCRTH),THAT2(MXRRTH,MXCRTH)
      COMMON/CMISC/ IMISC(20)
      COMMON/CWORK/ RADSKY(MXMU,MXPHI)
C
```

```
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NL = IMISC(3)
      NY = IMISC(4)
      IDBUG = IMISC(9)
      NRHAT = IMISC(10)
      IF(IDBUG.EQ.2) THEN
      CALL P2ARAY(THAT1,2*NMU,NMU,MXRRTH,2,'THAT1(A,X) IN AMPA0')
      CALL P2ARAY(THAT2,2*NMU,NMU,MXRRTH,2,'THAT2(A,X) IN AMPA0')
      CALL P2ARAY(RADSKY,NMU,NPHI,MXMU,2,'RADSKY IN AMPA0')
      ENDIF
C
C     LOOP OVER L AND MU TO DEFINE A0(A,-) VIA (4.8) AND (4.9)
C
C     LOOP OVER THE MU BANDS OTHER THAN THE POLAR CAP
C
      DO 100 I=1,NMU-1
C
C     DEFINE THE AMPLITUDES FOR EACH L VALUE FROM (4.8) AND (4.9)
C
C     L = 0 SPECIAL CASE
      SUM = 0.
      DO 310 J=1,NPHI
  310 SUM = SUM + RADSKY(I,J)
      A0AM(I) = SUM/FLOAT(NPHI)
      A0AM(I+NRHAT) = 0.
C
C     L = NL SPECIAL CASE
      SUM = 0.
      DO 320 J=1,NPHI
  320 SUM = SUM + RADSKY(I,J)*COS(FLOAT(NL)*PHI(J))
      A0AM(NMU*NL+I) = SUM/FLOAT(NPHI)
      A0AM(NMU*NL+I+NRHAT) = 0.
C
C     0 .LT. L .LT. NL GENERAL CASE
C
      DO 330 L=1,NL-1
      SUM1 = 0.
      SUM2 = 0.
      DO 332 J=1,NPHI
      SUM1 = SUM1 + RADSKY(I,J)*COS(FLOAT(L)*PHI(J))
  332 SUM2 = SUM2 + RADSKY(I,J)*SIN(FLOAT(L)*PHI(J))
      A0AM(NMU*L+I) = SUM1/FLOAT(NL)
  330 A0AM(NMU*L+I + NRHAT) = SUM2/FLOAT(NL)
C
  100 CONTINUE
C
C     POLAR CAP SPECIAL CASE
C
C     THE COSINE AMP IS JUST THE VALUE OF THE POLAR CAP QUAD-AVERAGED
C     RADIANCE, EQ. (5.4)
C
      A0AM(NMU) = RADSKY(NMU,1)
      A0AM(NMU + NRHAT) = 0.
      DO 340 L=1,NL
      A0AM(NMU*L+NMU) = 0.
  340 A0AM(NMU*L+NMU+NRHAT) = 0.
C
C     TRANSMIT A0(A,-) THROUGH THE UPPER BOUNDARY VIA 6.55 TO GET
C     A0(X,-) = A0YM(*,1)  (NOTE IN 6.55 THAT A0(X,+) = 0.  SEE PAGE 137)
C
      CALL RFMPAK(A0AM,THAT1,A0YM,MXRRTH,NMU,NL)
      CALL RFMPAK(A0AM(NRHAT+1),THAT2,A0YM(NRHAT+1,1),MXRRTH,NMU,NL)
C
C     TRANSMIT A0(X,-) TO ALL LOWER Y LEVELS, X .GT. Y .GE. Z, VIA 8.22
C
      IROW = 0
      DO 400 L=0,NL
      DO 400 J=1,NMU
      IROW = IROW + 1
      DO 400 IY=2,NY
      TEMP = EXP((Y(1) - Y(IY))/FMU(J))
      A0YM(IROW,IY) = TEMP*A0YM(IROW,1)
  400 A0YM(IROW+NRHAT,IY) = TEMP*A0YM(IROW+NRHAT,1)
C
      RETURN
      END
```

```
      SUBROUTINE AMPAP
C
C     ON NHM4/AMPAP
C
C     THIS ROUTINE DEFINES A(A,+) USING 6.56
C
C     SPECTRAL STORAGE ARRAYS MUST BE LOADED WITH
C        THAT1(X,A) IN THAT1
C        THAT2(X,A) IN THAT2
C        RHAT1(A,X) IN RHAT1
C        RHAT2(A,X) IN RHAT2
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
      PARAMETER(MXL=MXPHI/2, MXRRTH=MXMU*(MXL+1),
     1  MXCRTH=MXMU*((MXL+2)/2), MXAMP=2*MXRRTH)
C
      COMMON/CAMP/ AOAM(MXAMP),AAP(MXAMP),AYM(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/ AOAP(MXAMP),TEMP1(MXRRTH),TEMP2(MXRRTH),
     1 RHAT(MXRRTH,MXCRTH),THAT(MXRRTH,MXCRTH)
C
      DIMENSION AXP(MXAMP)
      EQUIVALENCE (AXP(1),AYP(1,1))
C
      NMU = IMISC(1)
      NL = IMISC(3)
      IDBUG = IMISC(9)
      NRHAT = IMISC(10)
      NCHAT = IMISC(11)
C
C     P = 1 (COSINE AMPLITUDES)
C
      IP = 1
C
  999 CONTINUE
      IPOFF = NRHAT*(IP-1)
C
C     LOAD THATP(X,A) INTO THAT
C     LOAD RHATP(A,X) INTO RHAT
C
      IF(IP.EQ.1) THEN
      DO 800 J=1,NCHAT
      DO 800 I=1,NRHAT
      RHAT(I,J) = RHAT1(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 THAT(I,J) = THAT2(I,J)
      ENDIF
C
      IF(IDBUG.GE.2) THEN
      WRITE(6,310) IP
      CALL P2ARAY(THAT,2*NMU,NMU,MXRRTH,2,'THATP(X,A) IN AMPAP')
      CALL P2ARAY(RHAT,2*NMU,NMU,MXRRTH,2,'RHATP(A,X) IN AMPAP')
      ENDIF
C
C     EVALUATE 6.56 AND SAVE AOAP FOR WRITING ONTO NUOUT
C
      CALL RFMPAK(AXP(IPOFF+1),THAT,TEMP1,MXRRTH,NMU,NL)
      CALL RFMPAK(AOAM(IPOFF+1),RHAT,TEMP2,MXRRTH,NMU,NL)
      DO 100 I=1,NRHAT
      AOAP(I+IPOFF) = TEMP2(I)
  100 AAP(I+IPOFF) = TEMP1(I) + TEMP2(I)
C
      IF(IP.GT.1) RETURN
C
C     REPEAT FOR P = 2 (SINE AMPLITUDES)
      IP = 2
      GO TO 999
  310 FORMAT(1H1,' SUBROUTINE AMPAP, P =',I2)
C
      END
```

```
      SUBROUTINE AMPINT
C
C     ON NHM4/AMPINT
C
C     THIS ROUTINE FINDS THE AMPLITUDES A(Y,-) AND A(Y,+) AT ALL INTERIOR
C     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             R1YB(MXMU,MXMU,MXY),R2YB(MXMU,MXMU,MXY)
      COMMON/CMISC/ IMISC(20)
      COMMON/CWORK/ TXYB(MXMU,MXMU),TEMP1(MXMU,MXMU),TEMP2(MXMU,MXMU),
     1              RPYB(MXMU,MXMU,MXY)
C
      DIMENSION AXM(MXAMP)
      EQUIVALENCE (AXM(1),AYM(1,1))
C
      DATA IDGT/10/, NUSCR1,NUSCR2,NUSCR3/45,46,47/
C
      NMU = IMISC(1)
      NL = IMISC(3)
      NY = IMISC(4)
      IDBUG2 = IMISC(9)
      NRHAT = IMISC(10)
C
      REWIND NUSCR1
      REWIND NUSCR2
      REWIND NUSCR3
C
      DO 100 L=0,NL
C
      IF(IDBUG2.EQ.2) THEN
           IF(L.LE.1 .OR. L.GE.NL-1) THEN
           IDBUG = 2
           ELSE
           IDBUG = 0
           ENDIF
      ELSE
      IDBUG = IDBUG2
      ENDIF
      LOFSET = NMU*L
C
C     READ IN RYX = R(Y,X,L), TXY = T(X,Y,L) AND RPYB = RP(Y,B,L) FOR ALL
C     Y LEVELS, FOR THIS L VALUE
C
      DO 300 IY=1,NY
      READ(NUSCR1) ((RYX(I,J,IY),I=1,NMU),J=1,NMU)
      READ(NUSCR2) ((TXY(I,J,IY),I=1,NMU),J=1,NMU)
      READ(NUSCR3) ((R1YB(I,J,IY),I=1,NMU),J=1,NMU)
  300 READ(NUSCR3) ((R2YB(I,J,IY),I=1,NMU),J=1,NMU)
C
      IF(IDBUG.GE.2) THEN
      WRITE(6,310) L
      CALL P3ARAY(RYX,NMU,NMU,NY,MXMU,MXMU,2,'R(Y,X,L)')
      CALL P3ARAY(TXY,NMU,NMU,NY,MXMU,MXMU,2,'T(X,Y,L)')
      CALL P3ARAY(R1YB,NMU,NMU,NY,MXMU,MXMU,2,'R1(Y,B,L)')
      CALL P3ARAY(R2YB,NMU,NMU,NY,MXMU,MXMU,2,'R2(Y,B,L)')
      ENDIF
C
C     INITIALIZE FOR COSINE AMPLITUDES
      IP = 1
C
  999 CONTINUE
C
C     LOAD RPYB FOR CURRENT P VALUE
      IF(IP.EQ.1) THEN
      IOFSET = LOFSET
      DO 205 IY=1,NY
      DO 205 J=1,NMU
      DO 205 I=1,NMU
  205 RPYB(I,J,IY) = R1YB(I,J,IY)
      ELSE
```

```
      IOFSET = LOFSET + NRHAT
      DO 206 IY=1,NY
      DO 206 J=1,NMU
      DO 206 I=1,NMU
  206 RPYB(I,J,IY) = R2YB(I,J,IY)
      ENDIF
C
C     COMPUTE THE AMPLITUDES AT EACH Y LEVEL
C
      DO 102 IY=2,NY
      IF(IDBUG.EQ.2) WRITE(6,311) IY,IP
C
C     COMPUTE TXYB = TP(X,Y,B,L) USING 6.33
C
C     COMPUTE TEMP1 = I - 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 + RPYB(I,K,IY)*RYX(K,J,IY)
      DELT = 0.
      IF(I.EQ.J) DELT = 1.
  210 TEMP1(I,J) = DELT - SUM
      IF(IDBUG.EQ.2) CALL P2ARAY(TEMP1,NMU,NMU,MXMU,2,
     1  'I - RP(Y,B) * R(Y,X)')
C
C     INVERT I - RP(Y,B) * R(Y,X)
      CALL LINV1F(TEMP1,NMU,MXMU,TEMP2,IDGT,TXYB,IER)
      IF(IDBUG.EQ.2) CALL P2ARAY(TEMP2,NMU,NMU,MXMU,2,
     1 '(I - RP(Y,B) * R(Y,X)) INVERSE')
C
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,IY)*TEMP2(K,J)
  220 TXYB(I,J) = SUM
      IF(IDBUG.EQ.2) CALL P2ARAY(TXYB,NMU,NMU,MXMU,2,
     1 'SCRIPT TP(X,Y,B,L)')
C
C     COMPUTE AP(Y,-) USING 7.6
C
      DO 230 J=1,NMU
      SUM = 0.
C
      DO 231 K=1,NMU
  231 SUM = SUM + AXM(K+IOFSET)*TXYB(K,J)
  230 AYM(J+IOFSET,IY) = SUM
C
C     COMPUTE A(Y,+) USING 7.7
C
      DO 240 J=1,NMU
      SUM = 0.
C
      DO 241 K=1,NMU
  241 SUM = SUM + AYM(K+IOFSET,IY)*RPYB(K,J,IY)
  240 AYP(J+IOFSET,IY) = SUM
C
  102 CONTINUE
C
C     REPEAT FOR THE SINE AMPLITUDES, IP = 2
C
      IP = IP + 1
      IF(IP.LT.3) GO TO 999
C
  100 CONTINUE
C
      IF(IDBUG.EQ.2) THEN
      CALL P2ARAY(AYM,2*NRHAT,NY,MXAMP,2,'A(Y,-)')
      CALL P2ARAY(AYP,2*NRHAT,NY,MXAMP,2,'A(Y,+)')
      ENDIF
  310 FORMAT(1H1,' SUBROUTINE AMPINT, L =',I3)
  311 FORMAT(1H0,' IY =',I2,3X,'IP =',I2)
C
      RETURN
      END
```

```
      SUBROUTINE AMPX
C
C     ON NHM4/AMPX
C
C     THIS ROUTINE COMPUTES A(X,-) AND A(X,+) USING 7.3 AND 7.5B
C
C     SPECTRAL STORAGE ARRAYS MUST BE LOADED WITH
C        THAT1(A,X) IN THAT1
C        THAT2(A,X) IN THAT2
C        RHAT1(X,A) IN RHAT1
C        RHAT2(X,A) IN RHAT2
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
      PARAMETER(MXL=MXPHI/2, MXRRTH=MXMU*(MXL+1),
     1  MXCRTH=MXMU*((MXL+2)/2), MXAMP=2*MXRRTH)
C
      COMMON/CAMP/ AOAM(MXAMP),AAP(MXAMP),AYM(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/ WORK(MXMU,MXMU),TEMP(MXRRTH,MXCRTH),
     1 RHAT(MXRRTH,MXCRTH),THAT(MXRRTH,MXCRTH),R1XBL(MXMU,MXMU,0:MXL),
     2 R2XBL(MXMU,MXMU,0:MXL),RXBL(MXMU,MXMU,0:MXL)
C
      DIMENSION AXM(MXAMP),AXP(MXAMP)
      EQUIVALENCE (AXM(1),AYM(1,1)),(AXP(1),AYP(1,1))
C
      NMU = IMISC(1)
      NL = IMISC(3)
      NY = IMISC(4)
      KINV = IMISC(8)
      IDBUG = IMISC(9)
      NRHAT = IMISC(10)
      NCHAT = IMISC(11)
      NUSCR3 = IMISC(20)
      IEVEN = (NL+2)/2
      IODD = (NL+1)/2
C
C     READ R1(X,B,L) AND R2(X,B,L) FROM SCRATCH FILE NUSCR3
C
      REWIND NUSCR3
      DO 100 L=0,NL
C
C     READ Y = X LEVEL
      READ(NUSCR3) ((R1XBL(I,J,L),I=1,NMU),J=1,NMU)
      READ(NUSCR3) ((R2XBL(I,J,L),I=1,NMU),J=1,NMU)
C
C     SKIP OTHER Y LEVELS FOR THIS L VALUE
      DO 100 IY=2,NY
      READ(NUSCR3) DUMREC
  100 READ(NUSCR3) DUMREC
      IF(IDBUG.EQ.2) THEN
      CALL P3ARAY(R1XBL,NMU,NMU,NL+1,MXMU,MXMU,2,'R1(X,B,L) IN AMPX')
      CALL P3ARAY(R2XBL,NMU,NMU,NL+1,MXMU,MXMU,2,'R2(X,B,L) IN AMPX')
      ENDIF
C
C     INITIALIZE FOR P = 1 (COSINE AMPLITUDES)
C
      IP = 1
      IOFSET = 0
C
  999 CONTINUE
C
C     LOAD RHATP(X,A) INTO RHAT AND RP(X,B,L) INTO RXBL
C
      IF(IP.EQ.1) THEN
      DO 800 J=1,NCHAT
      DO 800 I=1,NRHAT
  800 RHAT(I,J) = RHAT1(I,J)
      DO 801 L=0,NL
      DO 801 J=1,NMU
      DO 801 I=1,NMU
  801 RXBL(I,J,L) = R1XBL(I,J,L)
      ELSE
```

```
      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 P2ARAY(RHAT,2*NMU,NMU,MXRRTH,2,'RHATP(X,A) IN AMPX')
      ENDIF
C
C     COMPUTE TEMP = RP(X,B) * RHATP(X,A) AS NMU BY NMU BLOCKS
C
      DO 200 L=0,NL
      IROFF = NMU*L
      NCOL = IODD
      IF(MOD(L,2).EQ.0) NCOL = IEVEN
      DO 200 IR=1,NCOL
C
C     EXTRACT AN NMU BY NMU BLOCK FROM RHATP
      ICOFF = NMU*(IR-1)
      DO 210 I=1,NMU
      DO 210 J=1,NMU
  210 WORK(I,J) = RHAT(I+IROFF,J+ICOFF)
C
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
  214 SUM = SUM + RXBL(I,K,L)*WORK(K,J)
  212 RHAT(I+IROFF,J+ICOFF) = SUM
  200 CONTINUE
C
C     RHAT NOW CONTAINS RP(X,B) * RHATP
C
C     COMPUTE THE INVERSE FOR (7.3) USING THE APPROXIMATION (7.4)
C     (I + X) INVERSE = I + X + X**2 + ... + X**KINV
C
      IF(IDBUG.EQ.2) CALL P2ARAY(RHAT,2*NMU,NMU,MXRRTH,2,
     1'RP(X,B) * RHATP')
      CALL ADIPAK(RHAT,TEMP,MXRRTH,NMU,NL)
      DO 250 K=2,KINV
      CALL FFMPAK(TEMP,RHAT,THAT,MXRRTH,NMU,NL,WORK)
  250 CALL ADIPAK(THAT,TEMP,MXRRTH,NMU,NL)
C
C     TEMP NOW CONTAINS THE INVERSE
      IF(IDBUG.GE.1) THEN
      DO 804 I=1,NRHAT
      DO 804 J=1,NCHAT
  804 RHAT(I,J) = -RHAT(I,J)
      CALL ADIPAK(RHAT,THAT,MXRRTH,NMU,NL)
      CALL FFMPAK(TEMP,THAT,RHAT,MXRRTH,NMU,NL,WORK)
      CALL P2ARAY(RHAT,2*NMU,NMU,MXRRTH,2,'IDENTITY CHECK IN AMPX')
      ENDIF
C
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,NCHAT
      DO 810 I=1,NRHAT
  810 THAT(I,J) = THAT1(I,J)
      ELSE
      DO 812 J=1,NCHAT
      DO 812 I=1,NRHAT
  812 THAT(I,J) = THAT2(I,J)
      ENDIF
      IF(IDBUG.EQ.2) THEN
      CALL P2ARAY(THAT,2*NMU,NMU,MXRRTH,2,'THATP(A,X) IN AMPX')
      CALL P2ARAY(TEMP,2*NMU,NMU,MXRRTH,2,'(I - RP*RHATP) INVERSE')
      ENDIF
C
```

```
      CALL FFMPAK(THAT,TEMP,RHAT,MXRRTH,NMU,NL,WORK)
      IF(IDBUG.EQ.2) CALL P2ARAY(RHAT,2*NMU,NMU,MXRRTH,2,
     1 'SCRIPT T(A,X,B)')
C
C     RHAT NOW CONTAINS SCRIPT T(A,X,B)
C
C     COMPUTE AP(X,-) BY (7.3)
C
      CALL RFMPAK(A0AM(IOFSET+1),RHAT,AXM(IOFSET+1),MXRRTH,NMU,NL)
C
C     COMPUTE AP(X,+) BY 7.5B
C
      DO 400 L=0,NL
      LOFSET = NMU*L
      DO 400 I=1,NMU
      SUM = 0.
      DO 402 K=1,NMU
  402 SUM = SUM + AXM(K+LOFSET+IOFSET)*RXBL(K,I,L)
  400 AXP(I+LOFSET+IOFSET) = SUM
      IF(IDBUG.GE.2) THEN
      CALL P2ARAY(A0AM(IOFSET+1),1,NRHAT,1,2,'A0(A,-)')
      CALL P2ARAY(AXM(IOFSET+1),1,NRHAT,1,2,'A(X,-)')
      CALL P2ARAY(AXP(IOFSET+1),1,NRHAT,1,2,'A(X,+)')
      ENDIF
C
      IF(IP.GT.1) RETURN
C
C     REPEAT FOR P = 2 (SINE AMPLITUDES)
      IP = 2
      IOFSET = NRHAT
      GO TO 999
  901 FORMAT(1H1,' SUBROUTINE AMPX:  P =',I2)
C
      END
```

```
      SUBROUTINE BOTMBC(L)
C
C     ON NHM4/BOTMBC
C
C     THIS ROUTINE COMPUTES THE DISCRETE SPECTRAL RHATZB = RHAT1(Z,B,L)
C     FOR THE DESIRED BOTTOM BOUNDARY CONDITION.
C
C     IF IBOTM = 0, USE 3.26, 5.47, 5.50, 5.51 AND 5.53
C     IF IBOTM = 1, SET UP AND SOLVE THE EIGENVALUE PROBLEM 10.2
C                   AND THEN USE 10.8 AND 10.9 FOR RHAT(Z,INFINITY)
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
C
      COMMON/CBOTBC/ RHATZB(MXMU,MXMU)
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1 BNDPHI(MXPHI),OMEGA(MXMU)
      COMMON/CMISC/ IMISC(20),FMISC(20)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      IBOTM = IMISC(12)
      RMINUS = FMISC(14)
      CONST = RMINUS/FMISC(1)
C
```

```
      IF(IBOTM.EQ.0) THEN
C
      IF(L.EQ.0) THEN
C     FOR A MATTE BOTTOM AND L.EQ.0, GET THE GEOMETRIC R(Z,B) BY 3.26,
C     RHAT ELEMENT BY 5.53, RHAT ARRAY BY 5.50
C
      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
C
      ELSE
C     FOR MATTE BOTTOM AND L.GT.0, RHAT1(Z,B) = 0 BY 5.53B
      DO 102 IR=1,NMU
      DO 102 IU=1,NMU
  102 RHATZB(IR,IU) = 0.
      ENDIF
C
      ELSE
C
C     SET UP AND SOLVE THE EIGENVALUE PROBLEM FOR R(INFINITY,L)
C     EIGENR SETS RHATZB = R(INF,L)
C     IF RUNS ARE BEING MADE WITH A MATTE BOTTOM ONLY, THE CALL TO EIGENR
C     CAN BE COMMENTED OUT TO PREVENT LOADING THE LARGE IMSL ROUTINES IT CALLS
C
      CALL EIGENR(L)
      ENDIF
      RETURN
C
      END
```

```
      SUBROUTINE DRTAB(NRTAB,YNOW,RT,DERIV)
C
C     ON NHM4/DRTAB
C
C     THIS SUBROUTINE EVALUATES DERIV = D(RT)/DY AT Y = YNOW (THE RIGHT
C     HAND SIDE OF 6.43, 6.44 AND 6.48) FOR USE BY THE IMSL ROUTINE DVERK
C
C     RECALL THAT RYX AND TXY ARE STORED IN RT:
C
C     RYX(I,J) = RT(I + (J-1)*NIJ)
C     TXY(I,J) = RT(I + (J-1)*NIJ + NIJ*NIJ)
C
      PARAMETER(MXMU=10, MXSIGY=3)
      REAL RT(NRTAB),DERIV(NRTAB)
      DIMENSION WORK(MXMU,MXMU),RHOY(MXMU,MXMU),TAUY(MXMU,MXMU)
      COMMON/CRTSIG/ RHOHAT(MXMU,MXMU,MXSIGY),TAUHAT(MXMU,MXMU,MXSIGY)
      COMMON/CSIGY/ YSIG(MXSIGY)
      COMMON/CMISC/ IMISC(20)
C
      NMU = IMISC(1)
      NSIGY = IMISC(5)
      IDE = IMISC(13)
      NSQ = NMU*NMU
C
```

```
C     DETERMINE RHOHAT AND TAUHAT AT THE CURRENT Y VALUE
C
      IF(NSIGY.EQ.1 .OR. YNOW.LE.YSIG(1)) THEN
C
C     THE WATER IS UNIFORM, OR YNOW IS AT OR ABOVE THE FIRST DEPTH
C     WHERE SIGMA IS KNOWN
C
      DO 50 J=1,NMU
      DO 50 I=1,NMU
      RHOY(I,J) = RHOHAT(I,J,1)
   50 TAUY(I,J) = TAUHAT(I,J,1)
C
      ELSEIF(YNOW.GE.YSIG(NSIGY)) THEN
C
C     YNOW IS AT OR BELOW THE LAST DEPTH WHERE SIGMA IS KNOWN
C
      DO 52 J=1,NMU
      DO 52 I=1,NMU
      RHOY(I,J) = RHOHAT(I,J,NSIGY)
   52 TAUY(I,J) = TAUHAT(I,J,NSIGY)
C
      ELSE
C
C     DEFINE RHOHAT AND TAUHAT BY LINEAR INTERPOLATION OF THE VALUES FROM
C     THOSE DEPTHS WHERE SIGMA IS KNOWN
C
      DO 55 JY=2,NSIGY
      IF(YNOW.LT.YSIG(JY)) GO TO 56
   55 CONTINUE
C
   56 DY = (YNOW - YSIG(JY-1))/(YSIG(JY) - YSIG(JY-1))
C
      DO 58 J=1,NMU
      DO 58 I=1,NMU
      RHOY(I,J) = (1.0 - DY)*RHOHAT(I,J,JY-1) + DY*RHOHAT(I,J,JY)
   58 TAUY(I,J) = (1.0 - DY)*TAUHAT(I,J,JY-1) + DY*TAUHAT(I,J,JY)
C
      ENDIF
C
C     COMPUTE K = TAUY + RHOY*RYX
C
      DO 100 I=1,NMU
      DO 100 J=1,NMU
      WORK(I,J) = TAUY(I,J)
      DO 100 K=1,NMU
  100 WORK(I,J) = WORK(I,J) + RHOY(I,K)*RT(K + (J-1)*NMU)
C
C     COMPUTE D(RYX)/DY BY EQ. 6.43
C
      DO 200 I=1,NMU
      DO 200 J=1,NMU
      TEMP1 = 0.
      TEMP2 = 0.
      DO 201 K=1,NMU
      TEMP1 = TEMP1 + RT(I + (K-1)*NMU)*WORK(K,J)
  201 TEMP2 = TEMP2 + TAUY(I,K)*RT(K + (J-1)*NMU)
  200 DERIV(I + (J-1)*NMU) = RHOY(I,J) + TEMP1 + TEMP2
C
      IF(IDE.NE.2) THEN
C
C     COMPUTE D(TXY)/DY BY EQ. 6.44
C
      DO 300 I=1,NMU
      DO 300 J=1,NMU
      TEMP1 = 0.
      DO 301 K=1,NMU
  301 TEMP1 = TEMP1 + RT(I + (K-1)*NMU + NSQ)*WORK(K,J)
  300 DERIV(I + (J-1)*NMU + NSQ) = TEMP1
C
      ELSE
C
C     CHANGE OF SIGN TO GET EQ. 6.48
      DO 700 I=1,NRTAB
  700 DERIV(I) = -DERIV(I)
      ENDIF
C
      RETURN
      END
```

```
      SUBROUTINE EIGENR(L)
C
C     ON NHM4/EIGENR
C
C     THIS ROUTINE SETS UP AND SOLVES THE EIGENMATRIX PROBLEM KE = EL
C     AS DESCRIBED IN SECTION 10.
C     THE SUBMATRICES EP = E(+) AND EM = E(-) ARE EXTRACTED, AND
C     R(INFINITY,L) = -E(-) * E(+)INVERSE IS COMPUTED.
C
C     THE ASYMPTOTIC RADIANCE DISTRIBUTION AND ASSOCIATED QUANTITIES
C     ARE ALSO FOUND USING FROMULAS FROM TECH MEMO ERL-PMEL-76.
C
C     IF L = 0, THE FULL RHO AND TAU MATRICES ARE USED TO DEFINE K
C     IF L.GT.0, ROW NMU AND COLUMN NMU OF RHO AND TAU IS ZERO, AND
C                THUS IS OMITTED FROM K (SEE PAGE 174)
C
      PARAMETER(MXMU=10, MXPHI=24, MXY = 30, MXSIGY=3)
      PARAMETER(MXMU2=2*MXMU, MXMUSQ=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/CRTSIG/ RHOHAT(MXMU,MXMU,MXSIGY),TAUHAT(MXMU,MXMU,MXSIGY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CBOTBC/ RHATZB(MXMU,MXMU)
      COMMON/CMISC/ IMISC(20)
      COMMON/CWORK/ WERK(MXMUSQ,12),RPINF(MXMU),RMINF(MXMU),WORK(1)
C     ARRAY WORK(1) MUST HAVE 4*NMU*(NMU+1) WORDS AVAILABLE
C
      DIMENSION FP(MXMU,MXMU),FM(MXMU,MXMU),TEMP1(MXMU,MXMU),
     1  TEMP2(MXMU,MXMU),EMNV(MXMU,MXMU)
C
      DIMENSION AK(MXMU2,MXMU2),EMK(MXMU2,MXMU2),EP(MXMU,MXMU),
     1  EM(MXMU,MXMU),EPNV(MXMU,MXMU)
      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),EM(1,1)),(WERK(1,7),EPNV(1,1))
C
      NMU = IMISC(1)
      NSIGY = IMISC(5)
      IDBUG = IMISC(9)
      ALPHA = TOTALS(NSIGY)/ALBESS(NSIGY)
C
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(I.LE.M) THEN
      IF(J.LE.M) AK(I,J) = -TAUHAT(I,J,NSIGY)
      IF(M.LT.J) AK(I,J) = RHOHAT(I,J-M,NSIGY)
      ELSE
      IF(J.LE.M) AK(I,J) = -RHOHAT(I-M,J,NSIGY)
      IF(M.LT.J) AK(I,J) = TAUHAT(I-M,J-M,NSIGY)
      ENDIF
  100 CONTINUE
      IF(IDBUG.GT.1) CALL P2ARAY(AK,M2,M2,MXMU2,2,
     1 'K MATRIX FROM SUB EIGENR')
C
C     FIND EIGENVALUES AND EIGENVECTORS OF K
C
      CALL EIGRF(AK,M2,MXMU2,2,WEV,ZEV,MXMU2,WORK,IER)
      IF(IDBUG.GT.1) THEN
      WRITE(6,301) (WEV(I),I=1,M2)
      WRITE(6,304) WORK(1)
      ENDIF
```

```
C
C     SORT POSITIVE EIGENVALUES
C
      KPOS = 0
      DO 600 I=1,M2
      TMP = WEV(I)
      IF(TMP.LT.0.) GO TO 600
      KPOS = KPOS + 1
      EVALS(KPOS) = TMP
  600 CONTINUE
C
      CALL VSRTA(EVALS,KPOS)
C     DEFINE ORDERED EIGENVALUES
      DO 601 I=1,KPOS
      EIGV(I) = EVALS(I)
  601 EVALS(I+KPOS) = -EVALS(I)
C
C     CONSTRUCT 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.0 .OR. IDBUG.GT.0) WRITE(6,680) L,(EIGV(I),ALPHA*EIGV(I),
     1  I=1,M)
C
C     DEFINE REAL, ORDERED EIGENVECTOR MATRIX EMK
C
      DO 620 J=1,M2
      JJ = IP(J)
      DO 620 I=1,M2
  620 EMK(I,JJ) = ZEV(I,J)
      IF(IDBUG.GT.1) CALL P2ARAY(EMK,M2,M2,MXMU2,2,
     1 'SORTED EIGENVECTORS')
C
C     EXTRACT THE SUBMATRICES EP = E(+) AND EM = -E(-)
C
      DO 630 I=1,M
      DO 630 J=1,M
      EP(I,J) = EMK(I,J)
  630 EM(I,J) = -EMK(I+M,J)
      IF(IDBUG.GT.1) THEN
      CALL P2ARAY(EP,M,M,MXMU,2,'E(+)')
      CALL P2ARAY(EM,M,M,MXMU,2,'-E(-)')
      ENDIF
C
C     INVERT E(+) AND DEFINE R(INFINITY), USING 10.8 OR 10.9
C
      IDGT = 6
      CALL LINV2F(EP,M,MXMU,EPNV,IDGT,WORK,IER)
      CALL VMULFF(EM,EPNV,M,M,M,MXMU,MXMU,RHATZB,MXMU,IER)
C
C     FILL THE LAST ROW AND LAST COLUMN OF RHAT(Z,B) WITH ZEROS IF
C     L.GT.0
      IF(L.GT.0) THEN
      DO 649 I=1,NMU
      RHATZB(NMU,I) = 0.
  649 RHATZB(I,NMU) = 0.
      ENDIF
C
C     CONSTRUCT THE F(+) AND F(-) MATRICES AND GET THE ASYMPTOTIC
C     RADIANCE DISTRIBUTION USING 76/18.2.  NOTE THAT RADINF(+) IS OBTAINED
C     FROM F(-) AND THAT RADINF(-) IS OBTAINED FROM F(+).
C
      IF(L.EQ.0) THEN
C
C     EM IS -E(-)
      DO 652 J=1,NMU
      DO 652 I=1,NMU
  652 EM(I,J) = -EM(I,J)
C
```

```
      IDGT = 6
      CALL LINV2F(EM,NMU,MXMU,EMNV,IDGT,WORK,IER)
      IF(IDBUG.GT.1) THEN
      CALL P2ARAY(EPNV,NMU,NMU,MXMU,2,'E(+)-1')
      CALL P2ARAY(EMNV,NMU,NMU,MXMU,2,'E(-)-1')
      ENDIF
      CALL VMULFF(EM,EPNV,NMU,NMU,NMU,MXMU,MXMU,TEMP1,MXMU,IER)
      CALL VMULFF(TEMP1,EM,NMU,NMU,NMU,MXMU,MXMU,TEMP2,MXMU,IER)
      DO 650 I=1,NMU
      DO 650 J=1,NMU
  650 TEMP1(I,J) = EP(I,J) - TEMP2(I,J)
C
      IDGT = 6
      CALL LINV2F(TEMP1,NMU,MXMU,FP,IDGT,WORK,IER)
      CALL VMULFF(EP,EMNV,NMU,NMU,NMU,MXMU,MXMU,TEMP1,MXMU,IER)
      CALL VMULFF(TEMP1,EP,NMU,NMU,NMU,MXMU,MXMU,TEMP2,MXMU,IER)
      DO 651 I=1,NMU
      DO 651 J=1,NMU
  651 TEMP1(I,J) = EM(I,J) - TEMP2(I,J)
      IDGT = 6
      CALL LINV2F(TEMP1,NMU,MXMU,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)
  656 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
  670 CHM = CHM + RMINF(I)*FMU(I)*DMU
      DPINF = SHP/CHP
      DMINF = SHM/CHM
C     K(INFINITY) BY 76/19.2
      FKINF = ALPHA*EIGV(1)
C     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
      ENDIF
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
     1VEN BY'//'     I       RADINF(-)       RADINF(+)'/)
  657 FORMAT(1H ,I4,1P2E15.4)
  672 FORMAT(//' OTHER ASYMPTOTIC VALUES ARE'//'    D+(INFINITY) =',
     1 F7.4/'    D-(INFINITY) =',F7.4/'    R-(INFINITY) =',1PE11.4/
     2' EPS+(INFINITY) =',0PF8.5/' EPS-(INFINITY) =',F8.5)
  680 FORMAT(//' THE ORDERED POSITIVE EIGENVALUES OF K(L=',I2,
     1') ARE'//'      NONDIMEN       PER METER'//(1P2E15.6))
  681 FORMAT(/'     J    IP'//(2I5))
C
      END
```

```
      SUBROUTINE EPSINF(RPINF,RMINF, EPSP,EPSM)
C
C     ON NHM4/EPSINF
C
C     THIS ROUTINE COMPUTES THE ASYMPTOTIC BACKSCATTER ECCENTRICITIES
C     EPSILONB(+) AND EPSILONB(-) USING (8.15A) AND THE NORMALIZED
C     ASYMPTOTIC RADIANCE DISTRIBUTION.
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXL=MXPHI/2, MXGEOP=MXMU*(MXL+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)
      COMMON/CRTSIG/ RHOHAT(MXMU,MXMU,MXSIGY),TAUHAT(MXMU,MXMU,MXSIGY),
     1 GEOPP(MXMU,MXGEOP,MXSIGY),GEOPM(MXMU,MXGEOP,MXSIGY)
      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 IU=1,NMU
C
C     ACCUMULATE SCALAR IRRADIANCE SUMS
      SHPINF = SHPINF + RPINF(IU)*DELTMU(IU)
      SHMINF = SHMINF + RMINF(IU)*DELTMU(IU)
C
      QUV = OMEGA(IU)
      IVMAX = NPHI
      IF(IU.EQ.NMU) IVMAX = 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,ISMAX
C
C     COMPUTE THE STORAGE INDEX FOR P-(R,U,V) BY 12.7
C
      IVS = IABS(IV-IS)
      IF(IR.EQ.NMU) THEN
      KCOL = IU
      ELSE
           IF(IU.EQ.NMU) THEN
           KCOL = NMU
           ELSE
                IF(IVS.LE.NOPI) THEN
                KCOL = IU + NMU*IVS
                ELSE
                KCOL = IU + NMU*(NOPI - MOD(IVS,NOPI))
                ENDIF
           ENDIF
      ENDIF
C
      PM = GEOPM(IR,KCOL,NSIGY)
C
      EPSP = EPSP + QUV*RP*PM
  100 EPSM = EPSM + QUV*RM*PM
C
      EPSP = EPSP/(TWOPI*SHPINF)
      EPSM = EPSM/(TWOPI*SHMINF)
C
      RETURN
      END
```

```
      FUNCTION FALPHA(Y)
C
C     ON NHM4/FALPHA
C
C     GIVEN AN OPTICAL DEPTH Y, THIS FUNCTION RETURNS THE VALUE OF
C     1.0/ALPHA(Y), WHERE ALPHA IS THE ATTENUATION COEFFICIENT, FOR
C     USE IN INTEGRATING DY/ALPHA(Y) TO GET GEOMETRIC DEPTHS (SEE
C     SUBROUTINE Y2ZGEO).
C
      PARAMETER(MXSIGY=3)
C
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CMISC/ IMISC(20)
C
      NSIGY = IMISC(5)
C
      IF(NSIGY.EQ.1 .OR. Y.LE.YSIG(1)) THEN
C
      ALPHA1 = TOTALS(1)/ALBESS(1)
C
      ELSEIF(Y.GT.YSIG(NSIGY)) THEN
C
      ALPHA1 = TOTALS(NSIGY)/ALBESS(NSIGY)
C
      ELSE
C
      DO 55 JY=2,NSIGY
      IF(Y.LT.YSIG(JY)) GO TO 56
   55 CONTINUE
   56 DY = (Y - YSIG(JY-1))/(YSIG(JY) - YSIG(JY-1))
      ALPHA1 = (1.0 - DY)*TOTALS(JY-1)/ALBESS(JY-1) +
     1         DY*TOTALS(JY)/ALBESS(JY)
C
      ENDIF
C
      FALPHA = ALPHA1
      RETURN
      END
```

```
      SUBROUTINE FFMPAK(X,Y,Z,IROW,NMU,L,WORK)
C
C     ON NHM4/FFMPAK
C
C     THIS ROUTINE FORMS THE MATRIX PRODUCT X * Y = Z, WHERE X, Y, AND Z
C     ARE BLOCK MATRICES STORED ON THE PACKED FORMAT OF 12.4.
C
C     INDEXING IS SOMEWHAT COMPLICATED, DUE TO THE PACKING FORMAT.
C     THE VARIOUS INDICES USED ARE
C
C     IXB...BLOCK ROW INDEX OF X
C     I1....ELEMENT ROW OFFSET OF BLOCK ROW IXB OF X
C     J1....ELEMENT COLUMN OFFSET OF BLOCK COLUMN KX OF X
C     JYB...BLOCK COLUMN INDEX OF Y
C     I2....ELEMENT ROW OFFSET OF BLOCK ROW KY OF Y
C     J2....ELEMENT COLUMN OFFSET OF BLOCK JYB OF Y
C     KX....BLOCK COLUMN INDEX OF X
C     KY....BLOCK ROW INDEX OF Y
C     I,J,K.ELEMENT INDICES WITHIN AN NMU BY NMU BLOCK
C
C     WORK MUST HAVE AT LEAST NMU*NMU WORDS
      DIMENSION X(IROW,1),Y(IROW,1),Z(IROW,1),WORK(NMU,1)
C
```

```
      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
C     ZERO THE ACCUMULATION BLOCK
      DO 110 I=1,NMU
      DO 110 J=1,NMU
  110 WORK(I,J) = 0.
C
C     MULTIPLY BLOCK ROW IXB OF X BY BLOCK COLUMN JYB OF Y
C
      DO 300 KX=1,JXB2
      KY = 2*KX - 1
      IF(MOD(IXB,2).EQ.0) KY = 2*KX
      J1 = (KX - 1)*NMU
      I2 = (KY - 1)*NMU
C
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
  301 SUM = SUM + X(I1+I,J1+K)*Y(I2+K,J2+J)
  300 WORK(I,J) = WORK(I,J) + SUM
C
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(Y,COSPSI)
C
C     ON NHM4/PFLIMNE
C
C     THIS FUNCTION RETURNS THE VALUE OF THE PHASE FUNCTION AS DEFINED
C     FOR LAKE LIMNE.  (SEE RADIATIVE TRANSFER IN NATURAL WATERS, CHAPTER 5,
C     TABLE 5.3, WHEN PUBLISHED.  MEANWHILE, REGARD THIS PHASE FUNCTION
C     AS TYPICAL OF MODERATELY TURBID LAKE WATER.)
C
C     SINCE THE PHASE FUNCTION IS
C     NEARLY LINEAR ON A LOG-LOG PLOT, LINEAR INTERPOLATION IS
C     PERFORMED IN LOG(PHASE)-LOG(PSI).
C
      PARAMETER(MXPTS=22, MXSIGY=3)
C
      DIMENSION SIGMA(MXPTS),PSI(MXPTS),PLOG(MXPTS),PSILOG(MXPTS)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CMISC/ IMISC(20),FMISC(20)
C
```

```
      DATA PSI/0.0,0.01,0.1,1.0,
     1 10.,20.,30.,40.,50.,60.,70.,80.,90.,100.,110.,120.,130.,
     2 140.,150.,160.,170.,180./
      DATA SIGMA/7.92609E6,7.92609E6,315543.,12562.,315.55,90.62,
     1 30.89,13.2,6.41,3.47,2.08,1.37,1.0,0.811,0.716,0.691,
     2 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/, PSI0/0.01/, APSI/43.4197/, PPSI/1.4/
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)
      IMISC(5) = NSIGY
      YSIG(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) = ALOG10(S1*SIGMA(I))
  100 PSILOG(I) = ALOG10(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) PSI(I),SIGMA(I),PHASE
      WRITE(6,207) SIG90
      WRITE(6,208)
C
C     GET THE ANALYTIC INTEGRAL FROM PSI = 0 TO PSI = PSI0
C
      APSI = APSI*SIG90/S
      SOPSI0 = 2.0*PI*APSI/(2.0 - PPSI)
      SOPSI0 = SOPSI0*(PSI0*DEGRAD)**(2.0 - PPSI)
      WRITE(6,210) PSI0,SOPSI0,APSI,PPSI
C
      KALL = 1
      PHASEL = 0.
C
      ELSE
C
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
      PSIL = -8.0
      ELSEIF(PSIDEG.GT.180.) THEN
      PSIL = ALOG10(180.)
      ELSE
      PSIL = ALOG10(PSIDEG)
      ENDIF
C
      IF(PSIL.LE.PSILOG(2)) THEN
      PHASEL = PLOG(2)
      ELSE
      DO 300 I=2,MXPTS
      IF(PSIL.LT.PSILOG(I)) GO TO 302
  300 CONTINUE
  302 PHASEL = PLOG(I-1) + (PLOG(I) - PLOG(I-1))*
     1  (PSIL - PSILOG(I-1))/(PSILOG(I) - PSILOG(I-1))
      ENDIF
      ENDIF
```

```
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')
  202 FORMAT(1HO,' THE VOLUME ATTENUATION COEFFICIENT ALPHA IS',
     1F6.3,' PER METER'//'  THE TOTAL VOLUME SCATTERING FUNCTION S IS',
     2F6.3,' PER METER')
  204 FORMAT(1HO,' THE TABULATED VALUES DEFINING SIGMA(PSI) ARE'//
     1'      PSI     SIGMA/SIG90',9X,'PHASE'/)
  206 FORMAT(1H ,F7.2,F15.3,F18.6)
  207 FORMAT(1HO,' SIGMA(90) =',1PE13.6)
  208 FORMAT(1HO,' LINEAR INTERPOLATION IS DONE IN LOG(PSI)-LOG(PHASE)')
  210 FORMAT(1HO,' THE ANALYTIC INTEGRAL OF 2*PI*PHASE(PSI)*SIN(PSI)'//
     1'  FROM PSI = 0 TO PSI =',F5.3,' IS',1PE14.6,' FOR A =',E14.6,
     2' AND P =',0PF5.2)
      END
```

```
      FUNCTION PHASEF(Y,COSPSI)
C
C     ON NHM4/PFPELAG
C
C     THIS FUNCTION RETURNS THE VALUE OF THE PHASE FUNCTION AS DEFINED
C     FOR THE PELAGOS SEA.  (SEE RADIATIVE TRANSFER IN NATURAL WATERS,
C     CHAPTER 5, TABLE 5.5, WHEN PUBLISHED.  MEANWHILE, REGARD THIS
C     SCATTERING FUNCTION AS TYPICAL OF CLEAN, OPEN OCEAN WATERS.)
C
C     GIVEN THE WAVELENGTH IN NANOMETERS, WAVENM, THE FIRST CALL TO
C     THE ROUTINE LINEARLY INTERPOLATES IN LOG(NORM SIGMA) TO GET A NORMALIZED
C     SIGMA FUNCTION FOR THE DESIRED WAVELENGTH AT EACH TABULATED SCATTERING
C     ANGLE, PSI.  VALUES OF THE ABSORBTION AND TOTAL SCATTERING ARE
C     ALSO DETERMINED FOR THE REQUESTED WAVELENGTH.
C
C     N.B.:  THE REQUESTED WAVELENGTH, WAVENM, SHOULD BE ONE OF THE
C     DISCRETE NHM WAVELENGTHS (NAMELY, 400., 425., ..., 675. OR 700. NM).
C     THIS IS BECAUSE THE SIGMA(90) VALUES CANNOT BE OBTAINED BY
C     INTERPOLATION OF THE TABULATED VALUES.
C
C     SINCE THE PHASE FUNCTION IS NEARLY LINEAR ON A LOG-LOG PLOT,
C     LINEAR INTERPOLATION IS PERFORMED IN LOG(PHASE)-LOG(PSI) IN ORDER
C     TO DEFINE VALUES OF THE PHASE FUNCTION AT ARBITRARY PSI VALUES.
C
      PARAMETER(MXPTS=22, MXSIGY=3)
C
      DIMENSION SIGMA(MXPTS),PSI(MXPTS),PLOG(MXPTS),PSILOG(MXPTS)
      DIMENSION S400(MXPTS),S700(MXPTS),SIGLOG(MXPTS)
      DIMENSION WVLTAB(13),ABSORB(13),TSCAT(13),S90(13)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CMISC/ IMISC(20),FMISC(20)
C
      DATA PSI/0.0,0.01,0.1,1.0,
     1 10.,20.,30.,40.,50.,60.,70.,80.,90.,100.,110.,120.,130.,
     2 140.,150.,160.,170.,180./
      DATA S400/ 874689., 874689., 34822., 1386.29, 34.822, 10., 5.,
     1  2.8, 2.0, 1.5, 1.25, 1.1, 1.0, .96, .98, 1.05, 1.22, 1.5,
     2  1.9, 2.25, 2.55, 2.7/
      DATA S700/ 3.9361E6, 3.9361E6, 156700., 6238.3, 156.7, 45.,
     1  20., 9., 4.3, 2.65, 1.72, 1.22, 1.0, .94, .94, 1.0, 1.12,
     2  1.33, 1.6, 1.8, 1.9, 2.0/
C
```

```
      DATA WVLTAB/400., 425., 450., 475., 500., 525., 550., 575.,
     1  600., 625., 650., 675., 700./
      DATA ABSORB/ .05, .05, .05, .05, .05, .06, .08, .14, .25, .30,
     1 .37, .43, .60/
      DATA TSCAT/ .04, .0325, .0275, .0234, .02, .0175, .0152, .0136,
     1  .0122, .0115, .0108, .0102, .01/
      DATA S90/9.98348E-4, 7.45989E-4, 5.78771E-4, 4.50275E-4,
     1 3.50922E-4, 2.79280E-4, 2.201076E-4, 1.78304E-4, 1.44517E-4,
     2 1.22848E-4, 1.03858E-4, 8.815764E-5, 7.75634E-5/
C
      DATA KALL/0/, NSIGY/1/
      DATA APSI400,APSI700/4.791616, 21.562268/, PSI0/0.01/, PPSI/1.4/
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)
      WAVENM = FMISC(13)
      IMISC(5) = NSIGY
C
C     LINEARLY INTERPOLATE IN LOG(NORMALIZED SIGMA)-WAVELENGTH TO
C     DEFINE A NORMALIZED SIGMA AT THE REQUESTED WAVELENGTH, AT EACH
C     TABULATED PSI VALUE
C
      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)*ALOG10(S400(I)) +
     1 WAVEF*ALOG10(S700(I))
  400 SIGMA(I) = 10.0**SIGLOG(I)
C
C     LOOK UP THE ABSORBTION, TOTAL SCATTERING, AND SIGMA(90) VALUES
C     FOR THE REQUESTED WAVELENGTH
C
      IWAVE = IFIX(1.5 + AMOD(WAVENM,400.)/25.)
      ABSR = ABSORB(IWAVE)
      S = TSCAT(IWAVE)
      SIG90 = S90(IWAVE)
      ALPHA = ABSR + S
C
      YSIG(1) = 0.
      TOTALS(1) = S
      ALBESS(1) = S/ALPHA
C
C     CONVERT THE DEFINED SIGMA TO LOGS OF THE PHASE FUNCTION
      S1 = SIG90/S
      DO 100 I=2,MXPTS
      PLOG(I) = ALOG10(S1*SIGMA(I))
  100 PSILOG(I) = ALOG10(PSI(I))
      PLOG(1) = PLOG(2)
      PSILOG(1) = -1.0E200
C
      WRITE(6,200)
      WRITE(6,202) WAVENM,ABSR,S,ALPHA,ALBESS(1)
      WRITE(6,204)
      DO 102 I=2,MXPTS
      PHASE = 10.0**PLOG(I)
  102 WRITE(6,206) PSI(I),SIGMA(I),PHASE
      WRITE(6,207) SIG90
      WRITE(6,208)
```

```
C
C      GET THE ANALYTIC INTEGRAL FROM PSI = 0 TO PSI = PSI0
C
       APSI = (1.0 - WAVEF)*ALOG10(APSI400) + WAVEF*ALOG10(APSI700)
       APSI = 10.0**APSI
       APSI = APSI*SIG90/S
       SOPSI0 = 2.0*PI*APSI/(2.0 - PPSI)
       SOPSI0 = SOPSI0*(PSI0*DEGRAD)**(2.0 - PPSI)
       WRITE(6,210) PSI0,SOPSI0,APSI,PPSI
C
       KALL = 1
       PHASEL = 0.
C
       ELSE
C
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
       PSIL = -8.0
       ELSEIF(PSIDEG.GT.180.) THEN
       PSIL = ALOG10(180.)
       ELSE
       PSIL = ALOG10(PSIDEG)
       ENDIF
C
       IF(PSIL.LE.PSILOG(2)) THEN
       PHASEL = PLOG(2)
       ELSE
       DO 300 I=2,MXPTS
       IF(PSIL.LT.PSILOG(I)) GO TO 302
  300  CONTINUE
  302  PHASEL = PLOG(I-1) + (PLOG(I) - PLOG(I-1))*
      1  (PSIL - PSILOG(I-1))/(PSILOG(I) - PSILOG(I-1))
       ENDIF
       ENDIF
C
       PHASEF = 10.0**PHASEL
       KALL = KALL + 1
       RETURN
C
  200  FORMAT(1H1,' THE VOLUMNE SCATTERING FUNCTION DEFINED FOR THE PELAG
      1OS SEA IS USED AT ALL DEPTHS')
  202  FORMAT(1HO,' THE WAVELENGTH IS LAMBDA =',F6.1,' NANOMETERS'//
      1'  THE VOLUME ABSORBTION FUNCTION IS A =',F7.4,' PER METER'//
      2'  THE TOTAL VOLUME SCATTERING FUNCTION IS S =',F7.4,' PER METER'/
      3/'  THE VOLUME ATTENUATION FUNCTION IS ALPHA =',F7.4,' PER METER'/
      4/'  THE ALBEDO OF SINGLE SCATTERING IS OMEGA =',F7.4)
  204  FORMAT(1HO,' THE TABULATED VALUES DEFINING SIGMA(PSI) ARE'//
      1'      PSI     SIGMA/SIG90',9X,'PHASE'/)
  206  FORMAT(1H ,F7.2,F15.3,F18.6)
  207  FORMAT(1HO,' SIGMA(90) =',1PE13.6)
  208  FORMAT(1HO,' LINEAR INTERPOLATION IS DONE IN LOG(PSI)-LOG(PHASE)')
  210  FORMAT(1HO,' THE ANALYTIC INTEGRAL OF 2*PI*PHASE(PSI)*SIN(PSI)'//
      1'  FROM PSI = 0 TO PSI =',F5.3,' IS',1PE14.6,' FOR A =',E14.6,
      2' AND P =',0PF5.2)
       END
```

```
      FUNCTION PHASEF(Y,COSPSI)
C
C     ON NHM4/PFSPHER
C
C     THIS FUNCTION RETURNS A VALUE OF THE CONTINUOUS, POINT GEOMETRIC
C     PHASE FUNCTION P(Y, MU PRIME, PHI PRIME/ MU, PHI) = P(COSPSI,Y) =
C     SIGMA(COSPSI,Y)/S(Y) FOR ANY COS(PSI) AND Y VALUES.
C     PHASEF IS FOR USE IN THE COMPUTATION OF THE QUAD-AVERAGED,
C     GEOMETRIC SCATTERING FUNCTIONS P(Y, R,S/ U,V) = P(Y, R/ U,V") VIA
C      11.3.
C
C     THIS VERSION FOR FOR ISOTROPIC SCATTERING: SIGMA = S/(4*PI)
C     INDEPENDENT OF SCATTERING ANGLE AND DEPTH
C
      PARAMETER(MXSIGY=3)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DATA KALL/0/, NSIGY/1/, S,ALPHA/0.125, 0.736/
C
      IF(KALL.EQ.0) THEN
C
C     THE FIRST CALL TO PHASEF IS USED FOR INITIALIZATION ONLY
C
      PI = FMISC(1)
      SIG = 0.25*S/PI
      IMISC(5) = NSIGY
      YSIG(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
C
      ELSE
      PHASEF = SIG
      RETURN
      ENDIF
C
  100 FORMAT(1H1,' A SPHERICALLY SYMMETRIC VOLUME SCATTERING FUNCTION IS
     1 USED:'
     2//' SIGMA = S/(4*PI) =',F8.5,' FOR ALL ANGLES AND DEPTHS')
  102 FORMAT(1H0,'THE VOLUME ATTENUATION COEFFICIENT ALPHA IS',F6.3,
     1' PER METER'///' THE TOTAL VOLUME SCATTERING FUNCTION S IS',
     2F6.3,' PER METER')
      END
```

```
      FUNCTION PHASEF(Y,COSPSI)
C
C     ON NHM4/PFSPY
C
C     THIS FUNCTION RETURNS A VALUE OF THE CONTINUOUS, POINT GEOMETRIC
C     PHASE FUNCTION P(Y, MU PRIME, PHI PRIME/ MU, PHI) = P(Y,COSPSI) =
C     SIGMA(Y,COSPSI)/S(Y) FOR ANY Y AND COS(PSI) VALUES.
C     PHASEF IS FOR USE IN THE COMPUTATION OF THE QUAD-AVERAGED,
C     GEOMETRIC SCATTERING FUNCTIONS P(Y, R,S/ U,V) = P(Y, R/ U,V") VIA
C     (11.3)
C
C     THIS VERSION IS FOR DEPTH DEPENDENT SPHERICAL SCATTERING:
C     SIGMA(Y) = S(Y)/(4*PI), INDEPENDENT OF SCATTERING ANGLE
C     BUT DEPENDENT ON DEPTH Y.
C
      PARAMETER(MXSIGY=3)
      DIMENSION ALPHA(MXSIGY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DATA KALL/0/, NSIGY/3/
      DATA YSIG/1.,5.,10./, TOTALS/0.1,0.05,0.3/, ALPHA/0.2,0.6,0.4/
C
      IF(KALL.EQ.0) THEN
C
C     THE FIRST CALL TO PHASEF IS USED FOR INITIALIZATION ONLY
C
      PI = FMISC(1)
      IMISC(5) = NSIGY
      WRITE(6,100)
      DO 50 IY=1,NSIGY
      ALBESS(IY) = TOTALS(IY)/ALPHA(IY)
   50 WRITE(6,102) IY,YSIG(IY),TOTALS(IY),ALPHA(IY),ALBESS(IY)
      P = 0.25/PI
      KALL = 1
      PHASEF = 0.
C
      ELSE
      PHASEF = P
      ENDIF
      RETURN
C
  100 FORMAT(///' A DEPTH DEPENDENT, SPHERICAL VOLUME SCATTERING FUNCTIO
     1N IS USED;'//' SIGMA(Y,COS(PSI)) = S(Y)/(4*PI) WHERE'//
     2'   IY      Y       S(Y)     ALPHA(Y)     S/ALPHA'/)
  102 FORMAT(1H ,I4,F8.1,F8.3,F10.3,F11.3)
      END
```

```
      SUBROUTINE PNTAMP(Y,AMPA,AMPY,IROW)
C
C     ON NHM4/PNTAMP
C
C     THIS ROUTINE PRINTS THE AMPLITUDES AT Y = A, X,..., Z.
C     A TITLE GIVING THE APPROPRIATE COLUMN HEADINGS SHOULD BE WRITTEN
C     BEFORE CALLING PNTAMP.
C     IF ONLY AMPA IS TO BE PRINTED (THE CASE OF AMPA = AO(A,+)), GIVE
C     AMPY(1,1) A VALUE .GT. 1.0E200
C
      DIMENSION Y(1),AMPA(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
C     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) (Y(IY),IY=1,NY)
      DO 1602 I=1,2*NRHAT
      IF(I.EQ.NRHAT+1) THEN
      WRITE(6,1610)
      IL = 0
      ENDIF
      IMOD = MOD(I,NMU)
      IF(IMOD.EQ.1) THEN
      WRITE(6,1606) IL
      IL = IL + 1
      IMU = 0
      ENDIF
      IMU = IMU + 1
C     SELECT FULL OR PARTIAL PRINTOUT
      IF(IDBUG.EQ.1 .AND. IL.GT.2) GO TO 1602
      IF(IAOP.EQ.0) THEN
      IF(IMOD.EQ.1) WRITE(6,1612) IMU,AMPA(I),(AMPY(I,J),J=1,NY)
      IF(IMOD.NE.1) WRITE(6,1614) IMU,AMPA(I),(AMPY(I,J),J=1,NY)
      ELSE
      IF(IMOD.EQ.1) WRITE(6,1612) IMU,AMPA(I)
      IF(IMOD.NE.1) WRITE(6,1614) IMU,AMPA(I)
      ENDIF
 1602 CONTINUE
      RETURN
C
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(1H+,10X,I2,1P6E15.4/28X,5E15.4)
 1614 FORMAT(1H ,10X,I2,1P6E15.4/28X,5E15.4)
      END
```

```
      SUBROUTINE QAPHAS(NUQB,NVQB,INCBAS)
C
C     ON NHM4/QAPHAS
C
C     THIS ROUTINE COMPUTES THE QUAD-AVERAGED GEOMETRIC PHASE
C     FUNCTIONS GEOPP = P+(Y;R,U,V) AND GEOPM = P-(Y;R,U,V) USING 11.3.
C     VALUES ARE COMPUTED AT EACH Y LEVEL WHERE THE POINT GEOMETRIC
C     PHASE FUNCTION IS GIVEN (BY FUNCTION PHASEF)
C
C     NUQB AND NVQB ARE THE BASE NUMBER OF QUAD SUBDIVISIONS IN THE MU AND PHI
C     DIRECTIONS, USED FOR NUMERICAL INTEGRATION OF THE CONTINUOUS
C     PHASE FUNCTION.  THE NUMBER OF QUAD SUBDIVISIONS IS INCREASED
C     BY A FACTOR OF INCBAS IN THE FORWARD SCATTERING QUADS AND IN
C     THE ADJACENT QUADS.
C
C     THE ARRAY PHASE(I,IY) CONTAINS THE TABULATED VALUES OF
C     PHASEF(Y(IY),COS(PSI(I))
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXGEOP=MXMU*(MXPHI/2 + 1))
C
      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),
     1 BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      COMMON/CWORK/ RADSKY(MXMU,MXPHI),PHASE(2701,MXSIGY),
     1    CKSUM(MXMU,MXY)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NL = IMISC(3)
      NSIGY = IMISC(5)
      IDBUG = IMISC(9)
      TWOPI = 2.0*FMISC(1)
      RADEG = FMISC(3)
C
      DELPHI = TWOPI/FLOAT(NPHI)
      NOPI = NPHI/2
      NOPI1 = NOPI + 1
C
      DO 50 IY=1,MXSIGY
      DO 50 J=1,MXGEOP
      DO 50 I=1,MXMU
      GEOPP(I,J,IY) = 0.
   50 GEOPM(I,J,IY) = 0.
C
C     LOOP OVER THE DEPTHS (Y INDEX) WHERE THE OPTICAL PROPERTIES OF
C     THE WATER ARE DEFINED.
C
      DO 100 IY=1,NSIGY
C
C     LOOP OVER THE R, U, AND V QUAD INDICES
C
      DO 100 IU=1,NMU
C
      DO 100 IR=1,NMU
C
      NCOMPV = NOPI1
      IF(IU.EQ.NMU .OR. IR.EQ.NMU) NCOMPV = 1
      DO 100 IV=1,NCOMPV
C
C     SELECT THE SUBQUAD RESOLUTION.  IDENTICAL OR ADJACENT QUADS INVOLVE
C     FORWARD SCATTERING, AND NEED HIGHER RESOLUTION TO RESOLVE THE
C     PHASE FUNCTION ACCURATELY
C
      IF(IV.LE.2) THEN
           IF(IR.EQ.IU .OR. IR.EQ.IU+1 .OR. IR.EQ.IU-1) THEN
C          INPUT AND OUTPUT QUADS ARE IDENTICAL OR ADJACENT
           NUQ = INCBAS*NUQB
           NVQ = INCBAS*NVQB
           ELSE
           NUQ = NUQB
           NVQ = NVQB
           ENDIF
```

```
      ELSE
      NUQ = NUQB
      NVQ = NVQB
      ENDIF
C
C     BOUNDARITES OF THE MU (= IU) QUAD
      UMUMIN = 0.
      IF(IU.GT.1) UMUMIN = BNDMU(IU-1)
      DMU = DELTMU(IU)/FLOAT(NUQ)
      UO = 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
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*DMUP*DPHIP/OMEGA(IU)
C
C     BOUNDARIES OF THE PHI (= IV) QUAD
      PHIMIN = BNDPHI(NPHI)
      IF(IV.GT.1) PHIMIN = BNDPHI(IV-1)
      PHIO = PHIMIN + 0.5*DPHI
C
C     COMPUTE THE STORAGE INDEX BY(12.7)
C
      IF(IR.EQ.NMU) THEN
      KCOL = IU
      ELSE
           IF(IU.EQ.NMU) THEN
           KCOL = NMU
           ELSE
                IF(IV.LE.NL+1) THEN
                KCOL = IU + NMU*(IV-1)
                ELSE
                KCOL = IU + NMU*(NL - MOD(IV-1,NL))
                ENDIF
           ENDIF
      ENDIF
C
C     INTEGRATE OVER PHI PRIME ONLY FOR THE PHI PRIME = 0 QUADS (IS = 1)
      PHIOP = BNDPHI(NPHI) + 0.5*DPHIP
C
C     COMPUTE THE QUADRUPLE INTEGRAL (11.3) OVER THE SELECTED QUADS
C
      SUMP = 0.
      SUMM = 0.
      DO 110 JU=1,NUQ
C
C     DEFINE A MU VALUE
      UMU = UO + FLOAT(JU-1)*DMU
      ROOTJU = SQRT(1.0 - UMU*UMU)
C
      DO 110 JR=1,NUQ
C
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
C     DEFINE A PHI VALUE
      VPHI = PHIO + FLOAT(JV-1)*DPHI
C
```

```
      DO 110 JS=1,NVQ
C
C     DEFINE A PHI PRIME VALUE
      SPHIP = PHIOP + FLOAT(JS-1)*DPHIP
C
C     COMPUTE CONTRIBUTIONS TO INTEGRALS
      COSPPP = COS(VPHI - SPHIP)
C
      COSPSI = 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(PSI.LE.10.0) THEN
      IPSI = IFIX(PSI*100. + 1.5)
      ELSE
      IPSI = IFIX(PSI*10. + 901.5)
      ENDIF
      SUMP = SUMP + PHASE(IPSI,IY)
C
      COSPSI = -A1 + A2*COSPPP
      IF(ABS(COSPSI).GT.1.0) COSPSI = SIGN(1.0,COSPSI)
      PSI = RADEG*ACOS(COSPSI)
      IF(PSI.LE.10.0) THEN
      IPSI = IFIX(PSI*100. + 1.5)
      ELSE
      IPSI = IFIX(PSI*10. + 901.5)
      ENDIF
      SUMM = SUMM + PHASE(IPSI,IY)
  110 CONTINUE
C
      GEOPP(IR,KCOL,IY) = GEOPP(IR,KCOL,IY) + SUMP*FACT
  100 GEOPM(IR,KCOL,IY) = GEOPM(IR,KCOL,IY) + SUMM*FACT
C
C     COMPUTE THE CHECK ON THE TOTAL SCATTERING, (11.5)
C
      WRITE(6,208)
      DO 200 IY=1,NSIGY
      WRITE(6,212)
      DO 200 IR=1,NMU
C     POLAR CAP OUTPUT QUAD
      SUMP = (GEOPP(IR,NMU,IY) + GEOPM(IR,NMU,IY))*OMEGA(NMU)/OMEGA(IR)
      DO 202 IU=1,NMU-1
      FACTR = OMEGA(IU)/OMEGA(IR)
C     PHI = 0 VALUES
      SUMP = SUMP + (GEOPP(IR,IU,IY) + GEOPM(IR,IU,IY))*FACTR
C     PHI = PI VALUES
      KCOL = NMU*NOPI + IU
      IF(IR.EQ.NMU) KCOL = IU
      SUMP = SUMP + (GEOPP(IR,KCOL,IY) + GEOPM(IR,KCOL,IY))*FACTR
C     0 .LT. PHI .LT. PI VALUES
      DO 202 IV=2,NOPI
      KCOL = NMU*(IV-1) + IU
      IF(IR.EQ.NMU) KCOL = IU
  202 SUMP= SUMP + 2.0*(GEOPP(IR,KCOL,IY) + GEOPM(IR,KCOL,IY))*FACTR
      CKSUM(IR,IY) = SUMP
  200 WRITE(6,210) IY,IR,SUMP,GEOPP(IR,IR,IY)
C
C     USE THE CHECKSUMS TO REDEFINE THE FORWARD SCATTERING QUADS BY 11.7
C
      DO 300 IY=1,NSIGY
      DO 300 IR=1,NMU
  300 GEOPP(IR,IR,IY) = 1.0 - CKSUM(IR,IY) + GEOPP(IR,IR,IY)
C
      IF(IDBUG.GE.1) THEN
      CALL P3ARAY(GEOPP,NMU,4*NMU,NSIGY,MXMU,MXGEOP,2,
     1 'QUAD-AVERAGED P+(Y;R,1/U,V)')
      CALL P3ARAY(GEOPM,NMU,4*NMU,NSIGY,MXMU,MXGEOP,2,
     1 'QUAD-AVERAGED P-(Y;R,1/U,V)')
      ENDIF
C
  208 FORMAT(1H0,'CHECKSUMS ON QUAD-AVERAGED GEOMETRIC P+ AND P- FUNCTIO
     1NS'//1H ,2X,'Y     R     SUM (=1)       COMPUTED FWD SCAT')
  210 FORMAT(1H ,I3,I5,F11.5,1PE20.3)
  212 FORMAT(1H )
C
      RETURN
      END
```

```
      SUBROUTINE QASKY(RSKY,CARD,SHTOTL,THETAS,PHIS)
C
C     ON NHM4/QASKY
C
C     THIS ROUTINE COMPUTES THE INPUT SKY QUAD-AVERAGED RADIANCE
C     DISTRIBUTION, USING 3.3 EVALUATED AS IN APPENDIX B OF THIS TECH
C     MEMO.  SEE ALSO STEP 7A4 ON PAGE 130.
C
C     RSKY IS THE RATIO OF SKY TO TOTAL SCALAR IRRADIANCE.
C          RSKY = 0.0 FOR A BLACK SKY (SUN ONLY), RSKY = 1.0 FOR
C          NO SUN
C     CARD IS THE CARDIOIDAL PARAMETER.  CARD = 0. FOR A UNIFORM SKY,
C          CARD = 2. FOR A CARDIOIDAL SKY
C     SHTOTL IS THE TOTAL (SKY + SUN) SCALAR IRRADIANCE ON THE
C          WATER SURFACE FROM ABOVE
C     THETAS, PHIS ARE THE SUN SOURCE ANGLES (IN DEGREES, RELATIVE TO
C          PHI = 0. IN THE DOWNWIND DIRECTION)
C
C     UPON RETURN, RADSKY IN /CWORK/ HOLDS THE QUAD-AVERAGED SKY
C          RADIANCES FOR USE IN AMPA0 IN MAIN.
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
C
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1 BNDPHI(MXPHI),OMEGA(MXMU)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      COMMON/CWORK/ RADSKY(MXMU,MXPHI),THETAB(MXMU),PHIB(MXPHI)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      IDBUG = IMISC(9)
      PI = FMISC(1)
      RADEG = FMISC(3)
C
C     SET UP THE BACKGROUND SKY QUAD-AVERAGED RADIANCES USING B.7
C
      WRITE(6,500) SHTOTL,RSKY,CARD
      FN0 = RSKY*SHTOTL/(2.0*PI*(1.0 + 0.5*CARD))
      DO 100 I=1,NMU-1
      RAD = FN0*(1.0 + CARD*FMU(I))
      DO 100 J=1,NPHI
  100 RADSKY(I,J) = RAD
C     POLAR CAP
      RADSKY(NMU,1) = FN0*(1.0 + CARD*FMU(NMU))
C
C     ADD IN THE SUN TO THE APPROPRIATE QUAD USING B.8
C
      WRITE(6,502) THETAS,PHIS
C
C     CONVERT THE BOUNDARY MU AND PHI VALUES TO DEGREES
      DO 101 I=1,NMU
  101 THETAB(I) = RADEG*ACOS(BNDMU(I))
      DO 103 J=1,NPHI
  103 PHIB(J) = RADEG*BNDPHI(J)
C
C     DETERMINE THE (MU,PHI) INDICES OF THE QUAD CONTAINING THE SUN
C
      PH = AMOD(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(1)) IMUS = 1
C
      DO 202 J=1,NPHI
      IF(PH.LT.PHIB(J)) GO TO 206
  202 CONTINUE
      J = 1
  206 JPHIS = J
  200 CONTINUE
C
      THE = RADEG*ACOS(FMU(IMUS))
      WRITE(6,510) IMUS,JPHIS,THE,RADEG*PHI(JPHIS)
C
```

```
C     CHANGE PHI INDEX FROM SOURCE LOCATION TO BEAM DIRECTION
C
      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)
C
      IF(IDBUG.NE.0) CALL P2ARAY(RADSKY,NMU,NPHI,MXMU,2,
     1  'QUAD-AVERAGED SKY RADIANCES')
C
      RETURN
C
C     FORMATS
C
  500 FORMAT(1H1,' THE INPUT RADIANCE DISTRIBUTION HAS'//
     15X,'TOTAL SCALAR IRRADIANCE (SUN + SKY) =',1PE10.3,
     2' WATTS PER SQUARE METER'//
     35X,'RATIO OF SKY TO TOTAL SCALAR IRRADIANCE, R =',0PF6.3//
     45X,'CARDIOIDAL PARAMETER, C =',F6.3)
  502 FORMAT(//' THE SUN IS REQUESTED AT SKY LOCATION (THETA, PHI) = (',
     1 F4.1,',',F5.1,')')
  510 FORMAT(//' THE SUN IS PLACED IN QUAD Q(R,S) = Q(',I2,',',I2,
     1') CENTERED AT (THETA, PHI) = (',F6.3,',',F7.3,')')
      END
```

```
      SUBROUTINE RFMPAK(X,Y,Z,IROW,NMU,L)
C
C     ON NHM4/RFMPAK
C
C     THIS ROUTINE FORMS THE MATRIX PRODUCT X * Y = Z, WHERE X AND Z ARE
C     ROW VECTORS AND Y IS A BLOCK MATRIX STORED ON THE PACKED FORMAT
C     OF 12.4.
C
      DIMENSION X(1),Y(IROW,1),Z(1)
C
      LP1 = L + 1
      DO 100 JZB=1,LP1
      JZB2 = L/2 + 1
      IF(MOD(JZB,2).EQ.0) JZB2 = (L+1)/2
      J2 = (JZB-1)*NMU
      JY = ((JZB+1)/2 - 1)*NMU
C
      DO 200 I=1,NMU
      SUM = 0.
C
      DO 301 KX=1,JZB2
      KY = 2*KX - 1
      IF(MOD(JZB,2).EQ.0) KY = 2*KX
      I2 = (KY - 1)*NMU
      DO 301 K=1,NMU
  301 SUM = SUM + X(I2+K)*Y(I2+K,JY+I)
  200 Z(J2+I) = SUM
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE RHOTAU(L)
C
C     ON NHM4/RHOTAU
C
C     THIS ROUTINE COMPUTES THE DISCRETIZED SPECTRAL PHASE FUNCTIONS
C     RHOHAT AND TAUHAT FROM THE QUAD-AVERAGED GEOMETRIC SCATTERING
C     FUNCTIONS.  THE GOVERNING EQUATIONS ARE 5.6 AND 5.20B TO 5.20E.
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXGEOP=MXMU*(MXPHI/2 + 1))
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY)
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI)
      COMMON/CRTSIG/ RHOHAT(MXMU,MXMU,MXSIGY),TAUHAT(MXMU,MXMU,MXSIGY),
     1 GEOPP(MXMU,MXGEOP,MXSIGY),GEOPM(MXMU,MXGEOP,MXSIGY)
      COMMON/CMISC/ IMISC(20)
      COMMON/CWORK/ COSLPV(MXPHI)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NL = IMISC(3)
      NSIGY = IMISC(5)
C
      DO 100 IV=1,NPHI
  100 COSLPV(IV) = COS(FLOAT(L)*PHI(IV))
C
      IF(L.EQ.0 .OR. L.EQ.NL) THEN
      EPSL = FLOAT(NPHI)
      ELSE
      EPSL = FLOAT(NL)
      ENDIF
C
C     LOOP OVER THE DEPTHS WHERE THE INHERENT OPTICAL PROPERTIES ARE GIVEN
C
      DO 200 IY=1,NSIGY
      ALBEDO = ALBESS(IY)
C
C     POLAR CAP OUTPUT, IU = NMU
C
      IF(L.EQ.0) THEN
C
      FMU1 = 1.0/FMU(NMU)
C     QUAD INPUT:  USE 5.20D.  PHAT IS GIVEN BY 5.6C
      DO 300 IR=1,NMU-1
      RHOHAT(IR,NMU,IY) = ALBEDO*EPSL*GEOPM(IR,NMU,IY)*FMU1
  300 TAUHAT(IR,NMU,IY) = ALBEDO*EPSL*GEOPP(IR,NMU,IY)*FMU1
C
C     POLAR CAP INPUT: USE 5.20E.  PHAT IS GIVE BY 5.6D
      RHOHAT(NMU,NMU,IY) = ALBEDO*GEOPM(NMU,NMU,IY)*FMU1
      TAUHAT(NMU,NMU,IY) = (ALBEDO*GEOPP(NMU,NMU,IY) - 1.0)*FMU1
C
      ELSE
C
      DO 302 IR=1,NMU
      RHOHAT(IR,NMU,IY) = 0.
  302 TAUHAT(IR,NMU,IY) = 0.
      ENDIF
C
C     QUAD (NON-POLAR CAP) OUTPUT
C
      DO 310 IU=1,NMU-1
      FMU1 = 1.0/FMU(IU)
C
C     POLAR CAP INPUT, IR=NMU.  USE 5.20C.  PHAT IS GIVEN BY 5.6B
C
      IF(L.EQ.0) THEN
      RHOHAT(NMU,IU,IY) = ALBEDO*GEOPM(NMU,IU,IY)*FMU1
      TAUHAT(NMU,IU,IY) = ALBEDO*GEOPP(NMU,IU,IY)*FMU1
      ELSE
      RHOHAT(NMU,IU,IY) = 0.
      TAUHAT(NMU,IU,IY) = 0.
      ENDIF
C
```

```
C     QUAD (NON-POLAR CAP) INPUT; USE 5.20B.  PHAT MUST NOW BE
C     COMPUTED BY 5.6A
C
      DO 310 IR=1,NMU-1
      SUMP = 0.
      SUMM = 0.
      DO 400 IV=1,NPHI
C
C     COMPUTE STORAGE INDICES BY (12.7)
C
      IF(IV.LE.NL+1) THEN
      J = IU + NMU*(IV-1)
      ELSE
      J = IU + NMU*(NL - MOD(IV-1,NL))
      ENDIF
C
      SUMP = SUMP + GEOPP(IR,J,IY)*COSLPV(IV)
  400 SUMM = SUMM + GEOPM(IR,J,IY)*COSLPV(IV)
C
      RHOHAT(IR,IU,IY) = ALBEDO*SUMM*FMU1
      IF(IR.EQ.IU) THEN
      DELT = 1.
      ELSE
      DELT = 0.
      ENDIF
  310 TAUHAT(IR,IU,IY) = (ALBEDO*SUMP - DELT)*FMU1
C
  200 CONTINUE
C
      RETURN
C
      END
```

```
      SUBROUTINE RICATI(L)
C
C     ON NHM4/RICATI
C
C     THIS ROUTINE SOLVES FOR THE ARRAYS RYX = R(Y,X) AND TXY = T(X,Y)
C     BY INTEGRATING 6.43 AND 6.44 IN A DOWNWARD
C     SWEEP WITH INITIAL VALUES OF R(X,X) = 0 AND T(X,X) = I, BY 6.47.
C     RYB = R(Y,B) IS FOUND BY INTEGRATING 6.48 IN AN UPWARD SWEEP
C     WITH INITIAL CONDITION R(Z,B) = RHAT(Z,B), BY 6.58.
C
C     THE ARRAYS RYX AND TXY ARE STORED IN THE VECTOR RT AS FOLLOWS
C     (FOR A GIVEN Y VALUE):
C
C     RYX(I,J) IS RT(I + (J-1)*NMU)
C     TXY(I,J) IS RT(I + (J-1)*NMU + NMU*NMU)
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
      PARAMETER(MXEQN=2*MXMU*MXMU)
C
      DIMENSION CDVERK(24)
C
```

```
      COMMON/CRTR/ RYX(MXMU,MXMU,MXY),TXY(MXMU,MXMU,MXY),
     1  R1YB(MXMU,MXMU,MXY),R2YB(MXMU,MXMU,MXY)
      COMMON/CBOTBC/ RHATZB(MXMU,MXMU)
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      COMMON/CWORK/ WERK(MXEQN,9),RT(MXEQN)
C
C     SUBROUTINE DRTAB EVALUATES THE RHS OF 6.43, 6.44 AND 6.48
      EXTERNAL DRTAB
C
      NMU = IMISC(1)
      NY = IMISC(4)
      IDBUG = IMISC(9)
      IBOTM = IMISC(12)
      NMU2 = NMU*NMU
      NEQNS = 2*NMU2
      TOL = FMISC(7)
      IF(IDBUG.GE.1) WRITE(6,3002) TOL
C
C     INITIALIZE THE ARRAYS AT Y = X USING 6.47
C
      DO 500 I=1,NMU
      DO 500 J=1,NMU
      RYX(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)
      CDPREV = 0.
      IMISC(13) = 1
      IND = 1
C
C     INTEGRATE 6.43 AND 6.44 TO FIND R(Y,X) AND T(X,Y) AT EACH Y LEVEL
C
      DO 520 IY=2,NY
      YEND = YSTART + Y(IY) - Y(IY - 1)
      IF(IDBUG.GE.1) WRITE(6,3000) YSTART,YEND
C
      CALL DVERK(NEQNS,DRTAB,YSTART,RT,YEND,TOL,IND,CDVERK,MXEQN,
     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
C     SAVE THE SOLUTION AT Y = YEND
C
      DO 520 J=1,NMU
      DO 520 I=1,NMU
      RYX(I,J,IY) = RT(I+(J-1)*NMU)
  520 TXY(I,J,IY) = RT(I + (J-1)*NMU + NMU2)
C
C     INTEGRATE 6.48 FROM Z TO X TO FIND R1(Y,B) AT EACH Y LEVEL
C
C     INITIALIZE AT Y = Z WITH R1(Z,B) = RHAT1(Z,B), USING 6.58
C
      DO 550 J=1,NMU
      DO 550 I=1,NMU
      R1YB(I,J,NY) = RHATZB(I,J)
  550 RT(I+(J-1)*NMU) = RHATZB(I,J)
C
      YSTART = Y(NY)
      NEQNS = NMU2
      CDPREV = 0.
      IND = 1
      IMISC(13) = 2
C     INTEGRATE
      DO 570 IY=1,NY-1
      IYREV = NY - IY
      YEND = YSTART - Y(IYREV+1) + Y(IYREV)
      IF(IDBUG.GE.1) WRITE(6,3000) YSTART,YEND
C
```

```
      CALL DVERK(NEQNS,DRTAB,YSTART,RT,YEND,TOL,IND,CDVERK,MXEQN,
     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
C     SAVE THE SOLUTION AT YEND
      DO 570 J=1,NMU
      DO 570 I=1,NMU
  570 R1YB(I,J,IYREV) = RT(I+(J-1)*NMU)
C
C     USE R2(Y,B) = R1(Y,B) OR INTEGRATE 6.48 AGAIN, ACCORDING TO THE
C     BOTTOM TYPE
C
      IF(IBOTM.EQ.0 .AND. L.EQ.0) THEN
C
C     MATTE BOTTOM WITH L = 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 = Y(NY)
      NEQNS = NMU2
      CDPREV = 0.
      IND = 1
      IMISC(13) = 2
C     INTEGRATE
      DO 597 IY=1,NY-1
      IYREV = NY - IY
      YEND = YSTART - Y(IYREV+1) + Y(IYREV)
      IF(IDBUG.GE.1) WRITE(6,3000) YSTART,YEND
C
      CALL DVERK(NEQNS,DRTAB,YSTART,RT,YEND,TOL,IND,CDVERK,MXEQN,
     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
C     SAVE THE SOLUTION AT YEND
      DO 597 J=1,NMU
      DO 597 I=1,NMU
  597 R2YB(I,J,IYREV) = RT(I+(J-1)*NMU)
C
      ELSE
C
C     MATTE BOTTOM WITH L.GT.0 OR INFINITELY DEEP, HOMOGENEOUS LAYER.
C     USE R2(Y,B) = R1(Y,B)
C
      DO 600 IY=1,NY
      DO 600 J=1,NMU
      DO 600 I=1,NMU
  600 R2YB(I,J,IY) = R1YB(I,J,IY)
C
      ENDIF
C
      RETURN
C
 1060 FORMAT(/' SUB RICATI:  ERROR IN CALL TO DVERK:  IDE =',I5,5X,
     1 'IND = ,I5,5X,'IER =',I5)
 3000 FORMAT(' YSTART =',F8.4,5X, YEND =',F8.4)
 3001 FORMAT(1H+,T40,I4,' DERIVATIVE EVALUATIONS')
 3002 FORMAT(/////' OUTPUT FROM INTEGRATION ROUTINE DVERK (TOL =',
     1  1PE12.3,')')
      END
```

```
      SUBROUTINE Y2ZGEO
C
C     ON NHM4/Y2ZGEO
C
C     THIS ROUTINE COMPUTES THE GEOMETRICAL DEPTHS ZETA (IN METERS) WHICH
C     CORRESPOND TO THE OPTICAL DEPTHS Y (NONDIMENSIONAL) WHERE
C     OUTPUT IS REQUESTED.
C     EQUATION 7.1 IS INTEGRATED, WHEREIN ALPHA IS A FUNCTION OF OPTICAL
C     DEPTH Y
C
      PARAMETER (MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
C
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),YOUT(MXY),BNDMU(MXMU),
     1 BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CMISC/ IMISC(20)
C
      EXTERNAL FALPHA
C
      DATA AERR,RERR/0.0, 1.0E-8/
C
      NY = IMISC(4)
      NSIGY = IMISC(5)
      ALPHA1 = ALBESS(1)/TOTALS(1)
C
      IF(NSIGY.EQ.1) THEN
C
C     WATER COLUMN IS UNIFORM WITH DEPTH
C
      DO 100 IY=1,NY
  100 ZGEO(IY) = ALPHA1*YOUT(IY)
C
      ELSE
C
C     WATER COLUMN HAS VARIABLE OPTICAL PROPERTIES WITH DEPTH:  INTEGRATE
C     DZETA = DY/ALPHA(Y)
C
      ZGEO(1) = ALPHA1*YOUT(1)
      DO 200 IY=2,NY
  200 ZGEO(IY) = ZGEO(IY-1) + DCADRE(FALPHA,YOUT(IY-1),YOUT(IY),
     1  AERR,RERR,ERROR,IERR)
C
      ENDIF
      RETURN
      END
```

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-absorbtion 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) $\equiv$ x = 0.0 and YOUT(NY) $\equiv$ 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.

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,···,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 DO 302 I=IPRAD1, IPRAD2, IPRAD3 which select the μ-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 μ-bands in each hemisphere (μ_u, u = 1,···,m). If (IPRAD1, IPRAD2, IPRAD3) = (1, 10, 1) then all μ-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 ϕ-bands to be printed, ϕ_v, v = 1,···,2n. Referring again to 75/Fig. 4a, if (JPRAD1, JPRAD2, JPRAD3) = (1, 24, 1) then all ϕ-bands would be printed. If (JPRAD1, JPRAD2, JPRAD3) = (1, 24, 6) then only the ϕ-bands at v = 1, 7, 13, 19 (corresponding to ϕ = 0°, 90°, 180°, 270° 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,···,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.

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 ϕ 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 $\mu(I)$ is in Ξ_- if $I \le NMU$
and $\mu(I)$ is in Ξ_+ if $NMU < I \le 2*NMU$

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) $\le$ 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; $1 \le$ IYIRAD(j) $\le$ 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. $dy \doteq y_{j+1} - y_j$ = 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 specifiying 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),···,YOUT(NY-1)
IPKFCN	= 0	if the K-functions are to be computed at levels y_j, j = 1, 3, 5, 7,···. 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,···. One could then accurately compute K-functions at levels 0.005, 0.505, 1.005, 2.005, 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 μ-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 ϕ-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,···,IPKFCN, are the indices of the y_j-levels where the K-functions are to be computed; $1 \le$ IYKFCN(j) $\le$ NY-1.

C. File Management

Three files are read by Program 5, and one is written.

symbolic name	external name	description
NURAX	TAPE22	the quad-averaged geometric r(a,x) array from Program 2
NUTXA	TAPE25	the quad-averaged geometrix t(x,a) array from Program 2
NUIN	TAPE40	the radiance amplitudes and other information, generated by Program 4
NUOUT	TAPE50	a file containing discretized phase functions, radiances, and other information, for use by the graphics routines

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

D. Code Listing

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1             TAPE40,TAPE50,TAPE22,TAPE25)
C
C     ON NHM5/MAIN5
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                +
C     +  THIS IS PROGRAM 5 OF THE NATURAL HYDROSOL MODEL +
C     +                                                +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     THIS PROGRAM TAKES THE SPECTRAL AMPLITUDES GENERATED BY PROGRAM 4
C     AND SYNTHESIZES THE GEOMETRIC, QUAD-AVERAGED RADIANCE FIELDS.
C     VARIOUS DERIVED QUANTITIES ARE ALSO COMPUTED, IF DESIRED
C
C     RAXGEO (TAPE22) AND TXAGEO (TAPE25), THE QUAD-AVERAGED GEOMETRIC
C     R(A,X) AND T(X,A) ARRAYS, ARE REQUIRED IF THE CONTRAST
C     TRANSMITTANCE IS TO BE COMPUTED
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXL=MXPHI/2, MXGEOP=MXMU*(MXL+1), MXAMP=2*MXMU*(MXL+1))
      PARAMETER(MXRRTH=MXMU*(MXL+1), MXCRTH=MXMU*((MXL+2)/2))
      PARAMETER(MXWERK=MXMU*(MXL+1)*(1+2*MXMU)+5*MXY)
C
      DIMENSION IYRTE(MXY),IYPRAD(MXY),IYIRAD(MXY),IYKFCN(MXY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CAMPO/ AOAM(MXAMP),AOYM(MXAMP,MXY),AOAP(MXAMP)
      COMMON/CGEOP/ GEOPP(MXMU,MXGEOP,MXSIGY),GEOPM(MXMU,MXGEOP,MXSIGY)
      COMMON/CAMP/ AAM(MXAMP),AAP(MXAMP),AYM(MXAMP,MXY),AYP(MXAMP,MXY)
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1  BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(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/CIRRAD/ SHP(0:MXY),SHM(0:MXY),SCAPHP(0:MXY),SCAPHM(0:MXY),
     1               DPY(0:MXY),DMY(0:MXY)
      COMMON/CKRAD/ ISTART,ISTOP,ISTEP,JSTART,JSTOP,JSTEP
      COMMON/CMISC/ IMISC(20),FMISC(20)
      COMMON/CWORK/ WERK(MXWERK)
C
      DATA NUOUT/50/
C
C     INITIALIZE THE PROGRAM
C
      CALL INISHL(IRTECK,IYRTE,IPRAD,IYPRAD,IPIRAD,IYIRAD,IPKFCN,IYKFCN)
C
C     AAM AND AYM NOW CONTAIN DIFFUSE AMPLITUDES (SAME FOR AAP AND AYP)
C
C     COMPUTE IRRADIANCE QUANTITIES FROM THE L = 0 AMPLITUDES
C
      CALL IRRAD(IPIRAD,IYIRAD)
C
      NY = IMISC(4)
C
C     COMPUTE THE DIFFUSE RADIANCES AT Y = A, X,..., Z
C
      CALL SYNRAD(AAM,RADAM,MXMU)
      DO 100 K=1,NY
  100 CALL SYNRAD(AYM(1,K),RADM(1,1,K),MXMU)
C
      CALL SYNRAD(AAP,RADAP,MXMU)
      DO 110 K=1,NY
  110 CALL SYNRAD(AYP(1,K),RADP(1,1,K),MXMU)
C
C     COMPUTE THE DOWNWARD DIRECT RADIANCE AT Y = A, X,..., Z
```

```
C
      CALL SYNRAD(AOAM,RADOAM,MXMU)
      DO 120 K=1,NY
  120 CALL SYNRAD(AOYM(1,K),RADOM(1,1,K),MXMU)
C
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,MXMU)
C
C     PRINT SELECTED RADIANCES AND COMPUTE THE RADIANCE-IRRADIANCE RATIOS
C
      CALL RADY(IPIRAD,IYIRAD)
C
C     COMPUTE THE CONTRAST TRANSMITTANCE
C
      CALL CONTRM
C
C     COMPUTE THE K FUNCTIONS FOR IRRADIANCE
C
      CALL KFCN(IPKFCN,IYKFCN)
C
C     COMPUTE THE K-FUNCTIONS FOR RADIANCE
C
      IF(ISTART.GT.0) CALL KRAD(IPKFCN,IYKFCN)
C
C     COMPUTE THE BACKSCATTER AND FORWARD SCATTER FUNCTIONS
C
      CALL BFSCAT(IPIRAD,IYIRAD)
C
C     PRINT OUT THE RADIANCES
C
      IF(IPRAD.GT.0) CALL PRINT(IPRAD,IYPRAD)
C
C     CHECK THE BALANCE OF THE RTE AT INTERIOR Y VALUES
C
      IF(IRTECK.NE.0) CALL RTECK(IRTECK,IYRTE)
C
C     SAVE THE RADIANCE INFORMATION FOR ANALYSIS BY THE PLOTTING PROGRAM
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NY = IMISC(4)
      NSIGY = IMISC(5)
      KCOL = IMISC(10)
C
      REWIND NUOUT
      WRITE(NUOUT) IMISC,FMISC,FMU,PHI,Y,BNDMU,BNDPHI,OMEGA,DELTMU,
     1  YSIG,ALBESS,TOTALS,ZGEO
      WRITE(NUOUT) (((GEOPP(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
      WRITE(NUOUT) (((GEOPM(I,J,K),I=1,NMU),J=1,KCOL),K=1,NSIGY)
      WRITE(NUOUT) ((RADAP(I,J),I=1,NMU),J=1,NPHI)
      WRITE(NUOUT) (((RADP(I,J,K),I=1,NMU),J=1,NPHI),K=1,NY)
      WRITE(NUOUT) ((RADAM(I,J),I=1,NMU),J=1,NPHI)
      WRITE(NUOUT) (((RADM(I,J,K),I=1,NMU),J=1,NPHI),K=1,NY)
      WRITE(NUOUT) ((RADOAP(I,J),I=1,NMU),J=1,NPHI)
      WRITE(NUOUT) ((RADOAM(I,J),I=1,NMU),J=1,NPHI)
      WRITE(NUOUT) (((RADOM(I,J,K),I=1,NMU),J=1,NPHI),K=1,NY)
C     ZERO-MODE AMPLITUDES
      WRITE(NUOUT) (AOAM(I),I=1,NMU),(AOAP(I),I=1,NMU),
     1  ((AOYM(I,K),I=1,NMU),K=1,NY)
      WRITE(NUOUT) (AAM(I),I=1,NMU),(AAP(I),I=1,NMU),
     1  ((AYM(I,K),I=1,NMU),K=1,NY),((AYP(I,K),I=1,NMU),K=1,NY)
      ENDFILE NUOUT
      WRITE(6,200) NUOUT
C
  200 FORMAT(1H0,' TAPE',I2,' WRITTEN.')
      END
```

```
      SUBROUTINE INISHL(IRTECK,IYRTE,IPRAD,IYPRAD,IPIRAD,IYIRAD,
     1  IPKFCN,IYKFCN)
C
C     ON NHM5/INISHL5
C
C     THIS ROUTINE INITIALIZES PROGRAM 5 OF THE NHM
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXL=MXPHI/2, MXGEOP=MXMU*(MXL+1), MXAMP=2*MXMU*(MXL+1))
      PARAMETER(MXRRTH=MXMU*(MXL+1), MXCRTH=MXMU*((MXL+2)/2))
C
      DIMENSION IYRTE(MXY),IYPRAD(MXY),IYIRAD(MXY),IYKFCN(MXY)
      DIMENSION ITITLE(10)
      COMMON/CGEOP/ GEOPP(MXMU,MXGEOP,MXSIGY),GEOPM(MXMU,MXGEOP,MXSIGY)
      COMMON/CAMPO/ AOAM(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),
     1  BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CKRAD/ ISTART,ISTOP,ISTEP,JSTART,JSTOP,JSTEP
      COMMON/CPRAD/ IPRAD1,IPRAD2,IPRAD3,JPRAD1,JPRAD2,JPRAD3
      COMMON/CMISC/ IMISC(20),FMISC(20)
C
      DATA NUIN/40/, IDBUG/0/
C
C     READ HEADER RECORDS FROM THE AMPLITUDE FILE
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)
      NRHAT = IMISC(10)
      NRAMP = 2*NRHAT
      RADEG = FMISC(3)
      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
C     READ IN PARAMETERS FOR RADIANCE ANALYSIS
C
      READ(5,50) ITITLE
      WRITE(6,1000) ITITLE
C
C     READ SPECIFICATIONS FOR RADIANCE PRINTOUT
C
      READ(5,*) IPRAD,IPRAD1,IPRAD2,IPRAD3,JPRAD1,JPRAD2,JPRAD3
      IF(IPRAD.LT.0) THEN
      DO 106 IY=1,NY
  106 IYPRAD(IY) = IY
      IPRAD = NY
      ELSEIF(IPRAD.GT.0) THEN
      READ(5,*) (IYPRAD(IY),IY=1,IPRAD)
      ENDIF
C
C     READ SPECIFICATIONS FOR RTE CHECK
C
      READ(5,*) IRTECK,NIC,NJC
      IF(IRTECK.LT.0) THEN
      DO 108 IY=1,NY
  108 IYRTE(IY) = IY
      ELSEIF(IRTECK.GT.0) THEN
      READ(5,*) (IYRTE(IY),IY=1,IRTECK)
      ENDIF
C
```

```
C     READ SPECIFICATIONS FOR IRRADIANCE OUTPUT
C
      READ(5,*) IPIRAD
C
      IF(IPIRAD.LT.0) THEN
      DO 110 IY=1,NY
  110 IYIRAD(IY) = IY
      IPIRAD = NY
      ELSEIF(IPIRAD.EQ.0) THEN
      IYIRAD(1) = 1
      IPIRAD = 1
      DO 112 IY=2,NY,2
      IPIRAD = IPIRAD + 1
  112 IYIRAD(IPIRAD) = IY
      ELSE
      READ(5,*) (IYIRAD(IY),IY=1,IPIRAD)
      ENDIF
C
C     READ SPECIFICATIONS FOR K-FUNCTION OUTPUT AND RADIANCE K-FUNCTION DIRECTIONS
C
      READ(5,*) IPKFCN,ISTART,ISTOP,ISTEP,JSTART,JSTOP,JSTEP
C
      IF(IPKFCN.LT.0) THEN
      DO 114 IY=1,NY-1
  114 IYKFCN(IY) = IY
      IPKFCN = NY - 1
      ELSEIF(IPKFCN.EQ.0) THEN
      IPKFCN = 0
      DO 116 IY=1,NY-1,2
      IPKFCN = IPKFCN + 1
  116 IYKFCN(IPKFCN) = IY
      ELSE
      READ(5,*) (IYKFCN(IY),IY=1,IPKFCN)
      ENDIF
C
      IMISC(9) = IDBUG
      IMISC(15) = NIC
      IMISC(16) = NJC
C
C     RECORDS WRITTEN BY MAIN4 (DIRECT BEAM)
      READ(NUIN) (AOAM(I),I=1,NRAMP)
      DO 15 J=1,NY
   15 READ(NUIN) (AOYM(I,J),I=1,NRAMP)
      READ(NUIN) (AOAP(I),I=1,NRAMP)
C
C     (TOTAL BEAM)
      READ(NUIN) (AAM(I),I=1,NRAMP)
      READ(NUIN) (AAP(I),I=1,NRAMP)
      DO 16 J=1,NY
   16 READ(NUIN) (AYM(I,J),I=1,NRAMP)
      DO 17 J=1,NY
   17 READ(NUIN) (AYP(I,J),I=1,NRAMP)
C
C     PRINTOUT
C
      IF(IDBUG.GT.0) THEN
      WRITE(6,1018)
      WRITE(6,1022)
      DO 1020 I=1,NMU
      THETA = RADEG*ACOS(FMU(I))
 1020 WRITE(6,1024) I,THETA,FMU(I)
C
      WRITE(6,1026)
      DO 1028 J=1,NPHI
 1028 WRITE(6,1030) J,RADEG*PHI(J)
C
      WRITE(6,1032)
      DO 1034 K=1,NY
 1034 WRITE(6,1036) K,Y(K)
      ENDIF
C
```

```
      IF(IDBUG.GE.2) THEN
      WRITE(6,1038)
      CALL PNTAMP(Y,A0AM,A0YM,MXAMP)
      WRITE(6,1039)
      CALL PNTAMP(Y,A0AP,1.E201,MXAMP)
      WRITE(6,1040)
      CALL PNTAMP(Y,AAM,AYM,MXAMP)
      WRITE(6,1042)
      CALL PNTAMP(Y,AAP,AYP,MXAMP)
      ENDIF
C
C     CONVERT THE DOWNWARD TOTAL AMPLITUDES TO DIFFUSE AMPLITUDES
C     AT Y = A, X,..., Z BY 8.23
C     CONVERT THE UPWARD TOTAL AMPLITUDES TO DIFFUSE AMPLITUDES AT Y = A.
C     THE UPWARD TOTAL = THE UPWARD DIFFUSE FOR Y = X,..., Z.
C
      DO 600 I=1,NRAMP
      AAM(I) = AAM(I) - A0AM(I)
      AAP(I) = AAP(I) - A0AP(I)
      DO 600 K=1,NY
  600 AYM(I,K) = AYM(I,K) - A0YM(I,K)
      IF(IDBUG.GE.2) THEN
      WRITE(6,1044)
      CALL PNTAMP(Y,AAM,AYM,MXAMP)
      WRITE(6,1046)
      CALL PNTAMP(Y,AAP,AYP,MXAMP)
      ENDIF
C
      RETURN
C
C     FORMATS
C
   50 FORMAT(10A8)
 1000 FORMAT(1H1,' PROGRAM 5 OF THE NATURAL HYDROSOL MODEL'//
     1'  SYNTHESIS AND ANALYSIS OF THE RADIANCE FIELDS'//
     2'  RUN TITLE: ',10A8)
 1018 FORMAT(1H0,' THE RADIANCE FIELDS ARE COMPUTED AT THE FOLLOWING GRI
     1D VALUES:')
 1022 FORMAT(1H0,' THE THETA VALUES ARE'//'      I      THETA',6X,'MU')
 1024 FORMAT(1H ,I5,F10.3,F10.4)
 1026 FORMAT(1H0,' THE PHI VALUES ARE'//'      J      PHI')
 1030 FORMAT(1H ,I5,F10.3)
 1032 FORMAT(1H0,' THE Y VALUES ARE'//'      K    OPT DEPTH')
 1036 FORMAT(1H ,I5,4X,F7.4)
 1038 FORMAT(1H1,' THE DOWNWARD DIRECT BEAM RADIANCE AMPLITUDES ARE'//
     1 11X,'MU',7X,'A0(A,-)',8X,'A0(Y,-)')
 1039 FORMAT(1H1,' THE UPWARD DIRECT BEAM RADIANCE AMPLITUDES ARE'//
     1 11X,'MU',7X,'A0(A,+)')
 1040 FORMAT(1H1,' THE DOWNWARD TOTAL RADIANCE AMPLITUDES ARE'//
     1  11X,'MU',7X,'A(A,-)',9X,'A(Y,-)')
 1042 FORMAT(1H1,' THE UPWARD TOTAL RADIANCE AMPLITUDES ARE'
     1//11X,'MU',7X,'A(A,+)',9X,'A(Y,+)')
 1044 FORMAT(1H1,' THE DOWNWARD DIFFUSE RADIANCE AMPLITUDES ARE'//
     1 11X,'MU',7X,'A*(A,-)',8X,'A*(Y,-)')
 1046 FORMAT(1H1,' THE UPWARD DIFFUSE RADIANCE AMPLITUDES ARE'//
     1 11X,'MU',7X,'A*(A,+)',8X,'A*(Y,+)')
      END
```

```
      SUBROUTINE BFSCAT(IPIRAD,IYIRAD)
C
C     ON NHM5/BFSCAT
C
C     THIS ROUTINE COMPUTES THE BACKSCATTER FUNCTIONS B(Z,+) AND B(Z,-)
C     USING 8.15.  FORWARD SCATTER
C     FUNCTIONS F(Z,+) AND F(Z,-) ARE COMPUTED FROM EQ. 8.16.
C     COMPUTED VALUES ARE CHECKED USING EQ. 8.17.
C     THE ECCENTRICITIES ARE ALSO COMPUTED.
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXL=MXPHI/2, MXGEOP=MXMU*(MXL+1))
C
      DIMENSION IYIRAD(MXY)
C
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1  BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(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/CGEOP/ GEOPP(MXMU,MXGEOP,MXSIGY),GEOPM(MXMU,MXGEOP,MXSIGY)
      COMMON/CIRRAD/ HP(0:MXY),HM(0:MXY),CAPHP(0:MXY),CAPHM(0:MXY),
     1  DPY(0:MXY),DMY(0:MXY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      COMMON/CWORK/ GEOPPY(MXMU,MXGEOP),GEOPMY(MXMU,MXGEOP),BZPY(MXY),
     1 BZMY(MXY),FZPY(MXY),FZMY(MXY),SY(MXY)
      DATA EPS/1.E-12/
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NSIGY = IMISC(5)
      NRHAT = IMISC(10)
      NOPI = NPHI/2
      WRITE(6,300)
C
      DO 99 IIY=1,IPIRAD
      IY = IYIRAD(IIY)
      SUMBP = 0.
      SUMBM = 0.
      SUMFP = 0.
      SUMFM = 0.
      BZP = -1.
      BZM = -1.
      FZP = -1.
      FZM = -1.
      SMFMBP = -1.
      BBARP = -1.
      BBARM = -1.
      SMFMBM = -1.
C
      YNOW = Y(IY)
C
C     COMPUTE THE QUAD-AVERAGED GEOMETRIC PHASE FUNCTION AT THE NEEDED
C     Y VALUE BY LINEAR INTERPOLATION OF THE KNOWN VALUES
C
      IF(NSIGY.EQ.1 .OR. YNOW.LE.YSIG(1)) THEN
      S = TOTALS(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
      S = TOTALS(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.LT.YSIG(JY)) GO TO 56
   55 CONTINUE
   56 DY = (YNOW - YSIG(JY-1))/(YSIG(JY) - YSIG(JY-1))
C
      S = (1.0 - DY)*TOTALS(JY-1) + DY*TOTALS(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)
C
      ENDIF
C
      SY(IY) = S
      DO 100 IU=1,NMU
      QUV = OMEGA(IU)
      IVMAX = NPHI
      IF(IU.EQ.NMU) IVMAX = 1
      DO 100 IV=1,IVMAX
      SUMBP2 = 0.
      SUMBM2 = 0.
      SUMFP2 = 0.
      SUMFM2 = 0.
C
      DO 200 IR=1,NMU
      ISMAX = NPHI
      IF(IR.EQ.NMU) ISMAX = 1
      DO 200 IS=1,ISMAX
C
C     COMPUTE THE STORAGE INDEX FOR P-(R,U,V) AND P+(R,U,V) BY 12.7
C
      IVS = IABS(IV-IS)
      IF(IR.EQ.NMU) THEN
      KCOL = IU
      ELSE
           IF(IU.EQ.NMU) THEN
           KCOL = NMU
           ELSE
                IF(IVS.LE.NOPI) THEN
                KCOL = IU + NMU*IVS
                ELSE
                KCOL = IU + NMU*(NOPI - MOD(IVS,NOPI))
                ENDIF
           ENDIF
      ENDIF
C
      PP = GEOPPY(IR,KCOL)
      PM = GEOPMY(IR,KCOL)
C
      RPTOTL = RADP(IR,IS,IY)
      RMTOTL = RADM(IR,IS,IY) + RADOM(IR,IS,IY)
C
      SUMBP2 = SUMBP2 + RPTOTL*PM
      SUMBM2 = SUMBM2 + RMTOTL*PM
      SUMFP2 = SUMFP2 + RPTOTL*PP
  200 SUMFM2 = SUMFM2 + RMTOTL*PP
C
      SUMBP = SUMBP + QUV*SUMBP2
      SUMBM = SUMBM + QUV*SUMBM2
      SUMFP = SUMFP + QUV*SUMFP2
  100 SUMFM = SUMFM + QUV*SUMFM2
C
      IF(CAPHP(IY).GE.EPS*S*SUMBP) THEN
      CAP = S/CAPHP(IY)
      FZP = SUMFP*CAP
      BZP = SUMBP*CAP
      SDP = HP(IY)*CAP
      SMFMBP = SDP - FZP - BZP
      BBARP = BZP/DPY(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
C
   99 WRITE(6,302) IY,Y(IY),ZGEO(IY),BZP,BZM,FZP,FZM,SMFMBP,SMFMBM,
     1  BBARP,BBARM
C
C     ECCENTRICITIES
C
      WRITE(6,400)
      DO 402 IIY=1,IPIRAD
      IY = IYIRAD(IIY)
      DPS1 = 1.0/(DPY(IY)*SY(IY))
      DMS1 = 1.0/(DMY(IY)*SY(IY))
  402 WRITE(6,410) IY,Y(IY),ZGEO(IY),BZPY(IY)*DPS1,BZMY(IY)*DMS1,
     1  FZPY(IY)*DPS1,FZMY(IY)*DMS1
C
      RETURN
C
C     FORMATS
C
  300 FORMAT(1H1,//' BACKWARD AND FORWARD SCATTERING FUNCTIONS',
     1 ' (DIMENSIONS OF 1/METER)'//
     2'    IY      Y     ZGEO',6X,'B(Y,+)',8X,'B(Y,-)',8X,'F(Y,+)',8X,
     3 'F(Y,-)',6X,'(S-F-B)(+)',4X,'(S-F-B)(-)',6X,'BBAR(+)',7X,
     4 'BBAR(-)'/)
  302 FORMAT(I5,2F7.2,1P8E14.3)
  400 FORMAT(//' ECCENTRICITIES'//'    IY      Y     ZGEO',
     1'      EPSB(Y,+)       EPSB(Y,-)       EPSF(Y,+)       EPSF(Y,-)'/)
  410 FORMAT(I5,2F7.2,F13.4,3F15.4)
      END
```

```
      SUBROUTINE CONTRM
C
C     ON NHM5/CONTRM
C
C     THIS ROUTINE COMPUTES THE CONTRAST TRANSMITTANCE VIA 8.32.
C     THE QUAD-AVERAGED GEOMETRIC ARRAYS RAXGEO AND TXAGEO ARE REQUIRED.
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
      PARAMETER(MXROW=MXMU*MXPHI)
C
      COMMON/CRADIF/ RADAP(MXMU,MXPHI),RADP(MXMU,MXPHI,MXY),
     1  RADAM(MXMU,MXPHI)
      COMMON/CRADIR/ RADOAP(MXMU,MXPHI),RADOAM(MXMU,MXPHI)
      COMMON/CMISC/ IMISC(20)
      COMMON/CWERK/ RAXGEO(MXROW),TXAGEO(MXMU)
C
      DATA IDBUG/0/, NURAX/22/, NUTXA/25/
C
```

```
      NMU = IMISC(1)
      NPHI = IMISC(2)
C
C     READ THE FIRST NMU COLUMNS OF THE STORED TOP HALF OF RAXGEO,
C     BUT SAVE ONLY COLUMN NMU (THE POLAR CAP OUTPUT DIRECTION)
C
      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)
C
C     DEFINE THE BOTTOM HALF OF THE POLAR CAP OUTPUT DIRECTION FROM
C     THE TOP HALF
C
      DO 712 I=NROW2+1,NROW
  712 RAXGEO(I) = RAXGEO(I-NROW2)
C
C     READ THE FIRST NMU ROWS OF THE FIRST NMU COLUMNS, TO GET TXA(M,./M,.)
C
      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)
C
C     EQUATION 8.33
      RADOT = RADP(NMU,1,1)*TXAGEO(NMU)
C
      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+NMU*(JS-1))
C
      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
C
      CTRANS = RADOT/(RADOT + RADOR)
C
      WRITE(6,100) CTRANS
      RETURN
C
  100 FORMAT(////' THE CONTRAST TRANSMITTANCE IS T =',F6.3//)
  102 FORMAT(1H0,' ERROR IN SUB CONTRM:  NUNIT =',I3,' AND ',A6,' =',I3)
  400 FORMAT(1H0,' SUB CONTRM:  R(A,X:R,S/M,.) VALUES'/(2X,1P10E12.4))
  402 FORMAT(1H0,' T(X,A:R,1/M,.) VALUES'/(2X,1P10E12.4))
  404 FORMAT(1H0,' RADOT =',1PE12.4,5X,'RADOR =',E12.4)
      END
```

```
      SUBROUTINE IRRAD(IPIRAD,IYIRAD)
C
C     ON NHM5/IRRAD
C
C     THIS ROUTINE COMPUTES VARIOUS IRRADIANCE QUANTITIES FROM THE L = 0
C     TOTAL RADIANCE AMPLITUDES, USING 8.5 AND 8.8.
C     IRRADIANCES ARE COMPUTED AT ALL Y LEVELS, FOR POSSIBLE USE IN
C     COMPUTING K-FUNCTIONS, ETC., BUT PRINTOUT IS ONLY AT SELECTED Y
C     LEVELS.
C
C     THE ZERO ELEMENT OF IRRADIANCE ARRAYS HOLDS THE VALUES FOR Y = A
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
      PARAMETER(MXAMP=2*MXMU*(MXPHI/2 + 1))
C
      DIMENSION IYIRAD(MXY)
C
      COMMON/CAMPO/ AOAM(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),
     1  BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CIRRAD/ SHP(0:MXY),SHM(0:MXY),SCAPHP(0:MXY),SCAPHM(0:MXY),
     1  DPY(0:MXY),DMY(0:MXY)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      DATA EPS/1.E-12/, FTOTAL/1./
C     SET FTOTAL = 1. IF TOTAL RADIANCES ARE TO BE USED
C     SET FTOTAL = 0. IF DIFFUSE RADIANCES ARE TO BE USED
C
      NMU = IMISC(1)
      NY = IMISC(4)
      TWOPI = 2.*FMISC(1)
C
C     COMPUTE QUANTITIES AT Y = A
C
      HP = 0.
      HM = 0.
      CAPHP = 0.
      CAPHM = 0.
      DO 140 I=1,NMU
      AMPP = AOAP(I) + AAP(I)
      AMPM = AOAM(I)
      DMU = DELTMU(I)
      HP = HP + AMPP*DMU
      HM = HM + AMPM*DMU
      CAPHP = CAPHP + AMPP*FMU(I)*DMU
  140 CAPHM = CAPHM + AMPM*FMU(I)*DMU
C
      SHP(0) = TWOPI*HP
      SHM(0) = TWOPI*HM
      SCAPHP(0) = TWOPI*CAPHP
      SCAPHM(0) = TWOPI*CAPHM
C
      TOTH = 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(SCAPHM(0).GT.EPS*SCAPHP(0)) RM = SCAPHP(0)/SCAPHM(0)
      DPY(0) = DP
      DMY(0) = DM
C
      WRITE(6,200)
      WRITE(6,203) SHP(0),SHM(0),TOTH,SCAPHP(0),SCAPHM(0),DP,DM,RM
      IF(FTOTAL.NE.1.) WRITE(6,201
C
      DO 100 IY=1,NY
      HP = 0.
      HM = 0.
      CAPHP = 0.
      CAPHM = 0.
C
```

```
C      COMPUTE IRRADIANCES FROM AMPLITUDES
C
       DO 150 I=1,NMU
C      DEFINE THE TOTAL AMPLITUDES (DIFFUSE + DIRECT) FOR L = 0
       AMPP = AYP(I,IY)
       AMPM = AYM(I,IY) + FTOTAL*AOYM(I,IY)
       DMU = DELTMU(I)
       HP = HP + AMPP*DMU
       HM = HM + AMPM*DMU
       CAPHP = CAPHP + AMPP*FMU(I)*DMU
  150  CAPHM = CAPHM + AMPM*FMU(I)*DMU
C
       SHP(IY) = TWOPI*HP
       SHM(IY) = TWOPI*HM
       SCAPHP(IY) = TWOPI*CAPHP
       SCAPHM(IY) = TWOPI*CAPHM
C
       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(SCAPHM(IY).GT.EPS*SCAPHP(IY)) RM = SCAPHP(IY)/SCAPHM(IY)
       DPY(IY) = DP
       DMY(IY) = DM
C
C      CHECK FOR PRINTOUT
       IPRINT = 0
       DO 300 IIY=1,IPIRAD
       IF(IY.EQ.IYIRAD(IIY)) IPRINT = 1
  300  CONTINUE
       IF(IPRINT.NE.0) WRITE(6,202) IY,Y(IY),ZGEO(IY),SHP(IY),SHM(IY),
      1 TOTH,SCAPHP(IY),SCAPHM(IY),DP,DM,RM
  100  CONTINUE
C
       RETURN
C
  200  FORMAT(1H1,//' IRRADIANCE QUANTITIES COMPUTED FROM THE L = 0 AMPLI
      1TUDES'//'    IY      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(I5,2F7.2,1P5E15.4,0P2F9.4,1PE15.4)
  203  FORMAT(10X,'A       A',1X,1P5E15.4,0P2F9.4,1PE15.4/)
C
       END
```

```
      SUBROUTINE KFCN(IPKFCN,IYKFCN)
C
C     ON NHM5/KFCN
C
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     SEE 8.12 AND 8.13.
C
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(MXY)
C
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1 BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CIRRAD/ HP(0:MXY),HM(0:MXY),CAPHP(0:MXY),CAPHM(0:MXY)
C
      WRITE(6,300)
C
      DO 100 IIY=1,IPKFCN
      IY = IYKFCN(IIY)
      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) Y(IY),Y(IY+1),YMID,AKP,AKM,CAPKP,CAPKM
C
      WRITE(6,400)
C
      DO 500 IIY=1,IPKFCN
      IY = IYKFCN(IIY)
      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(IY),ZGEO(IY+1),ZMID,AKP,AKM,CAPKP,CAPKM
      RETURN
C
  300 FORMAT(1H1,//' OPTICAL DEPTH K-FUNCTIONS (NONDIMENSIONAL) FOR IRRA
     1DIANCES (VALID ONLY WHEN YUPPER AND YLOWER ARE CLOSELY SPACED)'//
     2'     YUPPER     YLOWER',7X,'Y',7X,
     3'K(+)       K(-)    CAP K(+)  CAP K(-)'/)
  302 FORMAT(3F10.3,4F10.5)
  400 FORMAT(///' GEOMETRIC DEPTH K-FUNCTIONS (UNITS OF 1/METER) FOR IRR
     1ADIANCES (VALID ONLY WHEN ZUPPER AND ZLOWER ARE CLOSELY SPACED)'//
     2'     ZUPPER     ZLOWER',6X,'ZGEO',5X,
     3'K(+)       K(-)    CAP K(+)  CAP K(-)'/)
      END
```

```
      SUBROUTINE KRAD(IPKFCN,IYKFCN)
C
C     ON NHM5/KRAD
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(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      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/CKRAD/ ISTART,ISTOP,ISTEP,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
C     NON-POLAR QUADS
      DO 200 I=ISTART,ISTOP,ISTEP
      THEDEG = RADEG*ACOS(FMU(I))
C
      WRITE(6,301)
      LINE = LINE + 1
      DO 200 IIY=1,IPKFCN
      IY = IYKFCN(IIY)
      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))
C
C     GET RADIANCES, RADIANCE DERIVATIVES, AND ATTENUATION FUNCTION
C     AT YMID
C
      RPMID = 0.5*(RADP(I,J,IY+1) + RADP(I,J,IY))
      RMMID = 0.5*(RADM(I,J,IY+1) + RADM(I,J,IY) + RADOM(I,J,IY+1) +
     1  RADOM(I,J,IY))
C
      DNPDY = C*(RADP(I,J,IY+1) - RADP(I,J,IY))
      DNMDY = C*(RADM(I,J,IY+1) - RADM(I,J,IY) + RADOM(I,J,IY+1) -
     1  RADOM(I,J,IY))
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
```

```
C     THE PATH FUNCTION AT YMID, USING 2.2
      PATHFP = -FMU(I)*ALPHA*DNPDY + ALPHA*RPMID
      PATHFM = -FMU(I)*ALPHA*DNMDY + ALPHA*RMMID
C
C     THE K-FUNCTIONS AT YMID, USING 8.26B
      FKP = -DNPDY/RPMID
      FKM = -DNMDY/RMMID
C
      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
C
C     POLAR CAP
C
      WRITE(6,301)
      LINE = LINE + 1
      DO 100 IIY=1,IPKFCN
      IY = IYKFCN(IIY)
      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))
C
C     GET RADIANCES, RADIANCE DERIVATIVES, AND ATTENUATION FUNCTION
C     AT YMID
C
      RPMID = 0.5*(RADP(NMU,1,IY+1) + RADP(NMU,1,IY))
      RMMID = 0.5*(RADM(NMU,1,IY+1) + RADM(NMU,1,IY) + RADOM(NMU,1,IY+1)
     1 + RADOM(NMU,1,IY))
C
      DNPDY = C*(RADP(NMU,1,IY+1) - RADP(NMU,1,IY))
      DNMDY = C*(RADM(NMU,1,IY+1) - RADM(NMU,1,IY) + RADOM(NMU,1,IY+1) -
     1  RADOM(NMU,1,IY))
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 57 JY=2,NSIGY
      IF(YMID.LT.YSIG(JY)) GO TO 58
   57 CONTINUE
   58 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
C     THE PATH FUNCTION AT YMID, USING 2.2
      PATHFP = -FMU(I)*ALPHA*DNPDY + ALPHA*RPMID
      PATHFM = -FMU(I)*ALPHA*DNMDY + ALPHA*RMMID
C
C     THE K-FUNCTIONS AT YMID, USING 8.26B
      FKP = -DNPDY/RPMID
      FKM = -DNMDY/RMMID
C
      I = NMU
      J = 0
      THEDEG = 0.
      PHIDEG = 0.
      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
  100 CONTINUE
C
      RETURN
C
```

```
  300 FORMAT(1H1,//' RADIANCES, PATH FUNCTIONS, AND RADIANCE K-FUNCTIONS
     1 FOR SELECTED DIRECTIONS (VALID ONLY WHEN YUPPER AND YLOWER ARE CL
     2OSELY SPACED)'//T90,'NONDIMENSIONAL',11X,'DIMENSIONAL (1/M)'/
     3'  I   J THETA     PHI YUPPER YLOWER      Y      RAD+(Y)',
     4 4X,'RAD-(Y)    PATHF+(Y)  PATHF-(Y)     K(+)      K(-)',6X,
     5  'ZGEO     K(+)      K(-)')
  301 FORMAT(1H )
  302 FORMAT(2I3,F6.1,F7.1,2F7.3,F8.4,1P4E11.3,0P2F9.4,F9.3,2F9.4)
      END
```

```
      SUBROUTINE PRINT(IPRAD,IYPRAD)
C
C     ON NHM5/PRINT
C
C     THIS ROUTINE PRINTS OUT THE FINAL RADIANCE FIELDS AT SELECTED Y LEVELS
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
C
      DIMENSION IYPRAD(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/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1 BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CPRAD/ IPRAD1,IPRAD2,IPRAD3,JPRAD1,JPRAD2,JPRAD3
      COMMON/CMISC/ IMISC(20),FMISC(20)
      COMMON/CWORK/ THEDEG(MXMU),PHIDEG(MXPHI)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      RADEG = FMISC(3)
C
C     CONVERT MU AND PHI TO DEGREES
C
      DO 50 I=1,NMU
   50 THEDEG(I) = RADEG*ACOS(FMU(I))
      DO 51 J=1,NPHI
   51 PHIDEG(J) = RADEG*PHI(J)
C
C     WRITE RADIANCES AT Y = A
C
      WRITE(6,300)
      DO 302 I=IPRAD1,IPRAD2,IPRAD3
      WRITE(6,102)
      DO 302 J=JPRAD1,JPRAD2,JPRAD3
  302 WRITE(6,304) I,J,THEDEG(I),PHIDEG(J),RADAP(I,J),RADOAP(I,J),
     1  RADOAM(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  RADOAM(I,J)
C
```

```
C     WRITE RADIANCES AT Y = X,...,Z
C
      WRITE(6,100)
      DO 101 IYY=1,IPRAD
      IY = IYPRAD(IYY)
      WRITE(6,110)
      DO 103 I=IPRAD1,IPRAD2,IPRAD3
      WRITE(6,102)
      DO 103 J=JPRAD1,JPRAD2,JPRAD3
  103 WRITE(6,104)I,J,IY,THEDEG(I),PHIDEG(J),Y(IY),ZGEO(IY),
     1  RADP(I,J,IY),RADM(I,J,IY),RADOM(I,J,IY)
      WRITE(6,102)
      I = NMU
      J = 1
      WRITE(6,104) I,J,IY,THEDEG(I),PHIDEG(J),Y(IY),ZGEO(IY),
     1  RADP(I,J,IY),RADM(I,J,IY),RADOM(I,J,IY)
  101 CONTINUE
C
      RETURN
C
  100 FORMAT(1H1,' THE FINAL DIFFUSE AND DIRECT RADIANCES AT INTERIOR Y
     1VALUES ARE'/)
  102 FORMAT(1H )
  104 FORMAT(3I4,2F9.3,2F8.3,1P3E15.5)
  110 FORMAT(1H0,'  I    J    K',4X,'THETA',4X,'PHI',7X,'Y       ZGEO',6X,
     1 7HRAD*(+),8X,7HRAD*(-),8X,7HRADO(-))
  300 FORMAT(1H1,' THE FINAL DIFFUSE AND DIRECT RADIANCES AT Y = A ARE'/
     1/'    I    J',4X,'THETA',4X,'PHI',8X,7HRAD*(+),8X,7HRADO(+),8X,
     2 7HRADO(-))
  304 FORMAT(2I4,2F9.3,1P3E15.5)
C
      END
```

```
      SUBROUTINE RADY(IPIRAD,IYIRAD)
C
C     ON NHM5/RADY
C
C     THIS ROUTINE PRINTS SELECTED RADIANCES (UP, DOWN, AND HORIZONTAL
C     ALONG-WIND AND CROSS-WIND).  RADIANCE-IRRADIANCE RATIOS ARE ALSO
C     COMPUTED.
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
C
      DIMENSION IYIRAD(MXY)
C
      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/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1 BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(MXY)
      COMMON/CIRRAD/ SHP(0:MXY),SHM(0:MXY),SCAPHP(0:MXY),SCAPHM(0:MXY)
      COMMON/CMISC/ IMISC(20)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
C
C     Y = A
C
      RADUP = RADOAP(NMU,1) + RADAP(NMU,1)
      RADDN = RADOAM(NMU,1) + RADAM(NMU,1)
      RH0 = 0.5*(RADOAP(1,1) + RADAP(1,1) + RADOAM(1,1) + RADAM(1,1))
      J90 = NPHI/4 + 1
      RH90 = 0.5*(RADOAP(1,J90) + RADAP(1,J90) + RADOAM(1,J90) +
     1  RADAM(1,J90))
      J180 = NPHI/2 + 1
      RH180 = 0.5*(RADOAP(1,J180) + RADAP(1,J180) + RADOAM(1,J180) +
     1  RADAM(1,J180))
      RN = RADUP/SCAPHM(0)
      QM = SCAPHP(0)/RADUP
      WRITE(6,100)
      WRITE(6,102) RADUP,RADDN,RH0,RH90,RH180,RN,QM
C
C     DEPTHS X .LE. Y .LE. Z
C
      DO 200 IIY=1,IPIRAD
      IY = IYIRAD(IIY)
      RADUP = RADP(NMU,1,IY)
      RADDN = RADOM(NMU,1,IY) + RADM(NMU,1,IY)
      RH0 = 0.5*(RADP(1,1,IY) + RADOM(1,1,IY) + RADM(1,1,IY))
      RH90 = 0.5*(RADP(1,J90,IY) + RADOM(1,J90,IY) + RADM(1,J90,IY))
      RH180 = 0.5*(RADP(1,J180,IY) + RADOM(1,J180,IY) + RADM(1,J180,IY))
      RN = RADUP/SCAPHM(IY)
      QM = SCAPHP(IY)/RADUP
  200 WRITE(6,104) IY,Y(IY),ZGEO(IY),RADUP,RADDN,RH0,RH90,RH180,RN,QM
C
      RETURN
C
  100 FORMAT('1'' SELECTED RADIANCES AND RADIANCE-IRRADIANCE RATIOS''//
     1'    IY      Y     ZGEO       N+(Y,M,.)       N-(Y,M,.)        NH(Y,0)',
     27X,'NH(Y,90)        NH(Y,180)        RN(Y,-)          Q(+)'/)
  102 FORMAT(10X,'A       A ',1P7E15.4/)
  104 FORMAT(I5,2F7.2,1P7E15.4)
      END
```

```
      SUBROUTINE RTECK(IRTECK,IYRTE)
C
C     ON NHM5/RTECK
C
C     THIS ROUTINE CHECKS THE FINAL TOTAL RADIANCES BY SEEING IF THEY
C     SATISFY THE QUAD-AVERAGED RADIATIVE TRANSFER EQUATION 3.12 AT
C     INTERIOR Y VALUES, X.LT.Y.LT.Z
C
C     IRTECK.LT.0 IF THE BALANCE OF THE RTE IS TO BE COMPUTED AT ALL
C                 POSSIBLE INTERIOR Y LEVELS, Y(2),...., Y(NY-1)
C     IRTECK.EQ.0 IF NO RTE BALANCE CHECK IS TO BE MADE
C     IRTECK.GT.0 IF THE RTE BALANCE IS COMPUTED AT THE Y LEVELS GIVEN BY
C                 Y(IYRTE(1)),...,Y(IYRTE(IRTECK))
C
C     NIC, NJC...ARE USED TO SELECT PARTICULAR MU AND PHI VALUES
C                WHERE THE RTE BALANCE CHECK IS TO BE MADE, IF IRTECK.NE.0
C                MU(I) AND PHI(J) ARE CHECKED, WHERE
C
C                     DO J=1,NPHI,NJC
C                     DO I=1,2*NMU,NIC
C
C                AND MU(I) IS IN XI(-) IF I.LE.NMU
C                    MU(I) IS IN XI(+) IF I.GT.NMU .AND. I.LE.2*NMU
C
C++++ WARNING:  DN/DY IS COMPUTED USING A CENTERED DIFFERENCE.  IF THE Y
C     VALUES ARE NOT EVENLY SPACED OR IF THEY ARE FAR APART, THIS
C     ESTIMATE OF THE DERIVATIVE MAY BE QUITE INACCURATE, CAUSING A POOR
C     BALANCE OF THE RTE EVEN THOUGH THE COMPUTED RADIANCES ARE CORRECT.
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXL=MXPHI/2, MXGEOP=MXMU*(MXL+1))
C
      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/CGRID/ FMU(MXMU),PHI(MXPHI),Y(MXY)
      COMMON/CGEOP/ GEOPP(MXMU,MXGEOP,MXSIGY),GEOPM(MXMU,MXGEOP,MXSIGY)
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY)
      COMMON/CMISC/ IMISC(20),FMISC(20)
      COMMON/CWORK/ GEOPPY(MXMU,MXGEOP),GEOPMY(MXMU,MXGEOP)
      DIMENSION IYRTE(MXY)
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NY = IMISC(4)
      NSIGY = IMISC(5)
      NRHAT = IMISC(10)
      NIC = IMISC(15)
      NJC = IMISC(16)
      NMU2 = NMU*2
      NOPI = NPHI/2
      IF(IRTECK.LT.0) THEN
      IYMIN = 2
      IYMAX = NY-1
      ELSEIF(IRTECK.GT.0) THEN
      IYMIN = 1
      IYMAX = IRTECK
      ELSE
      RETURN
      ENDIF
      WRITE(6,200)
C
C     IC, JC, AND IYC LABEL THE MU, PHI, AND Y VALUES FOR WHICH THE RTE
C     IS EVALUATED.
C     IC.GT.0 FOR UPWARD RADIANCE
C     IC.LT.0 FOR DOWNWARD RADIANCE
C
      DO 300 IY=IYMIN,IYMAX
      IYC = IYRTE(IY)
      YNOW = Y(IYC)
C
```

```
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.LT.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
      DO 300 IC2=1,NMU2,NIC
      IF(IC2.LE.NMU) THEN
      IC = -IC2
      ELSEIF(IC2.GT.NMU .AND. IC2.LE.NMU2) THEN
      IC = IC2 - NMU
      ELSE
C     IC = 0 NOT VALID
      GO TO 300
      ENDIF
      ICA = IABS(IC)
      IF(ICA.EQ.NMU .AND. JC.NE.1) GO TO 300
      IF(IC.GT.0) THEN
      FMUIC = FMU(ICA)
      ELSE
      FMUIC = -FMU(ICA)
      ENDIF
C
C     EVALUATE THE TERMS OF THE RTE
C
C     TERM1 = MU*DN/DY
C
      IF(IC.GT.0) THEN
      TERM1 = RADP(ICA,JC,IYC+1) - RADP(ICA,JC,IYC-1)
      ELSE
      TERM1 = RADM(ICA,JC,IYC+1) - RADM(ICA,JC,IYC-1) +
     1                    RADOM(ICA,JC,IYC+1) - RADOM(ICA,JC,IYC-1)
      ENDIF
      TERM1 = FMUIC*TERM1/(Y(IYC+1) - Y(IYC-1))
C
C     TERM2 = - N
C
      IF(IC.GT.0) THEN
      TERM2 = -RADP(ICA,JC,IYC)
      ELSE
      TERM2 = -RADM(ICA,JC,IYC) - RADOM(ICA,JC,IYC)
      ENDIF
C
```

```
C      TERM3 = (ALBEDO OF SINGLE SCATTERING) * INTEGRAL OF (RADIANCE *
C                                                          PHASE FUNCTION)
C            = ALBEDO * PATH FUNCTION
C
       TERM3 = 0.
       DO 700 IR=1,NMU
       ISMAX = NPHI
       IF(IR.EQ.NMU) ISMAX = 1
       DO 700 IS=1,ISMAX
C
       RP = RADP(IR,IS,IYC)
       RM = RADM(IR,IS,IYC) + RADOM(IR,IS,IYC)
C
C      COMPUTE STORAGE INDEX FOR GEOPP AND GEOPM
       IVS = IABS(JC-IS)
       IVINDX = IVS + 1
       IF(IVS.GT.NOPI) IVINDX = NOPI + 1 - MOD(IVS,NOPI)
       KCOL = NMU*(IVINDX - 1) + ICA
       IF(ICA.EQ.NMU) KCOL = NMU
       IF(IR.EQ.NMU) KCOL = ICA
       PP = GEOPPY(IR,KCOL)
       PM = GEOPMY(IR,KCOL)
       IF(IC.GT.0) THEN
       TERM3 = TERM3 + RM*PM + RP*PP
       ELSE
       TERM3 = TERM3 + RM*PP + RP*PM
       ENDIF
   700 CONTINUE
       TERM3 = ALBEDO*TERM3
C
C      OUTPUT
C
       SUM = TERM1 + TERM2 + TERM3
       WRITE(6,201) IC,JC,IYC,TERM1,TERM2,TERM3,SUM
   300 CONTINUE
C
C      FORMATS
C
   200 FORMAT(1H1,' FINAL CHECK ON COMPUTED TOTAL RADIANCES:'/
      16X,'EVALUATION OF THE RADIATIVE TRANSFER EQUATION FOR SELECTED MU,
      2 PHI AND Y VALUES'///
      3'    MU   PHI     Y',5X,'MU*DN/DY',9X,'- N',5X,'+ INT(N*SIGMA)/A',
      4 5X,'= ZERO'/)
   201 FORMAT(3I5,1P4E15.6)
C
       RETURN
       END
```

```
      SUBROUTINE SYNRAD(AMP,RAD,IROW)
C
C     ON NHM5/SYNRAD
C
C     THIS ROUTINE SYNTHESIZES THE RADIANCE FIELD R(MU,PHI) (FOR A
C     GIVEN Y VALUE) USING 5.3 AND 5.4.
C
      PARAMETER(MXMU=10, MXPHI=24)
      PARAMETER(MXL=MXPHI/2)
C
      DIMENSION AMP(1),RAD(IROW,1)
      DIMENSION COSLP(0:MXL,MXPHI),SINLP(0:MXL,MXPHI)
      COMMON/CGRID/ FMU(MXMU),PHI(MXPHI)
      COMMON/CMISC/ IMISC(20)
C
      DATA KALL/0/
C
      IF(KALL.EQ.0) THEN
C
C     THE FIRST CALL DOES INITIALIZATION
C
      NMU = IMISC(1)
      NPHI = IMISC(2)
      NL = IMISC(3)
      NRHAT = IMISC(10)
C
      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
C
C     LOOP OVER ALL MU AND PHI VALUES
C
      DO 100 I=1,NMU-1
      DO 100 J=1,NPHI
C
C     SUM OVER L VALUES, EQ 5.3
C
      SUM = 0.
      DO 200 L=0,NL
  200 SUM = SUM + AMP(NMU*L+I)*COSLP(L,J)
     1          + AMP(NRHAT+NMU*L+I)*SINLP(L,J)
  100 RAD(I,J) = SUM
C
C     POLAR CAP TERM BY 5.4
      RAD(NMU,1) = AMP(NMU)
      DO 102 J=2,NPHI
  102 RAD(NMU,J) = 0.
C
      RETURN
      END
```

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	≤ 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	≤ 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 alphanumeric 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,···,NY at which plots are to be made (cf. record 5 of Program 4).

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 θ. The specification record gives JPHI and JPI, which are the ϕ-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 θ. Otherwise as for ITYPE = 1.

If ITYPE = 3, plot the logarithm of the diffuse radiance as a function of θ. 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 θ, 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 ϕ. The specification record gives ITHETA, the index defining a particular θ-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 ϕ, otherwise as for ITYPE = 5.

If ITYPE = 7, plot the logarithm of the diffuse radiances as a function of ϕ. 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 ϕ, otherwise as for ITYPE = 7.

If ITYPE = 9, plot the logarithm of the total path function as a function of θ. 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.

2. Code Listing

```
      PROGRAM MPRAD(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE50,
     1 TAPE98,TAPE99)
C
C     ON NHM6/MPRAD
C
C     THIS PROGRAM CONTROLS THE PLOTTING OF THE RADIANCES, USING THE
C     FILE OF RADIANCE DATA WRITTEN BY PROGRAM 5 (TAPE50).
C
C     ALL PLOTTING IS DONE USING STANDARD CALCOMP CALLS
C     (TAPE98 AND TAPE99 ARE USED BY THE CALCOMP ROUTINES, AS
C     IMPLEMENTED ON THE AUTHOR'S CDC CYBER 855 COMPUTER.)
C
C     EACH PLOT IS GENERATED BY TWO TO FOUR FREE-FORMAT DATA RECORDS.
C
C     THE FIRST RECORD GIVES   ITYPE,NTIT,NYPLT  WHERE
C        ITYPE SPECIFIES THE TYPE OF PLOT TO BE MADE, AS DESCRIBED BELOW.
C        NTIT.LE.0 IF NO TITLE FOR THE TOP OF THE PLOT IS DESIRED
C            .GT.0 IF A TITLE FOR THE TOP OF THE PLOT IS TO BE READ IN
C        NYPLT.LE.0 IF ALL Y LEVELS ARE TO BE PLOTTED
C              .GT.0 IF NYPLT Y LEVELS ARE TO BE PLOTTED
C
C     IF NTIT.GT.0, THE NEXT RECORD GIVES THE DESIRED TITLE
C
C     IF NYPLT.GT.0, THE NEXT RECORD GIVES THE INDICES OF THE Y
C     LEVELS FOR THE PLOTS
C
C     THE LAST RECORD, THE SPECIFICATION RECORD, GIVES THE VALUES OF
C     THE PARAMETERS NEEDED TO SPECIFY THE DETAILS OF THE PLOT
C
C     IF ITYPE.EQ.1, MAKE POLAR PLOTS OF THE LOGARITHM OF THE DIFFUSE
C                    RADIANCE AS A FUNCTION OF THETA.  THE SPECIFICATION
C                    RECORD GIVES JPHI AND JPI, WHICH ARE THE
C                    PHI INDICES OF TWO HALF-PLANES.  NORMALLY PHI(JPI) =
C                    PHI(JPHI) + PI, SO THAT A PLANAR CROSS SECTION OF THE
C                    RADIANCE IS PLOTTED.
C                    A SEPARATE PLOT IS MADE FOR EACH DEPTH.
C
C     IF ITYPE.EQ.2, MAKE POLAR PLOTS OF THE LOGARITHM OF THE TOTAL
C                    RADIANCE AS A FUNCTION OF THETA, OTHERWISE AS FOR
C                    ITYPE = 1.
C
C     IF ITYPE.EQ.3, PLOT THE LOGARITHM OF THE DIFFUSE RADIANCE AS A FUNCTION OF
C                    THETA.  THE SPECIFICATION RECORD GIVES JPHI AND JPI
C                    AS FOR ITYPE = 1.
C                    ALL DEPTHS ARE ON THE SAME PLOT.
C
C     IF ITYPE.EQ.4, PLOT THE LOGARITHM OF THE TOTAL RADIANCE AS A FUNCTION OF
C                    THETA, OTHERWISE AS FOR ITYPE = 3.
C                    ALL DEPTHS ARE ON THE SAME PLOT.
C
C     IF ITYPE.EQ.5, MAKE A POLAR PLOT OF THE DIFFUSE RADIANCES AS A
C                    FUNCTION OF PHI.  THE SPECIFICATION RECORD GIVES
C                    ITHETA, THE INDEX DEFINING A PARTICULAR THETA CONE:
C                        IF ITHETA.GT.0, UPWARD RADIANCES ARE PLOTTED
C                        IF ITHETA.LT.0, DOWNWARD RADIANCES ARE PLOTTED
C                    A SEPARATE PLOT IS MADE FOR EACH DEPTH.
C
C     IF ITYPE.EQ.6, MAKE A POLAR PLOT OF THE TOTAL RADIANCES AS A
C                    FUNCTION OF PHI, OTHERWISE AS FOR ITYPE = 5.
C
C     IF ITYPE.EQ.7, PLOT THE LOGARITHM OF THE DIFFUSE RADIANCES AS A
C                    FUNCTION OF PHI.  THE SPEC REC GIVES ITHETA AS
C                    FOR ITYPE = 5.
C                    ALL DEPTHS ARE ON THE SAME PLOT.
C
C     IF ITYPE.EQ.8, PLOT THE LOGARITHM OF THE TOTAL RADIANCES AS A
```

```
C                     FUNCTION OF PHI, OTHERWISE AS FOR ITYPE = 7.
C
C     IF ITYPE.EQ.9, PLOT THE LOGARITHM OF THE TOTAL PATH FUNCTION AS A
C                     FUNCTION OF THETA.  THE SPEC REC GIVES JPHI AND
C                     JPI AS FOR ITYPE = 1.
C                     ALL DEPTHS ARE ON THE SAME PLOT.
C
C++++ WARNING:  ITYPE = 5, 6 OR 9 CANNOT BE USED, SINCE THE REQUIRED
C     SUBROUTINES PPHIPLR AND PPATH HAVE NOT YET BEEN WRITTEN (CM, 3 JUNE 88).
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30, MXSIGY=3)
      PARAMETER(MXL=MXPHI/2, MXGEOP=MXMU*(MXL+1))
C
      COMMON/CSIGY/ YSIG(MXSIGY),ALBESS(MXSIGY),TOTALS(MXSIGY)
      COMMON/CGEOP/ GEOPP(MXMU,MXGEOP,MXSIGY),GEOPM(MXMU,MXGEOP,MXSIGY)
      COMMON/CGRID/ THETA(MXMU),PHI(MXPHI),Y(MXY),BNDMU(MXMU),
     1  BNDPHI(MXPHI),OMEGA(MXMU),DELTMU(MXMU),ZGEO(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/ WORK(5000)
C
      DIMENSION FMU(MXMU),IYPLT(MXY)
C
      DATA NUIN/50/, EPS/1.0E-10/
C
C     INITIALIZE THE CALCOMP PLOTTING ROUTINES
      CALL PLOTS
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) (((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)
      READ(NUIN) ((RADAP(I,J),I=1,NMU),J=1,NPHI)
      READ(NUIN) (((RADP(I,J,K),I=1,NMU),J=1,NPHI),K=1,NY)
      READ(NUIN) ((RADAM(I,J),I=1,NMU),J=1,NPHI)
      READ(NUIN) (((RADM(I,J,K),I=1,NMU),J=1,NPHI),K=1,NY)
      READ(NUIN) ((RADOAP(I,J),I=1,NMU),J=1,NPHI)
      READ(NUIN) ((RADOAM(I,J),I=1,NMU),J=1,NPHI)
      READ(NUIN) (((RADOM(I,J,K),I=1,NMU),J=1,NPHI),K=1,NY)
C
C     DIRECT RADIANCES WHICH SHOULD BE ZERO ARE SOMETIMES NEGATIVE
C     DUE TO ROUNDOFF ERROR:  RESET TO ZERO
      DO 60 K=1,NY
      DO 60 J=1,NPHI
      DO 60 I=1,NMU
      IF(RADOM(I,J,K).LT.EPS) RADOM(I,J,K) = 0.
   60 CONTINUE
C
C     CONVERT FMU TO THETA
      DO 53 I=1,NMU
   53 THETA(I) = ACOS(FMU(I))
C
C     READ RECORDS DESCRIBING THE PLOTS
C
C     THE FIRST RECORD:
  100 READ(5,*,END=200) ITYPE,NYPLT,NTIT
C
C     THE TITLE RECORD, IF REQUESTED
      IF(NTIT.GT.0) THEN
      READ(5,70) ITITLE
      NTIT = NCHAR(ITITLE,8)
      ENDIF
C
```

```
      IF(NYPLT.LE.0) THEN
      DO 300 IY=1,NY
  300 IYPLT(IY) = IY
      ELSE
C     THE Y-INDEX RECORD, IF REQUESTED
      READ(5,*) (IYPLT(IY),IY=1,NYPLT)
      ENDIF
C
C     THE SPECIFICATION RECORD
C
      IF(ITYPE.GE.1 .AND. ITYPE.LE.4) THEN
      READ(5,*) JPHI,JPI
      ELSEIF(ITYPE.GE.5 .AND. ITYPE.LE.8) THEN
      READ(5,*) ITHETA
      ELSEIF(ITYPE.EQ.9) THEN
      READ(5,*) JPHI,JPI
      ENDIF
C
C     CALL THE APPROPRIATE PLOT SUBROUTINE
C
      IF(ITYPE.EQ.1 .OR. ITYPE.EQ.2) THEN
      CALL PTHEPLR(ITYPE,NYPLT,IYPLT,JPHI,JPI)
      ELSEIF(ITYPE.EQ.3 .OR. ITYPE.EQ.4) THEN
      CALL PTHELOG(ITYPE,NYPLT,IYPLT,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,NYPLT,IYPLT,ITHETA)
      ELSEIF(ITYPE.EQ.9) THEN
      CALL PPATH
      ENDIF
      GO TO 100
C
  200 CALL PLOT(0.,0.,-98)
C
   70 FORMAT(8A10)
      END
```

```
      FUNCTION NCHAR(ITITLE,NWORDS)
C
C     GIVEN AN ALPHANUMERIC TITLE, ITITLE, OF NWORDS (MAX 12), THIS
C     FUNCTION RETURNS THE NUMBER OF NON-BLANK CHARACTERS. (FOR USE IN
C     PLOTTING CENTERED TITLES)
C
      DIMENSION ITITLE(NWORDS),ICHAR(120)
      DATA IBLANK/10H          /
C
      MAXCHR = 10*NWORDS
      ENCODE(7,200,IFMT) MAXCHR
C
      DECODE(MAXCHR,IFMT,ITITLE) ICHAR
C
      DO 110 I=1,MAXCHR
      NCHAR = MAXCHR - I + 1
      IF(ICHAR(NCHAR).NE.IBLANK) RETURN
  110 CONTINUE
      NCHAR = 0
      RETURN
C
  200 FORMAT(1H(,I3,3HA1))
      END
```

```
      SUBROUTINE PPHILOG(ITYPE,NYPLT,IYPLT,ITHETA)
C
C     ON NHM6/PPHILOG
C
C     THIS ROUTINE PLOTS THE LOGARITHM OF THE DIFFUSE OR TOTAL RADIANCE
C     AS A FUNCTION OF PHI FOR A GIVEN THETA VALUE.
C     ALL Y VALUES ARE DISPLAYED ON THE SAME GRAPH.
C
C     IF ITYPE.EQ.7, THE DIFFUSE RADIANCES ARE PLOTTED
C     IF ITYPE.EQ.8, THE TOTAL RADIANCES ARE PLOTTED
C
C     IF ITHETA .GT. 0, PLOT UPWARD RADIANCES N(+THETA,PHI,Y) = RADP
C     IF ITHETA .LT. 0, PLOT DOWNWARD RADIANCES N(-THETA,PHI,Y) = RADM
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
      PARAMETER(MXPTS=MXPHI+3)
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,MXY),YPLT(MXPTS,MXY),BCD(5)
C
      DIMENSION IYPLT(MXY)
C
      DATA XINCH,YINCH/4.0,5.0/, H,BOX/0.15,1.0/, EPS/1.E-12/
      DATA ISYMBL/0/
C
      IT = IABS(ITHETA)
      NPHI = IMISC(2)
      NY = IMISC(4)
      NPHI1 = NPHI + 1
      PI = FMISC(1)
      RADEG = FMISC(3)
      TWOPI = 2.*PI
      PI2 = 0.5*PI
      HBOX = H*BOX
      IYMAX = NY
      IF(NYPLT.GT.0) IYMAX = NYPLT
C
      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)
C
C     DETERMINE THE ALLOWED RANGE OF Y VALUES
C
      IYT = 1
      IYB = NY
      IF(ITHETA.EQ.0) GO TO 99
C
C     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(IT,J,NY).GT.EPS*RADM(IT,J,NY)) IYB = NY
   90 CONTINUE
      GO TO 99
C
C     CHECK FOR ZERO DOWNWARD RADIANCE AT THE TOP (NAKED SLAB CASE)
   98 IYT = 2
      DO 91 J=1,NPHI
      IF(RADM(IT,J,1).GT.EPS) IYT = 1
   91 CONTINUE
   99 CONTINUE
C
```

```
C     DEFINE ARRAYS FOR PLOTTING
C
      DO 100 K=1,IYMAX
      IY = IYPLT(K)
      IF(IY.EQ.1 .AND. IYT.NE.1) GO TO 100
      IF(IY.EQ.NY .AND. IYB.NE.NY) GO TO 100
      DO 101 J=1,NPHI
      XPLT(J,IY) = PHI(J)
      IF(ITHETA.GT.0) THEN
      YPLT(J,IY) = ALOG10(RADP(IT,J,IY))
      ELSE
      YPLT(J,IY) = ALOG10(RADM(IT,J,IY) + FACT*RADOM(IT,J,IY))
      ENDIF
  101 CONTINUE
      XPLT(NPHI1,IY) = TWOPI
  100 YPLT(NPHI1,IY) = YPLT(1,IY)
C
C     FIND THE MAXIMUM AND MINIMUM VALUES TO BE PLOTTED
C
      RADMAX = -1.E30
      RADMIN = 1.E30
      DO 110 K=1,IYMAX
      IY = IYPLT(K)
      IF(IY.EQ.1 .AND. IYT.NE.1) GO TO 110
      IF(IY.EQ.NY .AND. IYB.NE.NY) GO TO 110
      DO 111 J=1,NPHI1
      RAD = YPLT(J,IY)
      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
      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(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)
      IDIV1 = IDIV + 1
C
      CALL PLOT(0.,YINCH,2)
      XX = -7.6*HBOX
      DO 310 I=1,IDIV1
      YY = YINCH - 0.45*H - FLOAT(I-1)*DINCH
      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*YINCH - 6.5*HBOX
      CALL SYMBOL(XX,YY,H,13HLOG(RADIANCE),90.0,13)
C
C     DEFINE SCALE FACTORS CONSISTENT WITH THE LABELS
C
      DO 200 K=1,IYMAX
      IY = IYPLT(K)
      IF(IY.EQ.1 .AND. IYT.NE.1) GO TO 200
      IF(IY.EQ.NY .AND. IYB.NE.NY) GO TO 200
      XPLT(NPHI1+1,IY) = 0.
      XPLT(NPHI1+2,IY) = PI2
      YPLT(NPHI1+1,IY) = FLOAT(MINH)
  200 YPLT(NPHI1+2,IY) = FLOAT(MAXH - MINH)/YINCH
C
```

```
C     LABEL THE HORIZONTAL AXIS IN DEGREES
C
      CALL PLOT(0.,0.,3)
      CALL PLOT(XINCH,0.,2)
      Y1 = -0.45*H
      Y2 = -2.35*H
      CALL SYMBOL(0.,Y1,H,13,0.,-1)
      CALL SYMBOL(0.-0.5*HBOX,Y2,H,1H0,0.,1)
      CALL SYMBOL(0.25*XINCH,Y1,H,13,0.,-1)
      CALL SYMBOL(0.25*XINCH-HBOX,Y2,H,2H90,0.,2)
      CALL SYMBOL(0.5*XINCH,Y1,H,13,0.,-1)
      CALL SYMBOL(0.5*XINCH-1.5*HBOX,Y2,H,3H180,0.,3)
      CALL SYMBOL(0.75*XINCH,Y1,H,13,0.,-1)
      CALL SYMBOL(0.75*XINCH-1.5*HBOX,Y2,H,3H270,0.,3)
      CALL SYMBOL(XINCH,Y1,H,13,0.,-1)
      CALL SYMBOL(XINCH-1.5*HBOX,Y2,H,3H360,0.,3)
      CALL SYMBOL(0.5*XINCH-7.*HBOX,-4.*H,H,14HPHI IN DEGREES,0.,14)
C
      IF(ITYPE.EQ.7) THEN
           IF(ITHETA.GT.0) THEN
           ENCODE(41,210,BCD) THETA(IT)*RADEG
           NCHAR = 41
           ELSE
           ENCODE(43,212,BCD) THETA(IT)*RADEG
           NCHAR = 43
           ENDIF
      ELSE
           IF(ITHETA.GT.0) THEN
           ENCODE(39,214,BCD) THETA(IT)*RADEG
           NCHAR = 39
           ELSE
           ENCODE(41,216,BCD) THETA(IT)*RADEG
           NCHAR = 41
           ENDIF
      ENDIF
      XX = 0.5*XINCH - 0.5*FLOAT(NCHAR)*HBOX
      CALL SYMBOL(XX,-0.9,H,BCD,0.0,NCHAR)
C
      IF(NTIT.GT.0) THEN
      XX = 0.5*XINCH - 0.5*FLOAT(NTIT)*HBOX
      CALL SYMBOL(XX,YINCH+3.0*H,H,ITITLE,0.,NTIT)
      ENDIF
C
C     PLOT THE RADIANCES
C
      DO 400 K=1,IYMAX
      IY = IYPLT(K)
      IF(IY.EQ.1 .AND. IYT.NE.1) GO TO 400
      IF(IY.EQ.NY .AND. IYB.NE.NY) GO TO 400
      ENCODE(10,401,BCD) Y(IY)
      CALL LINE(XPLT(1,IY),YPLT(1,IY),NPHI1,1,ISYMBL,1)
      YY = (YPLT(NPHI1,IY) - YPLT(NPHI1+1,IY))/YPLT(NPHI1+2,IY) - 0.5*H
  400 CALL SYMBOL(XINCH,YY,H,BCD,0.0,10)
C
      CALL PLOT(-1.,-2.,-3)
      CALL PLOT(10.0,0.0,-3)
      WRITE(6,802)
      RETURN
C
C     FORMATS
C
  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(35HTOTAL 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 =',I3)
  802 FORMAT(1H ,' END OF PPHILOG')
      END
```

```
      SUBROUTINE PTHELOG(ITYPE,NYPLT,IYPLT,JPHI,JPI)
C
C     ON NHM6/PTHELOG
C
C     THIS ROUTINE PLOTS THE LOGARITHM OF THE DIFFUSE OR TOTAL RADIANCE
C     AS A FUNCTION OF THETA FOR HALF PLANES DEFINED BY JPHI AND JPI.
C     ALL Y VALUES ARE DISPLAYED ON THE SAME PLOT.
C
C     IF(ITYPE.EQ.3, THE DIFFUSE RADIANCE IS PLOTTED
C     IF(ITYPE.EQ.4, THE TOTAL RADIANCE IS PLOTTED
C
      PARAMETER(MXMU=10, MXPHI=24, MXY=30)
      PARAMETER(MXPTS=4*MXMU+1, MXY1=MXY+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,MXY1),YPLT(MXPTS,MXY1),BCD(4),NPLT(MXY1)
C
      DIMENSION IYPLT(MXY)
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)
      RADEG = FMISC(3)
      PI2 = 0.5*PI
      IYMAX = NY
      IF(NYPLT.GT.0) IYMAX = NYPLT
      HBOX = H*BOX
      IF(ITYPE.EQ.3) THEN
      FACT = 0.
      ELSEIF(ITYPE.EQ.4) THEN
      FACT = 1.
      ELSE
      WRITE(6,800) ITYPE
      RETURN
      ENDIF
      CALL PLOT(1.,2.,-3)
C
C     DETERMINE THE RANGE OF THETA AT THE FIRST AND LAST IY 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
      IYTP = IYBP = 0
      EPSREL = EPS*RADM(1,JPHI,IYPLT(IYMAX))
      DO 700 I=1,NMU
      IF(RADM(I,JPHI,IYPLT(1)).GT.EPS) IYTP = 1
      IF(RADM(I,JPI,IYPLT(1)).GT.EPS) IYTP = 1
      IF(RADP(I,JPHI,IYPLT(IYMAX)).GT.EPSREL) IYBP = 1
      IF(RADP(I,JPI,IYPLT(IYMAX)).GT.EPSREL) IYBP = 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
C     SPECIAL CASE:  THE TOP BOUNDARY REQUIRES TWO PLOTS FOR A NAKED UPPER BOUNDARY
C
      LY = LY + 1
      L = 0
      L = L + 1
      XPLT(L,LY) = THETA(NMU) - PI
      YPLT(L,LY) = ALOG10(RADP(NMU,1,1))
```

```
      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,JPI,1))
      L = L + 1
      XPLT(L,LY) = PI - THETA(NMU)
      YPLT(L,LY) = ALOG10(RADP(NMU,1,1))
      NPLT(LY) = L
C
  200 DO 150 K=1,IYMAX
      IY = IYPLT(K)
      IF(IY.EQ.1 .AND. LY.GT.0) GO TO 150
      LY = LY + 1
      L = 0
      IF(IY.EQ.NY .AND. IYBP.EQ.0) GO TO 169
C
      L = L + 1
      XPLT(L,LY) = THETA(NMU) - PI
      YPLT(L,LY) = ALOG10(RADP(NMU,1,IY))
      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,IY))
C
  169 DO 170 I=1,NMU-1
      L = L + 1
      XPLT(L,LY) = -THETA(I)
  170 YPLT(L,LY) = ALOG10(RADM(I,JPHI,IY) + FACT*RADOM(I,JPHI,IY))
C
      L = L + 1
      XPLT(L,LY) = -THETA(NMU)
      YPLT(L,LY) = ALOG10(RADM(NMU,1,IY) + FACT*RADOM(NMU,1,IY))
      DO 180 I=1,NMU-1
      L = L + 1
      II = NMU - I
      XPLT(L,LY) = THETA(II)
  180 YPLT(L,LY) = ALOG10(RADM(II,JPI,IY) + FACT*RADOM(II,JPI,IY))
C
      IF(IY.EQ.NY .AND. IYBP.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,JPI,IY))
      L = L + 1
      XPLT(L,LY) = PI - THETA(NMU)
      YPLT(L,LY) = ALOG10(RADP(NMU,1,IY))
C
  149 NPLT(LY) = L
  150 CONTINUE
      NUMPLT = LY
C
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
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(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)
       IDIV1 = IDIV + 1
C
       CALL PLOT(0.,0.,3)
       CALL PLOT(0.,YINCH,2)
       XX = -7.6*BOX*H
       DO 310 I=1,IDIV1
       YY = YINCH - 0.45*H - FLOAT(I-1)*DINCH
       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*YINCH - 6.5*BOX*H
       CALL SYMBOL(XX,YY,H,13HLOG(RADIANCE),90.0,13)
C
C      LABEL THE HORIZONTAL AXIS
C
       CALL PLOT(0.,0.,3)
       CALL PLOT(XINCH,0.,2)
C
       Y1 = -0.45*H
       Y2 = -2.35*H
       CALL SYMBOL(0.,Y1,H,13,0.,-1)
       CALL SYMBOL(0.-2.*HBOX,Y2,H,4H-180,0.,4)
       CALL SYMBOL(0.25*XINCH,Y1,H,13,0.,-1)
       CALL SYMBOL(0.25*XINCH-1.5*HBOX,Y2,H,3H-90,0.,3)
       CALL SYMBOL(0.5*XINCH,Y1,H,13,0.,-1)
       CALL SYMBOL(0.5*XINCH-0.5*HBOX,Y2,H,1H0,0.,1)
       CALL SYMBOL(0.75*XINCH,Y1,H,13,0.,-1)
       CALL SYMBOL(0.75*XINCH-HBOX,Y2,H,2H90,0.,2)
       CALL SYMBOL(XINCH,Y1,H,13,0.,-1)
       CALL SYMBOL(XINCH-1.5*HBOX,Y2,H,3H180,0.,3)
       CALL SYMBOL(0.5*XINCH-15.*HBOX,-4.*H,H,
      1 30HVIEWING ANGLE THETA IN DEGREES,0.,30)
C
       H1 = 0.7*H
       H1BOX = 0.7*HBOX
       ENCODE(11,210,BCD) RADEG*PHI(JPI)
       CALL SYMBOL(0.25*XINCH-5.5*H1BOX,-0.95,H1,BCD,0.,11)
       ENCODE(11,210,BCD) RADEG*PHI(JPHI)
       CALL SYMBOL(0.75*XINCH-5.5*H1BOX,-0.95,H1,BCD,0.,11)
C
       IF(ITYPE.EQ.3) THEN
       CALL SYMBOL(0.5*XINCH-22.5*HBOX,-1.3,H,
      1  45HDIFFUSE FIELD RADIANCE AS A FUNCTION OF THETA,0.,45)
       ELSEIF(ITYPE.EQ.4) THEN
       CALL SYMBOL(0.5*XINCH-21.5*HBOX,-1.3,H,
      1 43HTOTAL FIELD RADIANCE AS A FUNCTION OF THETA,0.,43)
       ENDIF
C
C      PLOT THE REFERENCE LINE AT THETA = 0 AND THE TITLE
       CALL PLOT(0.5*XINCH,0.,3)
       CALL DASHPT(0.5*XINCH,YINCH,0.1)
       IF(NTIT.GT.0) THEN
       XX = 0.5*XINCH - 0.5*FLOAT(NTIT)*HBOX
       CALL SYMBOL(XX,YINCH+3.0*H,H,ITITLE,0.,NTIT)
       ENDIF
C
```

```
C      PLOT THE RADIANCES
C
       IY = 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)/YINCH
       CALL LINE(XPLT(1,LY),YPLT(1,LY),NPTS,1,ISYMBL,1)
       IF(LY.EQ.1 .AND. NUMPLT.GT.NYPLT) GO TO 400
       IY = IY + 1
       ENCODE(10,401,BCD) Y(IYPLT(IY))
       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
C
       CALL PLOT(-1.,-2.,-3)
       CALL PLOT(10.0,0.0,-3)
       WRITE(6,802)
       RETURN
C
C      FORMATS
C
  210 FORMAT(5HPHI =,F6.1)
  311 FORMAT(F6.2,2H -)
  401 FORMAT(4H Y =,F6.2)
  800 FORMAT(' ERROR:  SUB PTHELOG CALLED WITH ITYPE =',I3)
  802 FORMAT(' END OF PTHELOG')
       END
```

```
       SUBROUTINE PTHEPLR(ITYPE,NYPLT,IYPLT,JPHI,JPI)
C
C      ON NHM6/PTHEPLR
C
C      THIS ROUTINE MAKES A POLAR PLOT OF THE LOGARITHM OF THE DIFFUSE
C      OR TOTAL RADIANCE AS A FUNCTION OF THETA FOR HALF-PLANES DEFINED
C      BY JPHI AND JPI. A SEPARATE PLOT IS MADE FOR EACH DEPTH.
C
C      IF ITYPE.EQ.1, THE DIFFUSE RADIANCE IS PLOTTED
C      IF ITYPE.EQ.2, THE TOTAL RADIANCE IS PLOTTED
C
       PARAMETER(MXMU=10, MXPHI=24, MXY=30)
       PARAMETER(MXPTS=4*MXMU+1)
C
       COMMON/CGRID/ THETA(MXMU),PHI(MXPHI),YOD(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/ X(MXPTS),Y(MXPTS),RC(20)
C
```

```
      DIMENSION TITLE(6),IYPLT(MXY)
C
      DATA ISYMBL/1/, RINCH/3.0/, DICIRC/0.5/, H,BOX/0.15,1.0/
C
      CALL PLOT(4.,5.,-3)
C
      NMU = IMISC(1)
      NY = IMISC(4)
      PI = FMISC(1)
      DEGRAD = FMISC(2)
      RADEG = FMISC(3)
      PI2 = 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(6,800) ITYPE
      RETURN
      ENDIF
      IYMAX = NY
      IF(NYPLT.GT.0) IYMAX = NYPLT
C
      DO 888 K=1,IYMAX
      IY = IYPLT(K)
C
C     CONVERT RADIANCES AND NHM THETA VALUES (MEASURED FROM THE +Y
C     AXIS) TO X = LOG(RADIANCE) AND Y = CALCOMP THETA (MEASURED FROM
C     THE +X AXIS)
C     POLAR CAPS ALWAYS HAVE A PHI INDEX OF 1
C
      L = 0
      DO 200 I=1,NMU-1
      L = L + 1
      X(L) = ALOG10(RADP(I,JPHI,IY))
  200 Y(L) = PI2 - THETA(I)
      L = L + 1
      X(L) = ALOG10(RADP(NMU,1,IY))
      Y(L) = PI2
C
      DO 201 I=1,NMU-1
      L = L + 1
      II = NMU - I
      X(L) = ALOG10(RADP(II,JPI,IY))
  201 Y(L) = PI2 + THETA(II)
C
      DO 202 I=1,NMU-1
      L = L + 1
      X(L) = ALOG10(RADM(I,JPI,IY) + FACT*RADOM(I,JPI,IY))
  202 Y(L) = PI32 - THETA(I)
      L = L + 1
      X(L) = ALOG10(RADM(NMU,1,IY) + FACT*RADOM(NMU,1,IY))
      Y(L) = PI32
C
      DO 203 I=1,NMU-1
      L = L + 1
      II = NMU - I
      X(L) = ALOG10(RADM(II,JPHI,IY) + FACT*RADOM(II,JPHI,IY))
  203 Y(L) = PI32 + THETA(II)
C
      L = L + 1
      X(L) = X(1)
      Y(L) = Y(1)
      NPTS = L
C
C     FIND THE MAXIMUM AND MINIMUM LOG VALUES TO BE PLOTTED
C
      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
C
```

```
C     LABEL THE RADIAL (VERTICAL) AXIS FOR A LOG PLOT
C
      MINH = IFIX(RADMIN)
      IF(RADMIN.LT.0.) MINH = MINH - 2
      MAXH = IFIX(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(IFIX(DLABL)).EQ.DLABL) GO TO 303
      MRANGE = IDIV*IFIX(DLABL + 1.)
      GO TO 300
  303 DINCH = RINCH/FLOAT(IDIV)
C
      CALL PLOT(0.,-RINCH,3)
      CALL PLOT(0.,RINCH,2)
      CALL PLOT(-RINCH,0.,3)
      CALL PLOT(-1.1,0.,2)
      CALL PLOT(0.,0.,3)
      CALL PLOT(RINCH,0.,2)
      XX = -7.6*BOX*H
      DO 310 I=1,IDIV+1
      YY = RINCH - 0.45*H - FLOAT(I-1)*DINCH
      RC(I) = RINCH - FLOAT(I-1)*DINCH
      FLABL = FLOAT(MAXH) - FLOAT(I-1)*DLABL
      ENCODE(8,411,BCD) FLABL
  310 CALL SYMBOL(XX,YY,H,BCD,0.0,8)
      XX = -1.1
      YY = 0.5*RINCH - 6.5*BOX*H
      CALL SYMBOL(XX,YY,H,13HLOG(RADIANCE),90.0,13)
C
C     DRAW MAGNITUDE CIRCLES
C
      DO 260 I=1,IDIV,2
      R = RC(I)
      IF(R.GT.1.35) THEN
      THO = 90.0 + RADEG*ASIN(1.35/R)
      ELSEIF(R.LE.1.35 .AND. R.GT.0.1) THEN
      THO = 180. + RADEG*ASIN(0.1/R)
      ELSE
      THO = 180.0
      ENDIF
      XS = R*COS(DEGRAD*THO)
      YS = R*SIN(DEGRAD*THO)
      CALL CIRCLE(XS,YS,THO,360.,R,R,DICIRC)
  260 CALL CIRCLE(R,0.,0.,90.,R,R,DICIRC)
C
C     CONVERT LOG(RADIANCE) AND CALCOMP THETA TO X AND Y IN INCHES
C
      C1 = RINCH/FLOAT(MAXH - MINH)
      C2 = -FLOAT(MINH)*C1
      DO 500 L=1,NPTS
      RIN = C1*X(L) + C2
      X(L) = RIN*COS(Y(L))
  500 Y(L) = RIN*SIN(Y(L))
C
C     SCALE THE RADIANCES
C
      X(NPTS+1) = 0.
      Y(NPTS+1) = 0.
      X(NPTS+2) = 1.0
      Y(NPTS+2) = 1.0
C
C     PLOT RADIANCES
C
      CALL LINE(X,Y,NPTS,1,ISYMBL,1)
C
C     LABEL THE PLOT
C
```

```
      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.*HBOX,-3.5,H,TITLE,0.,26)
      IF(NTIT.GT.0) THEN
      XX = -0.5*FLOAT(NTIT)*HBOX
      CALL SYMBOL(XX,RINCH+3.0*H,H,ITITLE,0.,NTIT)
      ENDIF
C
  888 CALL PLOT(10.0,0.0,-3)
C
      CALL PLOT(-4.,-5.,-3)
      WRITE(6,802)
      RETURN
C
  400 FORMAT(46HDIFFUSE RADIANCE AS A FUNCTION OF THETA AT Y =,F6.2)
  402 FORMAT(44HTOTAL RADIANCE AS A FUNCTION OF THETA AT Y =,F6.2)
  404 FORMAT(5HPHI =,F6.1,4X,5HPHI =,F6.1)
  411 FORMAT(F6.2,2H -)
  800 FORMAT(1H ,' ERROR:  SUB PTHEPLR CALLED WITH ITYPE =',I3)
  802 FORMAT(1H ,' END OF PTHEPLR')
      END
```

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

LABPNT	= 0	if the plotted points are not labeled.
	= 1	if each plotted point is numbered on the chromaticity diagram, and a separate table of numbers is plotted, along with a label for each number to identify the plotted points.
IPLBLU	= 0	if only the full chromaticity diagram is to be drawn.
	= 1	if only the "blue corner" of the chromaticity diagram is to be drawn. This is often useful for plotting ocean color, which usually lies in the blue region.
	= 2	if both the full diagram and the blue corner are to be drawn.
IATMOS	= 0	if the raw data values, P(LAMBDA), are to be used in computing the color.
	= 1	if the raw data values, P(LAMBDA), are to be transformed by the atmospheric model of Appendix D before use.

Record 2: ITOP

This record gives a title for the top of the plot. Up to 80 alphanumeric 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) ⋮ P(13) 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,TAPE5=INPUT,TAPE6=OUTPUT,TAPE98,
     1  TAPE99)
C
C     ON NHM6/MPCHRO
C
C     THIS PROGRAM COMPUTES AND PLOTS CHROMATICITIES ON A STANDARD CIE
C     1931 CHROMATICITY DIAGRAM, GIVEN RADIANCES OR IRRADIANCES AT 13
C     WAVELENGTHS:  400. NM, 425. NM, 450. NM, ..., 675. NM, 700. NM
C
C     ALL PLOTTING IS DONE USING STANDARD CALCOMP CALLS
C     (TAPE98 AND TAPE99 ARE USED BY THE CALCOMP ROUTINES, AS
C     IMPLEMENTED ON THE AUTHOR'S CDC CYBER 855 COMPUTER.)
C
      PARAMETER(NWAVEL=13, MXPTS=50)
C
      DIMENSION P(NWAVEL),IWAVEL(NWAVEL),RAWP(NWAVEL)
      DIMENSION XCHR(MXPTS),YCHR(MXPTS),ITOP(8),ITITLE(8,MXPTS)
C
      DATA IWAVEL/400,425,450,475,500,525,550,575,600,625,650,675,700/
C
C     READ THE OVERALL PLOT SPECS, AND A TITLE FOR THE TOP OF THE PLOT
C
C     LABPNT = 0, IF POINTS ARE NOT TO BE LABELLED
C              1, IF EACH POINT IS NUMBERED AND LABELLED
C     IPLBLU = 0, IF ONLY THE FULL CHROMATICITY DIAGRAM IS TO BE DRAWN
C              1, IF ONLY THE BLUE CORNER IS TO BE DRAWN
C              2, IF BOTH FULL AND BLUE CORNER ARE TO BE DRAWN
C     IATMOS = 0, IF THE RAW P(LAMBDA) ARE TO BE USED
C              1, IF THE RAW P(LAMBDA) ARE TO BE SCALED BY THE ATMOSPHERIC MODEL
C     ITOP...A TITLE FOR THE TOP OF THE PLOT (80 CHAR MAX)
C
      NPTS = 0
      READ(5,*) LABPNT,IPLBLU,IATMOS
      READ(5,100) ITOP
      WRITE(6,90)
C
   99 NPTS = NPTS + 1
C
C     READ A LABEL AND A SET OF VALUES TO BE PROCESSED
C
      READ(5,100,END=900) (ITITLE(I,NPTS),I=1,8)
      READ(5,*,END=900) THETPS,(P(I),I=1,NWAVEL)
      DO 150 I=1,NWAVEL
  150 RAWP(I) = P(I)
C
C     SCALE THE P(I) ACCORDING TO THE ATMOSPHERIC MODEL
C
      IF(IATMOS.NE.0) CALL ATMOS(THETPS,P)
C
C     COMPUTE THE CHROMATICITY COORDINATES
C
      CALL CHRMXY(P,XCHR(NPTS),YCHR(NPTS),DOMWVL,PURITY)
C
      WRITE(6,200) NPTS,(ITITLE(I,NPTS),I=1,8)
      IF(IATMOS.NE.0) WRITE(6,201) THETPS
      WRITE(6,203)
      DO 202 I=1,NWAVEL
  202 WRITE(6,204) IWAVEL(I),P(I),RAWP(I)
      WRITE(6,206) XCHR(NPTS),YCHR(NPTS),DOMWVL,PURITY
C
      GO TO 99
C
C     DRAW A CHROMATICITY DIAGRAM AND PLOT POINTS ON IT
C
  900 NPTS = NPTS - 1
C
```

```
      CALL PLOTS
      CALL PLOT(2.,2.,-3)
      IF(IPLBLU.NE.0) CALL PLTBLU(XCHR,YCHR,NPTS,LABPNT,ITOP,ITITLE)
      CALL PLOT(10.,0.,-3)
      IF(IPLBLU.NE.1) CALL PLTCHR(XCHR,YCHR,NPTS,LABPNT,ITOP,ITITLE)
      CALL PLOT(0.,0.,-98)
C
C     FORMATS
C
   90 FORMAT(1H1)
  100 FORMAT(8A10)
  200 FORMAT(///' POINT NUMBER',I3,':  ',8A10/)
  201 FORMAT('  THE ATMOSPHERIC MODEL WITH THETPS =',F6.2,' IS USED'/)
  203 FORMAT('  THE FUNCTION OF WAVELENGTH GIVEN BY'//
     1'  LAMBDA     P(LAMBDA)         (RAW P)'/)
  204 FORMAT(I6,1P2E16.4)
  206 FORMAT(/' HAS CHROMATICITY COORDINATES (X,Y) = (',F5.4,',',
     1 F6.4,') OR (DOMINANT WAVELENGTH, PURITY) =(',F5.1,',',F6.4,')')
      END
```

```
      SUBROUTINE ATMOS(THETPS,P)
C
C     NHM6/ATMOS
C
C     GIVEN:  A SET OF VALUES P(I), I=1,...,NWAVEL, WHICH ARE THE
C     OUTPUT OF 13 NHM RUNS AT THE 13 NHM WAVELENGTHS, WHERE EACH
C     RUN WAS INITIALIZED WITH UNIT SCALAR IRRADIANCE.  THIS ROUTINE
C     SCALES THE P(I) VALUES TO REFLECT THE WAVELENGTH AND SOLAR ANGLE
C     DEPENDENCE OF THE INITIALIZING SCALAR IRRADIANCES, ACCORDING
C     TO THE MODEL ATMOSPHERE AND SOLAR SPECTRUM DESCRIBED IN
C     APPENDIX D OF THIS REPORT.
C
      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  1.62,1.53,1.44/
      DATA ALPHAL/.566, .428, .364, .293, .217, .210, .220, .206,
     1 .192, .165, .134, .114, .104/
      DATA DEGRAD/0.017453293/
C
      SECTH = 1.0/COS(DEGRAD*THETPS)
      DO 100 I=1,NWAVEL
  100 P(I) = P(I)*SOLARC(I)*EXP(-ALPHAL(I)*SECTH)
C
      RETURN
      END
```

```
      SUBROUTINE CHRMXY(P,X,Y,DOMWVL,PURITY)
C
C     ON NHM6/CHRMXY
C
C     GIVEN A SET OF RADIANCES OR IRRADIANCES, P, AT THE NHM WAVELENGTHS,
C     THIS ROUTINE COMPUTES THE CHROMATICITY COORDINATES (X,Y) FOR
C     PLOTTING ON A CHROMATICITY DIAGRAM.  THE CORRESPONDING DOMINANT
C     WAVELENGTH AND PURITY ARE ALSO COMPUTED.  SEE APPENDIX C OF THIS
C     REPORT FOR DETAILS.
C
      PARAMETER (NWAVEL=13, MXPURE=37)
      DIMENSION P(NWAVEL),XBAR(NWAVEL),YBAR(NWAVEL),ZBAR(NWAVEL)
      DIMENSION WAVEL(MXPURE),XPURE(MXPURE),YPURE(MXPURE),SLOPUR(MXPURE)
C
C     THE 13 TRISTIMULUS FUNCTION VALUES
      DATA XBAR/.0143,.2148,.3362,.1421,.0049,.1096,.4334,.8425,
     1  1.0622,.7514,.2835,.0636,.0114/
      DATA YBAR/.0004,.0073,.0380,.1126,.3230,.7932,.9950,.9154,
     1  .6310,.3210,.1070,.0232,.0041/
      DATA ZBAR/.0679,1.0391,1.7721,1.0419,.2720,.0573,.0087,.0018,
     1  .0008,.0001,0.,0.,0./
C
C     THE 37 SPECTRUM LOCUS VALUES
      DATA WAVEL/400.,450.,460.,470.,475.,480.,485.,490.,
     1495.,500.,505.,510.,515.,520.,525.,530.,535.,540.,545.,
     2550.,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,.0082,.0039,.0139,.0389,.0743,.1142,.1547,
     2 .1929,.2296,.2658,.3016,.3373,.3731,.4087,.4441,.4788,
     3 .5125,.5448,.5752,.6029,.6270,.6482,.6658,.6915,.7079,
     4 .7190,.7260,.7347/
      DATA YPURE/.0048,.0177,.0297,.0578,.0868,.1327,.2007,
     1 .2950,.4127,.5384,.6548,.7502,.8120,.8338,.8262,.8059,
     2 .7816,.7543,.7243,.6923,.6589,.6245,.5896,.5547,.5202,
     3 .4866,.4544,.4242,.3965,.3725,.3514,.3340,.3083,.2920,
     4 .2809,.2740,.2653/
      DATA KALL/0/
C
C     COMPUTE INTEGRALS BY SIMPSON'S RULE, C.2
C
      CAPX = P(1)*XBAR(1) + P(NWAVEL)*XBAR(NWAVEL)
      CAPY = P(1)*YBAR(1) + P(NWAVEL)*YBAR(NWAVEL)
      CAPZ = P(1)*ZBAR(1) + P(NWAVEL)*ZBAR(NWAVEL)
C
      DO 100 I=2,NWAVEL-1,2
      CAPX = CAPX + 4.0*P(I)*XBAR(I)
      CAPY = CAPY + 4.0*P(I)*YBAR(I)
  100 CAPZ = CAPZ + 4.0*P(I)*ZBAR(I)
C
      DO 110 I=3,NWAVEL-2,2
      CAPX = CAPX + 2.0*P(I)*XBAR(I)
      CAPY = CAPY + 2.0*P(I)*YBAR(I)
  110 CAPZ = CAPZ + 2.0*P(I)*ZBAR(I)
C
      F = 6800./FLOAT(NWAVEL-1)
      CAPX = F*CAPX
      CAPY = F*CAPY
      CAPZ = F*CAPZ
C
C     NORMALIZE THE INTEGRALS TO GET THE CHROMATICITY COORDINATES, BY C.3
C
      X = CAPX/(CAPX + CAPY + CAPZ)
      Y = CAPY/(CAPX + CAPY + CAPZ)
C
      IF(KALL.EQ.0) THEN
C
C     COMPUTE THE SLOPES NEEDED FOR FINDING THE DOMINANT WAVELENGTH
C     AND PURITY
C
      XW = 1.0/3.0
      YW = 1.0/3.0
      DO 50 I=1,MXPURE
   50 SLOPUR(I) = (YW - YPURE(I))/(XW - XPURE(I))
      KALL = 1
      ENDIF
C
```

```
C     COMPUTE THE DOMINANT WAVELENGTH AND PURITY OF THE CHROMATICITY
C     COORDINATES (X,Y)
C
      DY = YW - Y
      DX = XW - X
      SLOPE = DY/DX
C
   99 IF(DY.GE.0. .AND. DX.GE.0.) THEN
C
C     SEARCH LOWER LEFT OF SPECTRUM LOCUS, POINTS 2 TO 9
      DO 200 I=2,9
      IF(SLOPE.LT.SLOPUR(1) .AND. SLOPE.GE.SLOPUR(I)) GO TO 250
  200 CONTINUE
C
      ELSEIF(DY.LE.0. .AND. DX.GE.0) THEN
C
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
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
C     SEARCH LOWER RIGHT OF SPECTURM LOCUS, POINTS 33-37
      DO 206 I=33,37
      IF(SLOPE.GE.SLOPUR(I)) GO TO 250
  206 CONTINUE
C
      ENDIF
C
C     POINT IS IN PURPLE REGION, REVERSE (X,Y) AND THE WHITE POINT AND
C     SEARCH AGAIN
      DX = - DX
      DY = - DY
      GO TO 99
C
C     COMPUTE INTERSECTION POINT OF CHROMATICITY LINE AND SPECTRUM LOCUS
C
  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)
C
      XI = (S2*XW - S1*XPIM1 - S1*S2*(YW - YPIM1))/(S2 - S1)
      YI = (XW - XPIM1 - S1*YW + S2*YPIM1)/(S2 - S1)
C
C     GET DOMINANT WAVELENGTH BY INTERPOLATION
      DT = SQRT((XPI - XPIM1)**2 + (YPI - YPIM1)**2)
      DI = SQRT((XI - XPIM1)**2 + (YI - YPIM1)**2)
      F = DI/DT
C
      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
```

```
      SUBROUTINE PLTBLU(XPLT,YPLT,NPTS,LABPNT,ITOP,ITITLE)
C
C     ON NHM6/PLTBLU
C
C     THIS ROUTINE DRAWS THE 'BLUE CORNER' OF A CHROMATICITY DIAGRAM
C     AND PLOTS POINTS ON IT
C
C     STANDARD CALCOMP PLOTTING ROUTINES ARE USED
C
      PARAMETER (MXPURE=9)
      DIMENSION IWAVEL(MXPURE),XPURE(MXPURE),YPURE(MXPURE)
      DIMENSION XPLT(NPTS),YPLT(NPTS),ITOP(8),ITITLE(8,NPTS)
C
      DATA IWAVEL/400,450,460,470,475,480,485,490,494/
      DATA XPURE/.1733,.1566,.1440,.1241,.1096,.0913,.0687,.0454,.0259/
      DATA YPURE/.0048,.0177,.0297,.0578,.0868,.1327,.2007,.2950,.4/
C
      DATA XINCH,YINCH/4.0,4.0/, HTIC/0.1/
      DATA RADEG/57.295779513/
C
C     INITIALIZE AND DRAW AXES
C
      CALL AXIS(0.0,0.0,1HX,-1,XINCH,0.0,0.0,0.1)
      CALL AXIS(0.0,0.0,1HY,1,YINCH,90.0,0.0,0.1)
      CALL PLOT(0.,YINCH,3)
      CALL PLOT(XINCH,YINCH,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, YINCH+2.0*HTIC, HTIC,
     1 ITOP,0.0,NC)
      ENDIF
C
C     DRAW THE SPECTRUM LOCUS
C
      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
C     ADD TIC MARKS
      SLOPE = -(XPURE(I+1) - XPURE(I-1))/(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*0.1100,3)
      CALL PLOT(XSCALE*XPURE(1),YSCALE*YPURE(1),2)
C
C     ADD WAVELENGTH LABELS TO SELECTED TICS
C
      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)
C
C     PLOT THE WHITE POINT
C
      CALL SYMBOL(XSCALE/3.0,YSCALE/3.0,0.2,3,0.0,-1)
C
C     PLOT CHROMATICITY COORDINATES (XPLT,YPLT) ON THE DIAGRAM
C
      IF(NPTS.GT.0) THEN
      DO 200 I=1,NPTS
      X = XSCALE*XPLT(I)
      Y = YSCALE*YPLT(I)
      CALL SYMBOL(X,Y,0.5*HTIC,1,0.0,-1)
      IF(LABPNT.NE.0) THEN
      FPN = FLOAT(I)
      CALL NUMBER(X+0.5*HTIC,Y-0.5*HTIC,HTIC,FPN,0.0,-1)
```

```
      X = XINCH + 1.0
      Y = YINCH - FPN*2.0*HTIC
      CALL NUMBER(X,Y,HTIC,FPN,0.0,-1)
      NS = NCHAR(ITITLE(1,I),8)
      CALL SYMBOL(X+2.5*HTIC,Y,HTIC,ITITLE(1,I),0.0,NS)
      ENDIF
  200 CONTINUE
      ENDIF
C
      CALL PLOT(0.,0.,-3)
      RETURN
C
  120 FORMAT(1H ,I3)
      END
```

```
      SUBROUTINE PLTCHR(XPLT,YPLT,NPTS,LABPNT,ITOP,ITITLE)
C
C     ON NHM6/PLTCHR
C
C     THIS ROUTINE DRAWS A CHROMATICITY DIAGRAM AND PLOTS POINTS ON IT
C
C     STANDARD CALCOMP PLOTTING ROUTINES ARE USED
C
      PARAMETER (MXPURE=37)
      DIMENSION IWAVEL(MXPURE),XPURE(MXPURE),YPURE(MXPURE)
      DIMENSION XPLT(NPTS),YPLT(NPTS),ITOP(8),ITITLE(8,NPTS)
C
      DATA IWAVEL/400,450,460,470,475,480,485,490,
     1495,500,505,510,515,520,525,530,535,540,545,
     2550,555,560,565,570,575,580,585,590,595,600,
     3605,610,620,630,640,650,700/
      DATA XPURE/.1733,.1566,.1440,.1241,.1096,.0913,.0687,
     1 .0454,.0235,.0082,.0039,.0139,.0389,.0743,.1142,.1547,
     2 .1929,.2296,.2658,.3016,.3373,.3731,.4087,.4441,.4788,
     3 .5125,.5448,.5752,.6029,.6270,.6482,.6658,.6915,.7079,
     4 .7190,.7260,.7347/
      DATA YPURE/.0048,.0177,.0297,.0578,.0868,.1327,.2007,
     1 .2950,.4127,.5384,.6548,.7502,.8120,.8338,.8262,.8059,
     2 .7816,.7543,.7243,.6923,.6589,.6245,.5896,.5547,.5202,
     3 .4866,.4544,.4242,.3965,.3725,.3514,.3340,.3083,.2920,
     4 .2809,.2740,.2653/
C
      DATA XINCH,YINCH/4.0,4.5/, HTIC/0.1/
      DATA RADEG/57.295779513/
C
C     INITIALIZE AND DRAW AXES
C
      CALL AXIS(0.0,0.0,1HX,-1,XINCH,0.0,0.0,0.2)
      CALL AXIS(0.0,0.0,1HY,1,YINCH,90.0,0.0,0.2)
      CALL PLOT(0.,YINCH,3)
      CALL PLOT(XINCH,YINCH,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, YINCH+2.0*HTIC, HTIC,
     1 ITOP,0.0,NC)
      ENDIF
C
```

```
C      DRAW THE SPECTRUM LOCUS
C
       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)
C
C      ADD TIC MARKS
       SLOPE = -(XPURE(I+1) - XPURE(I-1))/(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*0.2463,YSCALE*0.0387,2)
       CALL PLOT(XSCALE*0.1845,YSCALE*0.0100,3)
       CALL PLOT(XSCALE*XPURE(1),YSCALE*YPURE(1),2)
C
C      ADD WAVELENGTH LABELS TO SELECTED TICS
C
       ENCODE(4,120,BCD) IWAVEL(1)
       CALL SYMBOL(XSCALE*XPURE(1),YSCALE*YPURE(1)-0.25*HTIC,HTIC,BCD,
      1  20.0,4)
       ENCODE(4,120,BCD) IWAVEL(3)
       CALL SYMBOL(XSCALE*XPURE(3)-4.0*HTIC, YSCALE*YPURE(3)-2.0*HTIC,
      1  HTIC,BCD,20.0,4)
       ENCODE(4,120,BCD) IWAVEL(4)
       CALL SYMBOL(XSCALE*XPURE(4)-4.0*HTIC, YSCALE*YPURE(4)-2.0*HTIC,
      1  HTIC,BCD,20.0,4)
       ENCODE(4,120,BCD) IWAVEL(6)
       CALL SYMBOL(XSCALE*XPURE(6)-4.0*HTIC, YSCALE*YPURE(6)-2.0*HTIC,
      1  HTIC,BCD,20.0,4)
       DO 110 I=8,34,2
       ENCODE(4,120,BCD) IWAVEL(I)
  110  CALL SYMBOL(XSCALE*XPURE(I),YSCALE*YPURE(I)-0.25*HTIC,HTIC,BCD,
      1  20.0,4)
       ENCODE(4,120,BCD) IWAVEL(37)
       CALL SYMBOL(XSCALE*XPURE(37),YSCALE*YPURE(37)-0.25*HTIC,HTIC,BCD,
      1  20.0,4)
C
C      PLOT THE WHITE POINT
C
       CALL SYMBOL(XSCALE/3.0,YSCALE/3.0,0.2,3,0.0,-1)
C
C      PLOT CHROMATICITY COORDINATES (XPLT,YPLT) ON THE DIAGRAM
C
       IF(NPTS.GT.0) THEN
       DO 200 I=1,NPTS
       X = XSCALE*XPLT(I)
       Y = YSCALE*YPLT(I)
       CALL SYMBOL(X,Y,0.5*HTIC,1,0.0,-1)
       IF(LABPNT.NE.0) THEN
       FPN = FLOAT(I)
       CALL NUMBER(X+0.5*HTIC,Y-0.5*HTIC,HTIC,FPN,0.0,-1)
       X = XINCH + 1.0
       Y = YINCH - FPN*2.0*HTIC
       CALL NUMBER(X,Y,HTIC,FPN,0.0,-1)
       NS = NCHAR(ITITLE(1,I),8)
       CALL SYMBOL(X+2.5*HTIC,Y,HTIC,ITITLE(1,I),0.0,NS)
       ENDIF
  200  CONTINUE
       ENDIF
C
       CALL PLOT(0.,0.,-3)
       RETURN
C
  120  FORMAT(1H ,I3)
       END
```

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

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.

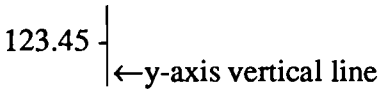
Record 3b: YFMT

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 ",2HΔ-)", which draws tic marks with the "-" in the 2HΔ-. The "Δ" symbol indicates a blank. The default is

(F6.2, 2HΔ-)

This generates tic mark labels of the form



where the "-" from the "2HΔ-" 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 2HΔ-).

The format

(F3.0, 2HΔ-)

would generate tic mark labels of the form

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.

2. Code Listing

```
      PROGRAM MPWAVE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE98,
     1  TAPE99)
C
C     ON NHM6/MPWAVE
C
C     THIS PROGRAM PLOTS DATA AS A FUNCTION OF WAVELENGTH.
C
C     ALL PLOTTING IS DONE USING STANDARD CALCOMP CALLS
C     (TAPE98 AND TAPE99 ARE USED BY THE CALCOMP ROUTINES, AS
C     IMPLEMENTED ON THE AUTHOR'S CDC CYBER 855 COMPUTER.)
C
      PARAMETER(MXWAVE=13, MXTRAC=20)
      PARAMETER(MXPLT=MXWAVE+2)
      DIMENSION DATA(MXWAVE,MXTRAC),LABTRC(8,MXTRAC)
      DIMENSION LABYAX(8),ITITLE(8)
      COMMON/CSCALE/ YINCH,YMAX,YMIN,IDIV,NCODE,YFMT(2)
      COMMON/CWORK/ X(MXPLT),Y(MXPLT),TRACE(MXPLT,MXTRAC)
C
      DATA NWAVEL/13/
C
      CALL PLOTS
      CALL PLOT(2.0,2.0,-3)
C
C     READ THE TITLE FOR THE TOP OF THE PLOT
  998 READ(5,100,END=1000) ITITLE
C
C     TITLE FOR THE Y AXIS
      READ(5,100) LABYAX
C
C     READ PLOT SPECIFIERS
C
C     NTRACE = THE NUMBER OF TRACES (CURVES) TO BE DRAWN
C     ILOG = 1, IF LOG(BASE 10) OF THE DATA IS TO BE PLOTTED
C            0, IF ACTUAL DATA VALUES ARE TO BE PLOTTED
C     IAUTOY = 1, IF THE PLOT PROGRAM EXAMINES THE DATA TO DETERMINE
C                 CONVENIENT Y AXIS (ORDINATE) SCALING
C     IPLABL = 1, IF THE PLOTTED CURVES ARE TO BE NUMBERED AND LABELLED
C              0, IF THE CURVES ARE NOT NUMBERED AND LABELLED
C     IATMOS = 1, IF THE RAW DATA VALUES ARE TO BE SCALED BY THE
C                 WAVELENGTH-SOLAR ANGLE DEPENDENT ATMOSPHERIC MODEL
C              0, IF THE RAW DATA ARE NOT SCALED
C
      READ(5,*) NTRACE,ILOG,IAUTOY,IPLABL,IATMOS
C
C     READ THE SPECIFICATIONS FOR SCALING THE Y (VERTICAL) AXIS, IF DESIRED
C
C     YINCH...THE LENGTH OF THE Y AXIS, IN INCHES
C     YMAX, YMIN...THE MAXIMUM AND MINIMUM Y VALUES TO BE USED FOR
C                  THE Y AXIS LABELS
C     IDIV...THE NUMBER OF DIVISIONS OF THE Y AXIS, FOR LABELLING TIC MARKS
C     NCODE...THE NUMBER OF CHARACTERS IN THE TOTAL WIDTH OF THE Y-AXIS
C              TIC MARK LABELS (USED IN ENCODE STATEMENTS).  E.G. IF THE
C              NEXT DATA RECORD HAS (F6.2,2H -) AS THE FORMAT, THEN
C              NCODE = 6 + 2 = 8
C     YFMT...A FORMAT FOR LABELLING THE Y-AXIS TICS.  IT SHOULD END
C            WITH ,2H -) WHICH DRAWS THE TIC MARKS WITH THE MINUS SIGN
      IF(IAUTOY.EQ.0) THEN
      READ(5,*) YINCH,YMAX,YMIN,IDIV,NCODE
      READ(5,100) YFMT
      ENDIF
C
C     READ THE TRACE LABELS AND THE DATA TO BE PLOTTED, ON STANDARD
C     WAVELENGTH FORMAT
C
      DO 200 NTR=1,NTRACE
      READ(5,100) (LABTRC(I,NTR),I=1,8)
```

```
      READ(5,*) THETPS,(DATA(I,NTR),I=1,NWAVEL)
C
      IF(IATMOS.NE.0) THEN
C     SCALE THE RAW DATA ACCORDING TO THE ATMOSPHERIC MODEL
      CALL ATMOS(THETPS,DATA(1,NTR))
      ENDIF
C
  200 CONTINUE
C
C     PLOT THE DATA
C
      CALL PLTWVL(DATA,NTRACE,ILOG,IAUTOY,LABTRC,IPLABL,LABYAX,ITITLE)
      CALL PLOT(18.0,0.0,-3)
      GO TO 998
C
 1000 CALL PLOT(0.,0.,-98)
      WRITE(6,999)
C
C
  100 FORMAT(8A10)
  999 FORMAT(1H ,' WAVELENGTH PLOTS COMPLETED')
      END
```

```
      SUBROUTINE PLTWVL(DATA,NTRACE,ILOG,IAUTOY,LABTRC,IPLABL,LABYAX,
     1  ITITLE)
C
C     ON NHM6/PLTWVL
C
C     THIS ROUTINE PLOTS DATA AS A FUNCTION OF THE 13 NHM WAVELENGTHS.
C
C     THE INPUT IS
C
C     DATA(13,NTRACE)...THE ARRAY OF DATA VALUES TO BE PLOTTED.  EACH
C                       COLUMN HOLDS ONE FUNCTION OF WAVELENGTH, TO BE
C                       PLOTTED AS ONE TRACE ON THE GRAPH.
C
C     ILOG...= 0, IF THE ACTUAL DATA VALUES ARE TO BE PLOTTED
C            = 1, IF LOG(BASE 10) OF THE DATA IS TO BE PLOTTED
C
C     IAUTOY...= 0, IF THE Y (VERTICAL) SCALING IS PREDETERMINED IN
C                   THE MAIN PROGRAM (COMMON BLOCK CSCALE)
C              = 1, IF THE DATA VALUES SHOULD BE EXAMINED TO DETERMINE
C                   APPROPRIATE Y SCALING VALUES
C
C     LABTRC(I,NTRACE)...LABELS USED TO IDENTIFY THE TRACES, UP TO 80
C                        CHARACTERS EACH.  THE LABELS ARE PLOTTED IF
C                        IPLABL.NE.0
C
C     LABYAX(8)...A LABEL FOR THE VERTICAL (DATA) AXIS, UP TO 80 CHAR
C
C     ITITLE(8)...A TITLE FOR THE TOP OF THE PLOT, UP TO 80 CHAR
C
```

```
      PARAMETER(MXWAVE=13, MXPLT=MXWAVE+2, MXTRAC=20)
C
      DIMENSION DATA(MXWAVE,NTRACE), LABTRC(8,NTRACE),LABYAX(8)
      DIMENSION ITITLE(8),FMT(2)
      COMMON/CSCALE/ YINCH,YMAX,YMIN,IDIV,NCODE,YFMT(2)
      COMMON/CWORK/ XPLT(MXPLT),YPLT(MXPLT),TRACE(MXPLT,MXTRAC)
C
      DATA JSYMB/1/, LSYMB/1/, NWAVEL/13/
      DATA XINCH/6./,YINCH/6./,H/0.15/
      DATA FMT/10H(F6.2,2H -,10H)          /
C
C     SELECT ACTUAL OR LOG VALUES
C
      IF(ILOG.EQ.0) THEN
      DO 100 J=1,NTRACE
      DO 100 I=1,NWAVEL
  100 TRACE(I,J) = DATA(I,J)
C
      ELSEIF(ILOG.EQ.1) THEN
      DO 102 J=1,NTRACE
      DO 102 I=1,NWAVEL
  102 TRACE(I,J) = ALOG10(DATA(I,J))
C
      IF(IAUTOY.EQ.0) THEN
      YMAX = ALOG10(YMAX)
      YMIN = ALOG10(YMIN)
      ENDIF
      ENDIF
C
C     SET UP Y-AXIS SCALING
C
      IF(IAUTOY.EQ.0) THEN
C
C     USE PREDETERMINED SCALING VALUES
C
      DINCH = YINCH/FLOAT(IDIV)
      DLABL = (YMAX - YMIN)/FLOAT(IDIV)
      FLABL0 = YMAX
      YZERO = YMIN
      YPINCH = (YMAX - YMIN)/YINCH
      ELSE
C
C     EXAMINE THE VALUES TO BE PLOTTED TO DETERMINE THE VERTICAL SCALING
C
      YMIN = 1.0E200
      YMAX = -1.0E200
      DO 110 J=1,NTRACE
      DO 110 I=1,NWAVEL
      YMIN = AMIN1(YMIN,TRACE(I,J))
  110 YMAX = AMAX1(YMAX,TRACE(I,J))
C
      MINV = IFIX(YMIN)
      IF(YMIN.LT.0.) MINV = MINV - 1
      MAXV = IFIX(YMAX)
      IF(YMAX.GT.0) MAXV = MAXV + 1
      MRANGE = MINV - MAXV
      IDIV = IABS(MRANGE)
  302 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)
      FLABL0 = FLOAT(MAXV)
      YZERO = FLOAT(MINV)
      YPINCH = FLOAT(MAXV - MINV)/YINCH
      NCODE = 8
      YFMT(1) = FMT(1)
      YFMT(2) = FMT(2)
      ENDIF
C
```

```
C      DRAW BORDER AND LABEL Y AXIS
C
C
       CALL PLOT(0.,0.,3)
       CALL PLOT(0.,YINCH,2)
       CALL PLOT(XINCH,YINCH,2)
       CALL PLOT(XINCH,0.,2)
       CALL PLOT(0.,0.,2)
C
       XX = -(FLOAT(NCODE) - .4)*H
       DO 310 I=1,IDIV+1
       YY = YINCH - 0.45*H - FLOAT(I-1)*DINCH
       FLABL = FLABL0 - FLOAT(I-1)*DLABL
       ENCODE(NCODE,YFMT,BCD) FLABL
  310  CALL SYMBOL(XX,YY,H,BCD,0.0,NCODE)
       XX = -1.2
       NC = NCHAR(LABYAX,8)
       YY = 0.5*YINCH - 0.5*FLOAT(NC)*H
       CALL SYMBOL(XX,YY,H,LABYAX,90.0,NC)
C
C      DRAW HORIZONTAL AXIS
C
       Y1 = -0.45*H
       Y2 = -2.35*H
       DO 120 I=1,NWAVEL
       XX = XINCH*FLOAT(I-1)/FLOAT(NWAVEL-1)
       CALL SYMBOL(XX,Y1,H,13,0.,-1)
       IF(MOD(I,2).NE.0) THEN
       LAMBDA = 400 + 25*(I-1)
       ENCODE(3,122,BCD) LAMBDA
       CALL SYMBOL(XX-1.5*H,Y2,H,BCD,0.,3)
       ENDIF
  120  CONTINUE
       CALL SYMBOL(0.5*XINCH - 8.0*H,-4.0*H,H,16HWAVELENGTH IN NM,0.,16)
C
C      DRAW TITLE AT TOP
C
       NC = NCHAR(ITITLE,8)
       XX = 0.5*XINCH - 0.5*FLOAT(NC)*H
       YY = YINCH + 2.0*H
       CALL SYMBOL(XX,YY,H,ITITLE,0.,NC)
C
C      PLOT THE TRACES
C
C      SET UP THE X COORDINATES, WITH SCALING FACTORS
       DO 200 I=1,NWAVEL
  200  XPLT(I) = 400.0 + 25.0*FLOAT(I-1)
       XPLT(NWAVEL+1) = 400.0
       XPLT(NWAVEL+2) = 50.
C
C      SET UP Y COORDNATES AND PLOT
C
       YPLT(NWAVEL+1) = YZERO
       YPLT(NWAVEL+2) = YPINCH
       DO 210 J=1,NTRACE
       DO 212 I=1,NWAVEL
  212  YPLT(I) = TRACE(I,J)
C
       CALL LINE(XPLT,YPLT,NWAVEL,1,JSYMB,LSYMB)
C
C      NUMBER THE TRACE
       YY = (YPLT(NWAVEL) - YPLT(NWAVEL+1))/YPLT(NWAVEL+2) - 0.5*H
       ENCODE(2,214,BCD) J
       CALL SYMBOL(XINCH+1.5*H,YY,H,BCD,0.0,2)
C
C      PLOT FULL LABELS IF DESIRED
       IF(IPLABL.EQ.0) GO TO 210
       FPN = FLOAT(J)
       XX = XINCH + 1.5
       YY = YINCH - FPN*2.0*H
       CALL NUMBER(XX,YY,H,FPN,0.0,-1)
       NC = NCHAR(LABTRC(1,J),8)
       CALL SYMBOL(XX+2.5*H,YY,H,LABTRC(1,J),0.0,NC)
  210  CONTINUE
C
```

```
      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.

IMSL routine	where called, program/subroutine	description of IMSL routine
DCADRE	4/Y2ZGEO	numerically integrates a function of one variable
DVERK	4/RICATI	solves systems of ordinary differential equations using a high-order Runge-Kutta scheme
EIGRF	4/EIGENR	finds eigenvalues and eigenvectors of a real-valued matrix
GGNML	1/INISHL	generates pseudo-random numbers from a Gaussian distribution
GGUBFS	1/MAIN	generates pseudo-random numbers from a uniform distribution
LINV1F	4/AMPINT	inverts a matrix
LINV2F	4/EIGENR	inverts a matrix (high-accuracy version)
VMULFF	4/EIGENR	multiplies two matrices
VSRTA	4/EIGENR	sorts an array by algebraic value
VSRTR	1/TIP	sorts an array by algebraic value and returns the permutations

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

$$N(\mu,\phi) = N_0(1 + C\mu) \qquad \text{(B.1)}$$

where N_0 and C are constants to be chosen. Note that this sky radiance distribution is independent of azimuthal angle ϕ or wavelength λ. The form (B.1) yields the quad-averaged radiances

$$N(u,v) = N_0(1 + C\,\mu_u) \qquad \text{(B.2)}$$

where, as always, μ_u is the average μ-value of quad Q_{uv}.

The scalar irradiance h(sky) of the radiance distribution (2) is given by 75/8.4:

$$
\begin{aligned}
h(\text{sky}) &= \sum_{u=1}^{m-1} \sum_{v=1}^{2n} N(u,v)\ \Omega_{uv} + N(m,\cdot)\ \Omega_m \\
&= \sum_{u=1}^{m-1} \sum_{v=1}^{2n} N_0 (1 + C\mu_u) \Delta\mu_u \left(\frac{2\pi}{2n}\right) + N_0 (1 + C\mu_m)\ 2\pi\Delta\mu_m \\
&= 2\pi N_0 \sum_{u=1}^{m} (1 + C\mu_u) \Delta\mu_u \\
&= 2\pi N_0 \Bigg[\underbrace{\sum_{u=1}^{m} \Delta\mu_u}_{1} + C \underbrace{\sum_{u=1}^{m} \mu_u \Delta\mu_u}_{\frac{1}{2}} \Bigg] \\
&= 2\pi N_0 \left(1 + \frac{C}{2}\right)
\end{aligned}
\tag{B.3}
$$

For a uniform sky, C = 0, and we get $h(\text{sky}) = 2\pi N_0$, as expected.

The plane irradiance H(sky) is given by 75/8.7, which reduces to

$$H(\text{sky}) = 2\pi N_0 \left(\frac{1}{2} + \frac{C}{3}\right) \tag{B.4}$$

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(sun) + h(sky) ≡ h(total), and define the ratio of sky scalar irradiance to total scalar irradiance as

$$R \equiv \frac{h(\text{sky})}{h(\text{sun}) + h(\text{sky})} = \frac{h(\text{sky})}{h(\text{total})} \tag{B.5}$$

Using (B.3) and (B.5) and solving for N_0 gives

$$N_0 = \frac{R \cdot h(total)}{2\pi\left(1 + \frac{C}{2}\right)} \tag{B.6}$$

and h(sun) = h(total)·(1-R).

Subroutine QASKY computes the quad-averaged input radiances by

$$N(u,v) = \frac{R(1 + C\mu_u)}{2\pi\left(1 + \frac{C}{2}\right)} h(total) \tag{B.7}$$

for "sky only" quads, and

$$N(u,v) = \left[\frac{R(1 + C\mu_u)}{2\pi\left(1 + \frac{C}{2}\right)} + \frac{1\text{-}R}{\Omega_{uv}}\right] h(total) \tag{B.8}$$

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(total)/2π.

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

$$X = 680 \int_{400}^{700} P(\lambda)\,\bar{x}(\lambda)\,d\lambda \tag{C.1}$$

with corresponding equations for Y and Z. Here λ is wavelength in nanometers, P(λ) 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,$\cdots$,$\lambda_{13} = 700$ nm. Then X is computed by

$$X = 680 \frac{\Delta\lambda}{3}[P(400)\ \bar{x}(400) + 4P(425)\ \bar{x}(425) + 2P(450)\ \bar{x}(450) + \cdots 2P(650)\ \bar{x}(650) + 4P(675)\ \bar{x}(675) + P(700)\ \bar{x}(700)] \tag{C.2}$$

* See, for example, *Hydrologic Optics, Vol. I, 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

$$x \equiv \frac{X}{X+Y+Z} \quad , \quad y \equiv \frac{Y}{X+Y+Z} \quad , \quad z \equiv \frac{Z}{X+Y+Z}. \tag{C.3}$$

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 $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ for the 13 λ-values used in the NHM.

Tristimulus (color matching) functions $\bar{x}$, $\bar{y}$ and $\bar{z}$*

λ	$\bar{x}(\lambda)$	$\bar{y}(\lambda)$	$\bar{z}(\lambda)$	weight for Simpson's rule integrations
400	.0143	.0004	.0679	1
425	.2148	.0073	1.0391	4
450	.3362	.0380	1.7721	2
475	.1421	.1126	1.0419	4
500	.0049	.3230	.2720	2
525	.1096	.7932	.0573	4
550	.4334	.9950	.0087	2
575	.8425	.9154	.0018	4
600	1.0622	.6310	.0008	2
625	.7514	.3210	.0001	4
650	.2835	.1070	0	2
675	.0636	.0232	0	4
700	.0114	.0041	0	1
sums	4.2699	4.2712	4.2617	
Integrals for $P(\lambda) = 1$	72319.	73005.	72170.	

Note that the integrals $680 \int_{400}^{700} \bar{x}(\lambda)\, d\lambda = 72319$, 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,···,37 to make a closed curve. The

* Taken from *Color Science* (2nd edition) by G. Wyszecki and W. Stiles, John Wyley & Sons, New York, 1982, Table II (3.31), pages 736-7.

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), λ, 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 $[x_p(I), y_p(I)]$, $I = 1,\cdots,37$.

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 λ will be somewhere between λ_I and λ_{I+1}, where λ_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 lows 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

$$\begin{cases} \dfrac{x_i - x_w}{y_i - y_w} = \dfrac{x - x_w}{y - y_w} \equiv s_1 & \text{the line determined by (x,y) and } (x_w, y_w) \\ \dfrac{x_i - x_p(I-1)}{y_i - y_p(I-1)} = \dfrac{x_p(I) - x_p(I-1)}{y_p(I) - y_p(I-1)} \equiv s_2 & \text{the line determined by } [x_p(I), y_p(I)] \text{ and } [x_p(I+1), y_p(I+1)] \end{cases}$$

which gives

$$x_i = \frac{s_2\, x_w - s_1\, x_p(I-1) - s_1\, s_2 [y_w - y_p(I-1)]}{s_2 - s_1}$$

$$y_i = \frac{x_w - x_p(I-1) - s_1\, y_w + s_2\, y_p(I-1)}{s_2 - s_1} \qquad \text{(C.4)}$$

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

$$\lambda = \left(1 - \frac{d_1}{d_2}\right)\lambda_I + \left(\frac{d_1}{d_2}\right)\lambda_{I+1} \qquad \text{(C.5)}$$

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

$$p = \frac{d_3}{d_4} \qquad \text{(C.6)}$$

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, θ_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 $\theta_s = 0°$ (the sun is at the zenith); it is 2 = sec 60° when $\theta_s = 60°$, and so on. The scalar irradiance at sea level, $h_{SL}(\lambda, \theta_s)$, is given by

$$h_{SL}(\lambda, \theta_s) = h_s(\lambda)\, e^{-\alpha_\lambda \sec\theta_s} \qquad (D.1)$$

where $h_s(\lambda)$ is the solar scalar irradiance at wavelength λ, outside the atmosphere, and α_λ includes all scattering and absorption effects of the model atmosphere. The table below gives the values of $h_s(\lambda)$† and α_λ‡.

---

* Taken from the *Handbook of Geophysics and Space Environments*, ed. by S.L. Valley, Air Force Cambridge Research Lab, 1965, page 16-19.

† These values are taken from *Hydrologic Optics, Vol. I*, 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 α_λ are rescaled to be consistent with 1396 W m^{-2}.

λ nm	$h_s(\lambda)$ W m^{-2} nm^{-1}	α_λ
400	1.54	.566
425	1.89	.428
450	2.20	.364
475	2.20	.293
500	1.98	.217
525	1.92	.210
550	1.95	.220
575	1.87	.206
600	1.81	.192
625	1.72	.165
650	1.62	.134
675	1.53	.114
700	1.44	.104

☆ U.S. Government Printing Office: 1988- 573-002/92,001