



-TR-0107

AD

Reports Control Symbol
OSD 1366



SUPPLEMENT TO EOSAEL 80
VOLUME II
USER'S MANUAL

PROGRAM LISTINGS FOR EOSAEL 80-B
AND ANCILLARY CODES AGAUS AND FLASH

FEBRUARY 1982

By

R. G. Steinhoff

DTIC FILE COPY

DTIC
ELECTE
MAY 14 1982
S D D

Approved for public release; distribution unlimited.



US Army Electronics Research and Development Command
Atmospheric Sciences Laboratory
White Sands Missile Range, NM 88002

82 05 11 016

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM												
1. REPORT NUMBER ASL-TR-0107	2. GOVT ACCESSION NO. AD-A114417	3. RECIPIENT'S CATALOG NUMBER 7												
4. TITLE (and Subtitle) SUPPLEMENT TO EOSAEL 80 Vol II USER'S MANUAL PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND FLASH	5. TYPE OF REPORT & PERIOD COVERED Final Report													
	6. PERFORMING ORG. REPORT NUMBER													
7. AUTHOR(s) R. G. Steinhoff	8. CONTRACT OR GRANT NUMBER(s)													
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Atmospheric Sciences Laboratory White Sands Missile Range, NM	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Task AH71612111 H710011													
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Electronics Research and Development Command Adelphi, MD 20783	12. REPORT DATE February 1982													
	13. NUMBER OF PAGES 510													
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED													
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE													
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		<table border="1"> <tr><td colspan="2">Accession For</td></tr> <tr><td>NTIS GRA&I</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>DTIC TAB</td><td><input type="checkbox"/></td></tr> <tr><td>Unannounced</td><td><input type="checkbox"/></td></tr> <tr><td>Justification</td><td></td></tr> </table>	Accession For		NTIS GRA&I	<input checked="" type="checkbox"/>	DTIC TAB	<input type="checkbox"/>	Unannounced	<input type="checkbox"/>	Justification			
Accession For														
NTIS GRA&I	<input checked="" type="checkbox"/>													
DTIC TAB	<input type="checkbox"/>													
Unannounced	<input type="checkbox"/>													
Justification														
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		<table border="1"> <tr><td colspan="2">By</td></tr> <tr><td colspan="2">Distribution/</td></tr> <tr><td colspan="2">Availability</td></tr> <tr><td colspan="2">Availability</td></tr> <tr><td>Ditt</td><td>Spec</td></tr> <tr><td>A</td><td></td></tr> </table>	By		Distribution/		Availability		Availability		Ditt	Spec	A	
By														
Distribution/														
Availability														
Availability														
Ditt	Spec													
A														
18. SUPPLEMENTARY NOTES		<div style="border: 1px solid black; border-radius: 50%; width: 40px; height: 40px; display: flex; flex-direction: column; align-items: center; justify-content: center; margin: 0 auto;"> <div style="font-size: 8px; margin-bottom: 2px;">DTIC</div> <div style="font-size: 8px; margin-bottom: 2px;">COPY</div> <div style="font-size: 8px; margin-bottom: 2px;">INSPECTED</div> <div style="font-size: 8px;">2</div> </div>												
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) EOSAEL 80 Computer Codes Transmission Transport and diffusion														
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Computer listing of the Electro-Optical Systems Atmospheric Effects Library, version 80-B.														

CONTENTS

INTRODUCTION	7
PROGRAM EOSAEL	9
SUBROUTINE RESET	18
SUBROUTINE COMPLT	19
SUBROUTINE ILLUM	20
FUNCTION SOLARS	21
FUNCTION SMOON	22
FUNCTION JPASCT	23
SUBROUTINE PFUNC	24
SUBROUTINE XSCALE	29
SUBROUTINE SLANT	33
SUBROUTINE TURB	35
FUNCTION DESUB	43
FUNCTION FALPH	44
SUBROUTINE FFT4	45
FUNCTION GAUSS	47
SUBROUTINE MEANVR	48
SUBROUTINE SPECT	49
SUBROUTINE SPREAD	50
SUBROUTINE THETO	51
SUBROUTINE BASCAT	52
SUBROUTINE PFUNC	54
SUBROUTINE BKWD	66
SUBROUTINE CONV	68
SUBROUTINE ELM	69
SUBROUTINE FIND	70
SUBROUTINE FWRD	71
SUBROUTINE GAS	72
SUBROUTINE GMAX	73
SUBROUTINE MATRX	74
SUBROUTINE ROTAT	75
SUBROUTINE SMOOZ	76
SUBROUTINE START	77
SUBROUTINE TRAVRS	79
SUBROUTINE USCA	81
SUBROUTINE SMOKE	82
FUNCTION JPASCT	93
SUBROUTINE CLSMOK	96
SUBROUTINE SCONST	96
SUBROUTINE SMASSP	98
SUBROUTINE STRANS	99
SUBROUTINE WGGEOM	100
FUNCTION QROOT	102
SUBROUTINE XYZINT	103
SUBROUTINE BRATE	108
SUBROUTINE GPUFF	109
SUBROUTINE LZTRAN	111
SUBROUTINE LZIDNM	115
SUBROUTINE DRTRAN	117
SUBROUTINE AMOUNT	130
SUBROUTINE ATMCAL	131
SUBROUTINE AVRG	135
SUBROUTINE CLIMB	136
SUBROUTINE CLDIM	138
SUBROUTINE COMPL	140
SUBROUTINE CONLEN	143
SUBROUTINE CONVRT	145
FUNCTION CSPHER	147
FUNCTION CWAKE	149
FUNCTION CWIN	151

SUBROUTINE	DIFEQ	1554
FUNCTION	DIFFUS	1556
FUNCTION	DOTPRD	1557
FUNCTION	DTERPI	1558
SUBROUTINE	DTERPS	1600
SUBROUTINE	DUSTCL	1611
FUNCTION	ERF	1644
SUBROUTINE	FIT	1659
FUNCTION	FUNCT	1666
SUBROUTINE	GAMMA	1674
FUNCTION	GFUN	1688
SUBROUTINE	GRAD2	1693
SUBROUTINE	GRAND	1700
SUBROUTINE	GREEN	1722
SUBROUTINE	GREEN1	1737
SUBROUTINE	MOMENT	1750
SUBROUTINE	PATH	1770
SUBROUTINE	PERP	1779
SUBROUTINE	PRECL	1808
SUBROUTINE	PRETRN	1808
SUBROUTINE	PREVEH	1808
SUBROUTINE	RISE	1808
SUBROUTINE	RKM	1900
SUBROUTINE	SETUP	1920
SUBROUTINE	SOURCE	1944
SUBROUTINE	TEMP	1977
FUNCTION	TMPCAL	1988
SUBROUTINE	TRAP	2000
SUBROUTINE	TRNCAL	2003
SUBROUTINE	TRNCHK	2007
SUBROUTINE	TRNCLD	2008
SUBROUTINE	UNIT	2110
SUBROUTINE	VEHCL	2111
SUBROUTINE	VEHTRN	2111
SUBROUTINE	VSRC	2155
SUBROUTINE	VSUM	2155
SUBROUTINE	WIN	2159
FUNCTION	WINDCAL	2159
SUBROUTINE	MMW	2201
FUNCTION	AB	2202
FUNCTION	DOP	2204
SUBROUTINE	INTRP	2206
SUBROUTINE	MMH20	2207
SUBROUTINE	MMIDX	2208
SUBROUTINE	MMOXY	2209
SUBROUTINE	MMRAN	2200
SUBROUTINE	MMSNO	2222
SUBROUTINE	MMWFG	2234
SUBROUTINE	MMWGS	2235
FUNCTION	PFR	2237
FUNCTION	PSAT	2238
FUNCTION	SUPK	2239
SUBROUTINE	CLTRAN	240
SUBROUTINE	LAYRXY	244
SUBROUTINE	CYLXY	245
SUBROUTINE	CLEXTN	247
SUBROUTINE	LISOUT	248
SUBROUTINE	DEFSET	2500
SUBROUTINE	CLREAD	2511
SUBROUTINE	SCREEN	253
SUBROUTINE	XSCALE	259
SUBROUTINE	SLANT	260
FUNCTION	JPASCT	263
SUBROUTINE	PFUNC	264
SUBROUTINE	CMIC	264
SUBROUTINE	CMIC1	264
SUBROUTINE	CMIC3	264

SUBROUTINE	CWIC4	265
SUBROUTINE	ITAM	267
SUBROUTINE	CINV	274
SUBROUTINE	INTAL	279
SUBROUTINE	CASE1	282
SUBROUTINE	CASE2	283
SUBROUTINE	CASE3	284
SUBROUTINE	CYCLE	285
SUBROUTINE	TREQ	286
SUBROUTINE	F_CLOUD	287
SUBROUTINE	ILLUM	290
FUNCTION	SOLARS	291
FUNCTION	SMOON	292
SUBROUTINE	THRMCL	293
SUBROUTINE	SSCLD	294
SUBROUTINE	PFNN	295
SUBROUTINE	MSCLD	297
FUNCTION	ETAINT	298
SUBROUTINE	OVRCS	299
SUBROUTINE	ILLUM	300
FUNCTION	SOLARS	301
FUNCTION	SMOON	302
FUNCTION	G2	303
FUNCTION	E1	304
SUBROUTINE	GRNADE	305
SUBROUTINE	CONCH	309
SUBROUTINE	GOGET	311
SUBROUTINE	PARMS	312
SUBROUTINE	EXTIN	313
SUBROUTINE	UMEAN	314
SUBROUTINE	SUMA	315
SUBROUTINE	LGCAT	316
SUBROUTINE	DATRD	317
SUBROUTINE	LT4M	321
SUBROUTINE	XSCALE	329
SUBROUTINE	SLANT	333
SUBROUTINE	ABSORB	336
SUBROUTINE	CKER	337
SUBROUTINE	FREQSL	337
SUBROUTINE	H2OVAP	339
SUBROUTINE	H2O410	340
SUBROUTINE	LTPATH	341
SUBROUTINE	MOLSC	343
SUBROUTINE	NH3	344
SUBROUTINE	NITRIC	345
SUBROUTINE	NITRO	346
SUBROUTINE	NO2	347
SUBROUTINE	OZONE	348
SUBROUTINE	POINT	349
FUNCTION	RESFN	350
SUBROUTINE	SO2	351
SUBROUTINE	UNIMIX	352
SUBROUTINE	UVOZNE	353
SUBROUTINE	SPOT	354
SUBROUTINE	LT4M	321
SUBROUTINE	XSCALE	329
SUBROUTINE	SLANT	333
SUBROUTINE	PFUNC	334
FUNCTION	SOLARS	321
FUNCTION	SMOON	322
FUNCTION	ALBEDO	361
FUNCTION	BLACK	362
SUBROUTINE	COEFS	363
SUBROUTINE	DIAG	364
SUBROUTINE	INDAT	366

SUBROUTINE	OUTPUT	369
SUBROUTINE	PATHRD	372
SUBROUTINE	ZERO	374
SUBROUTINE CLIMAT		375
PROGRAM	AGAUS	379
SUBROUTINE	AGXP1	385
SUBROUTINE	AGXP2	390
SUBROUTINE	AGXP3	396
SUBROUTINE	ANGLE	398
FUNCTION	GAMMA	399
SUBROUTINE	GUSET	400
SUBROUTINE	MIEGX	401
SUBROUTINE	WATER	405
SUBROUTINE	GAUS	408
SUBROUTINE	DIMER	409
BLOCK DATA		410
PROGRAM	FLASH1	411
SUBROUTINE	FLASH	412
SUBROUTINE	DATRD	414
SUBROUTINE	DATWT	416
SUBROUTINE	GETIM	418
SUBROUTINE	TRBLC	421
SUBROUTINE	YSBLC	422
EOSAEL TEST DATA (NEWRUN)		423
EOSAEL OUTPUT (EOOUT)		426

INTRODUCTION

This listing of EOSAEL 80-B is a supplement to Volume II¹ and supersedes all previous listings.² The current listing is complete as of 8 February 1982 and has revisions one through five incorporated into it.

EOSAEL 80-B differs from EOSAEL 80 in that modules SPOT, LT4M, NMMW, CLIMAT, BASCAT, and TURB have been extensively revised and, therefore, appear with new sequence numbers. All other modules have their original sequencing, except where revisions have been inserted or deleted.

The programs are listed by module with each module followed by its subroutines. Subroutines that have been listed for prior modules in the listing are not repeated in the source listing. The table of contents lists each module along with all its corresponding subroutines and the page number of each subroutine in the listing. The elements EOMAIN, COMPLT, and RESET, which are always to be resident, appear only at the beginning of the table of contents and the source listing.

Also included herein is a sample input file, NEWRUN, and an output file, E00UT, produced by using the aforementioned sample input file.

The supplemental codes AGAUS and FLASH are supplied with EOSAEL 80. FLASH is described in appendix A of volume II of the User's Manual¹ and is further described in the comments of the source listing. Operating instructions for the AGAUS code may be found in the comments of the source listing. Manuals for AGAUS are available upon written request from the US Army Atmospheric Sciences Laboratory, White Sands Missile Range, New Mexico.

¹Shirkey, R. C., and S. G. O'Brien, EOSAEL 80, Volume II, User's Manual, ASL-TR-0073, US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

²Steinhoff, R. G., Program Listings for EOSAEL 80 and Ancillary Codes AGAUS and FLASH, ASL-TR-0073 (Supplement), US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

```

PROGRAM EOSAEL
C MAIN PROGRAM FOR EOSAEL 80
C
REAL LOTRNS,LAZTRN,LZTRN,MMTRAN,MMWTRN,IPNAM,IAL,IALB1,IALB2
LOGICAL ISPOT,N16,LOREAD
COMMON /SPOTLO/ISPOT,LOREAD,N16
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
1 WNDVEL,WNDDIR,IPASCT
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /GEOMET/PTS(15),IGEOSW
C
C IOIN - CARD READER
C IOOUT - PRINTER
C IPHFUN - UNIT UPON WHICH PHASE FUNCTION DATA RESIDES
C LOUNIT - UNIT UPON WHICH LT4M ATM DATA RESIDES
C NDIRTU - UNIT UPON WHICH DRTRAN DATA RESIDES
C NCLIMT - UNIT FOR CLIMATOLOGICAL DATA
C KSTOR - AUXILLARY START/RESTART UNIT FOR BASCAT
C NPLOTU - OPTIONAL UNIT FOR WRITING RESULTS FOR SUBSEQUENT
C PLOTTING PURPOSES BY THE USER.
C
DIMENSION TRAN(16),RADAC(16),RADG(16),IPROGN(20)
DIMENSION IDOPGM(20),IPNAM(40)
DIMENSION IAL(12),DAT(10)
FOR UNIVAC
DATA IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR/
1 5,6,3,8,7,24,25/
C PROGRAM NAMES
DATA IPNAM,4HSPOT,4HTURB,4HBASC,4HLT4M,4HXSCA,4HSMOK,4HDRTR,
1 4HLZTR,4HMMW,4HCLTR,4HSCRE,4HFCLD,4HOVRC,4HGRNA,4H* ,
2 4H* ,4H* ,4H* ,4HCLIM,4H ,4H ,4HAT ,
3 4H ,4HLE ,4HE ,4HAN ,4HAN ,4H ,4HAN ,4HEN ,
4 4HUD ,4HST ,4HDE ,4H ,4H ,4H ,4H ,4H ,
5 4HATE /
C CARD MNEMONICS
DATA IAL,4HEGRU,4HVIS ,4HFREQ,4HWAVL,4HWVNU,4HRESF,
14HTARG,4HRCVR,4HDESG,4HOBSV,4HBFLC,4HGO /
DATA PI,TORRMB,CDEGK/3.14159265,1.33322,273.16/
DATA PTS,15*0.0/
ISTART=0
CLDAMT=0.
CLDHYT=0.
FOGPRB=0.
PI2=PI/2.
PIRAD=PI/180.
TWOPI=2.0*PI
WRITE (IOOUT,1060)
C***** I/O *****
C
C** INPUT TO EOSAEL IS CARD ORDER-INDEPENDENT, WITH EACH INPUT RECORD
C** HAVING A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4. THE ONLY EXCEPTION
C** TO THIS RULE IS THE GO SENTINEL CARD, WHICH MUST BE THE LAST RECORD
C** IN THE INPUT SEQUENCE. ALL RECORDS ARE READ IN UNDER THE
C** FORMAT (2A4,1X,10E7.4). INTEGERS MUST BE INPUT AS REAL
C** NUMBERS IN THIS COMMON FORMAT SCHEME. THEY ARE LATER FIXED TO
C** THE INTEGER TYPE. THE IDENTIFIERS FOR EACH OF THE INPUT
C** RECORDS ARE AS FOLLOWS :
C
C-----
C CARD IDENTIFIER : EORUN
C VARIABLES READ : NUMRUN
C NUMRUN - NUMBER OF TIMES EOSAEL DRIVER IS TO BE CYCLED
C DEFAULT IS 1.
C-----
C CARD IDENTIFIER : VIS
C VARIABLES READ : VIS,EXTN55,EXTN
C VIS - VISIBILITY AT WAVELENGTH OF 0.55 MICRONS (KM)
C EXTN55 - EXTINCTION COEFFICIENT AT 0.55 MICRONS (KM**-1)
C EXTN - EXTINCTION COEFFICIENT AT INPUT WAVELENGTH (KM**-1)
C** NOTE : IF THE VIS CARD IS NOT INPUT, A WARNING IS PRINTED
C AND THE VISIBILITY IS SET TO A DEFAULT VALUE OF 10 KM.

```

```

EOM00010
EOM00020
EOM00030
EOM00040
EOM00050
EOM00060
EOM00070
EOM00080
EOM00090
EOM00100
EOM00110
EOM00120
EOM00130
EOM00140
EOM00150
EOM00160
EOM00170
EOM00180
EOM00190
EOM00200
EOM00210
EOM00220
EOM00230
EOM00240
EOM00250
EOM00260
EOM00270
EOM00280
EOM00290
EOM00300
EOM00310
EOM00320
EOM00330
EOM00340
EOM00350
EOM00360
EOM00370
EOM00380
EOM00390
EOM00400
EOM00410
EOM00420
EOM00430
EOM00440
EOM00450
EOM00460
EOM00470
EOM00480
EOM00490
EOM00500
EOM00510
EOM00520
EOM00530
EOM00540
EOM00550
EOM00560
EOM00570
EOM00580
EOM00590
EOM00600
EOM00610
EOM00620
EOM00630
EOM00640
EOM00650
EOM00660
EOM00670

```



```

C** CARD IDENTIFIER : DESG
C** VARIABLES READ : PTS<7>,PTS<8>,PTS<9>
C** PTS<7-9> - COORDINATES OF THE DESIGNATOR OR SOURCE
EOM01380
EOM01390
EOM01400
EOM01410
EOM01420
EOM01430
EOM01440
EOM01450
EOM01460
EOM01470
EOM01480
EOM01490
EOM01500
EOM01510
EOM01520
EOM01530
EOM01540
EOM01550
EOM01560
EOM01570
EOM01580
EOM01590
EOM01600
EOM01610
EOM01620
EOM01630

C** CARD IDENTIFIER : OBSV
C** VARIABLES READ : PTS<10>,PTS<11>,PTS<12>
C** PTS<10-12> - COORDINATES OF THE OBSERVER USED BY DRTRAN
EOM01410
EOM01420
EOM01430
EOM01440
EOM01450
EOM01460
EOM01470
EOM01480
EOM01490
EOM01500
EOM01510
EOM01520
EOM01530
EOM01540
EOM01550
EOM01560
EOM01570
EOM01580
EOM01590
EOM01600
EOM01610
EOM01620
EOM01630

C** CARD IDENTIFIER : BFCL
C** VARIABLES READ : PTS<13>,PTS<14>,PTS<15>
C** PTS<13-15> - COORDINATES OF THE CENTER OF THE CLOUD ELLIPSOID
C** USED BY BASCAT AND FLOUD
EOM01430
EOM01440
EOM01450
EOM01460
EOM01470
EOM01480
EOM01490
EOM01500
EOM01510
EOM01520
EOM01530
EOM01540
EOM01550
EOM01560
EOM01570
EOM01580
EOM01590
EOM01600
EOM01610
EOM01620
EOM01630

-----
C** THE NEXT INPUT CARD IS THE CLIMATOLOGICAL OPTION CARD. THIS
C** OPTION ALLOWS USER INPUT OF METEOROLOGICAL PARAMETERS DIRECTLY
C** OR AUTOMATIC INPUT OF CLIMATOLOGY DATA CHARACTERISTIC OF
C** WEST GERMANY. IF THIS OPTION IS INVOKED ALL MODULES WILL USE
C** THIS DATA, I.E. MET DATA THAT HAS BEEN INPUT TO A SPECIFIC
C** MODULE WILL BE OVERRIDDEN.
EOM01510
EOM01520
EOM01530
EOM01540
EOM01550
EOM01560
EOM01570
EOM01580
EOM01590
EOM01600
EOM01610
EOM01620
EOM01630

CARD IDENTIFIER : CLIMAT
VARIABLES READ : ICLMAT,LOCAT,MONTH,NHOUR,IWIND,NPRT
** OR ** ICLMAT,IPASCT,TEMP,PRESS,RH,AH,DP,VIS,WNDVEL,
WINDIR
LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER
(1-4) FOR CENTRAL EUROPE AND
(5-10) FOR MID-EAST.

L = 1 - EUROPEAN LOWLANDS,
L = 2 - EUROPEAN RHINE VALLEY,
L = 3 - EUROPEAN HIGHLANDS,
L = 4 - EUROPEAN ALPINE,
L = 5 - MIDEAST DESERTS,
L = 6 - MIDEAST COASTAL,
L = 7 - MIDEAST PERSIAN GULF,
L = 8 - MIDEAST RED SEA,
L = 9 - MIDEAST EASTERN MOUNTAINS, AND
L = 10 - MIDEAST INDUS VALLEY.

MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR.
MONTH IS USED TO SELECT THE SEASON WHICH IS
APPLICABLE TO THE REGION LOCAT.
NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL
STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OF
FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14,
AND 15-19.
IWIND - *** NOT USED ***
NPRT - A PRINT SELECTOR.
NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA.
NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD
DEVIATIONS, AND PERCENT OCCURRENCES.
ICLMAT = 2.: USER INPUT QUANTITIES
IPASCT = PASQUILL STABILITY CATEGORY VALID RANGE = 1.-6.(A-F)
TEMP = TEMPERATURE IN DEGREES C
PRESS = PRESSURE IN MB (SEA LEVEL IF ICLMAT=1)
RH = RELATIVE HUMIDITY IN %
AH = ABSOLUTE HUMIDITY - DEFINED HERE AS THE H2O VAPOR
DENSITY IN G/M**3.
DP = DEW POINT TEMPERATURE IN DEGREES C
VIS = VISIBILITY IN KM
WINDVEL = WIND VELOCITY IN M/S (DEPT UPON IWIND IF ICLMAT=1)
WINDIR = WIND DIRECTION IN DEGREES (IF ICLMAT=1 WILL BE
MOST PROBABLE DIRECTION)
EOM01620
EOM01630
EOM01640
EOM01650
EOM01660
EOM01670
EOM01680
EOM01690
EOM01700
EOM01710
EOM01720
EOM01730
EOM01740
EOM01750
EOM01760
EOM01770
EOM01780
EOM01790
EOM01800
EOM01810
EOM01820
EOM01830
EOM01840
EOM01850
EOM01860
EOM01870
EOM01880
EOM01890
EOM01900
EOM01910
EOM01920
EOM01930
EOM01940
EOM01950
EOM01960
EOM01970
EOM01980

-----
C** THE FOLLOWING CARDS ARE ALSO READ IN UNDER THE COMMON FORMAT
C** USED ABOVE. THE INFORMATION ON EACH OF THESE RECORDS DETERMINES
C** WHICH MODULES ARE SELECTED AND HOW MANY TIMES THE MODULES ARE
EOM01920
EOM01930
EOM01940
EOM01950
EOM01960
EOM01970
EOM01980

```

C** TO BE CYCLED WITHIN EACH CYCLE OF THE EOSAEL DRIVER.

CARD IDENTIFIER : (SEE MODULE IDENTIFIERS BELOW)
VARIABLES READ : IDOPGM
IDOPGM - NUMBER OF TIMES THE SELECTED MODULE IS TO
BE CYCLED WITHIN EACH EOSAEL CYCLE - DEFAULT IS ONE.

NO.	MODULE IDENTIFIER	WAVE(UM)	RANGES	FREQ(GHZ)
1	SPOT	.25-2.,3.-5.,8.-12.	*	
2	TURB	LT 14.	**	
3	BASCAT	ANY WAVELENGTH IN DATA	FILE	IPHFUN
4	LT4M	.25-2.,3.-5.,8.-12.	*	
5	XSCALE	1,06,3-5,8-12	*	
6	SMOKE	.4-1.2,3-5,8-12	**	94.
7	DRTRAN	.4-1.1,3.5-4.,8.5-12	*	94.-140.
8	LZTRAN	.8-11.	*	
9	NMMW		*	10-350
10	CLTRAN	.2-2.,3.-5.,8.-12.	*	
11	SCREEN	N/A		
12	FCLLOUD	ANY WAVELENGTH IN DATA	FILE	IPHFUN
13	OVRCSST	ANY WAVELENGTH		
14	GRNADE	SAME AS SMOKE		

C** NOTE : THE DATA SPECIFIC TO EACH MODULE MUST BE INPUT IN THE SEQUENCE IN THE ABOVE LIST.

CARD IDENTIFIER : GO
VARIABLES READ : NONE
END OF READ SENTINEL (MUST BE LAST CARD READ).

C*****

```

NUMRUN=1
IRFLAG=0
READ (IOIN,1000) IALB1,IALB2,(DAT(L),L=1,10)
IF (IALB1.NE.IAL(1)) GO TO 10
NUMRUN=IFIX(DAT(1))
IF (NUMRUN.EQ.0) NUMRUN=1
GO TO 20
C SET FLAG IF EDRUN IS NOT THE FIRST CARD
10 IRFLAG=1
20 CONTINUE
C DO 580 JRUN=1,NUMRUN
INITIALIZATION
C DO 30 I=1,20
PROGRAM SELECTOR
C IPROGN(I)=0
PROGRAM CYCLE DEFAULT
C 30 IDOPGM(I)=1
TRANSMISSIONS
C LOTRNS=1.
XSTRN=1.
SMKTRN=1.
DRTRN=1.
LZTRN=1.
MMWTRN=1.
CLTRN=1.0
GRNTRN=1.0
IF (JRUN.GT.1) WRITE (IOOUT,1070) JRUN
C FREQUENCY, WAVELENGTH, WAVENUMBER INDICATOR
IFW=0
C GEOMETRICAL OPTION SWITCH
ICEOSW=0
C SENSOR RESPONSE FUNCTION OPTION SWITCH
NR=0
VIS=0.0
EXTN55=0.0
EXTN=0.0
DO 220 J=1,25

```

EOM01990
EOM02000
EOM02010
EOM02020
EOM02030
EOM02040
EOM02050
EOM02060
EOM02070
EOM02080
EOM02090
EOM02100
EOM02110
EOM02120
EOM02130
EOM02140
EOM02150
EOM02160
EOM02170
EOM02180
EOM02190
EOM02200
EOM02210
EOM02220
EOM02230
EOM02240
EOM02250
EOM02260
EOM02270
EOM02280
EOM02290
EOM02300
EOM02310
EOM02320
EOM02330
EOM02340
EOM02350
EOM02360
EOM02370
EOM02380
EOM02390
EOM02400
EOM02410
EOM02420
EOM02430
EOM02440
EOM02450
EOM02460
EOM02470
EOM02480
EOM02490
EOM02500
EOM02510
EOM02520
EOM02530
EOM02540
EOM02550
EOM02560
EOM02570
EOM02580
EOM02590
EOM02600
EOM02610
EOM02620
EOM02630
EOM02640
EOM02650
EOM02660
EOM02670
EOM02680

	IF (IRFLAG.EQ.1) WRITE (IOOUT,1010)	EOM02690
C	SUPPRESS READ IN CASE FIRST CARD PREVIOUSLY READ WASNT EORUN.	EOM02700
	IF (IRFLAG.EQ.0) READ (IOIN,1000) IALB1,IALB2,(DAT(L),L=1,10)	EOM02710
	IRFLAG=0	EOM02720
	INOPT=0	EOM02730
	IF (J.EQ.25) GO TO 230	EOM02740
	DO 40 KK=1,12	EOM02750
C	CHECK FOR CARD TYPES, NOT PROGRAM SELECTOR	EOM02760
	IF (IALB1.NE.IAL(KK)) GO TO 40	EOM02770
	INOPT=KK	EOM02780
	IF (INOPT.GE.3.AND.INOPT.LE.5) IFW=INOPT	EOM02790
C	GO CARD FOUND	EOM02800
	IF (INOPT.EQ.12) GO TO 250	EOM02810
	GO TO 80	EOM02820
	CONTINUE	EOM02830
C	40 SEARCH FOR PROGRAMS HERE	EOM02840
	DO 50 KK=1,20	EOM02850
	IF (IALB1.NE.IPNAM(KK)) GO TO 50	EOM02860
	IPROGM(KK)=KK	EOM02870
	IF (DAT(1).GT.1.0) IDOPGM(KK)=IFIX(DAT(1))	EOM02880
	IF (KK.EQ.20) GO TO 60	EOM02890
	GO TO 220	EOM02900
	CONTINUE	EOM02910
	GO TO 240	EOM02920
C	CLIMATOLOGICAL OPTION INVOKED	EOM02930
60	ICLMAT=IFIX(DAT(1))	EOM02940
	IF (ICLMAT.EQ.2) GO TO 70	EOM02950
	LOCAT=IFIX(DAT(2))	EOM02960
	MONTH=IFIX(DAT(3))	EOM02970
	NHOUR=IFIX(DAT(4))	EOM02980
	IWIND=IFIX(DAT(5))	EOM02990
	NPRT=IFIX(DAT(6))	EOM03000
	GO TO 220	EOM03010
70	IPASCT=IFIX(DAT(2))	EOM03020
	TEMP=DAT(3)	EOM03030
	PRESS=DAT(4)	EOM03040
	RH=DAT(5)	EOM03050
	AH=DAT(6)	EOM03060
	DP=DAT(7)	EOM03070
	VIS=DAT(8)	EOM03080
	WNDVEL=DAT(9)	EOM03090
	WINDDIR=DAT(10)	EOM03100
	GO TO 220	EOM03110
C	GEOMETRICAL OPTION INVOKED	EOM03120
80	IF (INOPT.GT.6) GO TO 90	EOM03130
C	CARD SETUP SWITCHING	EOM03140
	GO TO (210,160,170,180,190,200),INOPT	EOM03150
90	LPTSSW=INOPT-6	EOM03160
	IGEOSW=1	EOM03170
	DO 150 K=1,3	EOM03180
	GO TO (100,110,120,130,140),LPTSSW	EOM03190
100	PTS(K)=DAT(K)	EOM03200
	GO TO 150	EOM03210
110	PTS(K+3)=DAT(K)	EOM03220
	GO TO 150	EOM03230
120	PTS(K+6)=DAT(K)	EOM03240
	GO TO 150	EOM03250
130	PTS(K+9)=DAT(K)	EOM03260
	GO TO 150	EOM03270
140	PTS(K+12)=DAT(K)	EOM03280
150	CONTINUE	EOM03290
	GO TO 220	EOM03300
C	VISIBILITY CARD	EOM03310
160	VIS=DAT(1)	EOM03320
	EXTN55=DAT(2)	EOM03330
	EXTN=DAT(3)	EOM03340
	GO TO 220	EOM03350
C	FREQUENCY CARD	EOM03360
170	FREQ1=DAT(1)	EOM03370
	FREQ2=DAT(2)	EOM03380

```

      MULDV=IFIX(DAT(3))
      GO TO 220
C 180 WAVELENGTH CARD
      WAVE1=DAT(1)
      WAVE2=DAT(2)
      MULDV=IFIX(DAT(3))
      GO TO 220
C 190 WAVENUMBER CARD
      WVNUM1=DAT(1)
      WVNUM2=DAT(2)
      MULDV=IFIX(DAT(3))
      GO TO 220
C 200 SENSOR OPTION INVOKED
      NR=1
      GO TO 220
      WRITE (IOOUT,1010)
      CONTINUE
C 210 IF (IFW.NE.0) GO TO 250
      ERROR CHECK ON WAVENUMBER, WAVELENGTH, OR FREQUENCY
      WRITE (IOOUT,1020)
      GO TO 580
      WRITE (IOOUT,1030)
      GO TO 250
C 220 UNKNOWN CARD TYPE
      WRITE (IOOUT,1040) IALB1,IALB2
      CONTINUE
C 230 SELECT FREQUENCY, WAVELENGTH, OR WAVENUMBER
      IF (IFW-4) 260,270,280
      WVNUM1=FREQ1/30.
      WVNUM2=FREQ2/30.
      WAVE1=0.
      IF (FREQ2.GT..0001) WAVE1=3.E+05/FREQ2
      WAVE2=3.E+05/FREQ1
      GO TO 290
      WVNUM1=0.
      IF (WAVE2.GT..0001) WVNUM1=1.E+04/WAVE2
      WVNUM2=1.E+04/WAVE1
      FREQ1=30.*WVNUM1
      FREQ2=30.*WVNUM2
      GO TO 290
      FREQ1=30.*WVNUM1
      FREQ2=30.*WVNUM2
      WAVE1=0.
      IF (WVNUM2.GT..0001) WAVE1=1.E+04/WVNUM2
      WAVE2=1.E+04/WVNUM1
      CONTINUE
      IF (VIS.LT..0001.AND.EXTN55.LT.0001) WRITE (IOOUT,1050)
      IF (VIS.LT..0001.AND.EXTN55.LT.0001) VIS=10.
      IF (EXTN55.GT..0001) VIS=3.912/EXTN55
      IF (VIS.GT..0001) EXTN55=3.912/VIS
C 240 OUTPUT INFORMATION
      WRITE (IOOUT,1080)
      DO 300 I=1,20
      IF (IPROGN(I).EQ.I) WRITE (IOOUT,1090) IPNAM(I),IPNAM(I+20)
      WRITE (IOOUT,1100) WVNUM1,WVNUM2,WAVE1,WAVE2,FREQ1,FREQ2
C 250 CLIMAT USES UNIT NCLIMT
      IF (ICLMAT.EQ.1) CALL CLIMAT(LOCAT,MONTH,NHOUR,IWIND,NPRT,TEMP,
      1 PRESS,RH,AH,DP,VIS,WNDVEL,WNDDIR,IPASCT)
      WRITE (IOOUT,1110) VIS
      IF (ICLMAT.EQ.2) ICLMAT=1
C***** SPOT CONTRAST PGM *****
      IF (IPROGN(1).NE.1) GO TO 320
      IPGM1=IDOPGM(1)
      WRITE (IOOUT,1130)
      DO 310 I=1,IPGM1
C 310 SPOT USES UNITS: IPHFUN - PHASE FUNCTION; LOUNIT - LT4M DATA
      CALL SPOT(WVNUM1,WVNUM2,VIS,NR,IERR,MULDV)
      CALL RESET(IERR)
C***** TURBULENCE PGM *****
      320 IF (IPROGN(2).NE.2) GO TO 340

```

```

EOM03390
EOM03400
EOM03410
EOM03420
EOM03430
EOM03440
EOM03450
EOM03460
EOM03470
EOM03480
EOM03490
EOM03500
EOM03510
EOM03520
EOM03530
EOM03540
EOM03550
EOM03560
EOM03570
EOM03580
EOM03590
EOM03600
EOM03610
EOM03620
EOM03630
EOM03640
EOM03650
EOM03660
EOM03670
EOM03680
EOM03690
EOM03700
EOM03710
EOM03720
EOM03730
EOM03740
EOM03750
EOM03760
EOM03770
EOM03780
EOM03790
EOM03800
EOM03810
EOM03820
EOM03830
EOM03840
EOM03850
EOM03860
EOM03870
EOM03880
EOM03890
EOM03900
EOM03910
EOM03920
EOM03930
EOM03940
EOM03950
EOM03960
EOM03970
EOM03980
EOM03990
EOM04000
EOM04010
EOM04020
EOM04030
EOM04040
EOM04050
EOM04060
EOM04070
EOM04080

```

```

IPGM2=IDOPGM(2)
WRITE (IOOUT,1140)
DO 330 I=1,IPGM2
330 CALL TURB(WAVE1,IERR)
CALL RESET(IERR)
C***** LASER MULTIPLE SCATTERING PGM *****
340 IF (IPROGN(3).NE.3) GO TO 380
IPGM3=IDOPGM(3)
ISPOT=.FALSE.
WRITE (IOOUT,1150)
DO 350 I=1,IPGM3
C BASCAT USES UNIT IPHFUN FOR PHASE FUNCTION DATA
350 CALL BASCAT(WAVE1,EXTN,IERR)
CALL RESET(IERR)
C***** LT4M PGM *****
360 IF (IPROGN(4).NE.4) GO TO 380
C LT4M READS ATM DATA FROM LOUNIT
ISPOT=.FALSE.
LREAD=.TRUE.
IPGM4=IDOPGM(4)
DO 370 I=1,IPGM4
1 CALL LT4M(H1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN,MODEL,
VIS,WVNUM1,WVNUM2,T1,ICLMAT,IERR,NR,HAZE,MULDV)
370 LOTRNS=LOTRNS*TRAN(1)
CALL RESET(IERR)
C***** XSCALE EXTINCTION PGM *****
380 IF (IPROGN(5).NE.5) GO TO 400
WRITE (IOOUT,1160)
IPGM5=IDOPGM(5)
DO 390 I=1,IPGM5
CALL XSCALE(WAVE1,VIS,EXTN55,XTRN,IERR,0,0,0,0)
390 XSTRN=XSTRN*XTRN
CALL RESET(IERR)
C***** SMOKE PGM *****
400 IF (IPROGN(6).NE.6) GO TO 420
WRITE (IOOUT,1170)
IPGM6=IDOPGM(6)
DO 410 N=1,IPGM6
CALL SMOKE(WAVE1,ICLMAT,STRANS,IERR)
410 SMKTRN=SMKTRN*STRANS
CALL RESET(IERR)
C***** DRTRAN PGM *****
420 IF (IPROGN(7).NE.7) GO TO 440
WRITE (IOOUT,1180)
IPGM7=IDOPGM(7)
HOLDWV=WAVE1
IF (IFW.EQ.1) WAVE1=WAVE2
DO 430 N=1,IPGM7
C DRTRAN USES NDIRTU FOR DATA
IF (N.GT.1) WRITE (IOOUT,1120)
CALL DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)
430 DRTRN=DRTRN*TRNLOS
WAVE1=HOLDWV
CALL RESET(IERR)
C***** LASER TRANSMISSION PGM *****
440 IF (IPROGN(8).NE.8) GO TO 460
WRITE (IOOUT,1190)
IPGM8=IDOPGM(8)
DO 450 I=1,IPGM8
CALL LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)
450 LZTRN=LZTRN*LAZTRN
CALL RESET(IERR)
C***** NEAR MILLIMETER WAVE PGM *****
460 IF (IPROGN(9).NE.9) GO TO 480
WRITE (IOOUT,1200)
IPGM9=IDOPGM(9)
DO 470 I=1,IPGM9
CALL MMW(FREQ1,ICLMAT,MMTRAN,IERR)
470 MMWTRN=MMWTRN*MMTRAN
CALL RESET(IERR)

```

```

EOM04090
EOM04100
EOM04110
EOM04120
EOM04130
EOM04140
EOM04150
EOM04160
EOM04170
EOM04180
EOM04190
EOM04200
EOM04210
EOM04220
EOM04230
EOM04240
EOM04250
EOM04260
EOM04270
EOM04280
EOM04290
EOM04300
EOM04310
EOM04320
EOM04330
EOM04340
EOM04350
EOM04360
EOM04370
EOM04380
EOM04390
EOM04400
EOM04410
EOM04420
EOM04430
EOM04440
EOM04450
EOM04460
EOM04470
EOM04480
EOM04490
EOM04500
EOM04510
EOM04520
EOM04530
EOM04540
EOM04550
EOM04560
EOM04570
EOM04580
EOM04590
EOM04600
EOM04610
EOM04620
EOM04630
EOM04640
EOM04650
EOM04660
EOM04670
EOM04680
EOM04690
EOM04700
EOM04710
EOM04720
EOM04730
EOM04740
EOM04750
EOM04760
EOM04770
EOM04780

```



```

C***** CLOUD TRANSMISSION PGM *****EOM04790
480 IF (IPROGN(10).NE.10) GO TO 500EOM04800
      IPGM10=IDOPGM(10)EOM04810
      DO 490 I=1,IPGM10EOM04820
        WRITE (IOOUT,1210)EOM04830
        CALL CLTRAN(CTRANS,WAVE1,I,IERR)EOM04840
      CLTRN=CLTRN*CTRANS EOM04850
      CALL RESET(IERR)EOM04860
C**** CWIC MUNITION EXPENDITURES/INVERSE STATIC TARGET DETECTION PGM ***EOM04870
500 IF (IPROGN(11).NE.11) GO TO 520EOM04880
      WRITE (IOOUT,1220)EOM04890
      IPGM11=IDOPGM(11)EOM04900
      DO 510 I=1,IPGM11EOM04910
        CALL SCREEN(IERR,ICLMAT)EOM04920
      CALL RESET(IERR)EOM04930
C***** FINITE CLOUD RADIATIVE TRANSFER PGM *****EOM04940
520 IF (IPROGN(12).NE.12) GO TO 540EOM04950
      WRITE (IOOUT,1230)EOM04960
      IPGM12=IDOPGM(12)EOM04970
C FLOUD USES IPHFUN FOR PHASE FUNCTION DATAEOM04980
      DO 530 I=1,IPGM12EOM04990
        CALL FLOUD(WAVE1,FTRANS,IERR)EOM05000
      CALL RESET(IERR)EOM05010
C***** OVERCAST SKY RADIATIVE TRANSFER PGM *****EOM05020
540 IF (IPROGN(13).NE.13) GO TO 560EOM05030
      WRITE (IOOUT,1240)EOM05040
      IPGM13=IDOPGM(13)EOM05050
      DO 550 I=1,IPGM13EOM05060
        CALL OVRCAST(WAVE1,OTRANS,IERR)EOM05070
      CALL RESET(IERR)EOM05080
C***** SELF-SCREENING SMOKE GRENADE PGM *****EOM05090
560 IF (IPROGN(14).NE.14) GO TO 575EOM05100
      WRITE (IOOUT,1250)EOM05110
      IPGM14=IDOPGM(14)EOM05120
      DO 570 I=1,IPGM14EOM05130
        CALL GRNADE(WAVE1,ICLMAT,GRTRAN,IERR)EOM05140
      GRNTRN=GRNTRN*GRTRAN EOM05150
      CALL RESET(IERR)EOM05160
C*****EOM05170
575 IF (IPROGN(4).GT.0.OR. IPROGN(5).GT.0.OR. IPROGN(6).GT.0.OR. IPROGN(7)
+ .GT.0.OR. IPROGN(8).GT.0.OR. IPROGN(9).GT.0.OR. IPROGN(10).GT.0.OR.
+ IPROGN(14).GT.0) GO TO 576
      GO TO 580
576 CALL COMPT(LOTRNS,XSTRN,SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN)EOM05180
580 CONTINUEEOM05190
      WRITE (IOOUT,1260)EOM05200
      STOPEOM05210
C EOM05220
C EOM05230
1000 FORMAT(2A4,1X,10E7.4)EOM05240
1010 FORMAT(1H0,20X,75H***EOMAIN WARNING*** EORUN CYCLE CARD OUT OF SEQUEOM05250
      UENCE, DEFAULT TO ONE CYCLE //)EOM05260
1020 FORMAT(1H0,20X,74H***EOMAIN ERROR*** FREQ, WAVL, OR WVNUM CARD WASEOM05270
      1 NOT INPUT, RUN TERMINATED //)EOM05280
1030 FORMAT(1H0,20X,46H***EOMAIN ERROR*** END OF READ SENTINEL ABSENT /EOM05290
      1/,1X,20X,28HRESULTS MAY BE UNPREDICTABLE)EOM05300
1040 FORMAT(1H0,20X,80H***EOMAIN ERROR*** INPUT CARD DETECTED WHICH DQEEOM05310
      1S NOT MATCH CORRECT INPUT FORMAT//,1X,20X,13HTHE CARD WAS:,2X,2A4)EOM05320
1050 FORMAT (1H0,20X,24H*** EOSAEL WARNING ****,/,1X,20X,15HVISIBILITYEOM05330
      1 AND 47HEXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM//)EOM05340
1060 FORMAT (1H1,//////,1X,50X,30(1H*),/,1X,50X,1H*,28X,1H*,/EOM05350
      1,1X,50X,30H* ELECTRO-OPTICAL SYSTEMS *,/,1X,50X,1H*,EOM05360
      2 28X,1H*,/,1X,50X,30H* ATMOSPHERIC EFFECTS *,/EOM05370
      3 1X,50X,1H*,28X,1H*,/,1X,50X,25H* LIBRARYEOM05380
      4 5H *,/,1X,50X,1H*,28X,1H*,/,1X,50X,30(1H*))EOM05390
1070 FORMAT (1H1,//////,58X,11HRUN NUMBER ,12)EOM05400
1080 FORMAT (///1X,51X,28HINDIVIDUAL MODULES SELECTED)EOM05410
1090 FORMAT (1X,62X,2A4)EOM05420
1100 FORMAT (1H0,63X,9HBEGINNING,12X,6HENDING,/,/,39X,14HWAVENUMBER(CM+EOM05430
      1 4H*-1),6X,F10.3,10X,F10.3,/,/,39X,19HWAVELENGTH(MICRONS)),EOM05440

```

```

2          5X,F10.3,10X,F10.3,/,39X,14HFREQUENCY(GHZ),5X,F15.3,5X, EOM05450
3          F15.3,/) EOM05460
1110 FORMAT (1H0,62X,10HVISIBILITY,/,62X,F5.2,3H KM) EOM05470
1120 FORMAT(1H1) EOM05480
1130 FORMAT(1H1,40X,20HSPOT CONTRAST MODULE //) EOM05490
1140 FORMAT(1H1,40X,17HTURB LASER MODULE //) EOM05500
1150 FORMAT(1H1,40X,30HBASCAT LASER SCATTERING MODULE //) EOM05510
1160 FORMAT (1H1,40X,46HXSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE EOM05520
      //) EOM05530
1170 FORMAT (1H1,45X,19HSMOKE MODEL MODULE //) EOM05540
1180 FORMAT (1H1,40X,26H DIRT TRANSMISSION MODULE //) EOM05550
1190 FORMAT (1H1,40X,28H LASER TRANSMITTANCE MODULE //) EOM05560
1200 FORMAT (1H1,45X,29H NEAR MILLIMETER WAVE MODULE //) EOM05570
1210 FORMAT(1H1,40X,27H CLOUD TRANSMITTANCE MODULE //) EOM05580
1220 FORMAT(1H1,20X,43HCWIC MUNITION EXPENDITURES / INVERSE STATIC EOM05590
      1 24H TARGET DETECTION MODULE ) EOM05600
1230 FORMAT(1H1,40X,38HFINITE CLOUD RADIATIVE TRANSFER MODULE //) EOM05610
1240 FORMAT(1H1,40X,38HOVERCAST SKY RADIATIVE TRANSFER MODULE //) EOM05620
1250 FORMAT(1H1,40X,35HSELF-SCREENING SMOKE GRENADE MODULE //) EOM05630
1260 FORMAT (1X,/,/,1X,50X,14HEND EOSAEL RUN) EOM05640
      END EOM05650

```

```

SUBROUTINE RESET (IERR)
THE PURPOSE OF THIS ROUTINE IS: (1) TO RESET THE SEQUENCING OF
DATA CARDS DUE TO AN ERROR IN A PREVIOUS MODULE OR (2) TO READ
A SENTINAL CARD THAT DELINEATES THE END OF A DATA SET (SEE
BELOW FOR DEFINITION OF A DATA SET) OR (3) TO STOP THE PROGRAM -
THIS LAST MODE IS USALLY FOR DEBUGGING OR TO ONLY CHANGE A CARD
IN A COMPLETE RUN.
TO DELINEATE THE END OF A DATA SET A CARD THAT HAS JUST END
ON IT MUST BE INSERTED AS A SENTINAL CARD; A DATA SET IS
DEFINED AS THAT COMPLETE SET OF CARDS NECESSARY TO RUN THE
CALLED MODULE THE NUMBER OF TIMES AS SPECIFIED ON THE IDOPGM(I)
CARD. STOP MAY ALSO BE INSERTED AS A SENTINAL CARD, IN WHICH
CASE THE PROGRAM WILL BE TERMINATED AT THAT POINT - THIS IS
NOT THE NORMAL TERMINATION.
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUR
DATA ICHCK1, ICHCK2, ISTOP1, ISTOP2 /2HEN, 2HD, 2HST, 2HOP/
IF (IERR.EQ.1) GO TO 1
CONTINUE
FOR UNIVAC AND IBM
READ (IOIN,100,END=2) ISNTL1, ISNTL2
READ (IOIN,100) ISNTL1, ISNTL2
C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE
C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.
C3 READ (IOIN,100) ISNTL1, ISNTL2
C4 IF (EOF(IOIN)) 2,10
10 IF ((ISNTL1.NE.ICHCK1.AND.ISNTL2.NE.ICHCK2).AND.
1 (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 5
IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP
RETURN
WRITE (IOOUT,102)
FOR UNIVAC AND IBM
C6 READ (IOIN,100,END=2) ISNTL1, ISNTL2
6 READ (IOIN,100) ISNTL1, ISNTL2
C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE
C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.
C3 READ (IOIN,100) ISNTL1, ISNTL2
C4 IF (EOF(IOIN)) 2,20
20 CONTINUE
IF ((ISNTL1.NE.ICHCK1.AND.ISENTL.NE.ICHCK2).AND.
1 (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 6
IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP
IERR=0
RETURN
2 WRITE (IOOUT,101) IOIN
100 FORMAT (2A2)
101 FORMAT (1X,120(1H*),/,1X,29H ERROR IN INPUT CONTROL FILE ,I4,
+ 21H - PROGRAM TERMINATED,/,1X,120(1H*))
102 FORMAT(1H0,50H**** CARD SEQUENCE RESET DUE TO ERROR IN PREVIOUS ,
1 15HMODULE (IERR=1)///)
STOP
END

```

```

RES000010
RES000020
RES000030
RES000040
RES000050
RES000060
RES000070
RES000080
RES000090
RES000100
RES000110
RES000120
RES000130
RES000140
RES000150
RES000160
RES000170
RES000180
RES000190
RES000200
RES000210
RES000220
RES000230
RES000240
RES000250
RES000260
RES000270
RES000280
RES000290
RES000300
RES000310
RES000320
RES000330
RES000340
RES000350
RES000360
RES000370
RES000380
RES000390
RES000400
RES000410
RES000420
RES000430
RES000440
RES000450
RES000460
RES000470
RES000480
RES000490
RES000500
RES000510
RES000520
RES000530

```

CCCCCCCC

100

```
SUBROUTINE COMPLT<LOTRNS,XSTRN,                                COM00010
+ REAL LOTRNS,LZTRN,MMWTRN,SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN> COM00020
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU COM00030
LOWTRAN = LOTRNS                                             COM00040
XSCALE = XSTRN                                               COM00050
SMOKE = SMKTRN                                               COM00060
DRTRAN = DRTRN                                               COM00070
LZTRAN = LZTRN                                               COM00080
MMW = MMWTRN                                                 COM00090
CLTRAN = CLTRN                                               COM00100
GRNADE = GRNTRN                                             COM00110
TRAN=LOTRNS*XSTRN*SMKTRN*DRTRN*CLTRN*GRNTRN*MMWTRN*LZTRN COM00120
WRITE (IOOUT,100) TRAN                                       COM00130
FORMAT (///1X,20X,24HCOMBINED TRANSMISSION FO,15HR THE SELECTED , COM00150
1 10HMODULES = ,E10.4)                                       COM00160
RETURN                                                         COM00170
END                                                             COM00180
```

```

SUBROUTINE ILLUM(LAMBDA,LD,E0) ILL00010
REAL LAMBDA,LUNPHA ILL00020
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU ILL00030
C***** ILL00040
C ILL00050
C ILL00060
C ILL00070
C ILL00080
C ILL00090
C ILL00100
C ILL00110
C ILL00120
C ILL00130
C ILL00140
C***** ILL00150
C ILL00160
C ILL00170
C ILL00180
C ILL00190
C ILL00200
C ILL00210
C ILL00220
C ILL00230
C ILL00240
C ILL00250
C ILL00260
C ILL00270
C ILL00280
C ILL00290
SUBROUTINE ILLUM RETURNS THE EXTRATERRESTRIAL IRRADIANCE E0
AT WAVELENGTH LAMBDA. IF LD = 0, THE VALUE GIVEN IS SOLAR
IRRADIANCE. IF 1 < LD < 28 THE VALUE GIVEN IS LUNAR IRRAD-
IANCE ON LUNAR DAY LD, WITH DAY 28 CORRESPONDING TO FULL
MOON AND DAY 14 BEING NEW MOON.
SUBROUTINE COMPUTES VALUE OF LUNAR PHASE ANGLE, IF REQUIRED,
AND CALLS ONE OF THE EOSAEL ROUTINES SOLARS OR SMOON.
IF(ILD.GT.0) GO TO 10
E0=SOLARS(LAMBDA)
GO TO 100
LD GT 0 => E0 = LUNAR VALUE
10 ILD=LD
IF(ILD.GT.14) ILD=28-ILD
LUNPHA=180.0*FLOAT(ILD)/14.0
E0=SMOON(LAMBDA,LUNPHA)
100 WRITE(IOOUT,1000) E0
RETURN
1000 FORMAT(32H0 EXTRATERRESTRIAL IRRADIANCE= ,1PE10.4,11H W/M2-SR-MU)
END

```

CCCCC

```
FUNCTION SOLARS(WAVL) SOL00010
CALCULATE THE INTENSITY OF THE SOLAR SPECTRUM FOR WAVELENGTH (WAVL) SOL00020
UNITS: SOLARS ... WATTS M-2 MICRON-1 SOL00030
      WAVL ... MICRONS SOL00040
IF (WAVL.LT.0.15.OR.WAVL.GE.100.) GO TO 100 SOL00050
IF (WAVL.GE.0.15.AND.WAVL.LE.0.43) GO TO 200 SOL00060
IF (WAVL.GT.0.43.AND.WAVL.LT.0.58) GO TO 300 SOL00070
GO TO 400 SOL00080
100 SOLARS=0.0 SOL00090
    RETURN SOL00100
200 Z=(WAVL-.415)/0.68)**2 SOL00110
    SOLARS=1775.*EXP(Z/2) SOL00120
    RETURN SOL00130
300 CONTINUE SOL00140
    SOLARS=(-61142.*WAVL**4)+(1344477.*WAVL**3)-(110296.* SOL00150
    WAVL**2) SOL00160
    SOLARS=(SOLARS+39952.7*WAVL-5371.)-/100. SOL00170
    RETURN SOL00180
400 IF (WAVL.GE.2.5) GO TO 500 SOL00190
    SOLARS=5331.9*EXP(-1.959*WAVL) SOL00200
    RETURN SOL00210
500 SOLARS=2288.38*(WAVL**(-3.9765)) SOL00220
    RETURN SOL00230
END SOL00240
      SOL00250
      SOL00260
      SOL00270
```

CCCCCCCC

FUNCTION SMOON(WLAM,ANGLE)

CALCULATE THE INTENSITY OF MOONLIGHT FOR WAVELENGTH (WLAM)
AND PHASE ANGLE (ANGLE)

UNITS:

SMOON ... WATTS M-2 MICRON-1
ANGLE ... DEGREES
WLAM MICRONS

SMOON=0.0

IF (ANGLE.GT.160.) RETURN

SMOON=(3.426E-9*ANGLE**4-1.63E-6*ANGLE**3+3.01E-4*

1 ANGLE**2-.0266*ANGLE+.9882)*100.

ALBED=0.4

IF (WLAM.GE.5.) GO TO 200

IF (WLAM.GT.2.8) GO TO 100

IF (WLAM.LE.1.) ALBED=3.9633*WLAM**4-10.7306*WLAM**3+

1 10.2188*WLAM**2-3.8208*WLAM+.5512

IF (WLAM.GT.1.) ALBED=.0482*WLAM**4-.3283*WLAM**3+

1 .7584*WLAM**2-.5745*WLAM+.2808

GO TO 200

100 ALBED=.350+(.500-.350)*(WLAM-2.8)/2.2

200 SMOON=2.04472E-07*SOLARS(WLAM)*ALBED*SMOON

RETURN
END

SMN00010
SMN00020
SMN00030
SMN00040
SMN00050
SMN00060
SMN00070
SMN00080
SMN00090
SMN00100
SMN00110
SMN00120
SMN00130
SMN00140
SMN00150
SMN00160
SMN00170
SMN00180
SMN00190
SMN00200
SMN00210
SMN00220
SMN00230
SMN00240
SMN00250

C
C

```
FUNCTION JPASCT(ICAT)  
THIS FUNCTION CONVERTS THE INTEGER CODE FOR PASQUILL CATEGORY  
TO THE ALPHA CHARACTER  
DIMENSION NPASCT(6)  
DATA NPASCT/1HA,1HB,1HC,1HD,1HE,1HF/  
JPASCT=NPASCT(ICAT)  
RETURN  
END
```

```
JPA00010  
JPA00020  
JPA00030  
JPA00040  
JPA00050  
JPA00060  
JPA00070  
JPA00080
```



```

SUBROUTINE PFUNC(IDN)
LOGICAL ISPOT, LOREAD, N16
DIMENSION PFSPT(16), PFH(65)
COMMON /SPOTLO/ISPOT, LOREAD, N16
COMMON /IDUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, HDIRTU, NCLIMT, KSTOR, NPLOTU
COMMON /CONST/P1, P12, PIRAD, TWOPI, TORRMB, CDEGK
COMMON /CGEOM/COSGM, COSBT, COSIN
COMMON /BASPOT/ANG(65), SUM(65), WV(16), NWVL, ALB(16), BS(16),
+BE(16), SINGWV, PF(65), LMAX
THIS SUBROUTINE PERFORMS PHASE FUNCTION READING AND INTERPOL-
ATION OPERATIONS FOR THE SPOT AND BASCAT MODULES. THE FILE IN
WHICH THE PHASE FUNCTION DATA RESIDES IS PFNDAT. USERS MAY IN-
SERT PHASE FUNCTIONS OF THEIR OWN SPECIFICATION INTO PFNDAT
UNDER AN ID NUMBER 0. THIS PHASE FUNCTION MAY HAVE ANY AR-
BITRARY NORMALIZATION, SINCE PFUNC WILL RENORMALIZE IT TO
CONFORM TO THE NORMALIZATIONS USED IN SPOT AND BASCAT.
FOR FURTHER DETAILS ON THIS PROCEDURE,
THE USER IS REFERRED TO CHAPTER 16 OF THE EOSAEL 80 TECHNICAL
DOCUMENTATION MANUAL, WHERE THE STRUCTURE, USE, AND MODIFICATION
OF THE PFNDAT FILE IS DISCUSSED.
*** NOTE *** FOR USER-DEFINED PHASE FUNCTIONS (IDN=0), THIS ROUTINE
WILL INTERPOLATE OVER WAVELENGTH AND ANGLE FOR THE SPOT
MODULE. FOR THE BASCAT MODULE, HOWEVER, NO WAVELENGTH
INTERPOLATION IS PERFORMED FOR THE USER-DEFINED PHASE
FUNCTION. ONLY RENORMALIZATION IS PERFORMED IN THIS
LATTER CASE.
*** MAXID IS THE NUMBER OF DIFFERENT DISTRIBUTIONS - (PHASE FUNCTIONS)
MAXID=12
*** CHECK THE ALLOWABLE RANGE OF DISTRIBUTIONS
IF((IDN.GT.MAXID).OR.(IDN.LT.0))GO TO 491
*** DATA INITIALIZATIONS
DO 40 I=1,65
40 PF(I)=0.
ALBE=0.
BEX=0.
BSC=0.
*** ANGULAR READ BLOCK COMMON TO SPOT AND BASCAT
I=0
**** READ IN ANGLES AT WHICH PHASE FUNCTION IS DEFINED
50 I=I+1
L1=(I-1)*11+1
L11=L1+10
IF(L11.EQ.66)L11=65
IF(L11.GT.66)GO TO 492
READ(IPHFUN,60)(ANG(L),L=L1,L11)
60 FORMAT(11(F6.2,1X))
**** CHECK THIS ROW OF DATA FILE FOR TERMINATION SENTINEL
DO 100 K=L1,L11
IF((ANG(K).GE.999.99).AND.(L11.LT.65))GO TO 200
IF(K.EQ.65)GO TO 200
100 CONTINUE
GO TO 50
**** LMAX IS THE NBR OF ANGLE AND NBR OF PHASE FUNCTION VALUES+1 HERE.
200 LMAX=K

```

```

DO 250 L=1,LMAX
250 ANG(L)=COS(ANG(L))*PIRAD
C
C*** REDUCE DUE TO SENTINEL OF 999.99
C
IF(L11.LT.65)LMAX=K-1
IDNM=IDN-1
KMAX=IFIX(ALOG(FLOAT(LMAX-1))/ALOG(2.0)+0.1)
C
C*** RESET PARAMETERS FOR BASCAT PROCESSING IF APPROPRIATE.
C
IF(ISPOT) GO TO 260
NWVL=1
WVL(1)=SINGWV
260 CONTINUE
IF((IDN.EQ.1).OR.(IDN.EQ.0.)) GO TO 1050
C
C*** READ PAST AEROSOL DATA NOT OF CURRENT INTEREST
C
DO 1000 I=1,IDNM
DO 1000 II=1,16
READ(IPHFUN,300) IANG, ID, WAVE, ALBE, BEX, BSC
300 FORMAT(2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))
IF(IANG.NE.LMAX) GO TO 493
READ(IPHFUN,400) (PF(L),L=1,LMAX)
400 FORMAT(6(E12.6,1X))
1000 CONTINUE
1050 CONTINUE
C
C*** OMIT WAVELENGTH CHECKS FOR USER-DEFINED PHASE FUNCTION.
C
IF(IDN.EQ.0) GO TO 1070
C
C*** THE NEXT LOOP PERFORMS THE FOLLOWING OPERATIONS :
C
C (A) IT VERIFIES WHETHER OR NOT ALL INPUT WAVELENGTHS LIE
C WITHIN THE 0.2-12.0 MICROMETER RANGE (WITH LIMITS
C EXTENDED TO PLUS OR MINUS 5%).
C (B) IF THE PHASE FUNCTION IS NOT USER-SPECIFIED AND THE INPUT
C WAVELENGTH BAND LIES WITHIN THE 0.2-2.0 BAND, THEN
C INTERPOLATION IS NOT POSSIBLE DUE TO THE PRESENCE OF
C ONLY TWO DATA POINTS (AT 0.55 AND 1.06) IN THIS REGION.
C AS A RESULT, THE 0.55 AND 1.06 VALUES ARE ASSIGNED TO
C INDIVIDUAL POINTS IN THE INPUT WAVELENGTH BAND. THOSE
C POINTS WITH WAVELENGTH VALUES LESS THAN OR EQUAL TO 0.8
C MICROMETERS ARE ASSIGNED THE 0.55 DATA. ALL OTHER WAVE-
C LENGTHS ARE ASSIGNED THE 1.06 DATA.
C (C) IF AN INPUT WAVELENGTH LIES OUTSIDE OF THE 3-5 OR 8-12
C MICROMETER BANDS, BUT IS WITHIN 5% OF AN EXTREMUM FOR
C THESE BANDS, IT IS RESET TO THE EXTREMUM WAVELENGTH
C VALUE.
C (D) IF AN INPUT WAVELENGTH LIES BETWEEN BANDS, EXECUTION IS
C TERMINATED AND AN ERROR MESSAGE IS PRINTED.
C
DO 1060 I=1,NWVL
IF((WVL(I).LT.0.19).OR.(WVL(I).GT.12.049)) GO TO 504
IF(WVL(I).LE.0.8) WVL(I)=0.55
IF((WVL(I).GT.0.8).AND.(WVL(I).LE.2.1)) WVL(I)=1.06
IF(WVL(I).GT.12.0) WVL(I)=12.0
IF((WVL(I).GT.5.0).AND.(WVL(I).LE.5.25)) WVL(I)=5.0
IF((WVL(I).GE.7.6).AND.(WVL(I).LT.8.0)) WVL(I)=8.0
IF((WVL(I).GE.2.85).AND.(WVL(I).LT.3.0)) WVL(I)=3.0
IF((WVL(I).GT.2.1).AND.(WVL(I).LT.2.85)) GO TO 498
IF((WVL(I).GT.5.25).AND.(WVL(I).LT.7.6)) GO TO 498
1060 CONTINUE
1070 CONTINUE
C
C*** MAIN INTERPOLATION LOOP
C
DO 2000 I=1,NWVL
IF(I.GT.1) GO TO 1260
PFU00710
PFU00720
PFU00730
PFU00740
PFU00750
PFU00760
PFU00770
PFU00780
PFU00790
PFU00800
PFU00810
PFU00820
PFU00830
PFU00840
PFU00850
PFU00860
PFU00870
PFU00880
PFU00890
PFU00900
PFU00910
PFU00920
PFU00930
PFU00940
PFU00950
PFU00960
PFU00970
PFU00980
PFU00990
PFU01000
PFU01010
PFU01020
PFU01030
PFU01040
PFU01050
PFU01060
PFU01070
PFU01080
PFU01090
PFU01100
PFU01110
PFU01120
PFU01130
PFU01140
PFU01150
PFU01160
PFU01170
PFU01180
PFU01190
PFU01200
PFU01210
PFU01220
PFU01230
PFU01240
PFU01250
PFU01260
PFU01270
PFU01280
PFU01290
PFU01300
PFU01310
PFU01320
PFU01330
PFU01340
PFU01350
PFU01360
PFU01370
PFU01380
PFU01390
PFU01400

```

```

1100 CONTINUE
READ(IPHFUN,300) IANG, ID, WAVEH, ALBH, BEXH, BSCH
READ(IPHFUN,400) (PFH(L), L=1, LMAX)
IF((IDN.EQ.0).AND.(.NOT.ISPOT)) GO TO 1280
IF(WVL(I).LT.WAVEH) GO TO 1100
1150 CONTINUE
READ(IPHFUN,300) IANG, ID, WAVE, ALBE, BEX, BSC
READ(IPHFUN,400) (PF(L), L=1, LMAX)
IF(WVL(I).LE.WAVE) GO TO 1260
1160 CONTINUE
WAVEH=WAVE
ALBH=ALBE
BEXH=BEX
DO 1240 L=1, LMAX
1240 PFH(L)=PF(L)
GO TO 1150
1260 CONTINUE
C
C*** GO TO NEXT WAVELENGTH INTERPOLATION INTERVAL IF INPUT WAVE-
C*** LENGTH IS GREATER THAN THE MAXIMUM OF THE CURRENT ONE.
C
IF(WVL(I).GT.WAVE) GO TO 1160
C
C*** RENORMALIZE LOWER END OF INTERPOLATION INTERVAL
C
1280 CONTINUE
SUM(1)=0.0
DO 1200 L=2, LMAX
1200 SUM(L)=((ANG(L-1)-ANG(L))*(PFH(L-1)+PFH(L))/4.0)+SUM(L-1)
SUMT=SUM(LMAX)
DO 1250 L=1, LMAX
1250 PFH(L)=PFH(L)/SUMT
C
C*** BRANCH TO FINAL PROCEDURE FOR BASCAT. USER-DEFINED PHASE
C*** FUNCTION IF APPROPRIATE.
C
IF((IDN.EQ.0).AND.(.NOT.ISPOT)) GO TO 2500
C
C*** RENORMALIZE UPPER END OF INTERPOLATION INTERVAL.
C
SUM(1)=0.0
DO 1400 L=2, LMAX
1400 SUM(L)=((ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0)+SUM(L-1)
SUMT=SUM(LMAX)
DO 1450 L=1, LMAX
1450 PF(L)=PF(L)/SUMT
C
C*** BRANCH TO BASCAT WAVELENGTH INTERPOLATION PROCEDURE IF
C*** APPROPRIATE.
C
IF(.NOT.ISPOT) GO TO 2500
C
C*** PERFORM HALVING SEARCH FOR COSINES IN PHASE FUNCTION DATA FILE
C*** WHICH BRACKET COSINE INPUT FROM SPOT.
C
L=1
LL=LMAX-1
DO 1300 K=1, KMAX
LL=LL/2
L=L+LL
AT=COSIN-ANG(L)
IF(AT.GT.0.) L=L-LL
1300 CONTINUE
C
C*** PERFORM SPOT EXTINCTION COEFFICIENT AND PHASE FUNCTION
C*** INTERPOLATIONS OVER WAVELENGTH AND ANGLE.
C
FACANG=(COSIN-ANG(L))/(ANG(L+1)-ANG(L))
FACWVL=(WVL(I)-WAVEH)/(WAVE-WAVEH)
BE(I)=BEXH+(BEX-BEXH)*FACWVL

```

```

PFU01410
PFU01420
PFU01430
PFU01440
PFU01450
PFU01460
PFU01470
PFU01480
PFU01490
PFU01500
PFU01510
PFU01520
PFU01530
PFU01540
PFU01550
PFU01560
PFU01570
PFU01580
PFU01590
PFU01600
PFU01610
PFU01620
PFU01630
PFU01640
PFU01650
PFU01660
PFU01670
PFU01680
PFU01690
PFU01700
PFU01710
PFU01720
PFU01730
PFU01740
PFU01750
PFU01760
PFU01770
PFU01780
PFU01790
PFU01800
PFU01810
PFU01820
PFU01830
PFU01840
PFU01850
PFU01860
PFU01870
PFU01880
PFU01890
PFU01900
PFU01910
PFU01920
PFU01930
PFU01940
PFU01950
PFU01960
PFU01970
PFU01980
PFU01990
PFU02000
PFU02010
PFU02020
PFU02030
PFU02040
PFU02050
PFU02060
PFU02070
PFU02080
PFU02090
PFU02100

```

```

PFSPOT(I)=PFH(L)*(1.-FACANG-FACWVL+FACANG+FACWVL)+
+PFH(L+1)*(FACANG-FACANG+FACWVL)+PF(L)*(FACWVL-FACANG+FACWVL)+
+PF(L+1)*(FACANG+FACWVL)
2000 CONTINUE
C*** LOAD FIRST NWVL ANGLES OF OUTPUT ARRAY PF( ) WITH INTERPOLATED
C*** RESULTS FOR SPOT.
DO 2200 N=1,NWVL
2200 PF(N)=PFSPOT(N)/(4.*PI)
C*** FINAL EXIT FOR SPOT PROCESSING.
GO TO 500
2500 CONTINUE
FACWVL=0.
IF(IDN.EQ.0) GO TO 2700
C*** BASCAT ALBEDO, EXTINCTION COEFFICIENT, AND PHASE FUNCTION
C*** INTERPOLATION OVER WAVELENGTH.
FACWVL=(WVL(1)-WAVEH)/(WAVE-WAVEH)
2700 CONTINUE
DO 2800 L=1,LMAX
2800 PF(L)=PFH(L)+(PF(L)-PFH(L))*FACWVL
ALB(L)=ALBH+(ALBE-ALBH)*FACWVL
BE(L)=BEXH+(BEX-BEXH)*FACWVL
C*** FINAL EXIT FOR BASCAT USER-DEFINED PHASE FUNCTION PROCEDURE.
IF(IDN.EQ.0) GO TO 500
C*** FINAL BASCAT PHASE FUNCTION RENORMALIZATION.
SUM(1)=0.0
DO 2900 L=2,LMAX
2900 SUM(L)=(ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0+SUM(L-1)
SUMT=SUM(LMAX)
DO 2950 L=1,LMAX
2950 PF(L)=PF(L)/SUMT
C*** FINAL EXIT FOR BASCAT PROCESSING.
GO TO 500
C*** ERROR EXIT BLOCK COMMON TO SPOT AND BASCAT
491 CONTINUE
WRITE(IOOUT,495)
495 FORMAT(1H0,20X,58H***PFUNC ERROR*** AEROSOL ID NUMBER OUT OF ALLOW
+ABLE RANGE //)
STOP
492 CONTINUE
WRITE(IOOUT,496)
496 FORMAT(1H0,20X,83H***PFUNC ERROR*** READ TERMINATION SENTINEL NOT
+FOUND OR NUMBER OF ANGLES EXCEED 65 //)
STOP
493 CONTINUE
WRITE(IOOUT,497)
497 FORMAT(1H0,20X,93H***PFUNC ERROR*** NUMBER OF SPECIFIED ANGLES AND
+NUMBER OF PHASE FUNCTION VALUES DO NOT MATCH //)
STOP
498 CONTINUE
WRITE(IOOUT,499)
499 FORMAT(1H0,10X,79H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WVP
+L INPUT ARRAY DO NOT LIE WITHIN /1H,45HWAVELENGTH BANDS COVERED B
+Y PFNDAT DATA BASE //)
STOP
504 CONTINUE
WRITE(IOOUT,505)

```

```

PFU02120
PFU02130
PFU02140
PFU02150
PFU02160
PFU02170
PFU02180
PFU02190
PFU02200
PFU02210
PFU02220
PFU02230
PFU02240
PFU02250
PFU02260
PFU02270
PFU02280
PFU02290
PFU02300
PFU02310
PFU02320
PFU02330
PFU02340
PFU02350
PFU02360
PFU02370
PFU02380
PFU02390
PFU02400
PFU02410
PFU02420
PFU02430
PFU02440
PFU02450
PFU02460
PFU02470
PFU02480
PFU02490
PFU02500
PFU02510
PFU02520
PFU02530
PFU02540
PFU02550
PFU02560
PFU02570
PFU02580
PFU02590
PFU02600
PFU02610
PFU02620
PFU02630
PFU02640
PFU02650
PFU02660
PFU02670
PFU02680
PFU02690
PFU02700
PFU02710
PFU02720
PFU02730
PFU02740
PFU02750
PFU02760
PFU02770
PFU02780
PFU02790
PFU02800
PFU02810

```

```
505 FORMAT(1H0,10X,109H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WVPFU02820  
+L ARRAY DO NOT LIE WITHIN OVERALL ACCEPTABLE RANGE OF 0.2-12.0 /> PFU02830  
STOP PFU02840  
500 RETURN PFU02850  
END PFU02860
```

```

SUBROUTINE XSCALE(WAVE,VIS,EXT55,XSTRN,IERR,ISLT,IFOG,RANGE,ANGLE)XSC00010
THE PURPOSE OF THIS ROUTINE IS TO A) FIND THE HORIZONTAL EXTINCTIONXSC00020
IN FOG AT THE WAVELENGTHS SPECIFIED BELOW FROM THE EXTINCTIONXSC00030
AT .55 UM OR B) THE EXTINCTION ALONG A SLANT PATH AT ALLOWEDXSC00040
WAVELENGTHS FROM THE EXTINCTION AT .55 UM: FOG TYPE 1, 2, OR 3,XSC00050
MUST BE SPECIFIED FOR SLANT PATHS.XSC00060
*** VISIBILITY = 88, OR 89, IS NOT ALLOWED AS THIS IS USED AS ANXSC00070
INDICATOR THAT XSCALE IS BEING CALLED AS A SUBROUTINE FROM EITHERXSC00080
SPOT(88.), LOWTRAN(88.), OR CWIC(89.), NOT EOMAIN!XSC00090
WAVE=LAMDA IN UM - MUST BE .55, 1.06, 3.0-5.0, 8.0-12.05.XSC00100
*** ALL EXTN'S ARE IN KM**-1XSC00110
EXT55 = EXTINCTION AT .55 UMXSC00120
EXT106 = EXTINCTION AT 1.06 UMXSC00130
EXT35 = EXTINCTION FROM 3.0 TO 5.0 UMXSC00140
EXT812 = EXTINCTION FROM 8.0 TO 12.0 UMXSC00150
VIS= VISIBILITY IN KM -OR- EXT55 IN KM**-1XSC00160
EXT55 IS ** NOT ** CHANGED BY THIS ROUTINE.XSC00170
*****XSC00180
INPUT: THERE IS A MAXIMUM OF 3 CARDS TO EXECUTE THIS MODULEXSC00190
THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OFXSC00200
THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.XSC00210
THE CARDS ARE INPUT WITH FORMAT (A4,6X,5(F6.2,1X)).XSC00220
EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4XSC00230
FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 6 COL PERXSC00240
FIELD BEGINNING IN COL 11, WITH A BLANK BETWEEN EACH SUBSEQUENTXSC00250
FIELD. THE CARDS ARE NOT ORDER DEPENDENT.XSC00260
IF GEOMET OPTION IS BEING USED, THEN ONLY THE IDENTIFIER HORZ,XSC00270
SLNH, OR SLNS IS TO BE READ IN (NO ADDITIONAL PARAMETERS NEEDED).XSC00280
FOG FOG TYPE, RAIN RATE (MM/HR); RAIN RATE ONLY NEEDEDXSC00290
WHEN FOG TYPE=4.XSC00300
HORZ HORDIS (KM) ; HORIZONTAL PATH CALCULATIONXSC00310
SLNH HORDIS (KM), ANGLE (DEGREES); SLANT PATH CALCULATIONXSC00320
SLNS SLTDIS (KM), ANGLE (DEGREES); SLANT PATH CALCULATIONXSC00330
PLOT WRITE SLANT PATH EXTINCTION, AT INPUT WAVELENGTH, ANDXSC00340
ALTITUDE TO NPLTU (SEE COMMON BLOCK IOUNIT); THEXSC00350
FIRST RECORD WILL BE THE NUMBER OF POINTS TO BE WRITTEN.XSC00360
FORMATS: RECORD 1 (I5), SUBSEQUENT RECORDS (2(E10.4,1X))XSC00370
GO SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FORXSC00380
THIS CALL. NOTE THAT IF A DATA CARD IS NOT READXSC00390
THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLSXSC00400
TO THE MODULE ARE STILL IN EFFECT.XSC00410
ALL THE FOLLOWING FOG TYPES ARE RELEVANT TO HORIZONTAL PATHS,XSC00420
BUT ONLY FOG TYPES 1, 2, OR 3 ARE ALLOWED FOR SLANT PATH CALCULATIONSXSC00430
FOG TYPE=1. FOR MARITIME ARTICXSC00440
=2. FOR MARITIME POLARXSC00450
=3. FOR CONTINENTAL POLARXSC00460
=4. FOR RAINXSC00470
=5. FOR SNOWXSC00480
HORDIS - HORIZONTAL DISTANCE IN KM.XSC00490
SLTDIS - SLANT PATH DISTANCE IN KM.XSC00500
ANG= LOOK ANGLE FROM HORIZONTAL IN DEGREESXSC00510
N.B. ONE OF THE FOLLOWING COMBINATIONS MUST BE SUPPLIEDXSC00520
FOR SLANT PATH CALCULATIONS.XSC00530
HORDIS AND ANG ** OR ** SLTDIS AND ANGXSC00540
*****XSC00550
OUTPUTXSC00560
TRANSMISSION AT APPROPRIATE WAVELENGTH FOR SLANT OR HORIZONTAL PATHXSC00570
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGKXSC00580
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUXSC00590
COMMON /GEOMET/PTS(15),IGEOSW XSC00600
DIMENSION TYPE(6),DAT(6) XSC00610
LOGICAL NOLO XSC00620
DATA TYPE /4HFOG,4HHORZ,4HSLNH,4HSLNS,4HPLOT,4HGO / XSC00630
DATA A0,A1,A2,NPLT/0.1425,0.1475,-0.0017,0/ XSC00640
USE VIS=88, OR 89, AS AN INDICATOR THAT XSCALE HAS BEEN CALLED XSC00650
AS A SUBROUTINE FROM OTHER PROGRAMS - NOT EOMAIN! XSC00660
IF (VIS.LT.87.9.OR.VIS.GT.89.1) GO TO 8 XSC00670
ANG=ANGLE XSC00680
FIND ELEVATION ANGLE FROM ZENITH ANGLE IN SPOT AND LOWTRAN XSC00690

```

```

IF <VIS.GT.87.9.AND.VIS.LT.88.1> ANG=90.-ANGLE XSC00650
IF <VIS.GT.87.9.AND.VIS.LT.88.1.AND.ANGLE.GT.90.> ANG=ANGLE-90. XSC00660
HORDIS=0. XSC00670
SLTDIS=0. XSC00680
NOLO=.FALSE. XSC00690
ISLT=0=HORIZONTAL; ISLT.GT.0 = SLANT. XSC00700
IF <ISLT.EQ.0> HORDIS=RANGE XSC00710
IF <ISLT.EQ.0> ISLANT=0 XSC00720
IF <ISLT.GT.0> SLTDIS=RANGE XSC00730
IF <ISLT.GT.0> ISLANT=1 XSC00740
GO TO 6 XSC00750
CONTINUE XSC00760
NOLO=.TRUE. XSC00770
DO 9 I=1,3 XSC00780
READ <IOIN,500> <DAT<J>,J=1,5> XSC00790
IF <DAT<1>.EQ.TYPE<1>> GO TO 1 XSC00800
IF <DAT<1>.EQ.TYPE<2>> GO TO 2 XSC00810
IF <DAT<1>.EQ.TYPE<3>> GO TO 3 XSC00820
IF <DAT<1>.EQ.TYPE<4>> GO TO 4 XSC00830
IF <DAT<1>.EQ.TYPE<5>> GO TO 5 XSC00840
IF <DAT<1>.EQ.TYPE<6>> GO TO 6 XSC00850
C ERROR CHECK XSC00860
GO TO 7 XSC00870
C ADVERSE WEATHER INDICATOR AND OPTIONAL RAIN RATE: XSC00880
1 IFOG=IFIX<DAT<2>> XSC00890
RHRT=DAT<3> XSC00900
GO TO 9 XSC00910
C HORIZONTAL DISTANCE FOR HORIZONTAL PATH CALC. XSC00920
2 HORDIS=DAT<2> XSC00930
ISLANT=0 XSC00940
GO TO 9 XSC00950
C HORIZONTAL DISTANCE AND ANGLE FOR SLANT PATH CALC. XSC00960
3 HORDIS=DAT<2> XSC00970
ANG=DAT<3> XSC00980
ISLANT=1 XSC00990
GO TO 9 XSC01000
C SLANT DISTANCE AND ANGLE FOR SLANT PATH CALC. XSC01010
4 SLTDIS=DAT<2> XSC01020
ANG=DAT<3> XSC01030
ISLANT=1 XSC01040
GO TO 9 XSC01050
C SET PLOT FLAG XSC01060
5 NPLT=1 XSC01070
CONTINUE XSC01080
6 CONTINUE XSC01090
IF <NOLO> WRITE <IOOUT,600> XSC01100
IF <IGEOSW.NE.1>GO TO 88 XSC01110
HORDIS=SQR<<PTS<1>-PTS<4>>**2+<PTS<2>-PTS<5>>**2> XSC01120
SLTDIS=SQR<HORDIS**2+<PTS<3>-PTS<6>>**2> XSC01130
ANG=ACOS<HORDIS/SLTDIS>/PIRAD XSC01140
88 CONTINUE XSC01150
C WAVELENGTH ERROR CHECK XSC01160
IF <<WAVE.GT..4.AND.WAVE.LE.2.>.OR.<WAVE.GE.3..AND.WAVE.LE.5.> XSC01170
1 .OR.<WAVE.GE.8.AND.WAVE.LE.12.05>> GO TO 10 XSC01180
WRITE <IOOUT,1600> WAVE XSC01190
IERR=1 XSC01200
XSTRN=1. XSC01210
RETURN XSC01220
10 CONTINUE XSC01230
IF <NOLO.AND.IFOG.EQ.1> WRITE <IOOUT,800> XSC01240
IF <NOLO.AND.IFOG.EQ.2> WRITE <IOOUT,900> XSC01250
IF <NOLO.AND.IFOG.EQ.3> WRITE <IOOUT,950> XSC01260
IF <NOLO.AND.IFOG.EQ.4> WRITE <IOOUT,1000> XSC01270
IF <NOLO.AND.IFOG.EQ.5> WRITE <IOOUT,1100> XSC01280
IF <ISLANT.GT.0> GO TO 11 XSC01290
IF <NOLO> WRITE <IOOUT,1200> XSC01300
11 IF <NOLO.AND.ISLANT.GE.1> WRITE <IOOUT,1400> WAVE XSC01310
IF <ISLANT.GE.1.AND.<IFOG.LE.0.OR.IFOG.GE.4>> WRITE <IOOUT,2100> XSC01320
IF <ISLANT.GE.1.AND.<IFOG.LE.0.OR.IFOG.GE.4>> IFOG=1 XSC01330
EXTN=EXTSS

```

```

12 IF (ISLANT.GE.1.AND.WAVE.GE.8) GO TO 101
   IF (ISLANT.GE.1) CALL SLANT<EXTN,HORDIS,SLTDIS,ANG,AVEX55,IERR,
1 WAVE,NPLT>
   IF (IERR.EQ.1) XSTRN =1.
   IF (IERR.EQ.1) RETURN
   EXTN=EXT55
   IF (ISLANT.GE.1) EXTN=AVEX55
   XSPATH=HORDIS
   IF (ISLANT.GE.1) XSPATH=SLTDIS
   IF (ISLANT.EQ.0) GO TO 100
C SLANT PATH EXTINCTION
  XSTRN =EXP<-XSPATH*EXTN>
  IF (NOLO) WRITE (IOOUT,1500) WAVE,EXTN,XSPATH,XSTRN,ANG
  RETURN
100 IF (IFOG.NE.4) GO TO 101
C RAIN - ALL WAVELENGTHS
  IF (NOLO.AND.RNRT.LE.0.0) WRITE (IOOUT,550)
  IF (RNRT.LE.0.0) RNRT=1.
  RNEXTN=A0+A1*RNRT+A2*RNRT**2
  XSTRN =EXP<-XSPATH*RNEXTN>
  IF (NOLO) WRITE (IOOUT,1550) RNRT,RNEXTN,XSPATH,XSTRN
  RETURN
101 IF (ABS(WAVE-1.06).LT.01) GO TO 400
   IF (WAVE.GE.3.AND.WAVE.LE.5) GO TO 200
   IF (WAVE.GE.8.AND.WAVE.LE.12) GO TO 300
   XSTRN=EXP<-XSPATH*EXTN>
   RETURN
200 CONTINUE
C 3.0 TO 5.0 RANGE
C MA
  IF (IFOG.EQ.1) EXT35=10.**(<math>0.0345+1.03*\text{ALOG}_{10}</math><EXTN>)
C MP
  IF (IFOG.EQ.2) EXT35=10.**(<math>-0.38+1.32*\text{ALOG}_{10}</math><EXTN>)
C CP
  IF (IFOG.EQ.3) EXT35=10.**(<math>-0.82+1.58*\text{ALOG}_{10}</math><EXTN>)
  IF (IFOG.NE.5) EXTN=EXT35
  IF (ISLANT.GE.1) GO TO 12
C SNOW
  IF (IFOG.EQ.5) EXT35=10.0**(<math>1.05*\text{ALOG}_{10}</math><EXTN>+.021)
  XSTRN =EXP<-XSPATH*EXT35>
  IF (NOLO) WRITE (IOOUT,1700) EXT35,XSPATH,XSTRN
  RETURN
300 CONTINUE
C 8.0 TO 12.0 RANGE
C MA
  IF (IFOG.EQ.1) EXT812=10.**(<math>-.45+1.19*\text{ALOG}_{10}</math><EXTN>)
C MP
  IF (IFOG.EQ.2) EXT812=10.**(<math>-1.01+1.51*\text{ALOG}_{10}</math><EXTN>)
C CP
  IF (IFOG.EQ.3) EXT812=10.**(<math>-1.65+1.82*\text{ALOG}_{10}</math><EXTN>)
  IF (IFOG.NE.5) EXTN=EXT812
  IF (ISLANT.GE.1) GO TO 12
C SNOW
  IF (IFOG.EQ.5) EXT812=10.0**(<math>.993*\text{ALOG}_{10}</math><EXTN>+.114)
  XSTRN =EXP<-XSPATH*EXT812>
  IF (NOLO) WRITE (IOOUT,1800) EXT812,XSPATH,XSTRN
  RETURN
400 CONTINUE
C 1.06 RANGE
C MA, MP, AND CP
  EXT106=AMIN1<math>10.**(-0.14+1.16*\text{ALOG}_{10}</math><EXTN>),EXTN)
  EXTN=EXT106
  IF (ISLANT.GE.1) GO TO 12
C SNOW - ASSUME THAT THE EXTINCTION AT 1.06 IS THE SAME AS AT .55 UMX
  IF (IFOG.EQ.5) EXT106=EXT55
  XSTRN =EXP<-XSPATH*EXT106>
  IF (NOLO) WRITE (IOOUT,1900) EXT106,XSPATH,XSTRN
  RETURN
7 WRITE (IOOUT,2000) <DAT<J>,J=1,4)
  XSTRN=1.

```

```

XSC01280
XSC01290
XSC01300
XSC01310
XSC01320
XSC01330
XSC01340
XSC01350
XSC01360
XSC01370
XSC01380
XSC01390
XSC01400
XSC01410
XSC01420
XSC01430
XSC01440
XSC01450
XSC01460
XSC01470
XSC01480
XSC01490
XSC01500
XSC01510
XSC01520
XSC01530
XSC01540
XSC01550
XSC01560
XSC01570
XSC01580
XSC01590
XSC01600
XSC01610
XSC01620
XSC01630
XSC01640
XSC01650
XSC01660
XSC01670
XSC01680
XSC01690
XSC01700
XSC01710
XSC01720
XSC01730
XSC01740
XSC01750
XSC01760
XSC01770
XSC01780
XSC01790
XSC01800
XSC01810
XSC01870
XSC01880
XSC01890
XSC01900
XSC01910
XSC01920
XSC01930

```


	IERR=1		XSC01940
	RETURN		XSC01950
550	FORMAT (A4,6X,5(F6.2,1X))		XSC01960
550	FORMAT (1H0,50HXSCALE WARNING - RAIN RATE IS .LE. 0.0, RAIN RATE ,		XSC01970
	+20HSET EQUAL TO 1 MM/HR, /)		XSC01980
600	FORMAT (1H, /, /, /, 1X, 50X, 14HOPTIONS CHOSEN)		XSC02000
800	FORMAT (1H, 50X, 14HMARITIME ARTIC)		XSC02010
900	FORMAT (1H, 50X, 14HMARITIME POLAR)		XSC02020
950	FORMAT (1H, 50X, 17HCONTINENTAL POLAR)		XSC02030
1000	FORMAT (1H, 50X, 4HRRAIN)		XSC02040
1100	FORMAT (1H, 50X, 4HSNOW)		XSC02050
1200	FORMAT (1H, 50X, 15HHORIZONTAL PATH/)		XSC02060
1400	FORMAT (1H, 50X, 15HSLANT PATH FOR ,F8.3,8H MICRONS/)		XSC02070
1500	FORMAT (1X, 23X, 20HSLANT EXTINCTION AT ,F5.2,8H MICRONS,		XSC02080
	1 4X, 11HDISTANCE , 12HTRANSMISSION, 6X, 5HANGLE/, 1X, 40X,		XSC02090
	2 26HKM**-1 KM, /		XSC02100
	3 , 1X, 38X, F8.3, 12X, F8.3, 5X, E9.3, 7X, F7.2)		XSC02110
1550	FORMAT (1H, 20X, 27HEXTINCTION FOR RAIN RATE OF, F5.2, 6H MM/HR,		XSC02120
	1 5X, 11HDISTANCE , 12HTRANSMISSION/, 1X, 40X,		XSC02130
	2 26HKM**-1 KM, /		XSC02140
	3 , 1X, 38X, F8.3, 12X, F8.3, 5X, E9.3)		XSC02150
1600	FORMAT (1H, 18H***** WAVELENGTH (, F9.3, 10H) OUTSIDE ,		XSC02160
	1 10HALLOWABLE , 29HRANGE (1.06, 3.0-5.0, 8.0-12.0		XSC02170
	2 31HMICRONS) - CONTROL RETURNED TO ,		XSC02180
	3 17HMAIN FROM XSCALE.)		XSC02190
1700	FORMAT (1H, 25X, 37HEXTINCTION FROM 3.0 TO 5.0 MICRONS		XSC02200
	1 11HDISTANCE , 12HTRANSMISSION/, 1X, 40X,		XSC02210
	2 26HKM**-1 KM, /		XSC02220
	3 , 1X, 38X, F8.3, 13X, F8.3, 5X, E9.3)		XSC02230
1800	FORMAT (1H, 25X, 37HEXTINCTION FROM 8.0 TO 12.0 MICRONS		XSC02240
	1 12H DISTANCE , 12HTRANSMISSION/, 1X, 40X,		XSC02250
	2 26HKM**-1 KM, /		XSC02260
	3 , 1X, 38X, F8.3, 12X, F8.3, 5X, E9.3)		XSC02270
1900	FORMAT (1H, 25X, 37HEXTINCTION AT 1.06 MICRONS		XSC02280
	1 11HDISTANCE , 12HTRANSMISSION/, 1X, 40X,		XSC02290
	2 26HKM**-1 KM, /		XSC02300
	3 , 1X, 38X, F8.3, 12X, F8.3, 5X, E9.3)		XSC02310
2000	FORMAT (1H, 44HUNKNOWN CARD TYPE. CONTROL RETURNED TO MAIN,		XSC02320
	+13H FROM XSCALE, /, 1X, A4, 6X, 5(F6.2, 1X))		XSC02330
2100	FORMAT (1H, 40HINCORRECT FOG TYPE FOR SUBROUTINE SLANT, /, 1X,		
	1 21HFOG TYPE CHANGED TO 1/)		
	END		XSC02340

```

SUBROUTINE SLANT<EXT55,HORDIS,SLTDIS,ANG,AVEX55,IERR,WAVE,NPLT> SLNT0010
COMMON /IOUNIT,IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSLNT0020
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK SLNT0030
C ALL QUANTITIES IN THIS ROUTINE ARE FOR .55 UM SLNT0040
REAL KMTOM SLNT0050
C KILOMETERS TO METERS SLNT0060
KMTOM=1000. SLNT0070
IERR=0 SLNT0080
TOL=.0001 SLNT0090
100 IF <HORDIS,LT,TOL,OR,ANG,LT,TOL> GO TO 200 SLNT0100
C HORIZONTAL DISTANCE AND ANGLE INPUT SLNT0110
VERDIS=HORDIS*TAN<ANG*PIRAD> SLNT0120
SLTDIS=SQRT<HORDIS**2+VERDIS**2> SLNT0130
GO TO 300 SLNT0140
200 IF <SLTDIS,LT,TOL,OR,ANG,LT,TOL> GO TO 500 SLNT0150
C SLANT DISTANCE AND ANGLE INPUT SLNT0160
VERDIS=SLTDIS*SIN<ANG*PIRAD> SLNT0170
HORDIS=SQRT<SLTDIS**2-VERDIS**2> SLNT0180
C CONVERT TO 20 METER INCREMENTS SLNT0190
300 VERDIS=FLOAT<IFIX<<VERDIS+TOL>*KMTOM/20.>>*20./KMTOM SLNT0200
C LIMIT ON VERTICAL HEIGHT IS 500 METERS SLNT0210
IF <VERDIS*KMTOM,GT,500.> VERDIS=.5 SLNT0220
SD=SQRT<HORDIS**2+VERDIS**2> SLNT0230
IF <SD/SLTDIS,GT,1.01,OR,SD/SLTDIS,LT,.99>WRITE<IOOUT,700>SLTDIS,SD SLNT0240
SLTDIS=SD SLNT0250
C FIND NBR OF 20 METER INCREMENTS SLNT0260
ITOP=IFIX<<VERDIS+TOL>*KMTOM/20.> SLNT0270
IF <ITOP,LT,1> ITOP=1
VERDIS=FLOAT<ITOP>*20./KMTOM
EXTN=EXT55 SLNT0280
C BEGIN TRAPEZODIAL INTEGRATION FOR TAU < OPTICAL DEPTH> SLNT0290
TAU=EXTN/2, SLNT0300
ALT=0.0
NPTS=ITOP+1
IF <NPLT,EQ,1> WRITE <NPLOTU,352> NPTS
352 FORMAT <I5>
IF <NPLT,EQ,1> WRITE <NPLOTU,351> EXTN,ALT,WAVE
DO 400 I=1,ITOP SLNT0310
C THESE FORMULAS ARE GOOD ONLY IN 20M INCREMENTS SLNT0320
IF <EXTN,GE,7.0,AND,WAVE,LT,2.0> EXTN= SLNT0330
1 10.**<0.55+0.72*ALOG10<EXTN>> SLNT0340
IF <EXTN,LT,7.0,AND,WAVE,LT,2.0> EXTN= SLNT0350
1 10.**<0.1+1.25*ALOG10<EXTN>> SLNT0360
IF <EXTN,GE,3.3,AND,<WAVE,GE,3.0,AND,WAVE,LT,5.0>> EXTN= SLNT0370
1 10.**<0.55+.72*ALOG10<EXTN>> SLNT0380
IF <EXTN,LT,3.3,AND,<WAVE,GE,3.0,AND,WAVE,LT,5.0>> EXTN= SLNT0390
1 10.**<0.3+1.2*ALOG10<EXTN>> SLNT0400
IF <EXTN,GE,1.7,AND,<WAVE,GE,8.0,AND,WAVE,LT,12.0>> EXTN= SLNT0410
1 10.**<0.5+0.75*ALOG10<EXTN>> SLNT0420
IF <EXTN,LT,1.7,AND,<WAVE,GE,8.0,AND,WAVE,LT,12.0>> EXTN= SLNT0430
1 10.**<0.4+1.2*ALOG10<EXTN>> SLNT0440
ALT=FLOAT<I>*20,
IF <NPLT,EQ,1> WRITE <NPLOTU,351> EXTN,ALT
351 FORMAT <3<E10.4,1X>>
400 TAU=TAU+EXTN SLNT0450
C FINISH TRAP INTEGRATION SLNT0460
TAU=<TAU-EXTN/2.>*.02 SLNT0470
C FIND AVERAGE EXTINCTION VALUE FOR SLANT PATH. SLNT0480
AVEX55=TAU/VERDIS SLNT0490
RETURN SLNT0500
500 WRITE <IOOUT,600> SLNT0510
IERR=1 SLNT0520
RETURN SLNT0530
C SLNT0540
600 FORMAT <1X,38HERROR - IMPROPER INPUT FOR SUBROUTINE SLNT0550
1 34HSLANT; TRANSMISSION SET EQUAL TO 1> SLNT0560
700 FORMAT <1H,18HWARNING FROM SLANT,/,1X,22HTHE VERTICAL DISTANCE, SLNT0570
+38HEXCEEDS THE 500 METER UPPER LIMIT, OR,/,1X,10HIS NOT AN SLNT0580
+29HINTEGER MULTIPLE OF 20 METERS,/,1X,28HSLANT DISTANCE CHANGED FR SLNT0590
+0M,/,F7.4,4H TO,/,F7.4,3H KM/> SLNT0600

```

END

SLNT 0610

SUBROUTINE TURB(WAVE, IERR)
 CALCULATES TURBULENCE INDUCED POINTING JITTER AND POWER SPECTRUM
 FOR LASER TARGET DESIGNATOR AND TERMINAL HOMING SEEKER

TUP00010
 TUR00020
 TUR00030
 TUR00040
 TUR00050
 TUR00060
 TUR00070
 TUR00080
 TUR00090
 TUR00100
 TUR00110
 TUR00120
 TUR00130
 TUR00140
 TUR00150
 TUR00160
 TUR00170
 TUR00180
 TUR00190
 TUR00200
 TUR00210
 TUR00220
 TUR00230
 TUR00240
 TUR00250
 TUR00260
 TUR00270
 TUR00280
 TUR00290
 TUR00300
 TUR00310
 TUR00320
 TUR00330
 TUR00340
 TUR00350
 TUR00360
 TUR00370
 TUR00380
 TUR00390
 TUR00400
 TUR00410
 TUR00420
 TUR00430
 TUR00440
 TUR00450
 TUR00460
 TUR00470
 TUR00480
 TUR00490
 TUR00500
 TUR00510
 TUR00520
 TUR00530
 TUR00540
 TUR00550
 TUR00560
 TUR00570
 TUR00580
 TUR00590
 TUR00600
 TUR00610
 TUR00620
 TUR00630
 TUR00640
 TUR00650
 TUR00660
 TUR00670
 TUR00680
 TUR00690
 TUR00700

 CALCULATION FOR THE DESIGNATOR PATH ARE PERFORMED EACH
 TIME THIS ROUTINE IS REFERENCED. THE CALCULATIONS FOR THE
 SEEKER PATHS ARE PERFORMED ONLY WHEN THE DATA CARDS DVRV,
 CN2, OR V2 ARE INCLUDED IN THE INPUT SET.
 THE INPUT IS CARD ORDER INDEPENDENT, WITH THE SINGLE
 RESTRICTION THAT THE 'GO' CARD MUST BE THE LAST CARD
 OF THE DATA SET.
 THE DATA IS COMPLETELY IDENTIFIED BY THE ID IN COLUMNS
 1-4 OF EACH CARD, FOLLOWED BY UP TO 7<REAL> FIELDS AS
 NEEDED, WITH 10 COLUMNS PER FIELD BEGINNING IN COL. 11.
 COMMENTS BELOW.

THE INPUT FORMAT IS A4,6X,7(E10.4)

THE INPUT OF A CN1, CN2, V1 AND V2 TYPE DATA CARD IS
 TERMINATED WHEN THE FIRST COEFFICIENT OF VALUE ZERO
 IS ENCOUNTERED. THE REMAINING DATA COEFFICIENTS ON
 THE CARD, IF ANY, ARE IGNORED.

 THE FOLLOWING ARE REQUIRED RECORDS FOR AT THE FIRST INPUT SET.

IDENT.	VARS.	DESCRIPTION
PARM	DIAM	LASER TARGET DESIGNATOR APERTURE DIAMETER IN METERS
	THEI	LASER BEAMSPREAD ANGLE IN RADIANS
	TDOT	LASER BEAM SLUE RATE IN RADIANS/SECOND
	RANG	DISTANCE FROM DESIGNATOR TO TARGET IN METERS
	TIME	DURATION OF CALCULATIONS IN SECONDS
	M	NO. OF FREQUENCIES FOR WHICH POINTING JITTER POWER SPECTRUM IS TO BE CALCULATED, IF M=0, THEN DEFAULT TO M=512.
		***RANG AND R1V ARE RECALCULATED IF IGEOSW=1
		***N1 AND N2 ARE CALCULATED WITHIN THE ROUTINE.
	N1	NO OF SEGMENTS IN DESIGNATOR PATH
	N2	NO. OF SEGMENTS IN SEEKER TO TARGET PATH
		N1 AND N2 ARE SET EQUAL TO THE INDEX OF THE LAST NON-ZERO COEFFICIENT READ INTO CN1 AND CN2 RESPECTIVELY.
CN1	IR2	STARTING INDEX VALUE OF CN1(I)
	(CN1(I), I=IR2,IR2+5)	
	CN1(I)	VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT WITH ONE VALUE FOR EACH SEGMENT OF RANGE FROM LASER DESIGNATOR TO TARGET (METERS**(-2/3))
V1	IR2	STARTING INDEX VALUE OF V1(I)
	(V1(I), I=IR2,IR2+5)	
	V1(I)	SET OF VALUES OF CROSSWIND VELOCITY CORRESPONDING TO EACH SEGMENT OF RANGE FROM LASER DESIGNATOR TO TARGET (M/SEC)
DVRV	D1V	DIAMETER OF SEEKER APERTURE IN METERS.
	R1V	SEEKER RANGE TO TARGET IN METERS.
CN2	IR2	STARTING INDEX VALUE OF CN2(I)

```

(CN2(I), I=IR2,IR2+5)
CN2(I) VALUES OF REFRACTIVE INDEX STRUCTURE
CONSTANT FOR EACH SEGMENT OF RANGE FROM TARGET TO
SEEKER (METERS**(-2/3))
TUR00710
TUR00720
TUR00730
TUR00740
TUR00750
TUR00760
TUR00770
TUR00780
TUR00790
TUR00800
TUR00810
TUR00820
TUR00830
TUR00840
TUR00850
TUR00860
TUR00870
TUR00880
TUR00890
TUR00900
TUR00910
TUR00920
TUR00930
TUR00940
TUR00950
TUR00960
TUR00970
TUR00980
TUR00990
TUR01000
TUR01010
TUR01020
TUR01030
TUR01040
TUR01050
TUR01060
TUR01070
TUR01080
TUR01090
TUR01100
TUR01110
TUR01120
TUR01130
TUR01140
TUR01150
TUR01160
TUR01170
TUR01180
TUR01190
TUR01200
TUR01210
TUR01220
TUR01230
TUR01240
TUR01250
TUR01260
TUR01270
TUR01280
TUR01290
TUR01300
TUR01310
TUR01320
TUR01330
TUR01340
TUR01350
TUR01360
TUR01370
TUR01380
TUR01390
TUR01400

V2
IR2 STARTING INDEX VALUE OF V2(I)
V2(I) VALUES OF CROSSWIND VELOCITY FOR EACH SEGMENT
OF RANGE FROM TARGET TO SEEKER (M/SEC).
NPPS NONE PRINTS TABULAR VALUES OF POWER SPECTRUM VS FREQUENCY
NPPJ NONE PRINTS TABULAR VALUES OF POINTING JITTER VS TIME
NPAL NONE PRINTS TABULAR VALUES OF BOTH POWER SPECTRUM (PS)
AND POINTING JITTER (PJ).
**IF THIS CARD IS MISSING NO TABLES WILL BE PRINTED (DEFAULT VALUE).
THE FOLLOWING IDENT RECORD IS ALWAYS REQUIRED.
GO SIGNIFIES TO BEGIN EXECUTION FOR THIS DATA SET.
AFTER EXECUTION, ANOTHER SET OF INPUTS MAY BE
READ-IN FOLLOWED BY ANOTHER 'GO' CARD.
ANY VALUES ESTABLISHED FROM PREVIOUS INPUT SETS
TO THE ROUTINE ARE STILL IN EFFECT. THUS DATA
SUCH AS FROM CARD PARM NEED NOT BE READ AGAIN IF
THERE ARE TO BE NO CHANGES IN THE DATA ASSOCIATED
WITH THAT IDENTIFIER.
*****
++ CALLED PROGRAMS ++
DESUB
FALPH
FFT4
GAUSS
MEANVR
RAND
SPECT
SPREAD
THETO
*****
COMPLEX RAN
REAL LAMB
REAL INR(7),IR1,LABEL(11)
LOGICAL SETUP
COMMON /CONST/PI,PI2,PIRAD,TWOP1,TURRMB,CDEGK
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
THIS IS A COMPLEX NUMBER EACH NUMBER TAKES TWO WORDS
COMMON /MOS/RAN(2048)
COMMON /MO1/FR(1025),CN1(20),V1(20),FD(20),RD(20)
COMMON /LOWEX/PS(1025),V2(20),RR(10)
DIMENSION CN2(20),PJCHAR(4),PSCHAR(4)
COMMON /GEOMET/PTS(15),IGEOSW
EXTERNAL DESUB,FALPH
DATA SETUP/.TRUE./
DATA PJCHAR,PSCHAR /4HPJ(,4HRAD*,4H*2/S,4HEC),4HPS(,4HRAD*,
4H*2/H,4HZ)
1 DATA LABEL/4HGO,4H,4HPARM,4HCN1,4HV1,4HDVRY,4HCN2,
1 4HV2,4HNPPS,4HNPPJ,4HNPAL/
NPRINT=0
LAMB=WAVE
C SET THE SEED FOR THE RANDOM NUMBER GENERATOR
C*** NOTE, THIS SEED IS APPROPRIATE FOR THE RANDOM NUMBER GENERATOR
C*** USED AT THE ATMOSPHERIC SCIENCES LAB. USERS AT OTHER
C*** INSTALLATIONS WILL NEED TO SUPPLY THEIR OWN RANDOM NUMBER
C*** GENERATOR.
IF (.NOT.SETUP) GO TO 100
VARX=735.34829
VARX=RAND(VARX)
SETUP=.FALSE.

```

	IOPT=1	TUR01410
100	READ(IOIN,3000) IR1,(INR(I), I=1,7)	TUR01420
	IR2=IFIX(INR(1))	TUR01430
	IF(IR1.EQ.LABEL(1)) GO TO 180	TUR01440
	IF(IR1.EQ.LABEL(3)) GO TO 110	TUR01450
	IF(IR1.EQ.LABEL(4)) GO TO 120	TUR01460
	IF(IR1.EQ.LABEL(5)) GO TO 130	TUR01470
	IF(IR1.EQ.LABEL(6)) GO TO 140	TUR01480
	IF(IR1.EQ.LABEL(7)) GO TO 150	TUR01490
	IF(IR1.EQ.LABEL(8)) GO TO 160	TUR01500
C	PRINTING OPTIONS	TUR01510
	IF(IR1.EQ.LABEL(9)) NPRINT=1	TUR01520
	IF(IR1.EQ.LABEL(10)) NPRINT=2	TUR01530
	IF(IR1.EQ.LABEL(11)) NPRINT=3	TUR01540
	IF(NPRINT.EQ.1.OR.NPRINT.EQ.2.OR.NPRINT.EQ.3) GO TO 100	TUR01550
	WRITE(IOOUT,3001) IR1,(INR(I), I=1,7)	TUR01560
	GOTO 100	TUR01570
C		TUR01580
110	DIAM = INR(1)	TUR01590
	THET = INR(2)	TUR01600
	TDOT = INR(3)	TUR01610
	RANG = INR(4)	TUR01620
	TIME = INR(5)	TUR01630
	M = IFIX(INR(6))	TUR01640
	IF(M.EQ.0) M = 512	TUR01650
	GOTO 100	TUR01660
C		TUR01670
120	DO 125 I=2,7	TUR01680
	IF(INR(I).NE.0.0) GOTO 121	TUR01690
	N1=IR2-1	TUR01700
	GOTO 100	TUR01710
121	CN1(IR2)=INR(I)	TUR01720
	IR2=IR2+1	TUR01730
	IF(IR2.GT.20) GOTO 126	TUR01740
125	CONTINUE	TUR01750
126	IR1=IR2-1	TUR01760
	N1=MAX0(N1,IFIX(IR1))	TUR01770
	GOTO 100	TUR01780
C		TUR01790
130	DO 131 I=2,7	TUR01800
	V1(IR2)=INR(I)	TUR01810
	IR2=IR2+1	TUR01820
	IF(IR2.GT.20) GOTO 100	TUR01830
131	CONTINUE	TUR01840
	GOTO 100	TUR01850
C		TUR01860
140	IOPT = 2	TUR01870
	O1V = INR(1)	TUR01880
	R1V = INR(2)	TUR01890
	GOTO 100	TUR01900
C		TUR01910
150	IOPT = 2	TUR01920
	DO 155 I=2,7	TUR01930
	IF(INR(I).NE.0.0) GOTO 151	TUR01940
	N2=IR2-1	TUR01950
	GOTO 100	TUR01960
151	CN2(IR2) = INR(I)	TUR01970
	IR2=IR2+1	TUR01980
	IF(IR2.GT.20) GOTO 156	TUR01990
155	CONTINUE	TUR02000
156	IR1=IR2-1	TUR02010
	N2=MAX0(N2,IFIX(IR1))	TUR02020
	GOTO 100	TUR02030
C		TUR02040
160	IOPT = 2	TUR02050
	DO 161 I=2,7	TUR02060
	V2(IR2) = INR(I)	TUR02070
	IR2=IR2+1	TUR02080
	IF(IR2.GT.20) GOTO 100	TUR02090
161	CONTINUE	TUR02100

	GOTO 100	TUR02140
C	180 IF<IGEQSW,NE.1>GO TO 190	TUR02140
	RANG=SQRT<<PTS<7>-PTS<1>>**2+<PTS<8>-PTS<2>>**2+	TUR02140
	+<PTS<9>-PTS<3>>**2>	TUR02150
	DISKTM=1000.0	TUR02160
	RANG=RANG*DISKTM	TUR02170
	R1V=SQRT<<PTS<4>-PTS<1>>**2+<PTS<5>-PTS<2>>**2+<PTS<6>-PTS<3>>**2>	TUR02180
	R1V=R1V*DISKTM	TUR02190
	190 CONTINUE	TUR02200
	IF <LAMB,LE.14.> GO TO 200	TUR02210
	WRITE <IOOUT,2500> LAMB	TUR02220
	IERR=1	TUR02230
	RETURN	TUR02240
	200 WRITE <IOOUT,3100>	TUR02250
	IF <IOPT,EG.2> WRITE <IOOUT,3200>	TUR02260
	WRITE <IOOUT,3300> LAMB,DIAM,THET	TUR02270
C	CHANGE WAVELENGTH TO METERS	TUR02280
	LAMB=LAMB/1.0E+6	TUR02290
	IF <IOPT,EG.2> WRITE <IOOUT,3400> D1V,R1V	TUR02300
	WRITE <IOOUT,3500> TDOT,RANG	TUR02310
	WRITE <IOOUT,3600> TIME	TUR02320
	WRITE <IOOUT,3700> N1	TUR02330
	IF <IOPT,EG.2> WRITE <IOOUT,3800> N2	TUR02340
	WRITE <IOOUT,3900> M	TUR02350
	WRITE <IOOUT,4000>	TUR02360
	WRITE <IOOUT,4200>	TUR02370
	DO 400 I=1,N1	TUR02380
	WRITE <IOOUT,4300> I,CN1<I>,V1<I>	TUR02390
	400 CONTINUE	TUR02400
	IF <IOPT,EG.1> GO TO 600	TUR02410
	WRITE <IOOUT,4100>	TUR02420
	WRITE <IOOUT,4200>	TUR02430
	DO 500 I=1,N2	TUR02440
	WRITE <IOOUT,4300> I,CN2<I>,V2<I>	TUR02450
	500 CONTINUE	TUR02460
C	COMPUTATION OF TIME, FREQUENCY AND SPATIAL INCREMENTS	TUR02470
	600 DELT=TIME/M	TUR02480
	DELF=1./TIME	TUR02490
	DELZ=RANG/FLOAT<N1>	TUR02500
	IF <N2,NE.0> DEL1V=R1V/FLOAT<N2>	TUR02510
	MM=M+M	TUR02520
	M1=M+1	TUR02530
	MM1=MM+1	TUR02540
	MM2=MM+2	TUR02550
	MSG=SQRT<FLOAT<MM>>	TUR02560
	DO 700 I=2,M1	TUR02570
	FR<I>=<I-1>*DELF	TUR02580
	700 CONTINUE	TUR02590
	R2=DIAM/THET	TUR02600
	R=RANG+R2	TUR02610
	DT=THET*RANG	TUR02620
	D2=DIAM+DT	TUR02630
C	COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND	TUR02640
C	NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM LASER	TUR02650
C	DESIGNATOR TO TARGET	TUR02660
	ZI=0.0	TUR02670
	Z1=DELZ/2.	TUR02680
	ROT=0.0	TUR02690
	DEL=0.0	TUR02700
	DO 800 I=1,N1	TUR02710
	ZI=ZI+Z1	TUR02720
	Z1=DELZ	TUR02730
	VEI=V1<I>+TDOT*(ZI-R2)	TUR02740
	RO<I>=16.71*DELZ*CN1<I>*(1.-ZI/RANG)**1.66667/(<LAMB*LAMB>)	TUR02750
	FO<I>=VEI/(<PI*D2*ZI/R>)	TUR02760
	ROT=ROT+RO<I>	TUR02770
	DEL=DEL+DELZ*CN1<I>*(RANG-ZI)/RANG	TUR02780
	RO<I>=RO<I>*(.6)	TUR02790
	800 CONTINUE	TUR02800

```

ROT=ROT**(-.6)
CALL SPREAD(DIAM,ROT,LAMB,THET,RANG,DRO,THETDL,DDL,THET12,D12,
1 DTHET,D22)
WRITE (IOOUT,4500) R2,R
WRITE (IOOUT,4600) ROT,DRO,THET,DT,THETDL,DDL,THET12,D12,DTHET,D22
WRITE (IOOUT,4700)
DO 1000 I=1,N1
1000 WRITE (IOOUT,4800) I,RO(I),FO(I)
C COMPUTATION OF ANGLE OF ARRIVAL POWER SPECTRUM OF LASER DESIGNATOR
F2=0.
PS(I)=0.
DO 1100 J=2,M1
F=FR(J)
F1=0.
CALL SPECT(F,D2,N1,F1,LAMB)
PS(J)=F1
IF (IOPT.EQ.1) PS(J)=F1*(D2/DIAM)**2
F2=F2+PS(J)*DEL
1100 CONTINUE
IF (IOPT.EQ.1) WRITE (IOOUT,4400) F2
F2SQRT=SQRT(F2)
AJITT=F2SQRT*RANG
IF (IOPT.EQ.1) WRITE (IOOUT,4900) F2SQRT,AJITT
IF (IOPT.EQ.1) GO TO 1500
C
DTV=THET*R1V
C COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND
C NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM
C TARGET TO SEEKER
ZI=0.
Z1=DELIV/2.
ROT=0.0
DO 1200 I=1,N2
ZI=ZI+Z1
Z1=DELIV
VEI=V2(I)+TDOT*(R1V-ZI)
RO(I)=16.71*DELIV*CN2(I)*(ZI/R1V)**1.66667/(LAMB*LAMB)
ROT=ROT+RO(I)
RO(I)=RO(I)**(-.6)
FO(I)=VEI/(PI*DIV*ZI/R1V)
1200 CONTINUE
ROT=ROT**(-.6)
CALL SPREAD(D22,ROT,LAMB,THET,R1V,DRO,THETDL,DDL,THET12,D12,DTHET,
1 D22V)
WRITE (IOOUT,4550)
WRITE (IOOUT,4600) ROT,DRO,THET,DTV,THETDL,DDL,THET12,D12,DTHET,
1 D22V
WRITE (IOOUT,4700)
DO 1300 I=1,N2
1300 WRITE (IOOUT,4800) I,RO(I),FO(I)
C COMPUTATION OF TURBULENCE INDUCED POINTING JITTER POWER SPECTRUM
C FROM TARGET SPOT TO LASER SEEKER, COMPUTATION OF TOTAL POWER
C SPECTRUM FROM LASER DESIGNATOR TO SEEKER AND POWER SPECTRUM VARIANCE
CALL THETG(THETAO,FALPH,CN2,DIV,R1V,N2)
F2=0.
DO 1400 J=2,M1
F=FR(J)
F1=0.
CALL SPECT(F,DIV,N2,F1,LAMB)
PS(J)=PS(J)+F1/(1+(D2/(R1V*THETAO))**2)
PS(J)=PS(J)*(D2/DIV)**2
F2=F2+PS(J)*DEL
1400 CONTINUE
WRITE (IOOUT,4400) F2
1500 IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5000) DELF
FREQ=0.0
FINC=DELF*10.0
IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5600)
IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PSCHAR
K=1
TUR02810
TUR02820
TUR02830
TUR02840
TUR02850
TUR02860
TUR02870
TUR02880
TUR02890
TUR02900
TUR02910
TUR02920
TUR02930
TUR02940
TUR02950
TUR02960
TUR02970
TUR02980
TUR02990
TUR03000
TUR03010
TUR03020
TUR03030
TUR03040
TUR03050
TUR03060
TUR03070
TUR03080
TUR03090
TUR03100
TUR03110
TUR03120
TUR03130
TUR03140
TUR03150
TUR03160
TUR03170
TUR03180
TUR03190
TUR03200
TUR03210
TUR03220
TUR03230
TUR03240
TUR03250
TUR03260
TUR03270
TUR03280
TUR03290
TUR03300
TUR03310
TUR03320
TUR03330
TUR03340
TUR03350
TUR03360
TUR03370
TUR03380
TUR03390
TUR03400
TUR03410
TUR03420
TUR03430
TUR03440
TUR03450
TUR03460
TUR03470
TUR03480
TUR03490
TUR03500

```



```

1600 L=10
IF (L.GT.M1) L=M1
IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,2800) FREQ,(PS(J),J=K,L)
FREQ=FREQ+FINC
K=L+1
L=K+9
IF (K.LE.M1) GO TO 1600
DO 2400 L=1,IOPT
IF (L.EQ.1) WRITE (IOOUT,5100)
IF (L.EQ.2) WRITE (IOOUT,5200)
C GENERATION OF RANDOM SEQUENCE HAVING SAME POWER SPECTRUM VARIANCE
C AS INDUCED BY TURBULANCE. ADD SYMMETRIC TERMS FOR NEGATIVE
C FREQUENCIES. COMPUTE MEAN AND VARIANCE OF RANDOM ARRAY
RAN(I)=(0.,0.)
DO 1700 I=2,M1
MMM=MM2-I
C GENERATE RANDOM NUMBER WITH ROUTINE GAUSS
NORMAL DISTRIBUTION
MEAN = 0.0
STANDARD DEVIATION = 1.0
REAL PART = RANDOM NUMBER
IMAG PART = 0
RAN(I)=CMPLX(GAUSS(12,0.0,1.0),0.0)*SQRT(PS(I)/DELT)
RAN(MMM)=RAN(I)
1700 CONTINUE
C CALCULATE POWER OF 2 ,NPOW, FOR FFT4 SINCE ARRAY PASSED TO
C FFT4 MUST HAVE SIZE THAT IS A POWER OF 2. (NOTE LN(2)=0.693147.)
NPOW=IFIX(ALOG(FLOAT(MM)))/0.693147)
NMAX=2**NPOW
IF (MM.EQ.NMAX) GO TO 1900
C IF MM IS NOT A POWER OF 2 THEN RESET THE REST OF ARRAY RAN
NPOW=NPOW+1
ISTART=MM+1
NMAX=2**NPOW
DO 1800 I=ISTART,NMAX
RAN(I)=(0.0,0.0)
1800 CONTINUE
C COMPUTE AND WRITE MEAN AND VARIANCE OF RANDOM ARRAY
1900 WRITE (IOOUT,5300)
CALL MEANVR(1,M)
C FAST FOURIER TRANSFORM RANDOM ARRAY
C CDC ROUTINE CALL
CALL FFT(RAN,MM,+1)
CALL FFT4(1.0,RAN,NPOW,NMAX)
DO 2100 I=1,MM
RAN(I)=RAN(I)/MSG
2100 CONTINUE
C COMPUTE AND WRITE MEAN AND VARIANCE OF TIME SEQUENCE.
WRITE (IOOUT,5500)
CALL MEANVR(M1,MM)
C WRITE TRANSFORMED ARRAY VALUES CORRESPONDING TO TIME VALUES
C OF POINTING JITTER FOR ONE DIRECTION.
IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5400) DELT
DTIME=DELT
TINC=DELT*10.0
IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5700)
IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PJCHAR
DO 2300 I1=M1,MM,10
DO 2200 I2=1,10
I3=I1+I2-1
RR(I2)=REAL(RAN(I3))
2200 CONTINUE
IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,2800) DTIME,
1 (RR(I),I=1,10)
DTIME=DTIME+TINC
2300 CONTINUE
2400 CONTINUE
C
2500 FORMAT (1X,100(1H*),/,13H WAVELENGTH (<,F10.3,7H) GREAT,

```

```

TUR03510
TUR03515
TUR03520
TUR03530
TUR03540
TUR03550
TUR03560
TUR03570
TUR03580
TUR03590
TUR03600
TUR03610
TUR03620
TUR03630
TUR03640
TUR03650
TUR03660
TUR03670
TUR03680
TUR03690
TUR03700
TUR03710
TUR03720
TUR03730
TUR03740
TUR03750
TUR03760
TUR03770
TUR03780
TUR03790
TUR03800
TUR03810
TUR03820
TUR03830
TUR03840
TUR03850
TUR03860
TUR03870
TUR03880
TUR03890
TUR03900
TUR03910
TUR03920
TUR03930
TUR03940
TUR03950
TUR03960
TUR03970
TUR03980
TUR03990
TUR04000
TUR04010
TUR04020
TUR04030
TUR04040
TUR04050
TUR04060
TUR04070
TUR04080
TUR04090
TUR04100
TUR04110
TUR04120
TUR04130
TUR04140
TUR04150
TUR04160
TUR04170
TUR04180
TUR04190
TUR04200

```

```

1          40HER THAN 14 MICRONS: CONTROL RETURNED TO , TUR04210
2          21HMAIN FROM TURBULENCE. ,/,1X,100(1H*)) TUR04220
2700 FORMAT (5F16.6) TUR04230
2800 FORMAT (E10.4,10E12.4) TUR04240
3000 FORMAT (A4,6X,7E10.4) TUR04250
3001 FORMAT(1H0,20X,45HTHE FOLLOWING ID-FIELD NOT RECOGNIZED BY TURB,/, TUR04260
1 1X,A4,6X,7E10.4) TUR04270
3100 FORMAT (1H1,36H CALCULATION OF POWER SPECTRUM AND TUR04280
1 11HTURBULENCE 29HINDUCED POINTING JITTER OF A TUR04290
2 23HLASER TARGET DESIGNATOR) TUR04300
3200 FORMAT (1H+,101X,10HAND SEEKER) TUR04310
3300 FORMAT (1H0,41X,30HLASER WAVELENGTH (MICROMETERS),8X, TUR04320
1 F10.4,/,42X,26HDESIG. APERTURE DIAMETER ( TUR04330
2 7HMETERS),5X,F10.6,/,42X,18HBEAMSPREAD ANGLE ( TUR04340
3 8HRADIANS),12X,F10.6) TUR04350
3400 FORMAT (1H0,41X,33HSEEKER APERTURE DIAMETER (METERS),5X, TUR04360
1 F10.6,/,42X,36HRANGE FROM TARGET (METERS), TUR04370
2 2X,F10.2) TUR04380
3500 FORMAT (1H0,41X,24HBEAM SLUE RATE (RAD/SEC),14X,F10.6, TUR04390
1 ,/,42X,26HDESIGNATION RANGE (METERS),12X, TUR04400
2 F10.2) TUR04410
3600 FORMAT (1H0,41X,26HDURATION OF TEST (SECONDS),12X, TUR04420
1 F10.4) TUR04430
3700 FORMAT (1H0,41X,30HTOTAL DESIGNATOR PATH SEGMENTS,15X, TUR04440
1 I3) TUR04450
3800 FORMAT (1H0,41X,26HTOTAL SEEKER PATH SEGMENTS,19X,I3) TUR04460
3900 FORMAT (1H0,41X,27HTOTAL FREQUENCIES FOR WHICH,/,42X, TUR04470
1 22H POWER SPECTRUM IS TO,/,42X, TUR04480
2 15H BE CALCULATED,27X,I4) TUR04490
4000 FORMAT (1H0,/,26X,29H VALUES OF REFRACTIVE INDEX TUR04500
1 37HSTRUCTURE CONSTANT AND WIND SPEED IN TUR04510
2 15HDESIGNATOR PATH) TUR04520
4100 FORMAT (1H0,/,26X,29H VALUES OF REFRACTIVE INDEX TUR04530
1 37HSTRUCTURE CONSTANT AND WIND SPEED IN TUR04540
2 11HSEEKER PATH) TUR04550
4200 FORMAT (1H0,62X,5HCN**2,9X,9HWINDSPEED,16X,/,42X, TUR04560
1 11HSEGMENT NO.,5X,15H(METER**(-2/3)),3X, TUR04570
2 11H(METER/SEC),/,) TUR04580
4300 FORMAT (1H ,46X,I2,10X,E12.6,5X,F10.2) TUR04590
4400 FORMAT (1H0,/,36H THE VARIANCE OF THE POWER SPECTRUM TUR04600
1 3HIS ,E12.4) TUR04610
4500 FORMAT (1H1,/,56X,20HDESIGNATOR TO TARGET,/,38X, TUR04620
1 ,/,24X,41HVIRTUAL POINT SOURCE TO APERTURE DISTANCE, TUR04630
2 28X,F10.5,9H (METERS),/,24X,22HDISTANCE FROM VIRTUAL TUR04640
3 22HPOINT SOURCE TO TARGET,25X,F10.5,9H (METERS)) TUR04650
4550 FORMAT (1H1,/,58X,16HTARGET TO SEEKER,/,) TUR04660
4600 FORMAT (1H0,23X,27HINTEGRATED COHERENCE LENGTH,42X,F10.6, TUR04670
1 9H (METERS),/,24X,36HDIAMETER/INTEGRATED COHERENCE LENGTHTUR04680
2 33X,F10.6,/,24X,31HTRANSMITTER-INDUCED BEAM SPREAD TUR04690
3 13X,E12.5,10H (RADIANS),3X,F10.6,9H (METERS), TUR04700
4 ,/,24X,31HDIFFRACTION-LIMITED BEAM SPREAD,13X,E12.5, TUR04710
5 10H (RADIANS),3X,F10.6,9H (METERS),/,24X, TUR04720
6 38HDIFFRACTION AND TURBULENCE BEAM SPREAD,6X,E12.5, TUR04730
7 10H (RADIANS),3X,F10.6,9H (METERS),/,24X,6HTOTAL TUR04740
8 19HEFFECTIVE BEAM SIZE,19X,E12.5,10H (RADIANS),3X,F10.6, TUR04750
9 9H (METERS),/,) TUR04760
4700 FORMAT (1H0,34X,11HSEGMENT NO.,9X,16HCOHERENCE LENGTH, TUR04770
1 5X,27H REFERENCE FREQUENCY(HERTZ),/,) TUR04780
4800 FORMAT (1H ,38X,I2,10X,F16.6,10X,F16.6) TUR04790
4900 FORMAT (1H0,19H RMS SPOT JITTER = ,E10.4,10H RAD, OR = TUR04800
1 ,E10.4,7H METERS,/) TUR04810
5000 FORMAT (1H1,1X,40X,31H CALCULATED POWER SPECTRUM VS. TUR04820
1 9HFREQUENCY,/,47X,16H AT INTERVALS OF,F6.3,6H HERTZ) TUR04830
5100 FORMAT (1H0,37H OUTPUT FOR DESIGNATOR TO TARGET PATH) TUR04840
5200 FORMAT (1H0,33H OUTPUT FOR TARGET TO SEEKER PATH) TUR04850
5300 FORMAT (1H0,35X,35H MEAN AND VARIANCE OF RANDOM ARRAY TUR04860
1 ,/,) TUR04870
5400 FORMAT (1H1,49X,34HVALUES OF POINTING JITTER VS. TIME / TUR04880
1 52X,16H AT INTERVALS OF,F8.4,4H SEC) TUR04890
5500 FORMAT (1H0,/,35X,28H MEAN AND VARIANCE OF TIME TUR04900

```

```
      8HSEQUENCE//>
5600 FORMAT (1H ,10H BEGINNING,/,10H FREQ (HZ))
5700 FORMAT (1H ,2X,5H TIME,/,2X,6H (SEC))
5800 FORMAT (1H+,T13,46(1H-),1H ,4A4,55(1H-),/)
      RETURN
      END
```

```
TUR04910
TUR04920
TUR04930
TUR04940
TUR04950
TUR04960
```

```

C  FUNCTION DESUB(X,DRO)
    FUNCTION USED FOR INTEGRATION  F(X)
    FACTR=3.44*(DRO*X)**1.666667*(1.-X**0.333333)
    IF(FACTR.LT.160.) GO TO 10
    DESUB=0.
    GO TO 20
10  ARCCX=ATAN2(SQRT(1.-X**2),X)
    DESUB=X*((ARCCX-X*(1.-X**2)**.5)*EXP(-FACTR))
20  RETURN
    END

```

```

DES00010
DES00020
DES00030
DES00040
DES00050
DES00060
DES00070
DES00080
DES00090
DES00100

```

```

FUNCTION FALPH(XI1)
IF (XI1.GE..5623) GO TO 100
FALPH=10.66*((XI1)**2)
GO TO 900
100 IF (XI1.GE.1.0) GO TO 200
FALPH=4.025*XI1-.00659
GO TO 900
200 IF (XI1.GE.1.778) GO TO 300
FALPH=1.8547*XI1+2.164
GO TO 900
300 IF (XI1.GE.3.162) GO TO 400
FALPH=.8475*XI1+3.955
GO TO 900
400 IF (XI1.GE.5.623) GO TO 500
FALPH=.391*XI1+5.3977
GO TO 900
500 IF (XI1.GE.10.) GO TO 600
FALPH=.1814*XI1+6.578
GO TO 900
600 IF (XI1.GT.31.62) GO TO 700
FALPH=.0534*XI1+7.95
GO TO 900
700 IF (XI1.GT.1000.) GO TO 800
FALPH=7.8*(XI1**.06)
GO TO 900
800 FALPH=11.97
900 RETURN
END

```

```

FAL00010
FAL00020
FAL00030
FAL00040
FAL00050
FAL00060
FAL00070
FAL00080
FAL00090
FAL00100
FAL00110
FAL00120
FAL00130
FAL00140
FAL00150
FAL00160
FAL00170
FAL00180
FAL00190
FAL00200
FAL00210
FAL00220
FAL00230
FAL00240
FAL00250
FAL00260
FAL00270
FAL00280

```

```

SUBROUTINE FFT4(SIGN,X,NPOW,NMAX)
COOLEY-TUKEY METHOD OF FOURIER TRANSFORM
INCLUDES SINE COSINE COMPUTATION AND
REARRANGING DATA ACCORDING TO REVERSE BIT ADDRESSES

SIGN = FOURIER DIRECTION TRANSFORM FLAG
      -1. FOR DIRECT TRANSFORM, TO COEFFICIENTS FROM SERIES
      1. FOR INVERSE TRANSFORM, TO SERIES FROM COEFFICIENTS

X      = LOC. OF FOURIER TRANSFORM BLOCK

NPOW = POWER OF 2 TO OBTAIN NMAX

NMAX = LENGTH OF BLOCK X

COMPLEX X,CXCS,HOLD,XA
DIMENSION CS(2),MSK(13)
DIMENSION X(1)
EQUIVALENCE (CXCS,CS)
ZZ=6.283185306*SIGN/FLOAT(NMAX)
MSK(1)=NMAX/2
DO 100 I=2,NPOW
100 MSK(I)=MSK(I-1)/2
CONTINUE
NN=NMAX
MM=2
C LOOP OVER NPOW LAYERS
DO 800 LAYER=1,NPOW
NN=NN/2
NW=0
DO 700 I=1,MM,2
II=NN*I
C CXCS = CEXP(2*PI*NW*SIGN/NMAX)
W=FLOAT(NW)*ZZ
CS(1)=COS(W)
CS(2)=SIN(W)
C COMPUTE ELEMENTS FOR BOTH HALFS OF EACH BLOCK
DO 200 J=1,NN
II=II+1
IJ=II-NN
XA=CXCS*X(IJ)
X(IJ)=X(IJ)-XA
X(IJ)=X(IJ)+XA
200 CONTINUE
C BUMP UP SERIES BY 2
C COMPUTE REVERSE ADDRESS
DO 400 LOC=2,NPOW
LL=NW-MSK(LOC)
IF (LL) 500,600,300
300 NW=LL
400 CONTINUE
500 NW=MSK(LOC)+NW
GO TO 700
600 NW=MSK(LOC+1)
700 CONTINUE
MM=MM*2
800 CONTINUE
C DO FINAL REARRANGEMENT
NW=0
DO 1600 I=1,NMAX
NW1=NW+1
HOLD=X(NW1)
IF (NW1-I) 1100,1000,900
900 X(NW1)=X(I)
1000 X(I)=HOLD
C BUMP UP SERIES BY 1
C COMPUTE REVERSE ADDRESS
1100 DO 1300 LOC=1,NPOW
LL=NW-MSK(LOC)

```

```

FFT00010
FFT00020
FFT00030
FFT00040
FFT00050
FFT00060
FFT00070
FFT00080
FFT00090
FFT00100
FFT00110
FFT00120
FFT00130
FFT00140
FFT00150
FFT00160
FFT00170
FFT00180
FFT00190
FFT00200
FFT00210
FFT00220
FFT00230
FFT00240
FFT00250
FFT00260
FFT00270
FFT00280
FFT00290
FFT00300
FFT00310
FFT00320
FFT00330
FFT00340
FFT00350
FFT00360
FFT00370
FFT00380
FFT00390
FFT00400
FFT00410
FFT00420
FFT00430
FFT00440
FFT00450
FFT00460
FFT00470
FFT00480
FFT00490
FFT00500
FFT00510
FFT00520
FFT00530
FFT00540
FFT00550
FFT00560
FFT00570
FFT00580
FFT00590
FFT00600
FFT00610
FFT00620
FFT00630
FFT00640
FFT00650
FFT00660
FFT00670
FFT00680
FFT00690
FFT00700

```

```
IF (LL) 1400,1500,1200
1200 NW=LL
1300 CONTINUE
1400 NW=MSK(LOC)+NW
GO TO 1600
1500 NW=MSK(LOC+1)
1600 CONTINUE
IF (SIGN) 1900,1900,1700
1700 PTS=NMAX
DO 1800 I=1,NMAX
X(I)=X(I)/PTS
1800 CONTINUE
1900 RETURN
END
```

```
FFT00710
FFT00720
FFT00730
FFT00740
FFT00750
FFT00760
FFT00770
FFT00780
FFT00790
FFT00800
FFT00810
FFT00820
FFT00830
FFT00840
```

```

C      FUNCTION GAUSS(N,XBAR,SIGMA)
C      GENERATE RANDOM NUMBERS WITH NORMAL DISTRIBUTION
C      MEAN = XBAR
C      STANDARD DEVIATION = SIGMA
C      DATA NN /0/
C      IF (NN.GT.0) GO TO 1
C      NN=1
C      C=0.
1      CONTINUE
C      X=0.0
C      IF (C.EQ.0.0) C=735.34829
C      DO 100 J=1,N
C      C=RAND(C)
100     X=X+C
C      CONTINUE
C      XN=N
C      X=SQRT(12.0/XN)*(X-0.5*XN)
C      GAUSS=SIGMA*X+XBAR
C      RETURN
C      END

```

```

GAUC0010
GAU00030
GAU00040
GAU00050
GAU00060
GAU00070
GAU00080
GAU00090
GAU00100
GAU00110
GAU00120
GAU00130
GAU00140
GAU00150
GAU00160
GAU00170
GAU00180
GAU00190
GAU00200
GAUG0210

```



```

SUBROUTINE MEANVR(N1,N2)
COMPUTE AND WRITE MEAN AND VARIANCE OF COMPLEX ARRAY
OVER SOME RANGE OF THE ARRAY.
  ++ INPUT ++
N1 = STARTING INDEX OF RANGE
N2 = ENDING INDEX OF RANGE
  ++ COMMON ++
RAN = COMPLEX ARRAY CONTAINING DATA
  ++ OUTPUT ++
MEAN1 = MEAN OF REAL PART
MEAN2 = MEAN OF IMAGINARY PART
VAR1 = VARIANCE OF REAL PART
VAR2 = VARIANCE OF IMAGINARY PART

  COMPLEX RAN
  REAL MEAN1,MEAN2
  THIS IS A COMPLEX NUMBER EACH NUMBER TAKES TWO WORDS
  COMMON /M05/RAN(2048)
  COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUM
  MEAN1=0.
  MEAN2=0.
  VAR1=0.
  VAR2=0.
  DO 100 I=N1,N2
    MEAN1=REAL(RAN(I))+MEAN1
    MEAN2=AIMAG(RAN(I))+MEAN2
    VAR1=(REAL(RAN(I)))**2+VAR1
    VAR2=(AIMAG(RAN(I)))**2+VAR2
100 CONTINUE
  MEAN1=MEAN1/FLOAT(N2-N1)
  MEAN2=MEAN2/FLOAT(N2-N1)
  VAR1=VAR1/FLOAT(N2-N1)
  VAR2=VAR2/FLOAT(N2-N1)
  WRITE (IOOUT,200) MEAN1,MEAN2
  WRITE (IOOUT,300) VAR1,VAR2
  RETURN
200 FORMAT (35X,20H MEAN OF REAL PART =,E12.5,10H, MEAN OF
1 11HIMAG PART =,E12.5)
300 FORMAT (35X,20H VAR. OF REAL PART =,E12.5,10H, VAR. OF
1 11HIMAG PART =,E12.5)
END

```

```

MEA00010
MEA00020
MEA00030
MEA00040
MEA00050
MEA00060
MEA00070
MEA00080
MEA00090
MEA00100
MEA00110
MEA00120
MEA00130
MEA00140
MEA00150
MEA00160
MEA00170
MEA00180
MEA00190
MEA00200
MEA00210
MEA00220
MEA00230
MEA00240
MEA00250
MEA00260
MEA00270
MEA00280
MEA00290
MEA00300
MEA00310
MEA00320
MEA00330
MEA00340
MEA00350
MEA00360
MEA00370
MEA00380
MEA00390
MEA00400
MEA00410
MEA00420
MEA00430
MEA00440
MEA00450
MEA00460
MEA00470
MEA00480
MEA00490
MEA00500
MEA00510
MEA00520
MEA00530

```

	SUBROUTINE SPECT(F,D2,N,F1,LAMB)	SPE00010
C	THIS ROUTINE GENERATES THE APPROXIMATE FUNCTION	SPE00020
C	G<ALPHA> (F/F<SUB 0,1>)	SPE00030
C	USED IN THE POWER SPECTRUM OF ANGLE-OF-ARRIVAL EQUATION	SPE00040
	REAL LAMB	SPE00050
	COMMON /MO1/FR(1025),CN(20),V1(20),FO(20),RO(20)	SPE00060
	FACT=1.32E-2*(LAMB/D2)**2	SPE00070
	F1=0.	SPE00080
	DO 100 I=1,N	SPE00090
	IF (F.LE..332*FO(I)) G=1.	SPE00100
	IF (F.GT..332*FO(I)) G=1.12-.361*F/FO(I)	SPE00110
	IF (F.GE.3.10*FO(I)) G=0.	SPE00120
	F1=F1+FACT*((D2/RO(I))**5/(F*F*FO(I)))**.33333*G	SPE00130
100	CONTINUE	SPE00140
	RETURN	SPE00150
	END	SPE00160

	1	SUBROUTINE SPREAD(DIAM,ROT,WAVE,THET,RANG,DRO,THETDL,DDL,THET12, D12,DTHET,DTOT)	SPR00010
			SPR00020
C		COMPUTATION OF BEAM SPREAD ANGLE DUE TO DIFFRACTION AND DIFFRACTION	SPR00030
C		AND TURBULENCE AND SPOT DIAMETER ON TARGET (OR SEEKER).	SPR00040
			SPR00050
		DRO=DIAM/ROT	SPR00060
C	00	1/2 SIMPSON RULE INTEGRATION	SPR00070
		VARX=0.0	SPR00080
		DELTA=0.01	SPR00090
		RDRO=DESUB(VARX,DRO)/2.0	SPR00100
		DO 100 I=1,100	SPR00110
		VARX=VARX+DELTA	SPR00120
		RDRO=RDRO+DESUB(VARX,DRO)	SPR00130
100		CONTINUE	SPR00140
		RDRO=(RDRO-DESUB(VARX,DRO)/2.0)*DELTA	SPR00150
		RDRO=1.0/SQRT(5.092958*(DRO)**2*RDRO)	SPR00160
		THETDL=1.128*WAVE/DIAM	SPR00170
		DDL=THETDL*RANG	SPR00180
		THET12=THETDL*DRO*RDRO	SPR00190
		D12=THET12*RANG	SPR00200
		DTHET=SQRT(THET12**2+THET**2)	SPR00210
		DTOT=DIAM+DTHET*RANG	SPR00220
		RETURN	SPR00230
		END	SPR00240
			SPR00250

```

SUBROUTINE THETA(THETA0,FALPH,CN2,D1V,R1V,N2)
C ++ CALLED FUNCTIONS ++
C FALPH
C DIMENSION CN2(20)
COMMON /IOUNIT,IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C DEL1V=R1V/FLOAT(N2)
C CALCULATE D1INF
D1INF=0.
S1=DEL1V/2.
S=0.
D1V3=(D1V)*(-.3333)
DO 100 I=1,N2
S=S+S1
D1INF=DEL1V*CN2(I)*((S/R1V)**1.6667)+D1INF
S1=DEL1V
100 CONTINUE
D1INF=0.5*11.97*D1V3*D1INF
C INITIAL ESTIMATE FOR THETA0
C WRITE (IOOUT,500) D1INF
THETA0=1.E-4
200 XI1=0.
D1THE=0.
S=0.
S1=DEL1V/2.
DO 300 I=1,N2
S=S+S1
XI1=THETA0*(R1V-S)/D1V
D1THE=DEL1V*CN2(I)*((S/R1V)**1.6667)*FALPH(XI1)+D1THE
S1=DEL1V
300 CONTINUE
D1THE=D1THE*D1V3
IF (ABS((D1INF-D1THE)/D1INF).LT..001) GO TO 400
THETA0=THETA0*(1+.5*(D1INF-D1THE)/D1INF)
C WRITE (IOOUT,500) THETA0,D1THE
GO TO 200
400 CONTINUE
C WRITE (IOOUT,500) D1THE
C RETURN
C 500 FORMAT (2E16.8)
END
THE00010
THE00020
THE00030
THE00040
THE00050
THE00060
THE00070
THE00080
THE00090
THE00100
THE00110
THE00120
THE00130
THE00140
THE00150
THE00160
THE00170
THE00180
THE00190
THE00200
THE00210
THE00220
THE00230
THE00240
THE00250
THE00260
THE00270
THE00280
THE00290
THE00300
THE00310
THE00320
THE00330
THE00340
THE00350
THE00360
THE00370
THE00380
THE00390
THE00400
THE00410

```

SUBROUTINE BASCAT(WAVE,EXCO,IERR)

 THIS VERSION OF BASCAT (20 SEP 81) DIFFERS FROM THE EOSAEL 80
 VERSION IN ITS INTERNAL PROGRAM STRUCTURE AND OUTPUT CAPABILITY.
 INPUT FORMATTING HAS NOT BEEN CHANGED. BRIEFLY, THE INTERNAL
 STRUCTURAL CHANGES CONSIST OF THE FOLLOWING :

- (A) SUBROUTINE THIT HAS BEEN ELIMINATED. THE FUNCTIONS WHICH
 IT ONCE PERFORMED HAVE BEEN CONSOLIDATED INTO SUBROUTINE
 START.
- (B) A NEW LIDAR BIASING ALGORITHM HAS BEEN INSERTED.
- (C) DIRECT BEAM (I.E., UNSCATTERED) COMPUTATIONS IN
 SUBROUTINE START HAVE BEEN REVISED.
- (D) SUBROUTINE CONV HAS BEEN MODIFIED SO THAT THE DIFFER-
 ENCE OF TWO NUMBERS RETAINS MORE SIGNIFICANT DIGITS.
- (E) NUMERICAL CHECKS FOR IMPROPER ARGUMENTS OF FUNCTIONS
 (DIVISIONS, SQUARE ROOTS, LOGARITHMS, ETC.) HAVE BEEN
 REVISED.
- (F) ARGUMENT LISTS OF A FEW COMMON BLOCKS HAVE BEEN CHANGED.
- (G) WRITE STATEMENTS FOR OUTPUTTING BASCAT RESULTS TO A USER-
 DEFINED PLOT FILE (NPLOTU) HAVE BEEN INCLUDED. THESE STATE-
 MENTS MUST BE UNCOMMENTED IN ORDER TO ACTIVATE THEM.

 THE BASCAT MODULE ALONE USES THE FOLLOWING SUBROUTINES:

- BKWD - CONTAINS BACKWARD SCATTERING ALGORITHM
- CONV - CONVOLVES IMPULSE RESPONSE WITH SQUARE PULSE
- ELM - DETERMINES BIASING DISTANCES
- FIND - DETERMINES INTERPOLATED PHASE FUNCTION VALUE
- FWRD - FIRST ORDER SCATTERING ALGORITHM
- GAS - DETERMINES MONTE CARLO SCATTERING ANGLES FOR TRAVERSES
- GMAX - DETERMINES MAXIMUM OF AN INPUT ARRAY
- MATRX - GENERATES ROTATION MATRICES
- ROTAT - ROTATES VECTORS FROM ONE COORDINATE SYSTEM TO ANOTHER
- SMOOZ - DETERMINES START OF TRAILING ZEROS IN INPUT ARRAY
- START - INITIATES PHOTON TRAJECTORIES
- TRAVRS - MOVES PHOTONS BETWEEN SCATTERING POINTS AND FINDS OB-
 SERVED POWER CONTRIBUTIONS AT THOSE POINTS
- USCA - SELECTS RANDOM ANGLES WEIGHTED BY PHASE FUNCTION

TWO SUBROUTINES SHARED BY BASCAT WITH OTHER EOSAEL 80 MODULES
 ARE THE FOLLOWING :

- PFUNC - SELECTS AND RENORMALIZES PHASE FUNCTION DATA FROM
 EOSAEL 80 DATA BASE
- RAND - RANDOM NUMBER GENERATOR (GENERATES UNIFORM DISTRIBUTION
 OF RANDOM NUMBERS BETWEEN 0 AND 1)

*** NOTE** THE FOLLOWING ROUTINES UTILIZE THE RANDOM NUMBER GENERATOR
 WHICH IS INVOKED AS FUNCTION 'RAND(SEED)' :

- (A) BKWD - 3 OCCURRENCES OF FUNCTION RAND
- (B) FWRD - 2 OCCURRENCES
- (C) GAS - 3 OCCURRENCES
- (D) START - 2 OCCURRENCES
- (E) TRAVRS - 2 OCCURRENCES

USERS OF OTHER (NON-MP) COMPUTER SYSTEMS MUST REPLACE
 FUNCTION RAND(SEED) WITH A UNIFORM RANDOM NUMBER GENERATOR
 WHICH WORKS FOR THEIR SYSTEMS. THE RANDOM NUMBER SEED IS
 INITIALIZED IN SUBROUTINE BASCAT. THIS AND ALL SUBSEQUENT
 VALUES OF THE RANDOM NUMBER SEED ARE PASSED VIA COMMON
 BLOCK 'RNDM'. SHOULD THE RANDOM NUMBER SEED USED HERE
 (735.34829) BE INAPPROPRIATE FOR THE USER'S SYSTEM, IT IS
 SUGGESTED THAT THE INITIALIZATION USED BELOW (SEED0=
 735.34829) BE CHANGED TO AN APPROPRIATE VALUE.

SUBSTANTIAL MODIFICATIONS HAVE BEEN MADE TO THE FOLLOWING
 SUBROUTINES PRESENT IN EOSAEL 80 :
 CONV

BAS0000
 BAS0001
 BAS0002
 BAS0003
 BAS0004
 BAS0005
 BAS0006
 BAS0007
 BAS0008
 BAS0009
 BAS0010
 BAS0011
 BAS0012
 BAS0013
 BAS0014
 BAS0015
 BAS0016
 BAS0017
 BAS0018
 BAS0019
 BAS0020
 BAS0021
 BAS0022
 BAS0023
 BAS0024
 BAS0025
 BAS0026
 BAS0027
 BAS0028
 BAS0029
 BAS0030
 BAS0031
 BAS0032
 BAS0033
 BAS0034
 BAS0035
 BAS0036
 BAS0037
 BAS0038
 BAS0039
 BAS0040
 BAS0041
 BAS0042
 BAS0043
 BAS0044
 BAS0045
 BAS0046
 BAS0047
 BAS0048
 BAS0049
 BAS0050
 BAS0051
 BAS0052
 BAS0053
 BAS0054
 BAS0055
 BAS0056
 BAS0057
 BAS0058
 BAS0059
 BAS0060
 BAS0061
 BAS0062
 BAS0063
 BAS0064
 BAS0065
 BAS0066
 BAS0067
 BAS0068
 BAS0069
 BAS0070

```

C          TRAVRS
C          START
C*****
COMMON/RNDM/SEED
COMMON /GEOMET/PTS(15), IGEOSW
COMMON/ALL/AT, BT, CT, BLIM
COMMON/IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLINT, KSTOR, NPLOTJ
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
COMMON /MOS/V1, C1, S1, SA(3), EN(10,100), ENC(10), ELMIN, DELD, DTOT,
+NTMAX, NSCAM, KMAX, LMAX, LMM1
COMMON /BASPOT/U(65), SUM(65), WV(16), NWVL, ALBED(16), BS(16),
+BE(16), SINGWV, PF(65), LLMAX
COMMON/FAHT/SD(3), UV, GAMMA, ELD(2), STH, FAC, ALIM, THV, TAU, EL(2),
*ALB(2), ZG, DMAX
COMMON/FGEL/XA(3), D, NSCA
COMMON/FWD/AKM, R(3,3), AKSQ, XD(3), ASQ(3), RE(3,3), A(3)
COMMON/CONB/X(100), Y(100)
COMMON/HIT/UDS, THSP, RS(3,3), XS(3), DSA, XV(3)
DIMENSION NM(10), TPU(7), SS(3)
DIMENSION IAL(7), DAT(7), IOR(10)
DATA IAL/2HPA ,2HSQ ,2HDE ,2HCL ,2HGR ,2HPU ,2HGO /
DATA IZERO/0/
C*****
THIS SUBROUTINE CALCULATES STEADY STATE AND TIME-DEPENDENT DIRECT
AND MULTIPLY SCATTERED POWER INTO A DETECTOR BY AN ELLIPSOIDAL
AEROSOL CLOUD WITH GROUND PLANE, FOR A LASER SOURCE. THE DETECTOR
AND SOURCE MAY HAVE ANY LOCATIONS, LOOK ANGLES, AND CONE OF VIEW,
BEAM SPREAD/WAVELENGTH. THE AEROSOL CLOUD MAY HAVE ANY ORIENTA-
TION, SIZE, AND SCATTERING PHASE FUNCTION (ARBITRARY NORMALIZA-
TION), IN A COORDINATE SYSTEM WITH ORIGIN AT THE CLOUD CENTER,
WITH Z-AXIS VERTICAL, X-AXIS EAST, AND Y-AXIS NORTH. THE GROUND
PLANE, ASSUMED AN ISOTROPIC REFLECTOR, MAY HAVE ANY ALBEDO, AND
MAY OR MAY NOT INTERSECT THE AEROSOL CLOUD.
C*****
** INPUT DATA CARDS ARE READ IN AN ORDER-INDEPENDENT MANNER, WITH
** A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH RECORD. DATA
** ON EACH CARD IS READ IN UNDER THE FOLLOWING FORMAT :
** (A4,1X,7(E9.4,1X)). NOTE THAT INTEGER VARIABLES IN THE PROGRAM
** MUST BE INPUT AS REAL NUMBERS IN THIS INPUT SCHEME ... THEY ARE
** LATER FIXED TO THE INTEGER TYPE.
-----
CARD IDENTIFIER : PART
VARIABLES READ : N1,N2,ITIME
N1=NUMBER OF PARTIAL OUTPUTS DESIRED, FOR A GIVEN RUN
N2=NUMBER OF PHOTONS TO BE USED FOR EACH PARTIAL CALCULATION
** NOTE ** FOR CERTAIN DETECTOR CONDITIONS, AND WITH NORMAL BIASING,
AS MANY AS 100,000 PHOTONS MAY BE NEEDED TO OVERCOME LARGE
STATISTICAL FLUCTUATIONS IN FIRST ORDER SCATTERING RETURNS.
SUCH CONDITIONS ARE DEFINED BY THE FOLLOWING CHARACTERISTICS
(A) THE DETECTOR IS IN A MONOSTATIC LIDAR CONFIGURATION.
(B) THE DETECTOR IS WITHIN 10 METERS OF THE CLOUD (OR IS INSIDE
OF IT).
(C) THE CLOUD IS NOT OPTICALLY THICK ALONG THE LOOK DIRECTION
(OPTICAL DEPTHS LESS THAN 10).
*** IN ORDER TO ATTAIN MORE RAPID CONVERGENCE OF FIRST ORDER RETURN
POWER UNDER THE ABOVE CONDITIONS, A DIFFERENT BIASING SCHEME IS
USED. THE SPECIFIC CONDITIONS WHICH TRIGGER THIS ALTERNATE MODE
ARE THE FOLLOWING :
(A) THE DOT PRODUCT OF THE SOURCE APERTURE SURFACE NORMAL AND
THE DETECTOR APERTURE SURFACE NORMAL IS GREATER THAN 0.99.
(B) THE DISTANCE OF THE DETECTOR APERTURE FROM THE NEAREST
CLOUD SURFACE (AS SEEN ALONG THE DETECTOR NORMAL) IS LESS
THAN OR EQUAL TO 10 METERS (0.01 KM).
(C) THE SEPARATION OF SOURCE AND DETECTOR APERTURE CENTERS IS
LESS THAN OR EQUAL TO 10 TIMES THE DETECTOR APERTURE
RADIUS.

```

```

BAS00710
BAS00720
BAS00730
BAS00740
BAS00750
BAS00760
BAS00770
BAS00780
BAS00790
BAS00800
BAS00810
BAS00820
BAS00830
BAS00840
BAS00850
BAS00860
BAS00870
BAS00880
BAS00890
BAS00900
BAS00910
BAS00920
BAS00930
BAS00940
BAS00950
BAS00960
BAS00970
BAS00980
BAS00990
BAS01000
BAS01010
BAS01020
BAS01030
BAS01040
BAS01050
BAS01060
BAS01070
BAS01080
BAS01090
BAS01100
BAS01110
BAS01120
BAS01130
BAS01140
BAS01150
BAS01160
BAS01170
BAS01180
BAS01190
BAS01200
BAS01210
BAS01220
BAS01230
BAS01240
BAS01250
BAS01260
BAS01270
BAS01280
BAS01290
BAS01300
BAS01310
BAS01320
BAS01330
BAS01340
BAS01350
BAS01360
BAS01370
BAS01380
BAS01390
BAS01400

```

```

0*** THE MAIN DIFFERENCES WHICH WILL BE OBSERVED BETWEEN THE NORMAL AND ALTERNATE
0ALTERNATE BIASING MODES ARE THE FOLLOWING :
0(A) THE FIRST ORDER RETURN POWER IN THE EARLIEST TIME BOX WILL BE EXTREMELY STABLE
0IN THE ALTERNATE MODE.
0(B) ONLY THE EARLIEST FIRST ORDER SCATTERING RESULTS WILL BE STRONGLY AFFECTED.
0(C) LATER FIRST ORDER RETURNS AND ALL HIGHER ORDER RETURNS WILL HAVE NEGLIGIBLY
0POORER CONVERGENCE.
0*** IN ORDER TO OBSERVE CONVERGENCE OF THE FIRST ORDER STEADY-STATE RETURN POWER
0TOWARD A STABLE VALUE, IT IS SUGGESTED THAT THE USER UTILIZE THE PARTIAL OUTPUT
0OPTION. AS AN EXAMPLE, IF 10,000 PHOTONS ARE REQUIRED, SET N1=10 AND N2=1,000. THIS
0SELECTION WILL RUN 10 X 1,000 = 10,000 PHOTONS AND WILL OUTPUT RETURN POWER RESULTS
0AFTER EACH BATCH OF 1,000 PHOTONS.
0
0ITIME=OVERALL RUN NUMBER FOR THIS SET OF PARAMETERS. FOR EXAMPLE, ITIME=1 MEANS
0THE FIRST RUN, ITIME=3 MEANS THAT THE RESULTS OF THE TWO PREVIOUS RUNS WILL BE
0COMBINED WITH THIS 3RD RUN.
0-----
0CARD IDENTIFIER : SORC
0VARIABLES READ : XS(1),XS(2),XS(3),THES,PHIS,ASMM
0(XS(K),K=1,3)=SOURCE XYZ COORDINATES(KM)
0THES,PHIS=(POLAR,AZIMUTHAL)ANGLES(DEG) OF SOURCE BEAM AXIS
0ASMM=RADIUS OF SOURCE APERTURE(MM). THE SOURCE BEAM SPREAD ANGLE THSP IS TAKEN
0BY THE PROGRAM AS THE DIFFRACTION LIMIT FOR THIS APERTURE. IF YOU SET ASMM=0,
0THE PROGRAM PUTS THSP=0. IF YOUR SOURCE PHOTONS WOULD NOT INTERSECT THE CLOUD
0OR THE GROUND, THE SUBROUTINE NOTIFIES YOU, AND RETURNS.
0-----
0CARD IDENTIFIER : DETR
0VARIABLES READ : XD(1),XD(2),XD(3),THED,PHID,THEV,ACM
0(XD(K),K=1,3)=DETECTOR XYZ COORDINATES(KM)
0THED,PHID=(POLAR,AZIMUTHAL)ANGLES(DEG) OF DETECTOR LOOK AXIS
0THEV=DETECTOR CONE OF VIEW HALF-ANGLE(DEG)
0ACM=RADIUS OF DETECTOR DISK(CM)
0IF YOUR DETECTOR POINTS SKYWARD, OR IF NEITHER YOUR SOURCE NOR YOUR DETECTOR
0LOOK INTO THE CLOUD, THE SUBROUTINE NOTIFIES YOU, AND RETURNS.
0-----
0CARD IDENTIFIER : CLDS
0VARIABLES READ : A(1),A(2),A(3),THE,PHE,PSE,ISO
0(A(K),K=1,3)=ELLIPSOIDAL CLOUD PRINCIPAL HALF-AXES(KM)
0THE,PHE,PSE=ELLIPSOID EULER ANGLES(DEG), WHERE 'PHE' IS THE FIRST ROTATION,
0ABOUT THE Z-AXIS, 'THE' IS THE NEXT, ABOUT THE NEW Y-AXIS, 'PSE' IS THE LAST,
0ABOUT THE NEW Z-AXIS. ISO=AEROSOL IDENTIFICATION NUMBER, TO COMPARE WITH THE
0'ID' PARAMETER.
0-----
0CARD IDENTIFIER : GRND
0VARIABLES READ : ZG,ALBG
0ZG=Z-COORDINATE OF GROUND PLANE(KM)
0ALBG=GROUND PLANE REFLECTIVITY 0.<ALBG<1. IF YOUR ZG IS SUCH THAT THE GROUND
0PLANE IS ENTIRELY ABOVE THE CLOUD, THE SUBROUTINE INFORMS YOU, AND RETURNS.
0IF YOUR ZG IS NEGATIVE, AND SO LARGE THAT NO GROUND REFLECTIONS WILL RETURN TO
0THE DETECTOR WITHIN THE TIME LIMIT SET BY THE SUBROUTINE, THE SUBROUTINE SETS
0ALBG=0. IF YOUR SOURCE IS BELOW THE GROUND PLANE, THE SUBROUTINE PUTS THE SOURCE
0ON THE GROUND PLANE, AT YOUR XY COORDINATES, AND NOTIFIES YOU.
0-----
0CARD IDENTIFIER : PULS
0VARIABLES READ : TPU(1),TPU(2),...,TPU(7)
0(TPU(J),J=1,7)=SOURCE PULSE DURATIONS(USEC)
0YOU CAN INPUT AS MANY AS SIX DIFFERENT PULSE LENGTHS. THE LAST ENTRY MUST BE
0BLANK(ZERO). THE SUBROUTINE CONVOLUTES THE MONTE CARLO PROBABILITY PER UNIT TIME
0DATA WITH EACH OF THESE SQUARE PULSES, AND, FOR EACH PULSE, WRITES THE TIME-DEPENDENT
0POWER TO THE DETECTOR, FOR UNIT SOURCE PULSE POWER, FOR EACH SIGNIFICANT ORDER
0OF MULTIPLE SCATTERING, AND FOR THE TOTAL OF ALL ORDERS.
0-----

```

```

BAS01410
BAS01420
BAS01430
BAS01440
BAS01450
BAS01460
BAS01470
BAS01480
BAS01490
BAS01500
BAS01510
BAS01520
BAS01530
BAS01540
BAS01550
BAS01560
BAS01570
BAS01580
BAS01590
BAS01600
BAS01610
BAS01620
BAS01630
BAS01640
BAS01650
BAS01660
BAS01670
BAS01680
BAS01690
BAS01700
BAS01710
BAS01720
BAS01730
BAS01740
BAS01750
BAS01760
BAS01770
BAS01780
BAS01790
BAS01800
BAS01810
BAS01820
BAS01830
BAS01840
BAS01850
BAS01860
BAS01870
BAS01880
BAS01890
BAS01900
BAS01910
BAS01920
BAS01930
BAS01940
BAS01950
BAS01960
BAS01970
BAS01980
BAS01990
BAS02000
BAS02010
BAS02020
BAS02030
BAS02040
BAS02050
BAS02060
BAS02070
BAS02080
BAS02090
BAS02100

```

```

C      CARD IDENTIFIER : GO
C      VARIABLES READ : NONE
C      RUN TERMINATION CARD (MUST BE LAST CARD READ).
C-----
C      IERR=0
C*** THE FOLLOWING STATEMENT MAY NEED TO BE CHANGED OR ELIMINATED
C*** FOR USE ON NON-HP1000 COMPUTERS.
C      SEED0=735.34829
C      NDIM=100
C      ND1=NDIM-1
C      DTAUM=0.2
C      GAMMA=EXCO
C      SINGWV=WAVE
C** INPUT DATA INITIALIZATIONS.
C      IF(IZERO.NE.0) GO TO 600
C      N1=0
C      N2=0
C      ITIME=0
C      IS0=0
C      DO 599 LL=1,3
C      XS(LL)=0.
C      XD(LL)=0.
C      A(LL)=0.
C      TPU(LL)=0.
599  TPU(LL+3)=0.
C      TPU(7)=0.
C      THES=0.
C      PHIS=0.
C      ASMM=0.
C      THED=0.
C      PHID=0.
C      THEV=0.
C      ACM=0.
C      THE=0.
C      PHE=0.
C      PSE=0.
C      ZG=0.
C      ALBC=0.
C      LLMAX=65
C      IZERO=1
600  CONTINUE
C*** READ BASCAT DATA SET RECORDS UNDER CARD-INDEPENDENT FORMAT
C
C      DO 700 K=1,7
C      READ(IOIN,610) IA,IA1,(DAT(I),I=1,7)
610  FORMAT(2A2,1X,7(E9.4,1X))
C      DO 615 JJ=1,8
C      IF(IA.NE.IAL(JJ)) GO TO 615
C      INOPT=JJ
C      IF(INOPT.EQ.7) GO TO 701
C      GO TO 620
615  CONTINUE
C      IF((K.EQ.7).AND.(JJ.EQ.8)) GO TO 697
C      GO TO 695
620  CONTINUE
C      GO TO (621,622,623,624,625,626), INOPT
621  N1=IFIX(DAT(1))
C      N2=IFIX(DAT(2))
C      ITIME=IFIX(DAT(3))
C      GO TO 700
622  XS(1)=DAT(1)
C      XS(2)=DAT(2)
C      XS(3)=DAT(3)
C      THES=DAT(4)
C      PHIS=DAT(5)

```

```

BAS02110
BAS02120
BAS02130
BAS02140
BAS02150
BAS02160
BAS02170
BAS02180
BAS02190
BAS02200
BAS02210
BAS02220
BAS02230
BAS02240
BAS02250
BAS02260
BAS02270
BAS02280
BAS02290
BAS02300
BAS02310
BAS02320
BAS02330
BAS02340
BAS02350
BAS02360
BAS02370
BAS02380
BAS02390
BAS02400
BAS02410
BAS02420
BAS02430
BAS02440
BAS02450
BAS02460
BAS02470
BAS02480
BAS02490
BAS02500
BAS02510
BAS02520
BAS02530
BAS02540
BAS02550
BAS02560
BAS02570
BAS02580
BAS02590
BAS02600
BAS02610
BAS02620
BAS02630
BAS02640
BAS02650
BAS02660
BAS02670
BAS02680
BAS02690
BAS02700
BAS02710
BAS02720
BAS02730
BAS02740
BAS02750
BAS02760
BAS02770
BAS02780
BAS02790
BAS02800

```



```

ASMM=DAT(6)
GO TO 700
623 XD(1)=DAT(1)
XD(2)=DAT(2)
XD(3)=DAT(3)
THED=DAT(4)
PHID=DAT(5)
THEV=DAT(6)
ACM=DAT(7)
GO TO 700
624 A(1)=DAT(1)
A(2)=DAT(2)
A(3)=DAT(3)
THE=DAT(4)
PHE=DAT(5)
PSE=DAT(6)
ISO=IFIX(DAT(7))
GO TO 700
625 ZG=DAT(1)
ALBG=DAT(2)
GO TO 700
626 CONTINUE
DO 627 NN=1,7
627 TPU(NN)=DAT(NN)
GO TO 700
C
C*** ERROR RETURNS
C
695 CONTINUE
WRITE(IOOUT,696)
696 FORMAT(1H0,20X,89H***BASCAT ERROR*** INPUT CARD DETECTED WHICH DOES
+S NOT MATCH ANY CORPECT INPUT IDENTIFIERS ^)
IERR=1
GO TO 777
697 CONTINUE
WRITE(IOOUT,698)
698 FORMAT(1H0,20X,66H***BASCAT ERROR*** TOO MANY INPUT CARDS OR GO SE
+NTINEL NOT PRESENT ^)
700 CONTINUE
701 CONTINUE
C
C*** GEOMETRICAL OPTION DATA TRANSFER
C
IF(IGEOSM.NE.1) GO TO 111
DO 110 I=1,3
XS(I)=PTS(I+6)-PTS(I+12)
110 XD(I)=PTS(I+3)-PTS(I+12)
111 CONTINUE
C
C*** GENERATE INTERPOLATED, RENORMALIZED PHASE FUNCTION
C
CALL PFUNC(ISO)
IF(GAMMA.EQ.0.0) GAMMA=BE(1)
ALBEDO=ALBED(1)
REWIND IPHFUN
ALB(1)=ALBEDO
ALB(2)=ALBG
C
C*** DETERMINE POWER OF 2 (KMAX) CORRESPONDING TO NUMBER OF PHASE
C*** FUNCTION VALUES PRESENT.
C
LMAX=LLMAX
LMM1=LMAX-1
KMAX=IFIX(ALOG(FLOAT(LMM1))/.693147)
C
C*** CC = SPEED OF LIGHT (KM/MICROSECOND)
C
CC=0.3
THV=PIRAD*THEV
IF(THV.LE.1.E-30) THV=0.

```

```

BAS02810
BAS02820
BAS02830
BAS02840
BAS02850
BAS02860
BAS02870
BAS02880
BAS02890
BAS02900
BAS02910
BAS02920
BAS02930
BAS02940
BAS02950
BAS02960
BAS02970
BAS02980
BAS02990
BAS03000
BAS03010
BAS03020
BAS03030
BAS03040
BAS03050
BAS03060
BAS03070
BAS03080
BAS03090
BAS03100
BAS03110
BAS03120
BAS03130
BAS03140
BAS03150
BAS03160
BAS03170
BAS03180
BAS03190
BAS03200
BAS03210
BAS03220
BAS03230
BAS03240
BAS03250
BAS03260
BAS03270
BAS03280
BAS03290
BAS03300
BAS03310
BAS03320
BAS03330
BAS03340
BAS03350
BAS03360
BAS03370
BAS03380
BAS03390
BAS03400
BAS03410
BAS03420
BAS03430
BAS03440
BAS03450
BAS03460
BAS03470
BAS03480
BAS03490
BAS03500

```

```

UV=COS(THV)
C
C*** SET RANDOM NUMBER SEED
C
SEED=(2*ITIME-1)*SEED0
AKM=ACM*1.E-5
AKSQ=AKM**2
C
C*** SET MINIMUM DISTANCE ALIM (AND ITS SQUARE BLIM) SEPARATING
C*** THE DETECTOR DISK AND A PHOTON SCATTERING POINT.
C
ALIM=AKM*10.
BLIM=ALIM**2
FAC=THV*AKSQ/8.
BT=.61*WAVE*1.E-3
THSP=0.
C
C*** DETERMINE DIFFRACTION-LIMITED SOURCE BEAMSPREAD
C
IF(ASMM.GT.0.)THSP=BT/ASMM
C
C*** GENERATE ROTATION MATRIX RE( ) FOR CONVERSION FROM STANDARD
C*** FRAME OF REFERENCE TO CLOUD FRAME OF REFERENCE
C
CALL MATRX(THS,PHE,SS,R)
AT=PIRAD*PSE
IF(AT.LE.1.E-30) AT=0.
DO 1 I=1,3
DO 1 J=1,3
1 RS(I,J)=0.
RS(1,1)=COS(AT)
RS(1,2)=SIN(AT)
RS(2,1)=-RS(1,2)
RS(2,2)=RS(1,1)
RS(3,3)=1.
DO 2 I=1,3
DO 2 J=1,3
RE(I,J)=0.
DO 2 K=1,3
2 RE(I,J)=RE(I,J)+RS(I,K)*R(K,J)
C
C*** GENERATE ROTATION MATRIX RS( ) FOR CONVERSION FROM SOURCE
C*** CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE
C
CALL MATRX(THS,PHIS,SS,RS)
C
C*** ECHO INPUT PARAMETERS
C
WRITE (IOOUT,6800)
WRITE (IOOUT,4800)
WRITE (IOOUT,4900)
WRITE (IOOUT,5000)
WRITE (IOOUT,4800)
WRITE (IOOUT,4700)
WRITE (IOOUT,5100)
IF (ISO.EQ.0) WRITE (IOOUT,5200)
IF (ISO.EQ.1) WRITE (IOOUT,5201)
IF (ISO.EQ.2) WRITE (IOOUT,5202)
IF (ISO.EQ.3) WRITE (IOOUT,5203)
IF (ISO.EQ.4) WRITE (IOOUT,5204)
IF (ISO.EQ.5) WRITE (IOOUT,5205)
IF (ISO.EQ.6) WRITE (IOOUT,5206)
IF (ISO.EQ.7) WRITE (IOOUT,5207)
IF (ISO.EQ.8) WRITE (IOOUT,5208)
IF (ISO.EQ.9) WRITE (IOOUT,5209)
IF (ISO.EQ.10) WRITE (IOOUT,5210)
IF (ISO.EQ.11) WRITE (IOOUT,5211)
IF (ISO.EQ.12) WRITE (IOOUT,5212)
WRITE (IOOUT,5600) WAVE,ALBEDO
WRITE (IOOUT,5700) GAMMA

```

```

BAS03510
BAS03520
BAS03530
BAS03540
BAS03550
BAS03560
BAS03570
BAS03580
BAS03590
BAS03600
BAS03610
BAS03620
BAS03630
BAS03640
BAS03650
BAS03660
BAS03670
BAS03680
BAS03690
BAS03700
BAS03710
BAS03720
BAS03730
BAS03740
BAS03750
BAS03760
BAS03770
BAS03780
BAS03790
BAS03800
BAS03810
BAS03820
BAS03830
BAS03840
BAS03850
BAS03860
BAS03870
BAS03880
BAS03890
BAS03900
BAS03910
BAS03920
BAS03930
BAS03940
BAS03950
BAS03960
BAS03970
BAS03980
BAS03990
BAS04000
BAS04010
BAS04020
BAS04030
BAS04040
BAS04050
BAS04060
BAS04070
BAS04080
BAS04090
BAS04100
BAS04110
BAS04120
BAS04130
BAS04140
BAS04150
BAS04160
BAS04170
BAS04180
BAS04190
BAS04200

```

```

WRITE(IOOUT,5601)
WRITE(IOOUT,5602)
WRITE(IOOUT,5603)
WRITE(IOOUT,5701)
WRITE(IOOUT,5702)(XS(K),K=1,3)
WRITE(IOOUT,5704)THES
WRITE(IOOUT,5705)PHIS
WRITE(IOOUT,5703)ASMM
WRITE(IOOUT,5706)THSP
WRITE (IOOUT,5900)
WRITE (IOOUT,6000) THEY
WRITE (IOOUT,6100) ACM
WRITE(IOOUT,6201)(XD(K),K=1,3)
WRITE (IOOUT,6400) THED
WRITE (IOOUT,6500) PHID
WRITE(IOOUT,6501)
WRITE(IOOUT,6502)ZG,ALBG
DO 10 K=1,3
10 ASQ(K)=A(K)**2
DO 12 K=1,3
12 Y(K)=A(K)*2.*GAMMA
C
C*** DETERMINE LARGEST OPTICAL DEPTH PRESENT IN AEROSOL CLOUD
CALL GMAX(3,TAU)
C
C*** SET TIME AND DISTANCE INCREMENTS AND LIMITS
NTMAX=50
D=TAU/GAMMA
DELD=5.*D/NTMAX
DELT=DELD/CC
DMAX=5.5*D
C
C*** BEGIN CLOUD SUBBLOCK
C*****
C THE FOLLOWING BLOCK OF WRITE STATEMENTS MUST BE UNCOMMENTED
C IN ORDER TO OUTPUT DATA TO A USER-DEFINED PLOT FILE (NPLOTU).
C
C THE OUTPUT QUANTITIES IN THIS BLOCK ARE THE FOLLOWING :
C WAVE = WAVELENGTH (MICROMETERS)
C ISO = AEROSOL TYPE (VALID RANGE, 0-12)
C TAU = OPTICAL DEPTH ALONG LONGEST AXIS OF CLOUD ELLIPSOID
C N1 = NUMBER OF PARTIAL RUNS WITHIN THIS BASCAT RUN
C XS( ) = SOURCE XYZ POSITION ARRAY (KILOMETERS)
C THES = SOURCE VECTOR POLAR ANGLE (DEGREES)
C PHIS = SOURCE VECTOR AZIMUTHAL ANGLE (DEGREES)
C ASMM = SOURCE APERTURE RADIUS (MILLIMETERS)
C THSP = HALF-ANGLE OF SOURCE DIFFRACTION CONE (RADIAN)
C XD( ) = DETECTOR XYZ POSITION ARRAY (KILOMETERS)
C THED = DETECTOR VECTOR POLAR ANGLE (DEGREES)
C PHID = DETECTOR VECTOR AZIMUTHAL ANGLE (DEGREES)
C ACM = DETECTOR APERTURE RADIUS (CENTIMETERS)
C THEY = HALF-ANGLE OF DETECTOR FIELD OF VIEW (DEGREES)
C AK( ) = CLOUD ELLIPSOID PRINCIPAL HALF-AXIS ARRAY (KILOMETERS)
C ALB(1)= SINGLE-SCATTERING ALBEDO OF CLOUD AEROSOL
C ALB(2)= ALBEDO OF GROUND PLANE
C
C WRITE(NPLOTU,9111) WAVE,ISO,TAU,N1
C9111 FORMAT(E9.4,1X,12,1X,E9.4,1X,2(12,1X))
C WRITE(NPLOTU,9222)XS(1),XS(2),XS(3),THES,PHIS,ASMM,THSP
C WRITE(NPLOTU,9222)XD(1),XD(2),XD(3),THED,PHID,ACM,THEY
C9222 FORMAT(12(E9.4,1X))
C WRITE(NPLOTU,9222) A(1),A(2),A(3),ALB(1),ALB(2)
C*****
C WRITE (IOOUT,6900)
C WRITE (IOOUT,7000)(A(K),K=1,3)
C WRITE (IOOUT,7100)THE,PHE,PSE
BAS04210
BAS04220
BAS04230
BAS04240
BAS04250
BAS04260
BAS04270
BAS04280
BAS04290
BAS04300
BAS04310
BAS04320
BAS04330
BAS04340
BAS04350
BAS04360
BAS04370
BAS04380
BAS04390
BAS04400
BAS04410
BAS04420
BAS04430
BAS04440
BAS04450
BAS04460
BAS04470
BAS04480
BAS04490
BAS04500
BAS04510
BAS04520
BAS04530
BAS04540
BAS04550
BAS04560
BAS04570
BAS04580
BAS04590
BAS04600
BAS04610
BAS04620
BAS04630
BAS04640
BAS04650
BAS04660
BAS04670
BAS04680
BAS04690
BAS04700
BAS04710
BAS04720
BAS04730
BAS04740
BAS04750
BAS04760
BAS04770
BAS04780
BAS04790
BAS04800
BAS04810
BAS04820
BAS04830
BAS04840
BAS04850
BAS04860
BAS04870
BAS04880
BAS04890
BAS04900

```

```

C      WRITE (IOOUT,7200)(Y(K),K=1,3)
C***  GENERATE ROTATION MATRIX R( ) FOR CONVERSION FROM DETECTOR
C***  CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE
C      CALL MATRX(THED,PHID,SD,R)
C***  SET LIMIT NSCAM ON HIGHEST SIGNIFICANT SCATTERING ORDER (NSCAM-1)
C      NSCAM=3.0*ALBEDO*TAU+2.0
C      IF (NSCAM.GT.10) NSCAM=10
C      IF (NSCAM.LT.3) NSCAM=3
C      NSCA1=NSCAM-1
C      AT=0.
C      DO 4 K=1,3
C 4     AT=AT+RE(K,3)**2/ASQ(K)
C      BT=1.-AT*ZG
C      IF (BT.LE.0.) GO TO 4002
C      AT=ZG+D
C      IF (AT.GT.0.) GO TO 3
C      ZG=-D
C      WRITE (IOOUT,9004) ZG
C 3     CONTINUE
C      IF (XS(3).GE.ZG) GO TO 16
C      XS(3)=ZG
C      WRITE (IOOUT,9003) XS(3)
C 16    CONTINUE
C***  DETERMINE DISTANCES ELD(1), ELD(2) FROM DETECTOR TO NEAREST AND
C***  FARTHEST CLOUD BOUNDARIES ALONG DETECTOR AXIS
C      CALL ELM(XD,SD,ELD)
C***  DETERMINE DISTANCES EL(1), EL(2) FROM SOURCE TO NEAREST AND
C***  FARTHEST CLOUD BOUNDARIES ALONG SOURCE AXIS
C      CALL ELM(XS,SS,EL)
C***  PERFORM GEOMETRICAL ERROR CHECKS
C      IF ((EL(2).LE.0.) .AND. (ELD(2).LE.0.)) GO TO 4003
C      IF ((EL(2).GT.0.) .AND. (ELD(2).GT.0.)) GO TO 6
C      IF (EL(2).LE.0.) GO TO 7
C      IF (ALB(2).LE.0.) GO TO 4004
C      IF (SD(3).GE.0.) GO TO 4001
C      AT=EL(1)
C      CT=(ZG-XD(3))/SD(3)
C      GO TO 9
C 7     CONTINUE
C      IF (ALB(2).LE.0.) GO TO 4005
C      IF (SS(3).GT.0.) GO TO 4000
C      AT=(ZG-XS(3))/SS(3)
C      GO TO 8
C 6     AT=EL(1)
C***  DETERMINE MINIMUM POSSIBLE TRAVERSE DISTANCE ELMIN
C      8 CT=ELD(1)
C      9 BT=0.
C      DO 11 K=1,3
C      X(K)=XS(K)+AT*SS(K)
C      Y(K)=XD(K)+CT*SD(K)
C 11    BT=BT+(X(K)-Y(K))**2
C      IF (BT.LE.1.E-30) BT=0.
C      ELMIN=SQRT(BT)+AT+CT
C      IF (ELMIN.LT.ALIM) ELMIN=ALIM
C***  DETERMINE VIRTUAL SOURCE POINT XV( )
C      DSA=0.

```

```

BAS04910
BAS04920
BAS04930
BAS04940
BAS04950
BAS04960
BAS04970
BAS04980
BAS04990
BAS05000
BAS05010
BAS05020
BAS05030
BAS05040
BAS05050
BAS05060
BAS05070
BAS05080
BAS05090
BAS05100
BAS05110
BAS05120
BAS05130
BAS05140
BAS05150
BAS05160
BAS05170
BAS05180
BAS05190
BAS05200
BAS05210
BAS05220
BAS05230
BAS05240
BAS05250
BAS05260
BAS05270
BAS05280
BAS05290
BAS05300
BAS05310
BAS05320
BAS05330
BAS05340
BAS05350
BAS05360
BAS05370
BAS05380
BAS05390
BAS05400
BAS05410
BAS05420
BAS05430
BAS05440
BAS05450
BAS05460
BAS05470
BAS05480
BAS05490
BAS05500
BAS05510
BAS05520
BAS05530
BAS05540
BAS05550
BAS05560
BAS05570
BAS05580
BAS05590
BAS05600

```

```

IF(ASMM.GT.0.)DSA=ASMM*1.E-6/TAN(THSP)
DO 305 K=1,3
305 XV(K)=XS(K)-DSA*SS(K)
C
C*** DETERMINE WHETHER NEAR-CLOUD LIDAR BIASING IS NEEDED.
C
DOTSD=0.
DOTXD=0.
DO 47 KL=1,3
47 DOTXD=DOTXD+(XD(KL)-XS(KL))*2
DOTSD=DOTSD+SS(KL)*SD(KL)
IF(DOTXD.GT.BLIM) DOTSD=0.
REWIND KSTOR
C
C*** START PHOTON LOOPS
C
DO 1000 II=1,NI
C
C*** CHECK TO SEE IF THIS IS THE FIRST RUN WITH NO PRIOR RESULTS USED
C
IF((II.EQ.1).AND.(ITIME.EQ.1))GO TO 1303
C
C*** RELOAD WITH RESULTS FROM PREVIOUS PARTIAL RUN
C
DO 1301 NSCA=1,NSCAM
READ(KSTOR)ENC(NSCA)
DO 1301 II=1,NTMAX
1301 READ(KSTOR)EN(NSCA,II)
CONTINUE
REWIND KSTOR
GO TO 1302
1303 CONTINUE
C
C*** INITIALIZATION FOR FIRST RUN
C
DO 1300 NSCA=1,NSCAM
ENC(NSCA)=0.
DO 1300 NT=1,NTMAX
1300 EN(NSCA,NT)=0.0
1302 CONTINUE
C
C*** LOAD EXPECTED CUMULATIVE NUMBER OF PHOTONS
C
DO 1304 NS=1,NSCAM
1304 ENC(NS)=ENC(NS)+N2
C
C*** DETERMINE VECTOR FROM VIRTUAL SOURCE POINT TO DETECTOR
C
DO 555 K=1,3
555 Y(K)=XD(K)-XV(K)
C
C*** START INNER PHOTON LOOP
C
DO 900 I2=1,N2
C
C*** INITIALIZE PHOTON DIRECTION AND CALCULATE DIRECT BEAM
C*** CONTRIBUTIONS (IF ANY)
C
CALL START(SS)
IF(STH.GT.0.)GO TO 150
GO TO 900
150 NSCA=1
C
C*** START MULTIPLE SCATTERING LOOP
C
I2FLG=0
ICOND=0
1700 NSCA=NSCA+1
C
C*** NEAR-CLOUD LIDAR BIASING IS ACTIVE WHEN ICOND=1. 10 PERCENT OF ALL

```

```

BAS05610
BAS05620
BAS05630
BAS05640
BAS05650
BAS05660
BAS05670
BAS05680
BAS05690
BAS05700
BAS05710
BAS05720
BAS05730
BAS05740
BAS05750
BAS05760
BAS05770
BAS05780
BAS05790
BAS05800
BAS05810
BAS05820
BAS05830
BAS05840
BAS05850
BAS05860
BAS05870
BAS05880
BAS05890
BAS05900
BAS05910
BAS05920
BAS05930
BAS05940
BAS05950
BAS05960
BAS05970
BAS05980
BAS05990
BAS06000
BAS06010
BAS06020
BAS06030
BAS06040
BAS06050
BAS06060
BAS06070
BAS06080
BAS06090
BAS06100
BAS06110
BAS06120
BAS06130
BAS06140
BAS06150
BAS06160
BAS06170
BAS06180
BAS06190
BAS06200
BAS06210
BAS06220
BAS06230
BAS06240
BAS06250
BAS06260
BAS06270
BAS06280
BAS06290
BAS06300

```

```

C**** FIRST ORDER PHOTONS USE AN ALTERNATE MODE OF BIASING IN THIS          BAS 06310
C**** SITUATION. THE REMAINING 90 PERCENT ARE NORMALLY BIASED IN ALL        BAS 06320
C**** ORDERS.                                                                BAS 06330
C                                                                              BAS 06340
C                                                                              BAS 06350
C                                                                              BAS 06360
C                                                                              BAS 06370
C                                                                              BAS 06380
C                                                                              BAS 06390
C                                                                              BAS 06400
C                                                                              BAS 06410
C                                                                              BAS 06420
C                                                                              BAS 06430
C                                                                              BAS 06440
C                                                                              BAS 06450
C                                                                              BAS 06460
C                                                                              BAS 06470
C                                                                              BAS 06480
C                                                                              BAS 06490
C                                                                              BAS 06500
C                                                                              BAS 06510
C                                                                              BAS 06520
C                                                                              BAS 06530
C                                                                              BAS 06540
C                                                                              BAS 06550
C                                                                              BAS 06560
C                                                                              BAS 06570
C                                                                              BAS 06580
C                                                                              BAS 06590
C                                                                              BAS 06600
C                                                                              BAS 06610
C                                                                              BAS 06620
C                                                                              BAS 06630
C                                                                              BAS 06640
C                                                                              BAS 06650
C                                                                              BAS 06660
C                                                                              BAS 06670
C                                                                              BAS 06680
C                                                                              BAS 06690
C                                                                              BAS 06700
C                                                                              BAS 06710
C                                                                              BAS 06720
C                                                                              BAS 06730
C                                                                              BAS 06740
C                                                                              BAS 06750
C                                                                              BAS 06760
C                                                                              BAS 06770
C                                                                              BAS 06780
C                                                                              BAS 06790
C                                                                              BAS 06800
C                                                                              BAS 06810
C                                                                              BAS 06820
C                                                                              BAS 06830
C                                                                              BAS 06840
C                                                                              BAS 06850
C                                                                              BAS 06860
C                                                                              BAS 06870
C                                                                              BAS 06880
C                                                                              BAS 06890
C                                                                              BAS 06900
C                                                                              BAS 06910
C                                                                              BAS 06920
C                                                                              BAS 06930
C                                                                              BAS 06940
C                                                                              BAS 06950
C                                                                              BAS 06960
C                                                                              BAS 06970
C                                                                              BAS 06980
C                                                                              BAS 06990
C                                                                              BAS 07000

C**** IF<<NSCA.NE.2>>.OR.<<DOTSD.LE.0.99>>.OR.<<ELD<1>.GT.0.01>>GO TO 707
C      ICOND=1
C      IF<<FLOAT<I2>.LE.<<FLOAT<N2>/10.>>I2FLG=1
C 707 CONTINUE

C**** MOVE PHOTON TO NEXT EVENT POINT VIA BIASED TRAVERSE AND DETERMINE
C**** POWER CONTRIBUTIONS

C      CALL TRAVRS<JTYPE,I2FLG,ICOND>
C      IF<<NSCA.EQ.NSCA1>>.OR.<<STH.LE.0.>>GO TO 900

C**** DETERMINE SCATTERING DIRECTION FOR NEXT TRAVERSE
C      CALL GAS<JTYPE>

C**** DETERMINE BIASING DIRECTIONS FOR NEXT TRAVERSE
C      CALL ELM<XA,SA,EL>
C      IF<<EL<2>.LE.EL<1>>GO TO 900
C      GO TO 1700
C 900 CONTINUE

C**** END MULTIPLE SCATTERING LOOP, BEGIN CONVOLUTION BLOCK
C**** WRITE PARTIAL RUN RESULTS INTO STORAGE FILE FOR USE BY NEXT RUN

C      DO 2201 NSCA=1,NSCAM
C      WRITE<KSTOR>ENC<NSCA>
C      DO 2201 II=1,NTMAX
C      WRITE<KSTOR>ENC<NSCA,II>
C 2201 CONTINUE
C      REWIND KSTOR

C**** OUTPUT PARTIAL RUN RESULTS FOR STEADY STATE POWER

C      WRITE<IOOUT,7400>
C      WRITE<IOOUT,7500>
C      WRITE<IOOUT,7600>
C      PTOT=0.
C*****
C      THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO
C      A USER-DEFINED PLOT FILE <NPLOTU> IS DESIRED.
C*****
C      NSCAM= 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING
C*****
C      WRITE<NPLOTU,9333> NSCAM
C*****
C      DO 2110 NS=1,NSCAM
C      PE=0.
C      NS1=NS-1
C      DO 2115 NT=1,NTMAX
C 2115 PE=PE+ENC<NS,NT>
C      PE=PE/ENC<NS>
C      PTOT=PTOT+PE
C*****
C      THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO
C      A USER-DEFINED PLOT FILE <NPLOTU> IS DESIRED.
C*****
C      NS1 = ORDER OF SCATTERING
C      PE = OBSERVED STEADY STATE POWER FOR THIS ORDER
C      ENC<NS>= TOTAL NUMBER OF PHOTONS COUNTED FOR THIS ORDER
C*****
C      WRITE<NPLOTU,9333> NS1,PE,ENC<NS>
C*****
C      WRITE<IOOUT,7700>NS1,PE,ENC<NS>
C 2110 CONTINUE
C*****

```

```

C          THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO
C          A USER-DEFINED PLOT FILE (NPLTU) IS DESIRED.
C          PTOT = OBSERVED TOTAL STEADY STATE POWER (ALL ORDERS)
C9333  FORMAT(I2,1X,2(E9.4,1X))
C      WRITE(NPLTU, 9222) PTOT
C*****
C      WRITE(IOOUT,7701)PTOT
C      DO 2500 NSCA=1,NSCAM
C
C****  DETERMINE INDEX OF TIME OF LAST NONZERO POWER VALUE FOR EACH
C****  ORDER
C
C      CALL SMOOZ(NSCA,NO)
C      NM(NSCA)=1+NO
C      IOR(NSCA)=NSCA-1
C2500  Y(NSCA)=NM(NSCA)
C
C****  DETERMINE LATEST TIME INDEX OF NONZERO POWER FOR ALL ORDERS
C
C      CALL GMAX(NSCAM,YMAX)
C      NMA=YMAX
C      JP=1
C2900  JP=JP+1
C      AT=TPU(JP)
C      IF(AT.LE.0.0) GO TO 3000
C      GO TO 2900
C3000  JP=JP-1
C      WRITE(IOOUT,7400)
C      WRITE(IOOUT,7900) JPMAX
C
C****  BEGIN PULSE LOOP
C
C      DO 3800 JP=1,JPMAX
C      WRITE(IOOUT,7400)
C      TP=TPU(JP)
C      AT=(NMA+4)*DELT
C      IF(TP.GT.AT)TP=AT
C      IF(TP.LE.DELT)TP=DELT+1.E-3
C      NP=1.001+TP/DELT
C      NMAX=NMA+NP
C      IF(NMAX.GT.NDIM)NMAX=NDIM
C      NP=NMAX-NMA
C      TP=(NP-1)*DELT
C      TMAX=(NMAX-1)*DELT
C      WRITE(IOOUT,8100) JP,TP
C      WRITE(IOOUT,8200) JP,TMAX
C      IF(TP.LE.0.)GO TO 3999
C      DO 3400 NSCA=1,NSCAM
C      NMS=NM(NSCA)
C      NMS1=NMS-1
C
C****  NORMALIZE RETURN POWER BY DIVIDING CUMULATIVE POWER BY CUMULATIVE
C****  NUMBER OF PHOTONS
C
C      DO 3200 N=1,NMS1
C3200  EN(NSCA,N)=EN(NSCA,N)/ENC(NSCA)
C
C****  PERFORM SQUARE SOURCE PULSE CONVOLUTION WITH PROBABILITIES
C****  PER UNIT TIME
C
C      CALL CONV(NP,NMS,NMAX,NSCA)
C3400  CONTINUE
C      DO 3500 N=1,NMAX
C      Y(N)=0.0
C      DO 3500 NSCA=1,NSCAM
C3500  Y(N)=Y(N)+EN(NSCA,N)
C      XN=-DELT
C      DO 3600 N=1,NMAX

```

```

BAS07010
BAS07020
BAS07030
BAS07040
BAS07050
BAS07060
BAS07070
BAS07080
BAS07090
BAS07100
BAS07110
BAS07120
BAS07130
BAS07140
BAS07150
BAS07160
BAS07170
BAS07180
BAS07190
BAS07200
BAS07210
BAS07220
BAS07230
BAS07240
BAS07250
BAS07260
BAS07270
BAS07280
BAS07290
BAS07300
BAS07310
BAS07320
BAS07330
BAS07340
BAS07350
BAS07360
BAS07370
BAS07380
BAS07390
BAS07400
BAS07410
BAS07420
BAS07430
BAS07440
BAS07450
BAS07460
BAS07470
BAS07480
BAS07490
BAS07500
BAS07510
BAS07520
BAS07530
BAS07540
BAS07550
BAS07560
BAS07570
BAS07580
BAS07590
BAS07600
BAS07610
BAS07620
BAS07630
BAS07640
BAS07650
BAS07660
BAS07670
BAS07680
BAS07690
BAS07700

```

```

3600 XN=XN+DELT
X(N)=XN
C*** OUTPUT TIME-DEPENDENT RESULTS
WRITE (IOOUT,8300)
WRITE (IOOUT,8400)
WRITE(IOOUT,8500)(IOR(NS),NS=1,NSCAM)
WRITE(IOOUT,8501)
C*****
C THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO
A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.
C
C NMAX = NUMBER OF TIME BOXES USED FOR TIME-DEPENDENT DATA
C JP = INDEX OF INPUT PULSE
C NSCAM = 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING
C X( ) = ARRAY OF TIME VALUES FOR EACH TIME BOX
C Y( ) = ARRAY OF OBSERVED TIME-DEPENDENT TOTAL POWER
C EN(L, ) = ARRAY OF OBSERVED TIME-DEPENDENT POWER FOR ORDER L+1
C
C IF(I1.LT.N1) GO TO 9777
WRITE(NPLOTU,9444) NMAX,JP,NSCAM
C9444 FORMAT(3(I3,1X))
WRITE(NPLOTU,9222)(X(LLL),LLL=1,NMAX)
WRITE(NPLOTU,9222)(Y(LLL),LLL=1,NMAX)
DO 9666 LLL=1,NSCAM
C9666 WRITE(NPLOTU,9222)(EN(LLL,LLX),LLX=1,NMAX)
CONTINUE
C9777 CONTINUE
C*****
C DO 3700 N=1,NMAX
WRITE (IOOUT,8600) X(N),Y(N),EN(NSCA,N),NSCA=1,NSCAM)
3700 CONTINUE
C*** RELOAD PARTIAL RUN RESULTS FOR CONVOLUTION WITH NEXT PULSE
C
DO 3701 NS=1,NSCAM
READ(KSTOR)ENC(NS)
DO 3701 II=1,NTMAX
READ(KSTOR)EN(NS,II)
3701 CONTINUE
REWIND KSTOR
3800 CONTINUE
C*** END PULSE LOOP
C
1000 CONTINUE
C*** END OUTER PHOTON LOOP?
C
777 RETURN
C*** ERROR RETURN MESSAGES
C
3999 CONTINUE
WRITE(IOOUT,8700)NTMAX
RETURN
4000 CONTINUE
WRITE(IOOUT,8800)
RETURN
4001 CONTINUE
WRITE(IOOUT,9000)
RETURN
4002 CONTINUE
WRITE(IOOUT,9001)
RETURN
4003 CONTINUE
WRITE(IOOUT,9002)
RETURN
4004 CONTINUE

```

```

BAS07710
BAS07720
BAS07730
BAS07740
BAS07750
BAS07760
BAS07770
BAS07780
BAS07790
BAS07800
BAS07810
BAS07820
BAS07830
BAS07840
BAS07850
BAS07860
BAS07870
BAS07880
BAS07890
BAS07900
BAS07910
BAS07920
BAS07930
BAS07940
BAS07950
BAS07960
BAS07970
BAS07980
BAS07990
BAS08000
BAS08010
BAS08020
BAS08030
BAS08040
BAS08050
BAS08060
BAS08070
BAS08080
BAS08090
BAS08100
BAS08110
BAS08120
BAS08130
BAS08140
BAS08150
BAS08160
BAS08170
BAS08180
BAS08190
BAS08200
BAS08210
BAS08220
BAS08230
BAS08240
BAS08250
BAS08260
BAS08270
BAS08280
BAS08290
BAS08300
BAS08310
BAS08320
BAS08330
BAS08340
BAS08350
BAS08360
BAS08370
BAS08380
BAS08390
BAS08400

```



```

WRITE< IOOUT, 9005 >
RETURN
4005 CONTINUE
WRITE< IOOUT, 9006 >
RETURN
C*****FORMATS
4200 FORMAT (7F10.5)
4400 FORMAT (7I10)
4500 FORMAT (I10,2F10.5,I10)
4600 FORMAT (5E15.10)
4700 FORMAT (1H,34X,51(1H*))
4800 FORMAT (1H,34X,1H*,49X,1H*)
4900 FORMAT (1H,34X,1H*,9X,31HMONTE CARLO MULTIPLE SCATTERING,9X,1H*)
5000 FORMAT (1H,34X,1H*,16X,18HAEROSOL SCATTERING,15X,1H*)
5100 FORMAT (1H0,48X,23HPARAMETERS FOR THIS RUN)
5200 FORMAT(1H0,46X,28HUSER SUPPLIED PHASE FUNCTION)
5201 FORMAT(1H0,43X,34HMARITIME ARCTIC, VIS=0.1 TO 2.0 KM)
5202 FORMAT(1H0,47X,26HMARITIME POLAR, VIS=0.2 KM)
5203 FORMAT(1H0,47X,26HMARITIME POLAR, VIS=2.0 KM)
5204 FORMAT(1H0,42X,36HCONTINENTAL POLAR, VIS=0.2 TO 2.5 KM)
5205 FORMAT(1H0,52X,16HWHITE PHOSPHORUS)
5206 FORMAT(1H0,52X,16HHEXACHLOROETHANE)
5207 FORMAT(1H0,57X,7HFOG OIL)
5208 FORMAT(1H0,45X,31HDUST (MODERATE AEROSOL LOADING))
5209 FORMAT(1H0,46X,28HDUST (HEAVY AEROSOL LOADING))
5210 FORMAT(1H0,43X,34HMARITIME MODEL B, VIS=5 KM, RH=95%)
5211 FORMAT(1H0,43X,35HMARITIME MODEL B, VIS=10 KM, RH=90%)
5212 FORMAT(1H0,43X,35HMARITIME MODEL B, VIS=50 KM, RH=50%)
5500 FORMAT (1H0,46X,27HUSER SUPPLIED AEROSOL MODEL)
5600 FORMAT (1H,36X,11HWAVELENGTH=,F6.3,16H MICROMETERS
1 7HALBEDO=,F5.3)
5601 FORMAT(1H0,47X,25HELLIPSOIDAL AEROSOL CLOUD)
5602 FORMAT(1H,41X,36HCOORDINATE ORIGIN AT CENTER OF CLOUD)
5603 FORMAT(1H,38X,42HZ-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH)
5700 FORMAT (1H,36X,31HAEROSOL EXTINCTION COEFFICIENT=,
1 E10.4,7H KM**-1)
5701 FORMAT(1H0,51X,17HSOURCE PARAMETERS)
5702 FORMAT(1H,36X,27HSOURCE XYZ COORDINATES(KM)=,3(F8.4,1X))
5703 FORMAT(1H,36X,27HSOURCE APERTURE RADIUS(MM)=,F7.3)
5704 FORMAT(1H,36X,26HSOURCE AXIS POLAR ANGLE =,F7.3,8H DEGREES)
5705 FORMAT(1H,36X,26HSOURCE AXIS AZIMUTH ANGLE=,F7.3,8H DEGREES)
5706 FORMAT(1H,36X,26HSOURCE BEAM SPREAD ANGLE =,E10.4,8H RADIANS)
5900 FORMAT (1H0,50X,19HDETECTOR PARAMETERS)
6000 FORMAT (1H,35X,29H CONE OF VIEW HALF-ANGLE =,F7.3,
1 8H DEGREES)
6100 FORMAT (1H,35X,29H DETECTOR APERTURE RADIUS =,F7.3,
1 3H CM)
6201 FORMAT(1H,36X,29HDETECTOR XYZ COORDINATES(KM)=,3(F8.4,1X))
6400 FORMAT (1H,36X,28HDETECTOR AXIS POLAR ANGLE =,F7.3,
1 8H DEGREES)
6500 FORMAT (1H,36X,28HDETECTOR AXIS AZIMUTH ANGLE=,F7.3,
1 8H DEGREES)
6501 FORMAT(1H0,47X,23HGROUND PLANE PARAMETERS,/,40X,38HISOTROPIC REFLE
*CTION FROM GROUND PLANE)
6502 FORMAT(1H,36X,33HGROUND PLANE Z-COORDINATE ZG(KM)=,F7.3,/,
*37X,33HGROUND PLANE ALBEDO, ALBG, =,F7.3)
6800 FORMAT (1H0,34X,51(1H*))
6900 FORMAT (1H0,51X,16HCLLOUD PARAMETERS)
7000 FORMAT(1H,31X,43HELLIPSOID PRINCIPAL XYZ HALF-AXES(KM) =,
*3(F8.4,1X))
7100 FORMAT (1H,31X,43HEULER ANGLES THE,PHE,PSE OF ELLIPSOID( DEG)=,
*3(F8.4,1X))
7200 FORMAT (1H,31X,43HOPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES =,
*3(F8.4,1X))
7400 FORMAT (1H0,100X)
7500 FORMAT (1H0,32X,53HSTEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE
* POWER)
7600 FORMAT (1H0,37X,53HORDER,3X,19HSTEADY STATE POWER,3X,
1 17HNUMBER OF PHOTONS)
7700 FORMAT (1H,39X,12,8X,E10.5,8X,E12.6)

```

```

BAS0841
BAS0842
BAS0843
BAS0844
BAS0845
BAS0846
BAS0847
BAS0848
BAS0849
BAS0850
BAS0851
BAS0852
BAS0853
BAS0854
BAS0855
BAS0856
BAS0857
BAS0858
BAS0859
BAS0860
BAS0861
BAS0862
BAS0863
BAS0864
BAS0865
BAS0866
BAS0867
BAS0868
BAS0869
BAS0870
BAS0871
BAS0872
BAS0873
BAS0874
BAS0875
BAS0876
BAS0877
BAS0878
BAS0879
BAS0880
BAS0881
BAS0882
BAS0883
BAS0884
BAS0885
BAS0886
BAS0887
BAS0888
BAS0889
BAS0890
BAS0891
BAS0892
BAS0893
BAS0894
BAS0895
BAS0896
BAS0897
BAS0898
BAS0899
BAS0900
BAS0901
BAS0902
BAS0903
BAS0904
BAS0905
BAS0906
BAS0907
BAS0908
BAS0909
BAS0910

```

```

7701 FORMAT (1H ,37X,5HTOTAL,7X,E10.5) BAS09110
7900 FORMAT (1H0,32X,23HPower INTO DETECTOR FOR,12,6H PULSE BAS09120
1 23H(S) OF DIFFERENT LENGTH) BAS09130
8100 FORMAT (1H0,37X,12HPULSE NUMBER,12,12H HAS LENGTH , BAS09140
1 E10.4,13H MICROSECONDS) BAS09150
8200 FORMAT (1H ,23X,34HDETECTOR RESPONSE CUTOFF TIME FOR BAS09160
1 12HPULSE NUMBER,12,4H IS ,E10.4, BAS09170
2 13H MICROSECONDS) BAS09180
8300 FORMAT (1H0,25X,68HDETECTOR RESPONSE, POWER AS A FUNCTION OF TIME, BAS09190
* FOR UNIT PULSE POWER) BAS09200
8400 FORMAT (1H0,55X,21HPower FROM EACH ORDER/14X,5HTOTAL) BAS09210
8500 FORMAT (1H ,3X,4HTIME,6X,5HPower,2X,10(SX,12,4X)) BAS09220
8501 FORMAT(1H ,130(1H-)) BAS09230
8600 FORMAT (12(E10.4,1X)) BAS09240
8700 FORMAT(1H0,6HNTMAX=,13,80H SHOULD BE DECREASED TO 46. IT IS TOO LAB BAS09250
*RGE TO ALLOW CONVOLUTION WITH YOUR PULSE) BAS09260
8800 FORMAT(1H0,97HYOUR INCIDENT PHOTONS NEVER INTERSECT THE CLOUD OR TBAS09270
*HE GROUND. CHECK YOUR INPUT SOURCE PARAMETERS) BAS09280
9000 FORMAT(1H0,72HTHE DETECTOR LOOKS ABOVE THE CLOUD. CHECK YOUR INPUT BAS09290
* DETECTOR PARAMETERS) BAS09300
9001 FORMAT(1H0,65HYOUR GROUND PLANE IS ENTIRELY ABOVE YOUR CLOUD. CHECK BAS09310
*K YOUR INPUTS) BAS09320
9002 FORMAT(1H0,76HNEITHER YOUR SOURCE NOR YOUR DETECTOR LOOK INTO THE BAS09330
*CLOUD. CHECK YOUR INPUTS) BAS09340
9003 FORMAT(1H0,71HYOUR SOURCE WAS UNDERGROUND. IT HAS BEEN PUT AT THE BAS09350
*GROUND, WITH XS(3)=,F6.3,2HKM) BAS09360
9004 FORMAT(1H0,114HYOUR GROUND PLANE WAS TOO FAR AWAY FROM THE CLOUD TBAS09370
*O PRODUCE GROUND REFLECTIONS WITHIN THE MAX TIME DELAY ALLOWED.,/, BAS09380
*34H THE GROUND PLANE WAS MOVED TO ZG=,F6.3,2HKM) BAS09390
9005 FORMAT(1H0,91HGROUND PLANE WAS ABSENT AND DETECTOR DOES NOT LOOK ABAS09400
*T CLOUD. CHECK YOUR DETECTOR PARAMETERS) BAS09410
9006 FORMAT(1H0,79HGROUND PLANE WAS ABSENT AND SOURCE DOES NOT ILLUMINABAS09420
*TE CLOUD. CHECK YOUR INPUTS) BAS09430
END BAS09440

```

```

SUBROUTINE BKWD(JTYPE)
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON /RNDM/ SEED
COMMON /MOS/V1,C1,S1,SA(3),ENK(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
COMMON /FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
*ALB(2),ZG,DMAX
COMMON /FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
COMMON /FGEL/XA(3),D,NSCA
DIMENSION SBC(3),SBA(3)
NSCAP=NSCA+1
C
C*** DETERMINE MONTE CARLO BACKWARD TRAVERSE ANGLES THETA (AT)
C*** AND PHI (BT) INSIDE DETECTOR CONE.
C
BT=TWOPI*RAND ( SEED)
AT=THV*RAND ( SEED)
IF(AT.LE.1.E-30) AT=0.
IF(BT.LE.1.E-30) BT=0.
V=SIN(AT)
U23=COS(AT)
C
C*** ROTATE BACKWARD TRAVERSE VECTOR INTO STANDARD FRAME OF REFERENCE.
C
CALL ROTAT(AT,BT,R,SBC)
C
C*** DETERMINE BIASING DISTANCES FOR BACKWARD TRAVERSE.
C
CALL ELM(XD,SBC,EL)
C
C*** DETERMINE STATISTICAL WEIGHT REX OF BACKWARD TRAVERSE.
C
REX=1.-EXP(-GAMMA*(EL(2)-EL(1)))
REXRN=REX*RAND(SEED)
IF((1.-REXRN).LE.1.E-7) GO TO 15
C
C*** DETERMINE RANDOM DISTANCE FOR BIASED BACKWARD TRAVERSE.
C
ELBC=-ALOG(1.-REXRN)/GAMMA+EL(1)
CT=0.
DO 4 K=1,3
SBA(K)=XD(K)-XA(K)+SBC(K)*ELBC
4 CT=CT+SBA(K)**2
IF(CT.LE.1.E-30) CT=0.
ELAB=SQRT(CT)
IF(ELAB.LT.ALIM)GO TO 15
C
C*** DETERMINE TIME BOX INDEX NT FOR THE COMPLETE BACKWARD TRAVERSE.
C
NT=1.+(DTOT+ELAB+ELBC-ELMIN)/DELD
IF(NT.LT.0) GO TO 15
IF(NT.EQ.0) NT=1
IF(NT.GT.NTMAX)RETURN
U1=0.
U2=0.
DO 5 K=1,3
SBA(K)=SBA(K)/ELAB
U1=U1+SA(K)*SBA(K)
5 U2=U2-SBA(K)*SBC(K)
6 CALL FIND(U2,PF2)
IF(JTYPE.EQ.2)GO TO 9
CALL FIND(U1,PF1)
GO TO 10
9 CONTINUE
CALL ELM(XA,SBA,EL)
ELAB=ELAB-EL(1)
PF1=1.
10 DOM=FAC*ALB(1)*STH*REX*V*U23*PF1*PF2*EXP(-GAMMA*ELAB)/CT

```

```

BKW00010
BKW00020
BKW00030
BKW00040
BKW00050
BKW00060
BKW00070
BKW00080
BKW00090
BKW00100
BKW00110
BKW00120
BKW00130
BKW00140
BKW00150
BKW00160
BKW00170
BKW00180
BKW00190
BKW00200
BKW00210
BKW00220
BKW00230
BKW00240
BKW00250
BKW00260
BKW00270
BKW00280
BKW00290
BKW00300
BKW00310
BKW00320
BKW00330
BKW00340
BKW00350
BKW00360
BKW00370
BKW00380
BKW00390
BKW00400
BKW00410
BKW00420
BKW00430
BKW00440
BKW00450
BKW00460
BKW00470
BKW00480
BKW00490
BKW00500
BKW00510
BKW00520
BKW00530
BKW00540
BKW00550
BKW00560
BKW00570
BKW00580
BKW00590
BKW00600
BKW00610
BKW00620
BKW00630
BKW00640
BKW00650
BKW00660
BKW00670
BKW00680
BKW00690
BKW00700

```

```
EN<NSCAP, NT>=EN<NSCAP, NT>+DOM  
RETURN  
15 EN<NSCAP>=EN<NSCAP>-1.  
STH=0,  
RETURN  
END
```

```
BKW00710  
BKW00720  
BKW00730  
BKW00740  
BKW00750  
BKW00760
```

	SUBROUTINE CONV(NP,NMS,NMAX,NSCA)	CON00010
C*****	THIS SUBROUTINE CONVOLUTES A GIVEN SEQUENCE OF VALUES EN(NSCA,N),	CON00020
C	N=1,NMS, WITH THE UNIT SQUARE FUNCTION EXTENDING FROM TIME=0 TO	CON00030
C*****	TIME=(NP-1)*DELT.	CON00040
	DOUBLE PRECISION DBLE,XDBLE(100)	CON00050
	COMMON/CONB/X(100),Y(100)	CON00060
	COMMON /M05/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,	CON00070
	+NTMAX,NSCAN,KMAX,LMAX,LMM1	CON00080
	COMMON /BASPO/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),	CON00090
	+BE(16),SINGWV,PF(65),LLMAX	CON00100
	NMS1=NMS+1	CON00110
	NMA=NMS+NP-1	CON00120
	XDBLE(1)=0.E-00	CON00130
	DO 100 N=2,NMS	CON00140
100	XDBLE(N)=XDBLE(N-1)+DBLE(EN(NSCA,N-1))	CON00150
	NP1=NP+1	CON00160
	IF (NP.LT.NMS) GO TO 600	CON00170
	DO 200 N=1,NMS	CON00180
200	Y(N)=XDBLE(N)	CON00190
	IF (NP.EQ.NMS) GO TO 400	CON00200
	DO 300 N=NMS1,NP	CON00210
300	Y(N)=XDBLE(NMS)	CON00220
400	CONTINUE	CON00230
	DO 500 N=NP1,NMA	CON00240
500	Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)	CON00250
	GO TO 1000	CON00260
600	CONTINUE	CON00270
	DO 700 N=1,NP	CON00280
700	Y(N)=XDBLE(N)	CON00290
	DO 800 N=NP1,NMS	CON00300
800	Y(N)=XDBLE(N)-XDBLE(N-NP+1)	CON00310
	DO 900 N=NMS1,NMA	CON00320
900	Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)	CON00330
1000	CONTINUE	CON00340
	IF (NMA.EQ.NMAX) GO TO 1200	CON00350
	NMA1=NMA+1	CON00360
	DO 1100 N=NMA1,NMAX	CON00370
1100	Y(N)=0.0	CON00380
	DO 1150 N=1,NMAX	CON00390
1150	EN(NSCA,N)=Y(N)	CON00400
1200	RETURN	CON00410
	END	CON00420

<pre> SUBROUTINE ELM(X1,S1,EL) C*** THIS SUBROUTINE DETERMINES THE SMALLEST (EL(1)) AND THE LARGEST C*** (EL(2)) DISTANCES FROM A POINT X1(3) TO THE SURFACES OF AN C*** ELLIPSOID ALONG A LINE OF SIGHT DEFINED BY UNIT VECTOR S1(3). C COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3) DIMENSION X1(3),S1(3),EL(2) AT=0. BT=0. CT=-1. DO 1 K=1,3 X=0. Y=0. DO 2 L=1,3 X=X+RE(K,L)*X1(L) 2 Y=Y+RE(K,L)*S1(L) AT=AT+Y**2/ASQ(K) BT=BT+X*Y/ASQ(K) 1 CT=CT+X**2/ASQ(K) DISC=BT**2-AT*CT IF(DISC.LT.1.E-30)GO TO 10 DISC=SQRT(DISC) EL(1)=-<BT+DISC>/AT EL(2)=-<-BT+DISC>/AT IF(CT.LE.0.)EL(1)=0. IF((EL(1).LT.0.).OR.(EL(2).LT.0.))GO TO 10 RETURN 10 EL(1)=0. EL(2)=0. RETURN END </pre>	<pre> ELM00010 ELM00020 ELM00030 ELM00040 ELM00050 ELM00060 ELM00070 ELM00080 ELM00090 ELM00100 ELM00110 ELM00120 ELM00130 ELM00140 ELM00150 ELM00160 ELM00170 ELM00180 ELM00190 ELM00200 ELM00210 ELM00220 ELM00230 ELM00240 ELM00250 ELM00260 ELM00270 ELM00280 ELM00290 ELM00300 ELM00310 ELM00320 </pre>
--	--

```

SUBROUTINE FIND(U1,PFU)
COMMON /MOS/V1,C1,S1,SA(3),ENC(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASPT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
L=1
LL=LMM1
DO 1 K=1,KMAX
LL=LL/2
L=L+LL
AT=U1-U(L)
IF(AT.GT.1.E-7)L=L-LL
CONTINUE
PFU=PF(L)+(U1-U(L))*(PF(L+1)-PF(L))/(U(L+1)-U(L))
RETURN
END

```

```

FIN00010
FIN00020
FIN00030
FIN00040
FIN00050
FIN00060
FIN00070
FIN00080
FIN00090
FIN00100
FIN00110
FIN00120
FIN00130
FIN00140
FIN00150
FIN00160

```

SUBROUTINE FWRD(SCA,DOM,ELAC)	FWR00010
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK	FWR00020
COMMON/FGEL/XA(3),D,NSCA	FWR00030
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)	FWR00040
COMMON/RNDM/SEED	FWR00050
DIMENSION SCA(3)	FWR00060
PSI=TWOPI*RNDM(SEED)	FWR00070
TH=PI2	FWR00080
CALL ROTAT(TH,PSI,R,SCA)	FWR00090
ANR=RNDM(SEED)	FWR00100
IF(ANR.LE.1.E-30) ANR=0.	FWR00110
RHO=AKM*SQRT(ANR)	FWR00120
ELSQ=0.	FWR00130
DO 1 K=1,3	FWR00140
SCA(K)=XD(K)-XA(K)+SCA(K)*RHO	FWR00150
1 ELSQ=ELSQ+SCA(K)**2	FWR00160
IF(ELSQ.LE.1.E-30) ELSQ=1.E-30	FWR00170
ELAC=SQRT(ELSQ)	FWR00180
DO 3 K=1,3	FWR00190
3 SCA(K)=SCA(K)/ELAC	FWR00200
DOM=AKSQ/(4.*ELSQ)	FWR00210
RETURN	FWR00220
END	FWR00230


```

SUBROUTINE GAS(JTYPE)
C*** THIS SUBROUTINE DETERMINES A RANDOMLY-SELECTED SCATTERING
C*** DIRECTION USED IN PHOTON TRAVERSES WITHIN THE ELLIPSOIDAL
C*** AEROSOL CLOUD. JTYPE=1 SIGNIFIES THAT THE SCATTERING EVENT AT
C*** WHICH THIS ANGLE IS SELECTED IS WITHIN THE AEROSOL CLOUD.
C*** JTYPE=2 SIGNIFIES THAT THE SCATTERING EVENT IS ON THE GROUND
C
COMMON/ALL/AT,BT,CT,BLIM
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON/MO5/V1,C1,S1,SA(3),ENC(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON/BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
COMMON/RNDM/SEED
COMMON/FGEL/XA(3),D,NSCA
AT=Rand (< SEED)
IF(AT.LE.1.E-30) AT=0.
IF(JTYPE.EQ.1)GO TO 1
SA(3)=1.-AT
ARG=1.-SA(3)**2
IF(ARG.LT.1.E-30) ARG=0.
BT=TWOPI*Rand (< SEED)
IF(BT.LE.1.E-30) BT=0.
V1=SQRT(ARG)
C1=cos(BT)
S1=sin(BT)
SA(1)=V1*C1
SA(2)=V1*S1
RETURN
1 CALL USCA(AT,BT)
CT=TWOPI*Rand (< SEED)
IF(CT.LE.1.E-30) CT=0.
ARG=1.-BT**2
IF(ARG.LT.1.E-30) ARG=0.
V=SQRT(ARG)
C=cos(CT)
S=sin(CT)
SA(1)=BT*SA(1)+V*(C*C1*SA(3)-S*S1)
SA(2)=BT*SA(2)+V*(C*S1*SA(3)+S*C1)
SA(3)=BT*SA(3)-V*C*V1
AT=SA(1)**2+SA(2)**2
IF(AT.LT.1.E-10)GO TO 3
V1=SQRT(AT)
C1=SA(1)/V1
S1=SA(2)/V1
RETURN
3 C1=C
S1=S
V1=0.
SA(1)=0.
SA(2)=0.
BT=SA(3)
SA(3)=1.
IF(BT.LT.0.)SA(3)=-1.
RETURN
END
GAS00010
GAS00020
GAS00030
GAS00040
GAS00050
GAS00060
GAS00070
GAS00080
GAS00090
GAS00100
GAS00110
GAS00120
GAS00130
GAS00140
GAS00150
GAS00160
GAS00170
GAS00180
GAS00190
GAS00200
GAS00210
GAS00220
GAS00230
GAS00240
GAS00250
GAS00260
GAS00270
GAS00280
GAS00290
GAS00300
GAS00310
GAS00320
GAS00330
GAS00340
GAS00350
GAS00360
GAS00370
GAS00380
GAS00390
GAS00400
GAS00410
GAS00420
GAS00430
GAS00440
GAS00450
GAS00460
GAS00470
GAS00480
GAS00490
GAS00500
GAS00510
GAS00520
GAS00530
GAS00540
GAS00550
GAS00560
GAS00570
GAS00580

```

	SUBROUTINE GMAX(IMAX, YMAX)	GMA00010
C		GMA00020
C***	THIS SUBROUTINE DETERMINES THE MAXIMUM VALUE YMAX OF AN INPUT	GMA00030
C***	ARRAY Y() OF DIMENSION IMAX.	GMA00040
C		GMA00050
	COMMON/CONB/X(100), Y(100)	GMA00060
	I=0	GMA00070
	YMAX=Y(1)	GMA00080
i	I=I+1	GMA00090
	IF(I.EQ.IMAX)RETURN	GMA00100
	T=Y(I+1)-YMAX	GMA00110
	IF(T.GT.0.)YMAX=Y(I+1)	GMA00120
	GO TO I	GMA00130
	END	GMA00140

C	SUBROUTINE MATRX(TH,PH,S,R)	MAT00010
C***	THIS SUBROUTINE GENERATES UNIT VECTOR S() AND ROTATION MATRIX	MAT00020
C***	R() FOR A SET OF INPUT POLAR ANGLES (TH,PH).	MAT00030
C***	THE ROTATION MATRIX R() ROTATES A VECTOR DEFINED RELATIVE TO THE	MAT00040
C***	(TH,PH) DIRECTION INTO THE STANDARD SYSTEM OF COORDINATES.	MAT00050
C***	THE UNIT VECTOR S() POINTS IN THE (TH,PH) DIRECTION IN THE	MAT00060
C***	STANDARD SYSTEM OF COORDINATES.	MAT00070
C		MAT00080
	COMMON /CONST,PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK	MAT00090
	DIMENSION S(3),R(3,3)	MAT00100
	AT=PIRAD*TH	MAT00110
	BT=PIRAD*PH	MAT00120
	IF(AT.LE.1.E-30) AT=0.	MAT00130
	IF(BT.LE.1.E-30) BT=0.	MAT00140
	V1=SIN(AT)	MAT00150
	C1=COS(BT)	MAT00160
	S1=SIN(BT)	MAT00170
	S(1)=V1*C1	MAT00180
	S(2)=V1*S1	MAT00190
	S(3)=COS(AT)	MAT00200
	R(1,1)=C1*S(3)	MAT00210
	R(1,2)=S1*S(3)	MAT00220
	R(1,3)=-V1	MAT00230
	R(2,1)=-S1	MAT00240
	R(2,2)=C1	MAT00250
	R(2,3)=0.	MAT00260
	DO 1 K=1,3	MAT00270
1	R(3,K)=S(K)	MAT00280
	RETURN	MAT00290
	END	MAT00300
		MAT00310

```

SUBROUTINE ROTAT(TH,PH,R,S)
DIMENSION R(3,3),X(3),S(3)
IF(TH.LE.1.E-30) TH=0.
IF(PH.LE.1.E-30) PH=0.
V=SIN(TH)
X(1)=V*COS(PH)
X(2)=V*SIN(PH)
X(3)=COS(TH)
DO 1 J=1,3
S(J)=0.
DO 1 K=1,3
1 S(J)=S(J)+R(K,J)*X(K)
RETURN
END

```

```

ROT00010
ROT00020
ROT00030
ROT00040
ROT00050
ROT00060
ROT00070
ROT00080
ROT00090
ROT00100
ROT00110
ROT00120
ROT00130
ROT00140

```

	SUBROUTINE SMOOZ(NSCA,NO)	SMO00010
C*****	THIS SUBROUTINE DETERMINES 'NO', THAT VALUE OF I BEYOND WHICH ALL	SMO00020
C*****	Y(I) IN A SEQUENCE ARE ZERO.	SMO00030
	COMMON /MO5/YI,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,	SMO00040
	+NTMAX,NSCAM,KMAX,LMAX,LMM1	SMO00050
	COMMON /BASPT/UK(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),	SMO00060
	+BE(16),SINGWV,PF(65),LLMAX	SMO00070
	I=NTMAX+1	SMO00080
100	I=I-1	SMO00090
	IF(I.EQ.0)GO TO 200	SMO00100
	YI=EN(NSCA,I)	SMO00110
	IF (YI.GT.0.0) GO TO 200	SMO00120
	GO TO 100	SMO00130
200	NO=I	SMO00140
	RETURN	SMO00150
	END	SMO00160

```

SUBROUTINE START(SS)
C*** THIS SUBROUTINE INITIALIZES PHOTON LAUNCH DIRECTION AND CALCULATES
C*** DIRECT BEAM (ZEROth ORDER) CONTRIBUTIONS TO RECEIVED POWER.
COMMON/ALL/AT,BT,CT,BLIM
COMMON/CONB/X(100),Y(100)
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
COMMON/HIT/UDS,THSP,RS(3,3),XS(3),DSA,XV(3)
COMMON/FGEL/XA(3),D,NSCA
COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
*ALB(2),ZG,DMAX
COMMON/RNDM/SEED
COMMON/MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAP,KMAX,LMAX,LMM1
COMMON/BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DIMENSION SS(3)
C*** GENERATE RANDOM THETA (AT) AND PHI (BT) PHOTON LAUNCH ANGLES
C*** CONFINED WITHIN SOURCE CONE.
ANR=RAND(SEED)
IF(ANR.LE.1.E-30) ANR=0.
AT=THSP*SQRT(ANR)
BT=TWOPI*RAND (SEED)
C*** ROTATE LAUNCH VECTOR SA( ) INTO STANDARD FRAME OF REFERENCE.
CALL ROTAT(AT,BT,RS,SA)
C*** DETERMINE HORIZONTAL COMPONENT CT OF LAUNCH VECTOR AND BRANCH
C*** TO VERTICAL TREATMENT (WHICH INCLUDES GROUND PLANE) IF THIS
C*** COMPONENT IS VERY SMALL.
CT=SA(1)**2+SA(2)**2
IF(CT.LE.1.E-24)GO TO 1
V1=SQRT(CT)
C1=SA(1)/V1
S1=SA(2)/V1
GO TO 2
C*** DEFINE VERTICAL UNIT VECTOR
1 V1=0.
CT=SA(3)
SA(3)=1.
IF((CT.LT.0.).AND.(ALB(2).GT.0.)) SA(3)=-1.
C1=1.
S1=0.
SA(1)=0.
SA(2)=0.
C*** INITIALIZE TOTAL TRAVERSE DISTANCE DTOT AND STATISTICAL
C*** STRENGTH STH.
2 DTOT=0.
STH=1.
BT=0.
IF(AT.LE.1.E-30) GO TO 11
DAA=DSA/COS(AT)
DAD=0.
DO 3 K=1,3
XA(K)=XV(K)+DAA*SA(K)
DAD=DAD+Y(K)**2
3 BT=BT+Y(K)*SS(K)
IF(DAD.LE.1.E-30) DAD=0.
DAD=SQRT(DAD)

```

```

STA00010
STA00020
STA00030
ST.00040
STA00050
STA00060
STA00070
STA00080
STA00090
STA00100
STA00110
STA00120
STA00130
STA00140
STA00150
STA00160
STA00170
STA00180
STA00190
STA00200
STA00210
STA00220
STA00230
STA00240
STA00250
STA00260
STA00270
STA00280
STA00290
STA00300
STA00310
STA00320
STA00330
STA00340
STA00350
STA00360
STA00370
STA00380
STA00390
STA00400
STA00410
STA00420
STA00430
STA00440
STA00450
STA00460
STA00470
STA00480
STA00490
STA00500
STA00510
STA00520
STA00530
STA00540
STA00550
STA00560
STA00570
STA00580
STA00590
STA00600
STA00610
STA00620
STA00630
STA00640
STA00650
STA00660
STA00670
STA00680
STA00690
STA00700

```

```

C*** DETERMINE DIRECT BEAM ATTENUATION DISTANCE AND BIASING DISTANCES
C*** FOR INITIAL PHOTON TRAVERSE.
C
CALL ELM(XA,SA,EL)
IF(BT.LE.ALIM) GO TO 4
IF(DAD.LE.(1.001*DSA)) GO TO 4
BT=0.
CT=0.
DO 5 K=1,3
CT=CT+SA(K)*SD(K)
5 BT=BT+Y(K)*SD(K)
CTSQ=CT**2
IF(CTSQ.LE.1.E-30) GO TO 4
ELSD=ABS(BT/CT)-DAA
CT=0.
DO 6 K=1,3
6 CT=CT+(XA(K)+SA(K)*ELSD-XD(K))**2
IF(CT.GT.AKSQ) GO TO 4
IF(EL(2).LE.EL(1)) STH=0.
IF(ELD(1).LE.0.) EL(2)=ELSD
IF(DAD.LT.ALIM) GO TO 4
UDS=-BT/DAD
IF(UDS.LT.UV) RETURN
NT=1.+(ELSD-ELMIN)/DELD
IF(NT.LT.0) GO TO 11
IF(NT.EQ.0) NT=1
EN(1,NT)=EN(1,NT)+EXP(-GAMMA*(EL(2)-EL(1)))
RETURN
4 CONTINUE
IF((EL(2)-EL(1)).GT.1.E-20) RETURN
IF((SA(3).GE.0.).OR.(ALB(2).LE.1.E-20)) STH=0.
RETURN
11 STH=0.
RETURN
END

```

```

STA00710
STA00720
STA00730
STA00740
STA00750
STA00760
STA00770
STA00780
STA00790
STA00800
STA00810
STA00820
STA00830
STA00840
STA00850
STA00860
STA00870
STA00880
STA00890
STA00900
STA00910
STA00920
STA00930
STA00940
STA00950
STA00960
STA00970
STA00980
STA00990
STA01000
STA01010
STA01020
STA01030
STA01040
STA01050

```

SUBROUTINE TRAVRS(JTYPE, I2FLG, ICOND)	TRA00010
COMMON/RNDM/ SEED	TRA00020
COMMON/ALL/AT, BT, CT, BLIM	TRA00030
COMMON/FAHT/SD(3), UV, GAMMA, ELD(2), STH, FAC, ALIM, THV, TAU, EL(2),	TRA00040
*ALB(2), ZG, DMAX	TRA00050
COMMON /M05/V1, C1, S1, SA(3), EN(10, 100), ENC(10), ELMIN, DELD, DTOT,	TRA00060
+NTMAX, NSCAM, KMAX, LMAX, LMM1	TRA00070
COMMON /BASPOT/U(65), SUN(65), WVL(16), NWVL, ALBED(16), BS(16),	TRA00080
+BE(16), SINGWV, PF(65), LLMAX	TRA00090
COMMON/FGEL/XA(3), D, NSCA	TRA00100
COMMON/FWD/AKM, R(3, 3), AKSQ, XD(3), ASQ(3), RE(3, 3), A(3)	TRA00110
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK	TRA00120
COMMON/CONB/X(100), Y(100)	TRA00130
DIMENSION SDA(3)	TRA00140
DGAM=EL(2)-EL(1)	TRA00150
REXIT=.99999	TRA00160
RX=.99999	TRA00170
RXX=.99999	TRA00180
IF((DGAM.LE.0.).AND.(ALB(2).GT.0.))GO TO 9	TRA00190
IF(DGAM.LE.1.E-7) GO TO 70	TRA00200
IF(I2FLG.EQ.1) GO TO 5	TRA00210
IF((SA(3).GE.0.).OR.(ALB(2).LE.0.))REXIT=1.-EXP(-GAMMA*DGAM)	TRA00220
AT=EL(1)-(ALOG(1.-REXIT)*RAND (SEED))/GAMMA	TRA00230
GO TO 4	TRA00240
5 CONTINUE	TRA00250
IF(ELD(1).GT.ALIM)ALIM=ELD(1)	TRA00260
RXX=(1./GAMMA)*((1./ALIM)-(1./DGAM))	TRA00270
DENOM=((1./ALIM)-GAMMA*RXX*RAND(SEED))	TRA00280
IF(DENOM.LE.1.E-7) GO TO 70	TRA00290
AT=1./DENOM	TRA00300
RX=EXP(-(GAMMA*AT))*(AT*GAMMA)**2	TRA00310
4 BT=XA(3)+SA(3)*AT	TRA00320
IF((BT.GT.ZG).AND.(AT.LE.EL(2)))GO TO 2	TRA00330
9 JTYPE=2	TRA00340
AT=(ZG-XA(3))/SA(3)	TRA00350
GO TO 1	TRA00360
2 JTYPE=1	TRA00370
1 STH=ALB(JTYPE)*REXIT*STH*RXX	TRA00380
IF(STH.LE.0.)RETURN	TRA00390
DO 3 K=1,3	TRA00400
3 XA(K)=XA(K)+AT*SA(K)	TRA00410
DTOT=DTOT+AT	TRA00420
IF(NSCA.NE.2) GO TO 50	TRA00430
AT=0.	TRA00440
DO 8 K=1,3	TRA00450
8 SDA(K)=XD(K)-XA(K)	TRA00460
AT=AT+SDA(K)*SDA(K)	TRA00470
IF(AT.LT.BLIM)GO TO 15	TRA00480
AT=SQRT(AT)	TRA00490
BT=0.	TRA00500
DO 32 K=1,3	TRA00510
SDA(K)=SDA(K)/AT	TRA00520
32 BT=BT-SDA(K)*SD(K)	TRA00530
IF(BT.LT.UV) GO TO 50	TRA00540
CALL FWRD(SDA, DOM, AT)	TRA00550
NT=1.+(DTOT+AT-ELMIN)/DELD	TRA00560
IF(NT.LT.0) RETURN	TRA00570
IF(NT.EQ.0) NT=1	TRA00580
IF((NT.EQ.1).AND.(I2FLG.EQ.0).AND.(ICOND.EQ.1))GO TO 50	TRA00590
IF(NT.GT.NTMAX)RETURN	TRA00600
CT=0.	TRA00610
BT=0.	TRA00620
DO 7 K=1,3	TRA00630
BT=BT-SDA(K)*SD(K)	TRA00640
7 CT=CT+SDA(K)*SA(K)	TRA00650
PFI=1.	TRA00660
IF(JTYPE.EQ.1)CALL FIND(CT, PFI)	TRA00670
CALL ELM(XA, SDA, EL)	TRA00680
AT=AT-EL(1)	TRA00690
IF(ELD(1).GT.0.)AT=EL(2)-EL(1)	TRA00700


```

DOM=DOM*PFI*STH*EXP(-GAMMA*AT)*BT*RX
IF(I2FLG.NE.1) GO TO 13
IF(NT.EQ.1) DOM=DOM*10.
IF(NT.GT.1) DOM=0.
GO TO 14
13 CONTINUE
IF(ICOND.EQ.1) DOM=DOM/0.9
14 EN(NSCA,NT)=EN(NSCA,NT)+DOM
GO TO 50
15 ENC(2)=ENC(2)-1.
STH=0.
RETURN
50 CONTINUE
IF(I2FLG.EQ.1) GO TO 60
CALL BKWD(JTYPE)
60 RETURN
70 ENC(NSCA)=ENC(NSCA)-1.
STH=0.
RETURN
END

```

```

TRA00710
TRA00720
TRA00730
TRA00740
TRA00750
TRA00760
TRA00770
TRA00780
TRA00790
TRA00800
TRA00810
TRA00820
TRA00830
TRA00840
TRA00850
TRA00860
TRA00870
TRA00880
TRA00890
TRA00900

```

SUBROUTINE USCA(SC,US)	USC00010
COMMON /M05/V1,C1,S1,SA(3),ENC(10,100),ENC(10),ELMIN,DELD,DTOT,	USC00020
+NTMAX,NSCAM,KMAX,LMAX,LMM1	USC00030
COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),	USC00040
+BE(16),SINGWV,PF(65),LLMAX	USC00050
L=1	USC00060
LL=LMM1	USC00070
DO 1 K=1,KMAX	USC00080
LL=LL/2	USC00090
L=L+LL	USC00100
IF(SUM(L).GT.SC)L=L-LL	USC00110
1 CONTINUE	USC00120
US=U(L)+(SC-SUM(L))*(U(L+1)-U(L))/(SUM(L+1)-SUM(L))	USC00130
RETURN	USC00140
END	USC00150

PER FIELD BEGINNING IN COL 11. THE CARDS ARE NOT ORDER
DEPENDENT ALTHOUGH VALUES ON EACH RECORD MUST FOLLOW THE ORDER
SHOWN IN COMMENTS BELOW. FORMAT (2A2,6X,7F10.3)

SMK00430
SMK00440
SMK00450
SMK00460
SMK00470
SMK00480
SMK00490
SMK00500
SMK00510
SMK00520

THE FOLLOWING ARE REQUIRED RECORDS FOR AT LEAST ONE INPUT SET.

IDENT.	VARS.	DESCRIPTION	
MUNC	XM,YM,ZM,TM (M,M,M,SEC)	MUNITION COORDINATES AND EVENT TIME	SMK00530
BART	STO ETO DIO ANGLX	COMPUTATION TIMES AND X-AXIS DEFINITION: OUTPUT STARTING TIME (SEC. SINCE IGNITION) ENDING TIME FOR CALCULATION (SEC.) TIME INCREMENT FOR OUTPUT TABLES (SEC) ANGLE OF POSITIVE X-AXIS WRT NORTH (DEG. CLOCKWISE WRT NORTH) ASSUMED 90 DEG. IF IGEOSW IS 1 FROM EOMAIN	SMK00540 SMK00550 SMK00560 SMK00570 SMK00580 SMK00590 SMK00600 SMK00610 SMK00620 SMK00630 SMK00640

OPTIONAL DEPENDING ON PARAMETERS CHOSEN:

IDENT.	VARS.	DESCRIPTION	
OBSC	XO,YO,ZO	OBSERVER COORDINATES (IGNORED IF IGEOSW IS 1 FROM EOMAIN). (M,M,M)	SMK00650 SMK00660 SMK00670
TARC	XT,YT,ZT	TARGET COORDINATES (IGNORED IF IGEOSW IS 1 FROM EOMAIN) (M,M,M)	SMK00680 SMK00690
OUTP		OPTIONAL RECORD TO SELECT AMOUNT OF PRINT OUTPUT AND TO SELECT CRITERIA FOR RETURNED TRANSMISSION.	SMK00700 SMK00710 SMK00720
PRNT		IF 0., ALL FULL OUTPUT LISTINGS CREATED. IF 1. IS ENTERED AT ANY POINT, THEN ALL FURTHER OUTPUT IS SUPPRESSED, EXCEPT FOR THE FINAL ACCUMULATED EFFECTS LISTING OF TOTAL CL AND TOTAL TRANSMITTANCE FOR COMBINED MULTIPLE INPUT SETS. (DEFAULT IS PRNT = 0.)	SMK00730 SMK00740 SMK00750 SMK00760 SMK00770 SMK00780
CRITER		SELECTS CHOICE OF TRANSMISSION RETURNED FROM SMOKE. 0. = RETURN TOTAL TRANSMITTANCE COMPUTED AT LAST TIME ETO. (DEFAULT CASE) 1. = RETURN THE MINIMUM VALUE OF TOTAL TRANSMITTANCE COMPUTED FOR WAVELENGTH WAVE1. 2. = RETURN VALUE OF TOTAL TRANSMITTANCE FOR WAVELENGTH WAVE1 COMPUTED AT USER- SPECIFIED TIME TIMTRN BELOW.	SMK00790 SMK00800 SMK00810 SMK00820 SMK00830 SMK00840 SMK00850 SMK00860 SMK00870 SMK00880
TIMTRN		REQUIRED ONLY IF CRITER IS 2., TIME FOR WHICH TRANSMITTANCE RETURNED IS COMPUTED. SHOULD BE CLOSE OR EQUAL TO A TABLE TIME (AS DETERMINED BY THE BART RECORD) FOR ACCURACY.	SMK00890 SMK00900
IPLT		PLOT CODE ADDED IN ORDER TO PLOT OUTPUT ON UNIT NPLTU IF IPLT = 1.	SMK00910 SMK00920
MUNT		REQUIRED IF BURN RECORD IS NOT USED. OTHERWISE, OPTIONAL. ANY NON-ZERO VALUES INPUT WILL OVER- RIDE PREVIOUS SOURCE DEFINITIONS (INCLUDING THOSE FROM THE BURN RECORD.)	SMK00930 SMK00940 SMK00950 SMK00960 SMK00970 SMK00980 SMK00990
XN		NUMBER OF MUNITIONS IGNITED AT THE SAME LOCATION AND AT THE SAME TIME (DIMENSIONLESS)	SMK01000
FW		FILL WEIGHT OF ONE MUNITION (LBS.) FOR WP,PWP,HC OR RP (BUT RATE OF BURN IN GAL./HR. FOR FOG OIL. NOTE 1 GAL/HR= 0.93 G/S)	SMK01010 SMK01020 SMK01030 SMK01040
TBURN TYPE		BURN DURATION FOR THIS MUNITION (SEC) TYPE OF SMOKE (DIMENSIONLESS) 1.=WP, 2.=PWP OR WP WICK/WEDGE, 3.=HC, 4.=FOG OIL, 5.=RP	SMK01050 SMK01060 SMK01070 SMK01080
EFF		MUNITION BURN EFFICIENCY. (PERCENT)	SMK01090

	IF INPUT AS 0., THEN DEFAULT IS USED.	SMK01100
YF	YIELD FACTOR (DIMENSIONLESS). IF 0., DEFAULTS TO RELATIVE HUMIDITY DEPENDENT STRAIGHT LINE FIT FOR WP, PWP, RP FROM JOHNSON AND FORNEY. SIMILARLY FOR HC. FOG OIL IS SET TO 1.	SMK01110 SMK01120 SMK01130 SMK01140 SMK01150 SMK01160 SMK01170
METR	REQUIRED IF ICLMAT IS ZERO. OTHERWISE, NEEDED ONLY FOR E OR F PASQUILL CATEGORY TO PROVIDE TGRAD WHICH IS NOT AVAILABLE IN /CLYMAT/.	SMK01180 SMK01190 SMK01200 SMK01210 SMK01220 SMK01230 SMK01240
RELHUM	RELATIVE HUMIDITY (PERCENT)	SMK01250
UW	WIND VELOCITY (M/SEC)	SMK01260
WDIR	WIND DIRECTION (USUAL MET CONVENTION, ANGLE IN DEG. CLOCKWISE FROM NORTH OF DIRECTION FROM WHICH WIND ORIGINATES.)	SMK01270 SMK01280 SMK01290 SMK01300
PCAT	PASQUILL CATEGORY (DIMENSIONLESS) 1.-A, 2.-B, 3.-C, 4.-D, 5.-E, 6.-F	SMK01310 SMK01320 SMK01330
AIRT	SURFACE AIR TEMPERATURE (DEG C)	SMK01340
TGRAD	VERT TEMP GRADIENT (C DEG/M) EXAMPLE: TGRAD=(AIRT(10 M)-AIRT(.5 M))/9.5 M (USED ONLY FOR PASQUILL CATEGORIES E, F)	SMK01350 SMK01360 SMK01370 SMK01380 SMK01390 SMK01400 SMK01410 SMK01420 SMK01430 SMK01440 SMK01450 SMK01460 SMK01470 SMK01480 SMK01490
EXTC	OPTIONAL USER OVERRIDE FOR EXTINCTION COEFFICIENTS. IF RECORD NOT USED, OR FOR ANY VALUES READ IN AS 0., THE EXTINCTION COEFF. DEFAULTS TO ALPHA ARRAY VALUE IN STRANS. INPUT EXTINCTION COEFF. (M**2/G) ORDER ON CARD CORRESPONDS TO THE BANDS: 0.4-0.7 MICROMETERS 0.7-1.2 MICROMETERS 1.06 MICROMETERS 3.0-5.0 MICROMETERS 8.0-12. MICROMETERS 10.6 MICROMETERS 94.0 GHZ.	SMK01500 SMK01510 SMK01520 SMK01530 SMK01540 SMK01550 SMK01560 SMK01570 SMK01580 SMK01590 SMK01600 SMK01610 SMK01620 SMK01630 SMK01640 SMK01650 SMK01660 SMK01670 SMK01680 SMK01690 SMK01700 SMK01710 SMK01720 SMK01730 SMK01740 SMK01750 SMK01760 SMK01770 SMK01780 SMK01790
BURN	OPTIONAL - SELECTS BUILT-IN MUNITION CHARACTERISTICS FROM THE BRATE ROUTINE FOR ONE (XN=1) MUNITION. VALUES ARE FOR FILL WEIGHT (FW), BURN DURATION (TBURN), SMOKE TYPE (ITYPE), EFFICIENCY (EFF). YIELD FACTOR IS SET TO ZERO SO THAT RH MODEL DEPENDENT VALUES ARE USED. ANY VALUES READ IN AS NON-ZERO ON A MUNT RECORD (WHICH IS OPTIONAL IF A BURN CARD IS USED) WILL OVERRIDE THE DEFAULTS STORED IN BRATE.	SMK01800 SMK01810 SMK01820 SMK01830 SMK01840 SMK01850 SMK01860 SMK01870 SMK01880 SMK01890 SMK01900 SMK01910 SMK01920 SMK01930 SMK01940 SMK01950 SMK01960 SMK01970 SMK01980 SMK01990
TYPM	MUNITION TYPE: 0. = USER DEFINED MUNITION SOURCE CHAR. 1. = 155MM HC, M1 CANISTER. 2. = 155MM HC, M2 CANISTER. 3. = 105MM HC CANISTER. 4. = 155MM HC M116B1 PROJ. 5. = 105MM HC M84A1 PROJ. 6. = SMOKE POT HC M5 7. = 60MM WP M302 CARTRIDGE 8. = 81MM WP M375A2 9. = 4.2 IN WP M328A1 10. = 155MM WP M110E2 11. = 105MM WP M60A2 12. = 4.2 IN PWP M328A1 13. = 5. IN PWP ZUNI MK4 14. = 2.75 IN WP WEDGE SUB-MUNITION. 15. = 3. IN WP WICK SUB-MUNITION 16. = 6. IN WP WICK SUB-MUNITION 17. = 155MM WP WEDGE XM825 (92 SUB-MUN.) 18. = 81MM RP WEDGE NAVY SUB-MUNITION	SMK02000 SMK02010 SMK02020 SMK02030 SMK02040 SMK02050 SMK02060 SMK02070 SMK02080 SMK02090 SMK02100 SMK02110 SMK02120 SMK02130 SMK02140 SMK02150 SMK02160 SMK02170 SMK02180 SMK02190 SMK02200 SMK02210 SMK02220 SMK02230 SMK02240 SMK02250 SMK02260 SMK02270 SMK02280 SMK02290 SMK02300 SMK02310 SMK02320 SMK02330 SMK02340 SMK02350 SMK02360 SMK02370 SMK02380 SMK02390 SMK02400


```

* 2HPW,2HP,2HOR,2H W,2HP,2HWI,2HCK,2H/W,2HED,2HGE,2H,
* 2HHC,2H S,2HMO,2HKE,2H,2H,2H,2H,2H,2H,2H,
* 2HFO,2HG,2HOI,2HL,2HSM,2HOK,2HE,2H,2H,2H,2H,
* 2HRE,2HD,2HPP,2HQS,2HPPH,2HOR,2HUS,2H,2H,2H,2H,
DATA IR,2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA,
*2HRC,2HEX,2HTC,2HGO,2H,2HBU,2HRN,2HMI,2HSC,2HOU,2HTP,2HDO,
*2HNE,2HNA,2HME /
DATA BRAT1,BRAT2,BRAT3,BRAT4,BRAT5/1.,0.,0.,0.,1./
DATA TYPM,XN,FW,TBURN,EFF,YF,RELHUM,UW,WDIR,ANGLX,PCAT,
*AIRT,TGRAD /13*0./
DATA ITGRAD/0/
BFUNCT)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.))
**BRAT5*.07818288*ALOG(1.+358800.*T)
NUMDIV=8
NRUNS=0
TIMTRN=1.
ICRTR=0
TRANS=1.
MCUOPT=0
NOPRNT=0
NOMORE=0
NCY=0
DO 1 J=1,MAXS
1 CLTOT(J)=0.
CONTINUE
DO 2 J=1,8
5 2 EXTC(J)=0.
CONTINUE
NRUNS=NRUNS+1
MUNRD=0
KWAWE=0
IF (KWAWE1.GE.0.4.AND.WAVE1.LT.0.7) KWAWE=1
IF (KWAWE1.GE.0.7.AND.WAVE1.LE.1.2) KWAWE=2
IF (KWAWE1.GE.3.0.AND.WAVE1.LE.5.0) KWAWE=4
IF (KWAWE1.GE.8.0.AND.WAVE1.LE.12.0) KWAWE=5
IF (KWAWE1.GT.1.059.AND.WAVE1.LT.1.061) KWAWE=3
IF (KWAWE1.GT.10.59.AND.WAVE1.LT.10.61) KWAWE=6
IF (KWAWE1.GT.93.9.AND.WAVE1.LT.94.1) KWAWE=7
IF (KWAWE1.GT.3188.AND.WAVE1.LT.3195.) KWAWE=7
IF (KWAWE.EQ.0) GOTO 998
C*** BEGINNING OF READ LOOP
NCHK=0
DO 70 I = 1, 15
IF (I.EQ.15) GO TO 310
20 READ(IOIN,20) IR1,IR2,(R1(J),J=1,7)
FORMAT(2A2,6X,7F10,3)
C*** RELATING INPUT DATA TO VARIABLE NAMES.
IF (IR1.EQ.IR(1).AND.IR2.EQ.IR(2)) GOTO 90
IF (IR1.EQ.IR(3).AND.IR2.EQ.IR(4)) GOTO 100
IF (IR1.EQ.IR(5).AND.IR2.EQ.IR(6)) GOTO 110
IF (IR1.EQ.IR(7).AND.IR2.EQ.IR(8)) GOTO 120
IF (IR1.EQ.IR(9).AND.IR2.EQ.IR(10)) GOTO 130
IF (IR1.EQ.IR(11).AND.IR2.EQ.IR(12)) GOTO 140
IF (IR1.EQ.IR(13).AND.IR2.EQ.IR(14)) GOTO 150
IF (IR1.EQ.IR(15).AND.IR2.EQ.IR(16)) GOTO 155
IF (IR1.EQ.IR(17).AND.IR2.EQ.IR(18)) GOTO 105
IF (IR1.EQ.IR(19).AND.IR2.EQ.IR(20)) GOTO 70
IF (IR1.EQ.IR(21).AND.IR2.EQ.IR(22)) GOTO 115
IF (IR1.EQ.IR(23).AND.IR2.EQ.IR(24)) GOTO 154
IF (IR1.EQ.IR(25).AND.IR2.EQ.IR(26)) GOTO 121
WRITE(IOOUT,80)
80 FORMAT(1H,72HINVALID DATA CARD-DOES NOT CONFORM TO PROPER CO
*NVENTION IN SMOKE ROUTINE)
WRITE(IOOUT,30) IR1,IR2,(R1(J),J=1,7)
30 FORMAT(1H,2A2,6X,7F10,3)
GO TO 999
90 IF (ICLMAT.EQ.1) GOTO 92
IF (R1(1).NE.0.) RELHUM= R1(1)
IF (R1(2).NE.0.) UW = R1(2)
IF (R1(3).NE.0..OR.R1(2).NE.0.) WDIR = R1(3)
SMK02450
SMK02451
SMK02452
SMK02453
SMK02454
SMK02455
SMK02456
SMK02457
SMK02458
SMK02459
SMK02460
SMK02461
SMK02462
SMK02463
SMK02464
SMK02465
SMK02466
SMK02467
SMK02468
SMK02469
SMK02470
SMK02471
SMK02472
SMK02473
SMK02474
SMK02475
SMK02476
SMK02477
SMK02478
SMK02479
SMK02480
SMK02481
SMK02482
SMK02483
SMK02484
SMK02485
SMK02486
SMK02487
SMK02488
SMK02489
SMK02490
SMK02491
SMK02492
SMK02493
SMK02494
SMK02495
SMK02496
SMK02497
SMK02498
SMK02499
SMK02500
SMK02501
SMK02502
SMK02503
SMK02504
SMK02505
SMK02506
SMK02507
SMK02508
SMK02509
SMK02510
SMK02511
SMK02512
SMK02513
SMK02514
SMK02515
SMK02516
SMK02517
SMK02518
SMK02519
SMK02520
SMK02521
SMK02522
SMK02523
SMK02524
SMK02525
SMK02526
SMK02527
SMK02528
SMK02529
SMK02530
SMK02531
SMK02532
SMK02533
SMK02534
SMK02535
SMK02536
SMK02537
SMK02538
SMK02539
SMK02540
SMK02541
SMK02542
SMK02543
SMK02544
SMK02545
SMK02546
SMK02547
SMK02548
SMK02549
SMK02550
SMK02551
SMK02552
SMK02553
SMK02554
SMK02555
SMK02556
SMK02557
SMK02558
SMK02559
SMK02560
SMK02561
SMK02562
SMK02563
SMK02564
SMK02565
SMK02566
SMK02567
SMK02568
SMK02569
SMK02570
SMK02571
SMK02572
SMK02573
SMK02574
SMK02575
SMK02576
SMK02577
SMK02578
SMK02579
SMK02580
SMK02581
SMK02582
SMK02583
SMK02584
SMK02585
SMK02586
SMK02587
SMK02588
SMK02589
SMK02590
SMK02591
SMK02592
SMK02593
SMK02594
SMK02595
SMK02596
SMK02597
SMK02598
SMK02599
SMK02600
SMK02601
SMK02602
SMK02603
SMK02604
SMK02605
SMK02606
SMK02607
SMK02608
SMK02609
SMK02610
SMK02611
SMK02612
SMK02613
SMK02614
SMK02615
SMK02616
SMK02617
SMK02618
SMK02619
SMK02620
SMK02621
SMK02622
SMK02623
SMK02624
SMK02625
SMK02626
SMK02627
SMK02628
SMK02629
SMK02630
SMK02631
SMK02632
SMK02633
SMK02634
SMK02635
SMK02636
SMK02637
SMK02638
SMK02639
SMK02640
SMK02641
SMK02642
SMK02643
SMK02644
SMK02645
SMK02646
SMK02647
SMK02648
SMK02649
SMK02650
SMK02651
SMK02652
SMK02653
SMK02654
SMK02655
SMK02656
SMK02657
SMK02658
SMK02659
SMK02660
SMK02661
SMK02662
SMK02663
SMK02664
SMK02665
SMK02666
SMK02667
SMK02668
SMK02669
SMK02670
SMK02671
SMK02672
SMK02673
SMK02674
SMK02675
SMK02676
SMK02677
SMK02678
SMK02679
SMK02680
SMK02681
SMK02682
SMK02683
SMK02684
SMK02685
SMK02686
SMK02687
SMK02688
SMK02689
SMK02690
SMK02691
SMK02692
SMK02693
SMK02694
SMK02695
SMK02696
SMK02697
SMK02698
SMK02699
SMK02700
SMK02701
SMK02702
SMK02703
SMK02704
SMK02705
SMK02706
SMK02707
SMK02708
SMK02709
SMK02710
SMK02711
SMK02712
SMK02713
SMK02714
SMK02715
SMK02716
SMK02717
SMK02718
SMK02719
SMK02720
SMK02721
SMK02722
SMK02723
SMK02724
SMK02725
SMK02726
SMK02727
SMK02728
SMK02729
SMK02730
SMK02731
SMK02732
SMK02733
SMK02734
SMK02735
SMK02736
SMK02737
SMK02738
SMK02739
SMK02740
SMK02741
SMK02742
SMK02743
SMK02744
SMK02745
SMK02746
SMK02747
SMK02748
SMK02749
SMK02750
SMK02751
SMK02752
SMK02753
SMK02754
SMK02755
SMK02756
SMK02757
SMK02758
SMK02759
SMK02760
SMK02761
SMK02762
SMK02763
SMK02764
SMK02765
SMK02766
SMK02767
SMK02768
SMK02769
SMK02770
SMK02771
SMK02772
SMK02773
SMK02774
SMK02775
SMK02776
SMK02777
SMK02778
SMK02779
SMK02780
SMK02781
SMK02782
SMK02783
SMK02784
SMK02785
SMK02786
SMK02787
SMK02788
SMK02789
SMK02790
SMK02791
SMK02792
SMK02793
SMK02794
SMK02795
SMK02796
SMK02797
SMK02798
SMK02799
SMK02800
SMK02801
SMK02802
SMK02803
SMK02804
SMK02805
SMK02806
SMK02807
SMK02808
SMK02809
SMK02810
SMK02811
SMK02812
SMK02813
SMK02814
SMK02815
SMK02816
SMK02817
SMK02818
SMK02819
SMK02820
SMK02821
SMK02822
SMK02823
SMK02824
SMK02825
SMK02826
SMK02827
SMK02828
SMK02829
SMK02830
SMK02831
SMK02832
SMK02833
SMK02834
SMK02835
SMK02836
SMK02837
SMK02838
SMK02839
SMK02840
SMK02841
SMK02842
SMK02843
SMK02844
SMK02845
SMK02846
SMK02847
SMK02848
SMK02849
SMK02850
SMK02851
SMK02852
SMK02853
SMK02854
SMK02855
SMK02856
SMK02857
SMK02858
SMK02859
SMK02860
SMK02861
SMK02862
SMK02863
SMK02864
SMK02865
SMK02866
SMK02867
SMK02868
SMK02869
SMK02870
SMK02871
SMK02872
SMK02873
SMK02874
SMK02875
SMK02876
SMK02877
SMK02878
SMK02879
SMK02880
SMK02881
SMK02882
SMK02883
SMK02884
SMK02885
SMK02886
SMK02887
SMK02888
SMK02889
SMK02890
SMK02891
SMK02892
SMK02893
SMK02894
SMK02895
SMK02896
SMK02897
SMK02898
SMK02899
SMK02900
SMK02901
SMK02902
SMK02903
SMK02904
SMK02905
SMK02906
SMK02907
SMK02908
SMK02909
SMK02910
SMK02911
SMK02912
SMK02913
SMK02914
SMK02915
SMK02916
SMK02917
SMK02918
SMK02919
SMK02920
SMK02921
SMK02922
SMK02923
SMK02924
SMK02925
SMK02926
SMK02927
SMK02928
SMK02929
SMK02930
SMK02931
SMK02932
SMK02933
SMK02934
SMK02935
SMK02936
SMK02937
SMK02938
SMK02939
SMK02940
SMK02941
SMK02942
SMK02943
SMK02944
SMK02945
SMK02946
SMK02947
SMK02948
SMK02949
SMK02950
SMK02951
SMK02952
SMK02953
SMK02954
SMK02955
SMK02956
SMK02957
SMK02958
SMK02959
SMK02960
SMK02961
SMK02962
SMK02963
SMK02964
SMK02965
SMK02966
SMK02967
SMK02968
SMK02969
SMK02970
SMK02971
SMK02972
SMK02973
SMK02974
SMK02975
SMK02976
SMK02977
SMK02978
SMK02979
SMK02980
SMK02981
SMK02982
SMK02983
SMK02984
SMK02985
SMK02986
SMK02987
SMK02988
SMK02989
SMK02990
SMK02991
SMK02992
SMK02993
SMK02994
SMK02995
SMK02996
SMK02997
SMK02998
SMK02999
SMK03000
SMK03001
SMK03002
SMK03003
SMK03004
SMK03005
SMK03006
SMK03007
SMK03008
SMK03009
SMK03010
SMK03011
SMK03012
SMK03013
SMK03014
SMK03015
SMK03016
SMK03017
SMK03018
SMK03019
SMK03020
SMK03021
SMK03022
SMK03023
SMK03024
SMK03025
SMK03026
SMK03027
SMK03028
SMK03029
SMK03030
SMK03031
SMK03032
SMK03033
SMK03034
SMK03035
SMK03036
SMK03037
SMK03038
SMK03039
SMK03040
SMK03041
SMK03042
SMK03043
SMK03044
SMK03045
SMK03046
SMK03047
SMK03048
SMK03049
SMK03050
SMK03051
SMK03052
SMK03053
SMK03054
SMK03055
SMK03056
SMK03057
SMK03058
SMK03059
SMK03060
SMK03061
SMK03062
SMK03063
SMK03064
SMK03065
SMK03066
SMK03067
SMK03068
SMK03069
SMK03070
SMK03071
SMK03072
SMK03073
SMK03074
SMK03075
SMK03076
SMK03077
SMK03078
SMK03079
SMK03080
SMK03081
SMK03082
SMK03083
SMK03084
SMK03085
SMK03086
SMK03087
SMK03088
SMK03089
SMK03090
SMK03091
SMK03092
SMK03093
SMK03094
SMK03095
SMK03096
SMK03097
SMK03098
SMK03099
SMK03100
SMK03101
SMK03102
SMK03103
SMK03104
SMK03105
SMK03106
SMK03107
SMK03108
SMK03109
SMK03110
SMK03111
SMK03112
SMK03113
SMK03114
SMK03115
SMK03116
SMK03117
SMK03118
SMK03119
SMK03120

```

	IF (R1(4).NE.0.) PCAT = R1(4)	SMK03130
	ICAT=IFIX(PCAT+.0001)	SMK03140
92	AIRT = R1(5)	SMK03150
	TGRAD = R1(6)	SMK03160
	ITGRAD=1	SMK03170
	NCHK=1	SMK03180
	GO TO 70	SMK03190
100	IF (R1(1).NE.0.) XN = R1(1)	SMK03200
	IF (R1(2).NE.0.) FW = R1(2)	SMK03210
	IF (R1(3).NE.0.) TBURN = R1(3)	SMK03220
	IF (R1(4).NE.0.) ITYPE = IFIX(R1(4)+.0001)	SMK03230
	IF (MUNRD.EQ.0.OR.R1(5).NE.0.) EFF = R1(5)	SMK03240
	YF=R1(6)	SMK03250
	MUNRD=1	SMK03260
	NCHK=1	SMK03270
	GO TO 70	SMK03280
105	TYPM=R1(1)	SMK03290
	CALL BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,BRAT1,	SMK03300
	*BRAT2,BRAT3,BRAT4,BRAT5)	SMK03310
	IF (IERR.NE.0) WRITE (IOOUT,95) TYPM	SMK03320
95	FORMAT(37H IN SMOKE, ILLEGAL MUNITION TYPE READ ,F5.0)	SMK03330
	IF (IERR.NE.0) GOTO 999	SMK03340
	IF (R1(2).EQ.0..AND.R1(3).EQ.0..AND.R1(4).EQ.0..AND.R1(5).EQ.0.)	SMK03350
	GOTO 93	SMK03360
	*BRAT1=R1(2)	SMK03370
	BRAT2=R1(3)	SMK03380
	BRAT3=R1(4)	SMK03390
	BRAT4=R1(5)	SMK03400
	BRAT5=R1(6)	SMK03410
93	IF (TYPM.GT.0.) MUNRD=1	SMK03420
	NCHK=1	SMK03430
	GO TO 70	SMK03440
110	ISTO = IFIX(R1(1)+.0001)	SMK03450
	IETO = IFIX(R1(2)+.0001)	SMK03460
	IDTO = IFIX(R1(3)+.0001)	SMK03470
	ANGLX=R1(4)	SMK03480
	NCHK=1	SMK03490
	GO TO 70	SMK03500
115	NOPRNT=0	SMK03510
	IF (R1(1).NE.0.) NOPRNT=1	SMK03520
	CRITER=R1(2)	SMK03530
	ICRTR=IFIX(CRITER+.001)	SMK03540
	IF (ICRTR.GT.2) ICRTR=2	SMK03550
	IF (ICRTR.LT.0) ICRTR=0	SMK03560
	IF (ICRTR.EQ.2) TIMTRN=R1(3)	SMK03570
	IPLT=IFIX(R1(4))	
	GOTO 70	SMK03580
120	XM = R1(1)	SMK03590
	YM = R1(2)	SMK03600
	ZM = R1(3)	SMK03610
	TM = R1(4)	
	NCHK=1	SMK03620
	GO TO 70	SMK03630
121	MODE=1	
	STIME =R1(1)	
	ISRN =IFIX(STIME+0.0001)	
	FRONT =R1(2)	
	DELX =R1(3)	
	IF(DELX.LE.0.0) DELX=5.0	
	XXX=FRONT/DELX	
	NPTS=IFIX(XXX)+1	
	MCUOPT=IFIX(R1(4))	
	GO TO 70	
130	XO = R1(1)	SMK03640
	YO = R1(2)	SMK03650
	ZO = R1(3)	SMK03660
	NCHK=1	SMK03670
	GO TO 70	SMK03680
140	XT = R1(1)	SMK03690
	YT = R1(2)	SMK03700


```

      ZT      = R1(3)
      NCHK=1
      GO TO 70
150 DO 152 J=1,7
152 EXTC(J)=R1(J+1)
      EXTC(8)=0.
70 CONTINUE
154 NOMORE=1
      IF (NCHK.EQ.0) NRUNS=NRUNS-1
      IF (NCHK.EQ.0) GOTO 980
155 CONTINUE
C*****REDEFINE MUNITION EFFICIENCY IF INPUT AS ZERO
      IF (EFF.GT.0.0) GO TO 11
      IF (ITYPE.EQ.1) EFF=100.
      IF (ITYPE.EQ.2) EFF=65.0
      IF (ITYPE.EQ.3) EFF=40.0
      IF (ITYPE.EQ.4) EFF=100.0
      IF (ITYPE.EQ.5) EFF=50.
11 CONTINUE
      IF (ICLMAT.NE.1) GOTO 12
      RELHUM=RH
      UW      =WINDVEL
      WNDIR=WNDDIR
      ICAT   =IPASCT
      PCAT=FLOAT(ICAT)
      AIRT  =TEMP
12 CONTINUE
      IF (IGEOSW.NE.1) GOTO 13
      DISKTM=1000.
      CONVERT UNITS FROM KM TO M.
      XT=PTS(1)*DISKTM
      YT=PTS(2)*DISKTM
      ZT=PTS(3)*DISKTM
      XO=PTS(4)*DISKTM
      YO=PTS(5)*DISKTM
      ZO=PTS(6)*DISKTM
      ANGLX=90.
13 CONTINUE
      IF (ITYPE.LT.1.OR.ITYPE.GT.5) IERR=1
      IF (IERR.EQ.1) WRITE (IOOUT,180) ITYPE
180 FORMAT(1X,31HIN SMOKE, INVALID SMOKE TYPE = ,I4)
      IF (IERR.EQ.1) GOTO 999
C*** CHECK BURN RATE FOR 100 PERCENT BURN AT TBURN...
      IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0..AND.BRAT5.EQ.0.)
*BRAT1=1.
      VNORM=BFUN(1.)
      BRAT1=BRAT1/VNORM
      BRAT2=BRAT2/VNORM
      BRAT3=BRAT3/VNORM
      BRAT4=BRAT4/VNORM
      BRAT5=BRAT5/VNORM
      IF (XN.LE.0.) XN=1.
      IF (ITYPE.EQ.1.AND.TBURN.GT.1.) TBURN=1.
C*** SET UP EXTINCTION COEFF TO BE USED...
      CALL STRANS(CL,SMTRAN,ITYPE,EXTC,0)
      CALL SMASSP(XN,EFF,FW,RELHUM,W,ITYPE,YF)
      TGR=TGRAD
      TGRAD=ABS(TGRAD)
      IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 255
      IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 255
      IF (NRUNS.GT.1) WRITE (IOOUT,172) NRUNS
      IF (NRUNS.EQ.1) WRITE (IOOUT,1720) NRUNS
172 FORMAT(1H1,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,
*1H*,15X,9HEXECUTION ,13/51X,1H*,15X,1H*/51X,17(1H*)/)
1720 FORMAT(1H0,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,
*1H*,15X,9HEXECUTION ,13/51X,1H*,15X,1H*/51X,17(1H*)/)
C*** REPORTING INPUT DATA.
      IF (ICAT.GE.5.AND.ITYPE.NE.4.AND.ITGRAD.EQ.0) WRITE (IOOUT,98) TGRAD
98 FORMAT(1X,44HIN SMOKE ROUTINE PASQUILL CATEGORIES E AND F,
*29H REQUIRE TEMPERATURE GRADIENT/10X,23HIF SMOKE IS EXOTHERMIC.,

```

```

SMK03710
SMK03719
SMK03720
SMK03740
SMK03750
SMK03760
SMK03770
SMK03780
SMK03790
SMK03800
SMK03810
SMK03820
SMK03830
SMK03840
SMK03850
SMK03860
SMK03870
SMK03880
SMK03890
SMK03900
SMK03910
SMK03920
SMK03930
SMK03940
SMK03950
SMK03960
SMK03970
SMK03980
SMK03990
SMK04000
SMK04010
SMK04020
SMK04030
SMK04040
SMK04050
SMK04060
SMK04070
SMK04080
SMK04090
SMK04100
SMK04110
SMK04120
SMK04130
SMK04140
SMK04150
SMK04160
SMK04170
SMK04180
SMK04190
SMK04200
SMK04210
SMK04220
SMK04230
SMK04240
SMK04250
SMK04260
SMK04270
SMK04280
SMK04290
SMK04300
SMK04310
SMK04320
SMK04330
SMK04340
SMK04350
SMK04360
SMK04370

```

```

*20H A VALUE OF TGRAD = ,F7.2,25H C DEG/M WILL BE ASSUMED./) SMK04380
WRITE(IOOUT,170) SMK04390
170 FORMAT(5X,15HSMOKE MUNITIONS,22X,25HMETEOROLOGICAL CONDITIONS,25X, SMK04400
123HEXTINCTION COEFFICIENTS) SMK04410
WRITE(IOOUT,190) (ITTL(J,ITYPE),J=1,11),UW,EXTC(1) SMK04420
190 FORMAT(3X,11A2,11X,10HWINDSPEED,14X,F5.1,2X,3HM/S,16X, SMK04430
120H0.4-0.7 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04440
WRITE(IOOUT,200) XN,EXTC(2) SMK04450
200 FORMAT(3X,10HNO. ROUNDS,1X,F5.0,17X,22HWIND DIRECTION (USUAL ,28X, SMK04460
120H0.7-1.2 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04470
IF (ITYPE.NE.4) WRITE(IOOUT,210) FW,WDIR,EXTC(3) SMK04480
210 FORMAT(3X,11HFILL WEIGHT,F8.3,3H LB,11X,23HMET CONVENTION AZIMUTH)SMK04490
1,F6.1,2X,7HDEGREES,12X,20H1.06 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04500
IF (ITYPE.EQ.4) WRITE(IOOUT,215) FW,WDIR,EXTC(3) SMK04510
215 FORMAT(3X,9HBURN RATE,2X,F6.1,2X,6HGAL/HR, 8X,23HMET CONVENTION AZ SMK04520
11MUTH> ,F6.1,2X,7HDEGREES,12X,20H1.06 MICROMETERS ,F7.3,2X, SMK04530
27HM**2/GM) SMK04540
WRITE(IOOUT,220)TBURN,RELHUM,EXTC(4) SMK04550
220 FORMAT(3X,9HBURN TIME,F8.1,2X,3HSEC,11X,17HRELATIVE HUMIDITY,7X, SMK04560
1F5.1,2X,7HPERCENT,12X,20H3.0-5.0 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04570
LCAT=JPASCT(ICAT) SMK04580
WRITE(IOOUT,230) EFF,LCAT,EXTC(5) SMK04590
230 FORMAT(3X,10HEFFICIENCY,1X,F6.1,2X,7HPERCENT,7X,17HPASQUILL CATEGO SMK04600
1RY,9X,A1,23X,20H8.0-12. MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04610
WRITE (IOOUT,235) YF,AIR,EXTC(6) SMK04620
235 FORMAT(3X,12HYIELD FACTOR,F6.2,15X,15HAIR TEMPERATURE,8X,F6.1,2X, SMK04630
18HDEGREE C,11X,20H10.6 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04640
WRITE(IOOUT,240)TGRAD,EXTC(7) SMK04650
240 FORMAT(36X,14HTEMP. GRADIENT,10X,F6.2,9H C DEG./M,11X, SMK04660
111H94.0 GHZ,9X,F7.3,2X,7HM**2/GM) SMK04670
IF (ICAT.GE.5.AND.TGR.LT.0.) WRITE(IOOUT,250) SMK04680
250 FORMAT(36X,24H(ASSUMED POSITIVE INPUT)) SMK04690
255 NRAT=0 SMK04700
IF (BRAT1.NE.0.) NRAT=1 SMK04710
IF (BRAT2.NE.0.) NRAT=2 SMK04720
IF (BRAT3.NE.0.) NRAT=3 SMK04730
IF (BRAT4.NE.0.) NRAT=4 SMK04740
IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 258 SMK04750
IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 258
IF (NRAT.EQ.1) WRITE (IOOUT,175) BRAT1 SMK04760
IF (NRAT.EQ.2) WRITE (IOOUT,176) BRAT1,BRAT2 SMK04770
IF (NRAT.EQ.3) WRITE (IOOUT,177) BRAT1,BRAT2,BRAT3 SMK04780
IF (NRAT.EQ.4) WRITE (IOOUT,178) BRAT1,BRAT2,BRAT3,BRAT4 SMK04790
IF (NRAT.EQ.0) WRITE (IOOUT,181) SMK04800
181 FORMAT(40X,19HBURN RATE PROFILE =) SMK04810
IF (BRAT5.NE.0.) NRAT=5 SMK04820
NPWP=0 SMK04830
IF ((ITYPE.EQ.2.OR.ITYPE.EQ.5).AND.NRAT.GT.1) NPWP=1 SMK04840
IF (NRAT.EQ.5) WRITE (IOOUT,179) BRAT5 SMK04850
179 FORMAT(40X,2H+ ,F8.4,41H*0.0781829*(358800./((1.+358800.*T/TBURN))) SMK04860
175 FORMAT(1H0,40X,19HBURN RATE PROFILE =,F8.4) SMK04870
176 FORMAT(1H0,31X,19HBURN RATE PROFILE =,F8.4,2H +, SMK04880
*F8.4,10H (T/TBURN)) SMK04890
177 FORMAT(1H0,20X,19HBURN RATE PROFILE =,F8.4,2H +, SMK04900
*F8.4,12H (T/TBURN) +,F8.4,13H (T/TBURN)**2) SMK04910
178 FORMAT(1H0, 9X,19HBURN RATE PROFILE =,F8.4,2H+ , SMK04920
*F8.4,12H (T/TBURN) +,F8.4,15H (T/TBURN)**2 +,F8.4, SMK04930
*13H (T/TBURN)**3) SMK04940
C*** PROVIDE COORDINATES... SMK04950
258 ANGL=ANGLX+180.-WDIR SMK04960
IF (ANGL.GT.360.) ANGL=ANGL-360. SMK04970
IF (ANGL.LT.0.) ANGL=ANGL+360. SMK04980
IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 285 SMK04990
XP1=XQ-XM SMK05000
XP2=XT-XM SMK05010
YP1=YQ-YM SMK05020
YP2=YT-YM SMK05030
ZPP1=ZQ-ZM SMK05040
ZPP2=ZT-ZM SMK05050
CA=COS(ANGL*PIRAD) SMK05060

```

```

SA=SIN(ANGL*PIRAD)
XPP3=0.
YPP3=0.
ZPP3=0.
XPP1=XP1*CA+YP1*SA
YPP1=YP1*CA-XP1*SA
XPP2=XP2*CA+YP2*SA
YPP2=YP2*CA-XP2*SA
IF (MCUOPT.EQ.1 AND.NRUNS.GT.1) GO TO 285
WRITE (IOOUT,260)
WRITE (IOOUT,270) XM,YM,ZM,XPP3,YPP3,ZPP3,XO,YO,ZO,XPP1,YPP1,
12ZPP1,XT,YT,ZT,XPP2,YPP2,ZPP2
260 FORMAT(1H0,30X,17HFIELD COORDINATES,20X,43HROTATED COORD.(WIND X-A
1XIS,MUNITION ORIGIN)/27X,3H(X),8X,3H(Y),8X,3H(Z),20X,4H(XW),
27X,4H(YW),7X,4H(ZW))
270 FORMAT(1X,22HMUNITION COORDINATES=,3(F9.2,2X),6HMETERS,6X,
13(F9.2,2X),6HMETERS/1X,22HOBSERVER COORDINATES=,3(F9.2,2X),
26HMETERS,6X,3(F9.2,2X),6HMETERS/1X,22HTARGET COORDINATES=,
33(F9.2,2X),6HMETERS,6X,3(F9.2,2X),6HMETERS)
WRITE (IOOUT,275) ANGLX, TM
275 FORMAT(1X,40HANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = ,
*F7.2,5H DEG.,/,5X,12HEVENT TIME =,F6.1,4H SEC)
IF(MODE.GT.0) WRITE(IOOUT,1000) MODE,STIME,FRONT,DELX
1000 FORMAT(5X,5HMODE=,I2,/,5X,12HSCREEN TIME=,F6.1,/,5X,6HFRONT=,F6.1,
*/,5X,11HINCREMENT=,F6.1)
IF (NOPRNT.EQ.1 OR.MCUOPT.EQ.1) GO TO 285
IF(MODE.EQ.0) WRITE(IOOUT,280)
IF(MODE.GT.0) WRITE(IOOUT,281)
280 FORMAT(1H0,6X,4HTIME,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,
110HPATHLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,7X,5H(SEC),3(1X,8H(METERS
2ERS)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETERS
3ERS),/,60X,39H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12.,
415H 10.6 94.GHZ,/)
281 FORMAT(1H0,6X,5H*LOS*,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,10HPAT
1HLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,4X,8H(METERS),3(1X,8H(METERS
2)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETERS
3),/,60X,40H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12.,
415H 10.6 94.GHZ,/)
C*** BEGINNING OF CALCULATIONS.
285 IF (W.EQ.0.) GOTO 999
IF (ICAT.LT.1 OR.ICAT.GT.6) GOTO 999
IF(IDTO.EQ.0) IDTO = 1
IF (ISTO.LE.0) ISTO=IDTO
IF (IETO.LE.ISTO) IETO=ISTO
C*** CALCULATIONS DEPENDENT ON TIME.
C*** SET COMPUTATION TIME STEP DTIME TO 1 SEC. FOR HC/FOG OIL/PWP/
C RP AND WP WICKS/WEDGES
C*** BUT TO TABLE REPORT TIME INCREMENT FOR WP
DTIME=FLOAT(IDTO)
TRANS=1.0
TSUB(1)=0.
TSUB(2)=TBURN
IF (NPWP.EQ.0) GOTO 620
IS=IFIX(5.*TBURN+.0001)
IF (IS.LT.1) GOTO 610
DO 286 I=2,NUMDIV
TSUB(I)=1./5.
286 CONTINUE
DO 600 I=1,IS
FI=FLOAT(I)/5.
T=FI/TBURN
TMS=BFUN(T)
DO 601 JDIV=2,NUMDIV
IF (TMS.LE.QDIV(JDIV)) TSUB(JDIV)=FI
601 CONTINUE
TSUB(NUMDIV)=TBURN
600 CONTINUE
GOTO 620
610 NPWP=0
620 CONTINUE

```

```

IF (ITYPE.GE.2) DTIME=1.
NDIV=NUMDIV-1
DO 700 JPWP=1,NDIV
IF (JPWP.GT.1.AND.NPWP.EQ.0) GOTO 700
TSTAGE=TSUB(JPWP)
TBRN=TSUB(JPWP+1)-TSTAGE
PCNT=QDIV(JPWP+1)-QDIV(JPWP)
IF (NPWP.EQ.0) PCNT=1.
IF (TBRN.LE.0.) GOTO 700
CALL SCONST(ICAT,UW,ITYPE,FW,EFF,TBRN,NPWP,AIRT,TGRAD,C1,C2,C3,
*XN,XLIM,YLIM,ZLIM,HLIM,TLIM,XS,CNEUT,PCNT,W)
PCT=PCNT*100
IF (NPWP.NE.0.AND.NOPRNT.NE.1) WRITE (IOOUT,705) PCT,TSUB(JPWP),
*TSUB(JPWP+1)
IF (NPWP.NE.0.AND.MCUOPT.NE.1) WRITE (IOOUT,705) PCT,TSUB(JPWP),
*TSUB(JPWP+1)
705 FORMAT(IX,30H *** CLOUD PORTION CONTAINING ,F7.3,35H PERCENT OF SMSMK05800
*DKE DURING BURN FROM ,F7.2,3H TO,F7.2,4H SEC//) SMK05810
CL=0. SMK05820
ICODE=0 SMK05830
TO=0. SMK05840
NCY=0 SMK05850
ICALL=0 SMK05860
ISTT=ISRN-IFIX(TM+0.0001)
II=IFIX(TM)
XPP0=XPP1
I1=ISTO
I2=IETO
I3=IDTO
NNN=1
IF(MODE.EQ.0) GO TO 2000
I1=ISTT
I2=ISTT
I3=1
II=0
NNN=NPTS
TO=0.
2000 CONTINUE
DO 6 J=1,NNN
IF (ITYPE.EQ.1) TO=FLOAT(I1)-DTIME SMK05870
DO 6 I=I1,I2,I3 SMK05880
L=I-I1
IF(MODE.EQ.1) XPP1=XPP0+(J-1)*DELX
IF(MODE.EQ.1) XPP2=XPP1
X=0.0
Y=0.0
Z=0.0
PATHL=0.0
CL=0.0
IF(L.LT.I1) GO TO 2001
C*** TAB IS NEXT TABLE REPORT TIME. TO IS NEXT COMPUTATION TIME. SMK05890
TAB=FLOAT(L)-TSTAGE SMK05900
IF (TAB.GT.0.) GOTO 3 SMK05910
IF (NCY.GE.MAXS) GOTO 6 SMK05920
NCY=NCY+1 SMK05930
GOTO 6 SMK05940
3 TO=TO+DTIME SMK05950
CALL WGEOM(ICALL,CLGAUS,ITYPE,XPP1,YPP1,ZPP1,XPP2,YPP2,ZPP2,C1, SMK05960
1C2,C3,TO,UW,ICAT,HLIM,TLIM,CNEUT,XS,PATHL,X,Y,Z,XLIM,YLIM,ZLIM) SMK05970
CALL CLSMOK(ICODE,CLGAUS,ITYPE,CL,W,PATHL,TBRN,TBRN,PCNT,TSTAGE, SMK05980
*NPWP,XLIM,YLIM,ZLIM,TO,TLIM,DTIME,X,Y,Z,BRAT1,BRAT2,BRAT3,BRAT4, SMK05990
*BRAT5) SMK06000
C*** REPORT OUTPUT DATA IF TIME TO .EQ. TAB, OTHERWISE, LOOP SMK06010
C*** BACK FOR NEXT TIME STEP. SMK06020
IF (TO.LT.(TAB-.001)) GOTO 3 SMK06030
2001 CALL STRANS(CL,SMTRAN,ITYPE,EXTC,1) SMK06040
YFULL=2.*Y SMK06050
TV=FLOAT(I)-TM SMK06060
XWRIT=FLOAT(I)
IF(MODE.GT.0) XWRIT=XPP2+XM

```

```

PRNTOP=NOPRNT
IF (MODE.GT.0) PRNTOP=MCUOPT
IF (PRNTOP.NE.1) WRITE(IOOUT,290) XWRIT,X,YFULL,Z,PATHL,CL,
290 *(SMTRAN<K>,K=1,7) SMK06070
FORMAT(6X,F5.0,2X,F6.0,3X,F6.0,3X,F6.0,4X,F6.2,4X,F7.2,7<3X,F5.3>) SMK06080
IF (ICRTR.EQ.2.AND.TV.GT.(TIMTRN+.5)) GOTO 7 SMK06090
IF (ICRTR.EQ.1.AND.SMTRAN<KWAVE>.GE.TRANS) GOTO 7 SMK06100
TRTM=TV SMK06110
TRANS=SMTRAN<KWAVE> SMK06120
7 CONTINUE SMK06130
IF (NCY.GE.MAXS) GOTO 6 SMK06140
NCY=NCY+1 SMK06150
CLTOT<NCY>=CLTOT<NCY>+CL SMK06160
6 CONTINUE SMK06170
700 CONTINUE SMK06180
C*** FINAL OUTPUT. SMK06190
IF (NOMORE.EQ.0) GOTO 5 SMK06200
IF (NCY.LE.0) GOTO 999 SMK06210
IF (NRUNS.LE.1.AND.NPWP.EQ.0) GOTO 997 SMK06220
IF (MODE.EQ.0) WRITE(IOOUT,982) NRUNS SMK06230
IF (MODE.GT.0) WRITE(IOOUT,984) NRUNS SMK06240
982 FORMAT<1H1,40X,19HCOMBINED EFFECT OF ,13,21H EXECUTIONS IN SMOKE:/ SMK06250
*10X,10HTIME (SEC),5X,11HCL (G/M**2),21X,12HTRANSMISSION/ SMK06260
*10X,10<1H->,5X,11<1H->,4X, SMK06270
*54H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12. 10.6 94.GHZ// SMK06280
984 FORMAT<1H1,40X,19HCOMBINED EFFECT OF ,13,21H EXECUTIONS IN SMOKE:/
*10X,11HLOS(METERS),4X,11HCL (G/M**2),21X,12HTRANSMISSION/
*10X,10<1H->,5X,11<1H->,4X,
*54H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12. 10.6 94.GHZ//
TRANS=1.0 SMK06290
DO 985 I=1,NCY SMK06300
CL=CLTOT<I> SMK06310
TU=FLOAT<ISTO+(I-1)*IDTU> SMK06320
CALL STRANS<CL,SMTRAN,ITYPE,EXTC,1> SMK06330
YWRIT=TO
IF (MODE.GT.0) YWRIT=XPP0+XM+(I-1)*DELX
WRITE (IOOUT,983) YWRIT,CL,<SMTRAN<J>,J=1,7> SMK06340
IF (IPLT.EQ.1) WRITE(NPLOTU,883) YWRIT,CL
883 FORMAT<1X,F6.1,2X,F8.3>
983 FORMAT<12X,F6.0,8X,F8.2,6X,7<1X,F5.3,2X>> SMK06350
IF (ICRTR.EQ.2.AND.TO.GT.(TIMTRN+.5)) GOTO 985 SMK06360
IF (ICRTR.EQ.1.AND.SMTRAN<KWAVE>.GE.TRANS) GOTO 985 SMK06370
TRTM=TO SMK06380
TRANS=SMTRAN<KWAVE> SMK06390
985 CONTINUE SMK06400
997 CONTINUE SMK06410
WRITE (IOOUT,3100) WAVE1,TRANS,TRTM SMK06420
TRANS=SMTRAN<KWAVE> SMK06430
RETURN SMK06440
998 WRITE (IOOUT,3200) SMK06450
GOTO 999 SMK06460
3100 FORMAT<1H0,5X,37H***TRANSMISSION RETURNED TO MAIN FOR SMK06470
1 14HWAVELENGTH OF ,F8.3,16H MICROMETERS IS ,F5.3, SMK06480
2 8H AT TIME ,F7.0> SMK06490
3200 FORMAT<1H0,10X,35HINVALID WAVELENGTH PASSED FROM MAIN SMK06500
1 //,10X,27H TRANS=1.0 RETURNED TO MAIN,//> SMK06510
310 CONTINUE SMK06520
C*** ERROR CONDITION SMK06530
WRITE<IOOUT,320> SMK06540
320 FORMAT<1H ,105HMORE THAN 13 DATA CARDS HAVE BEEN INPUT , PLEASE CHSMK06550
*ECK THAT THERE ARE NO MORE THAN 13 DATA CARDS PER RUN.) SMK06560
999 CONTINUE SMK06570
IERR=1 SMK06580
TRANS=1.0 SMK06590
RETURN SMK06600
END SMK06610

```

```

SUBROUTINE CLSMOK(ICODE,CLGAUS,ITYPE,CL,W,PATHL,TBURN,TBRN,PCNT, CLS00010
*STAGE,NPWP,XLIM,YLIM,ZLIM,TO,TLIM,DTIME,X,Y,Z,BRAT1,BRAT2, CLS00020
*BRAT3,BRAT4,BRAT5) CLS00030
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK CLS00040
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU CLS00050
NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION CLS00060
AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC) CLS00070
COMMON /MOS/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9), CLS00080
*EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2) CLS00090
NOTE: TO CHANGE MAXIMUM BURN OR OBSCURATION DURATION, CHANGE CLS00100
PARAMETER MAXS AND DIMENSIONS OF SMAS,PVOL,CLTOT TO MAXS CLS00110
THIS SUBROUTINE CALCULATES THE VOLUME OF THE SMOKE CLOUD AND CLS00120
THE CONCENTRATION * LENGTH(CL) AT THE INTERSECTION OF THE OBSERVER CLS00130
TARGET LOS OF THE CLOUD FORMED BY THE TOTAL NUMBER OF SMOKE CLS00140
MUNITIONS WHICH WERE DETONATED AT THE SAME TIME AND FROM THE SAME CLS00150
POINT. FOR CONTINUOUS TYPE BURNS (HC, FOG OIL, PWP, RP, AND CLS00160
WP WICKS OR WEDGES), THE BURN MASS INCREMENT AND PATH LENGTH CLS00170
TO VOLUME RATIO ARE STORED FOR EACH PUFF. THESE PUFFS ARE CLS00180
SUBSEQUENTLY ADDED TO FIND THE TOTAL EFFECT OF OBSCURANT. CLS00190
INPUTS CLS00200
ICODE A FLAG TO BE SET BY USER TO 0 ON FIRST CALL CLS00210
TO PROGRAM, WHICH WILL THEN RESET IT TO 1. CLS00220
ITYPE SMOKE TYPE 1=WP, 2=PWP OR WP WICK/WEDGE CLS00230
3=HC, 4=FOG OIL, 5=RP CLS00240
W SMOKE MASS PRODUCED BY XN MUNITIONS (G FOR CLS00250
TYPES 1-3 AND 5, G/S FOR TYPE 4) CLS00260
PATHL PATHLENGTH OF SMOKE CLOUD AS IT INTERSECTS THE CLS00270
OBSERVER-TARGET LINE OF SIGHT (M) CLS00280
TBURN TOTAL LENGTH OF TIME OF BURN (S) CLS00290
NPWP FLAG FOR PWP/RP/WP WICKS OR WEDGES. IF NON-ZERO, CLS00300
THEN BURN IS IN 25 PERCENT (OF W) STAGES. CLS00310
TBRN PARTIAL BURN DURATION THIS STAGE (SEC) CLS00320
TSTAGE START OF PARTIAL BURN (SEC) CLS00330
XLIM,YLIM,ZLIM DIMENSIONS (LENGTH, HALF-WIDTH, HEIGHT) CLS00340
OF CLOUD AT END OF EXOTHERMIC RISE TIME (M) CLS00350
TO TIME AFTER IGNITION (S) CLS00360
TLIM TIME OF TERMINATION OF HEAT RISE (S) CLS00370
DTIME TIME INCREMENT OF COMPUTATION (S) CLS00380
ENDTIM ENDING TIME OF COMPUTATION (S) CLS00390
X,Y,Z CURRENT LEADING EDGE CO-ORDINATES (M) CLS00400
BRAT1,2,3,4 POLYNOMIAL BURN RATE COEFF. CLS00410
CLGAUS UNIT CONTRIBUTION FROM GAUSSIAN PUFF. CLS00420
OUTPUTS CLS00430
CL COMPUTED CL IN (G/M**2) FOR THIS MUNITION SET. CLS00440
C*** SIMPLE FOR INSTANTANEOUS BURN OF WP CLS00450
BFUN(T)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.)))+BRAT5* CLS00460
*0.07818288*ALOG(1.+358800.*T) CLS00470
MAXS=960 CLS00480
IF (ITYPE.GT.1) GOTO 100 CLS00490
VOL=0.25*(4.*PI/3.) *X*Y*Z CLS00500
CL=W*(.25*PATHL/VOL +.75 *CLGAUS) CLS00510
RETURN CLS00520
C*** INITIALIZE CLS00530
100 CL=0. CLS00540
IF (ICODE.NE.0) GOTO 110 CLS00550
ICODE=1 CLS00560
IB=0 CLS00570
IPL=0 CLS00580
START=0. CLS00590
BRNOUT=FLOAT(IFIX(.9999+TBRN)) CLS00600
NRAT=2 CLS00610
IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0.) NRAT=1 CLS00620
IF (BRAT5.NE.0.) NRAT=5 CLS00630
CLS00640
CLS00650
CLS00660
CLS00670
CLS00680
CLS00690
CLS00700

```

```

KSUB=0
TOTPV=0.
BR=W
IF (ITYPE.NE.4) BR=W/BRNOUT
IF (NPWP.NE.0) BR=BR*PCNT
IF (BRNOUT.LE.1.) NRAT=1
IF (NRAT.GT.1) BTHEN=BFUN(TSTAGE/TBURN)
C*** STORE PUFF MASS EMITTED AT TIME TO
110 IF (TO.GT.BRNOUT) GOTO 120
IF (IB.GE.MAXS) GOTO 120
IB=IB+1
IF (NRAT.EQ.1) GOTO 120
IF (IB.EQ.MAXS) WRITE (IOOUT,900) IB
900 FORMAT(1X,61H*** WARNING - IN SMOKE, MAXIMUM STORAGE FOR BURN DURACLS00840
*TION OF ,15,55H SEC IS FULL. ACCURACY BEYOND THIS POINT DECREASES.CLS00850
* *** )
TM=TO
IF (TO.GT.TBRN) TM=TBRN
T=(TM+TSTAGE)/TBURN
BNOW=BFUN(T)
SMAS(IB)=W*(BNOW-BTHEN)
BTHEN=BNOW
IF (ITYPE.EQ.4) SMAS(IB)=SMAS(IB)*TBURN
C*** COMPUTE VOLUME AT TIME TO OF FIRST PUFF
120 IF (ITYPE.EQ.3) GOTO 130
IF (ITYPE.EQ.4) GOTO 140
C*** PWP, RP OR WP WICKS/WEDGES CLOUD
VOL=0.25*(4.*PI/3.)*X*Y*Z
GOTO 200
130 IF (TO.LE.TLIM) GOTO 140
C*** POST RISE REGION HC CONE.
VOL=0.5*(PI/3.)*XLIM*YLIM*ZLIM
C*** POST-RISE FRUSTRUM OF APPROXIMATED ELLIPTIC CONE.
XZPROJ=Z*(X-XLIM)/(Z-ZLIM)
VFRUST=0.5*(PI/3.)*(XZPROJ*(Y*Z-YLIM*ZLIM)+(X-XLIM)*YLIM*ZLIM)
VOL=VOL+VFRUST
GOTO 200
C*** HC BEFORE THE END OF EXOTHERMIC RISE AND FOG OIL CASE
140 VOL=0.5*(PI/3.)*X*Y*Z
C*** STORE PATH LENGTH/VOLUME RATIO AT TIME TO FOR FIRST EXPANDING
C UNIFORM AND GAUSSIAN PUFF CONTRIBUTION.
200 IF (ITYPE.NE.4) PV=.25*(PATHL/VOL)+.75*CLGAUS
IF (ITYPE.EQ.4) PV=PATHL/VOL
IF (PV.LE.0.)AND.(START.EQ.0.) RETURN
IF (PV.GT.0.)AND.(START.EQ.0.) START=TO
IF (IPL.GE.MAXS) GOTO 300
IPL=IPL+1
IF (IPL.EQ.MAXS) WRITE (IOOUT,910) TO,IPL
910 FORMAT(1X,58H*** WARNING - IN SMOKE MAX. STORAGE FOR CLOUD VOLUME
*S OF ,15,34H SEC. IS FULL. COMPUTATION TIME = ,F6.0,5H SEC./
*15X,40HACCURACY BEYOND THIS POINT DECREASES ***)
PVOL(IPL)=PV
C*** SUM CL FOR PUFFS
300 IF (NRAT.EQ.1) GOTO 400
LMIN=MAX(1,(IPL-IB+1))
IF (IPL.LT.LMIN) RETURN
N=0
DO 320 J=LMIN,IPL
K=IPL-J+1
CDEL=SMAS(K)*PVOL(J)
CL=CL+CDEL
N=N+1
IF (N.LT.120) GOTO 320
IF (ABS(CDEL).LT.1.E-5) RETURN
320 CONTINUE
RETURN
C*** FAST COMPUTATION FOR CONSTANT BURN RATES
400 TOTPV=TOTPV+PVOL(IPL)
C INDEX OF LAST PUFF EMITTED
IF (IPL.LT.MAXS) KSUB=IPL-IB

```

```
C KEEP REMOVING LAST PUFF IF POSSIBLE          CLS01410
  IF (IPL.EQ.MAXS) KSUB=KSUB+1                 CLS01420
  IF (KSUB.GT.IPL) KSUB=IPL                   CLS01430
C ONLY REMOVE PUFFS AFTER BURN HAS STOPPED (IE LAST PUFF OUT)
  IF (KSUB.GT.0) TOTPV=TOTPV-PVOL(KSUB)        CLS01440
  CL=TOTPV*BR                                  CLS01450
  RETURN                                       CLS01460
  END                                         CLS01470
                                           CLS01480
```



```

SUBROUTINE SCONST(ICAT,UW,ITYPE,FW,EFF,TBRN, NPWP,AIRT,TGRAD,C1,C2,SC000010
* C3, XN,XLIM,YLIM,ZLIM,HLIM,TLIM,XS,CNEUT,PCNT,W) SC000020
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK SC000030
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUS SC000040
SMOKE MODEL PARAMETERS FOR CLOUD DIMENSIONS SC000050
THIS SUBROUTINE CALCULATES THE PARAMETERS FOR THE CLOUD WIDTH AND SC000060
CLOUD HEIGHT(MOMENTUM) AS A FUNCTION OF PASQUILL CATEGORY AND THE SC000070
HEAT RISE PARAMETER FOR THE HEIGHT(EXOTHERMIC). HEAT RISE IS SC000080
TERMINATED AS A FUNCTION OF ATMOSHERIC STABILITY SC000090
INPUTS ICAT PASQUILL CATEGORY SC000100
1-A, 2-B, 3-C, 4-D, 5-E, 6-F SC000110
UW WIND VELOCITY (M/S) SC000120
ITYPE TYPE OF SMOKE. 1=WP, 2=PWP OR WP WICK/WEDGE, SC000130
3=HC, 4=FOG OIL, 5=RP SC000140
FW FILL WEIGHT (LBS) SC000150
EFF EFFICIENCY (PERCENT) SC000160
XN NUMBER OF MUNITIONS SC000170
TBRN BURN TIME FOR THIS STAGE (OR ENTIRE MUNITION)SEC. SC000180
NPWP FLAG FOR LONG-BURN PHOSPHORUS. IF NON-ZERO, THEN SC000190
BURNS ARE IN 25 PERCENT STAGES TO ALLOW SEPARATE SC000200
BUOYANCIES FOR EACH. SC000210
AIRT SURFACE AIR TEMPERATURE (DEG C) SC000220
TGRAD VERT TEMP GRADIENT (C DEG./M) USED ONLY FOR SC000230
CATEGORIES E,F (IE. 5,6) IN WHICH IT MUST SC000240
BE POSITIVE. SC000250
OUTPUTS C1 EQUATION PARAMETER FOR CLOUD WIDTH Y=9.1+C1*X SC000260
C2 PARAMETER FOR HEIGHT (MOM) Z=2.73+C2*X SC000270
C3 PARAMETER FOR HEIGHT (EXO) Z=2.73+C2*X+C3*X**2/3 SC000280
XLIM CLOUD LENGTH ALONG WIND DIRECTION (M) SC000290
AT RISE TERMINATION. SC000300
YLIM CLOUD BASE HALF-WIDTH PERP. WIND DIRECTION (M) SC000310
AT RISE TERMINATION. SC000320
ZLIM TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE. SC000330
HLIM TOTAL ADDED RISE AT TERMINATION OF HEAT RISE. SC000340
TLIM TIME OF TERMINATION OF HEAT RISE SC000350
XS TERM FOR CALCULATION OF HEIGHT FOR NEUTRAL CONDITIS SC000360
CNEUT TERM FOR CALCULATION OF HEIGHT FOR NEUTRAL CONDITIS SC000370
DIMENSION C1(6),C2(6) SC000380
DATA C1/.419,.328,.238,.2,.18,.146/ SC000390
DATA C2/.137,.11,.073,.066,.055,.046/ SC000400
TLIM=600. SC000410
XS=0.0 SC000420
CNEUT=0.0 SC000430
XLIM=0. SC000440
YLIM=0. SC000450
ZLIM=0. SC000460
HLIM=1.0 SC000470
C*** ERROR CONDITION SC000480
IF(ICAT.LT.1.OR.ICAT.GT.6) WRITE(IOOUT,50) SC000490
50 FORMAT('H,62HERROR IN SUBROUTINE SCONST. PASQUILL CATEGORY IS NOT SC000500
ACCEPTABLE) SC000510
IF (ICAT.LT.1.OR.ICAT.GT.6) RETURN SC000520
C*** SELECTION OF CLOUD WIDTH COEFFICIENT AS A FUNCTION OF PASQUILL SC000530
CATEGORY. SC000540
C1 = C11(ICAT) SC000550
C*** SELECTION OF CLOUD HEIGHT(MOM) COEFFICIENT AS A FUNCTION OF SC000560
PASQUILL CATEGORY. SC000570
C2 = C22(ICAT) SC000580
IF(ICAT.GT.1)GO TO 1 SC000590
IF(UW.GT.2.0)GO TO 1 SC000600
C2=0.15 SC000610
IF(UW.GT.1.5)GO TO 1 SC000620
C2=0.18 SC000630
IF(UW.GT.1.0)GO TO 1 SC000640
C2=0.25 SC000650
IF(UW.GT.0.5)GO TO 1 SC000660
C2=0.39 SC000670
1 CONTINUE SC000680
SC000690
SC000700

```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G H/1-
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES ABAUS AND --ETC(11)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

2-6

21

21

21

The table is a grid with approximately 10 columns and 15 rows. The first row has a small black square in the first column. The rest of the grid is filled with black, indicating redacted or illegible data.

```

120 TB=TBRN
IF (ITYPE.EQ.1.AND.TBRN.GT.1.) TB=1.
C*** IF (NPWP.NE.0) TB=TB/PCNT
IF NON-EXOTHERMIC, RETURN.
IF (ITYPE.EQ.4) RETURN
EN=800.0
IF(ITYPE.EQ.3)EM=500.0
IF(ITYPE.EQ.5)EM=660.
F=0.00001*3.59*453.59*EM*FW*XM*(EFF/100.)/TB
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT FOR UNSTABLE ATM.
C*** CONDITIONS (A,B,C)
C3=1.6/UW*(F**.3333)
IF(ITYPE.EQ.3) GO TO 200
IF(ICAT.EQ.4) GO TO 131
IF(ICAT.GT.4) GO TO 141
TLIM=0.0
C*** DETERMINE TIME AND TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE.
123 TLIM=TLIM+2.
X=UW*TLIM
Y=(9.1+C1*X)/2.
V=.75*(2.*3.14159)**1.5*(Y/2.15)**2*(2.73+C3*X)/4.3
C=W/(V*TB)
IF(C.GT.0.11) GO TO 123
HLIM=C3*X**.667
RETURN
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENTS FOR NEUTRAL ATM.
C*** CONDITIONS (D).
131 XS=10.0*F**.4
CNEUT=1.6*(F**.3333)*XS**.667/UW
RETURN
141 CONTINUE
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT AND TOTAL CLOUD
C*** HEIGHT AT TERMINATION OF HEAT RISE FOR STABLE ATM. CONDITIONS(E,F)
SBAR=9.8/(AIRT+273.0)*(TGRAD+0.0098)
HLIM=1.4*(F/(UW*SBAR))**.333
RETURN
C *** CALCULATE PARAMETERS FOR HC 10 SEC. RISE TIME.
200 TLIM=10.
XLIM=TLIM*UW
YLIM=(9.1+C1*XLIM)/2.
ZLIM=2.73+C2*XLIM
IF (ICAT.EQ.4) GOTO 231
HLIM=C3*XLIM**.667
ZLIM=ZLIM+HLIM
IF (ICAT.LT.5) RETURN
SBAR=9.8/(AIRT+273.0)*(TGRAD+0.0098)
HTEST=1.4*(F/(UW*SBAR))**.333
IF (HLIM.LE.HTEST) RETURN
C*** IF UNSTABLE ATMOSPHERE REACHES MAX. BEFORE 10 SEC. COMPUTE
C*** TLIM.
XLIM=(HTEST/C3)**1.5
TLIM=XLIM/UW
YLIM=(9.1+C1*XLIM)/2.
HLIM=HTEST
ZLIM=2.73+C2*XLIM+HLIM
RETURN
231 XS=10.0*F**.4
CNEUT=C3*(XS**.667)
HLIM=CNEUT*(0.4+0.64*(XLIM/XS)+2.2*(XLIM/XS)**2)/
*(1.+0.8*XLIM/XS)**2
ZLIM=ZLIM+HLIM
RETURN
END

```

```

SC000710
SC000720
SC000730
SC000740
SC000750
SC000760
SC000770
SC000780
SC000790
SC000800
SC000810
SC000820
SC000830
SC000840
SC000850
SC000860
SC000870
SC000880
SC000890
SC000900
SC000910
SC000920
SC000930
SC000940
SC000950
SC000960
SC000970
SC000980
SC000990
SC001000
SC001010
SC001020
SC001030
SC001040
SC001050
SC001060
SC001070
SC001080
SC001090
SC001100
SC001110
SC001120
SC001130
SC001140
SC001150
SC001160
SC001170
SC001180
SC001190
SC001200
SC001210
SC001220
SC001230
SC001240
SC001250
SC001260
SC001270
SC001280
SC001290
SC001300
SC001310
SC001320
SC001330

```

CCCCCCCCCCCCCCCC

```

SUBROUTINE SMASSP(XN, EFF, FW, RELHUM, W, ITYPE, YF) SMA00010
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUSMA00020
SMOKE MODEL SMA00030
THIS SUBROUTINE CALCULATES THE SMOKE MASS PRODUCED BY XN SMA00040
MUNITIONS AT THE SAME POINT AND SAME TIME. SMA00050
INPUTS ITYPE SMOKE TYPE SMA00060
          1 WHITE PHOSPHOROUS SMA00070
          2 PLASTICIZED WHITE PHOSPHORUS OR WP WICK WEDGE SMA00080
          3 HC SMA00090
          4 FOG OIL SMA00100
          5 RED PHOSPHORUS SMA00110
XN NUMBER OF MUNITIONS AT THIS POINT AND TIME. SMA00120
RELHUM RELATIVE HUMIDITY (PERCENT) SMA00130
EFF MUNITION EFFICIENCY (PERCENT) SMA00140
FW FILL WEIGHT (LBS FOR TYPES 1-3, GAL/HR FOR TYPE 4) SMA00150
W SMOKE MASS PRODUCED FOR XN MUNITIONS. SMA00160
  G FOR TYPES 1-3 AND 5, G/S FOR TYPE 4 SMA00170
YF OPTIONAL USER SUPPLIED YIELD FACTOR SMA00180
SMA00190
SMA00200
W=0.0 SMA00210
IF (ITYPE.LT.1.OR.ITYPE.GT.5) GO TO 90 SMA00220
GO TO (101,101,102,103,101),ITYPE SMA00230
90 WRITE(IOOUT,50)ITYPE SMA00240
50 FORMAT(//,26H ERROR IN SMASSP:SMOKE TYPE ,I4,10H UNDEFINED) SMA00250
RETURN SMA00260
101 Y=.028*RELHUM+3.4 SMA00270
CONVER=453.592 SMA00280
GO TO 110 SMA00290
102 Y=1.17+.014*RELHUM SMA00300
CONVER=453.592 SMA00310
GO TO 110 SMA00320
C*** FOR FOG OIL, W IS A RATE, WHERE FW IS IN GAL/HR AND .93 SMA00330
C*** CONVERTS TO G/S. SMA00340
103 Y=1.0 SMA00350
CONVER=0.93 SMA00360
110 IF (YF.NE.0.) Y=YF SMA00370
YF=Y SMA00380
W=XN*Y*FW*CONVER*(EFF/100.0) SMA00390
RETURN SMA00400
END SMA00410

```

CC

```

SUBROUTINE STRANS(CL, SMTRAN, ITYPE, EXTC, ICALL)                                STR00010
                                                                                                                                 STR00020
                                                                                                                                 STR00030
                                                                                                                                 STR00040
                                                                                                                                 STR00050
                                                                                                                                 STR00060
                                                                                                                                 STR00070
                                                                                                                                 STR00080
                                                                                                                                 STR00090
                                                                                                                                 STR00100
                                                                                                                                 STR00110
                                                                                                                                 STR00120
                                                                                                                                 STR00130
                                                                                                                                 STR00140
                                                                                                                                 STR00150
                                                                                                                                 STR00160
                                                                                                                                 STR00170
                                                                                                                                 STR00180
                                                                                                                                 STR00190
                                                                                                                                 STR00200
                                                                                                                                 STR00210
                                                                                                                                 STR00220
                                                                                                                                 STR00230
                                                                                                                                 STR00240
                                                                                                                                 STR00250
                                                                                                                                 STR00260
                                                                                                                                 STR00270
                                                                                                                                 STR00280
                                                                                                                                 STR00290
                                                                                                                                 STR00300
                                                                                                                                 STR00310
                                                                                                                                 STR00320
                                                                                                                                 STR00330
                                                                                                                                 STR00340
                                                                                                                                 STR00350
                                                                                                                                 STR00360
                                                                                                                                 STR00370
                                                                                                                                 STR00380
                                                                                                                                 STR00390
                                                                                                                                 STR00400
                                                                                                                                 STR00410
                                                                                                                                 STR00420
                                                                                                                                 STR00430
                                                                                                                                 STR00440
                                                                                                                                 STR00450
                                                                                                                                 STR00460
                                                                                                                                 STR00470
                                                                                                                                 STR00480
                                                                                                                                 STR00490
                                                                                                                                 STR00500
                                                                                                                                 STR00510
                                                                                                                                 STR00520
                                                                                                                                 STR00530
                                                                                                                                 STR00540
                                                                                                                                 STR00550
                                                                                                                                 STR00560

SMOKE MODEL           CLOUD TRANSMISSION
THIS SUBROUTINE CALCULATES TRANSMISSION IN 7 SPECTRAL REGIONS.
< 0.4-0.7, 0.7-1.2, 1.06, 3.0-5.0, 8.0-12.0, 10.6 MICROMETERS AND 94.GHZ >
FOR A GIVEN SMOKE TYPE AND CONCENTRATION LENGTH
INPUTS  ITYPE      SMOKE TYPE
                1  WHITE PHOSPHOROUS
                2  PLASTICIZED WHITE PHOSPHORUS, WP WICK WEDGE
                3  HC
                4  FOG OIL
                5  RED PHOSPHORUS
CL      COMPUTED CL IN (G/M**2)
ICALL   = 0 SETS UP EXTC ARRAY USED FOR COMPUTATIONS AND
        = 1 EXECUTES TRANSMISSION CALCULATION.

OUTPUTS SMTRAN TRANSMISSION THROUGH SMOKE (DECIMAL)
        EXTC  ARRAY OF EXTINCTION COEFF. ACTUALLY USED
        IN TRANSMISSION CALCULATION. EXTC(8) IS
        USED AS A FLAG FOR ICALL=0 REPLACEMENT.

        IF EXTC(8)=ITYPE, NO CHANGES ARE MADE IN EXTC ARRAY IE.,
        ALPHA VALUES DO NOT REPLACE EXTC VALUES.
        IF EXTC(8)=0, ONLY THOSE VALUES IN EXTC WHICH ARE
        ZERO ARE REPLACED BY THE STORED VALUES IN
        ALPHA (ITYPE COLUMN).
        IF EXTC(8) IS NOT 0, OR ITYPE, THEN ALL EXTC VALUES
        ARE REPLACED BY CORRESPONDING ALPHA VALUES AND
        EXTC(8) IS SET TO ITYPE.

DIMENSION ALPHA(7,5), SMTRAN(7), EXTC(8)
DATA ALPHA              /4.304,2.166,1.541,0.350,0.338,0.364,0.001,
*                      4.304,2.166,1.541,0.350,0.338,0.364,0.001,
*                      4.579,2.186,2.040,0.190,0.052,0.051,0.001,
*                      6.851,4.592,3.497,0.245,0.020,0.018,0.001,
*                      4.304,2.166,1.541,0.350,0.338,0.364,0.001/
C*** TRANSMISSION CALCULATED BY BEER'S LAW APPROXIMATION
C*** IF ICALL=0, EXTC ARRAY IS FORMED OR MODIFIED...
    IF (ICALL.NE.0) GOTO 20
    IF (EXTC(8).EQ.FLOAT(ITYPE)) GOTO 18
    IF (EXTC(8).EQ.0.) GOTO 15
    DO 13 J=1,7
13  EXTC(J)=0.
15  EXTC(8)=FLOAT(ITYPE)
    DO 17 J=1,7
    IF (EXTC(J).EQ.0.) EXTC(J)=ALPHA(J, ITYPE)
17  CONTINUE
18  RETURN
C*** FOR ICALL NON-ZERO, COMPUTE TRANSMISSION USING EXTC VALUES.
20  DO 30 I=1,7
30  SMTRAN(I)=EXP(-EXTC(I)*CL)
    RETURN
END

```

```

SUBROUTINE WGGEOM(ICALL,CLGAUS,ITYPE,XPP1,YPP1,ZPP1,XPP2,YPP2,
1ZPP2,C1,C2,C3,TO,UW,ICAT,HLIM,TLIM,CNEUT,XS,PATHL,X,Y,Z,XLIM,
2YLIM,ZLIM)
NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION
AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)
COMMON /M05/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),
*EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DATA MAXS /960/
NOTE: TO CHANGE LENGTH OF BURN OR OBSCURATION, RESET
MAXS AND REDIMENSION SMAS, PVOL AND CLTOT ARRAYS
SMOKE MODEL GEOMETRY
INPUT:
ITYPE = SMOKE TYPE CODE
XPP1,YPP1,ZPP1 = OBSERVER IN MUNITION CENTERED COORDINATES
YPP1,YPP2,ZPP2 = TARGET IN MUNITION CENTERED COORD. (X AXIS ALONG
WIND VECTOR.)
C1,C2,C3 = CLOUD GROWTH PARAMETERS (SEE PROGRAM SCNST)
TO = TIME SINCE DETONATION (SECONDS)
UW = WIND SPEED (METERS/SECOND)
ICAT=PASQUILL CATEGORY
HLIM,TLIM,CNEUT,XS= EXOTHERMIC RISE PARAMETERS
ICALL = FLAG SET BY USER TO 0 FOR 1ST CALL. RESET BY PROGRAM
TO 1 THEREAFTER.
OUTPUT:
X = CLOUD LENGTH (METERS)
Y = CLOUD HALF-WIDTH (METERS)
Z = CLOUD HEIGHT (METERS)
PATHL = PATHLENGTH OF LOS THROUGH SMOKE CLOUD
SUBROUTINES CALLED... DIRECTLY: XYZINT, GPUFF, INDIRECTLY: QROOT
*** TRANSLATION TO PLACE MUNITION AT (0,0,0)
IF (ICALL.NE.0) GOTO 10
ICALL=1
KCALL=0
C*** CALCULATE LEADING EDGE LOCATION AT TO.
10 A=UW*TO
B=(9.1/2.0)+A*C1/2.
IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) GOTO 33
C*** WP/PWP/RP COMPUTATION.
GO TO (11,11,11,21,31,31),ICAT
11 CONTINUE
C=2.73+C2*A+C3*A**0.667
CLIM=2.73+C2*A+HLIM
IF (TO.GT.TLIM.AND.C.GT.CLIM)C=CLIM
GO TO 41
21 CONTINUE
C=2.73+C2*A
IF (A.LE.XS) C=C+C3*A**0.667
IF (A.GT.XS) C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/
1(1.0+0.8*(A/XS)**2)
GO TO 41
31 CONTINUE
C=2.73+C2*A+C3*A**0.667
CLIM=2.73+C2*A+HLIM
IF (C.GT.CLIM)C=CLIM
GOTO 41
C*** HC, FOG OIL COMPUTATION.
33 C=2.73+C2*A
IF (ITYPE.EQ.4) GOTO 41
IF (TO.LT.TLIM) GOTO 35
C=C+HLIM
GOTO 41
35 IF (ICAT.NE.4.OR.A.LE.XS)C=C+C3*A**0.667
IF (ICAT.EQ.4 .AND. A.GT.XS)

```

WGG00010
WGG00020
WGG00030
WGG00040
WGG00050
WGG00060
WGG00070
WGG00080
WGG00090
WGG00100
WGG00110
WGG00120
WGG00130
WGG00140
WGG00150
WGG00160
WGG00170
WGG00180
WGG00190
WGG00200
WGG00210
WGG00220
WGG00230
WGG00240
WGG00250
WGG00260
WGG00270
WGG00280
WGG00290
WGG00300
WGG00310
WGG00320
WGG00330
WGG00340
WGG00350
WGG00360
WGG00370
WGG00380
WGG00390
WGG00400
WGG00410
WGG00420
WGG00430
WGG00440
WGG00450
WGG00460
WGG00470
WGG00480
WGG00490
WGG00500
WGG00510
WGG00520
WGG00530
WGG00540
WGG00550
WGG00560
WGG00570
WGG00580
WGG00590
WGG00600
WGG00610
WGG00620
WGG00630
WGG00640
WGG00650
WGG00660
WGG00670
WGG00680
WGG00690
WGG00700

```

1 C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/
1<1.0+0.8*(A/XS)**2
41 CONTINUE
C*** NOW COMPUTE PATH LENGTHS -- SELECT ELLIPSOID FOR PHOSPHORUS
C*** OR ELLIPTICAL CONES FOR HC, FOG OIL.
C*** FIRST GAUSSIAN PUFF AND IMAGE PUFF CONTRIBUTION, THEN FULLY MIXED
NCODE=2
IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) NCODE=1
IF (ITYPE.NE.4)
*CALL GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1,ZPP2)
X=A
Y=B
Z=C
100 IF (ITYPE.EQ.3) GOTO 200
X0=A
Y0=B
Z0=C
150 XL(1)=XPP1
XL(2)=XPP2
YL(1)=YPP1
YL(2)=YPP2
ZL(1)=ZPP1
ZL(2)=ZPP2
CALL XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)
PATHL=SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+
*(ZINT(1)-ZINT(2))**2)
IF (ITYPE.EQ.3) GOTO 220
RETURN
200 ICONE=1
IF (TO.LE.TLIM) GOTO 100
ICONE=0
C*** FOR HC, BREAK PATH UP INTO TWO PARTS. FIRST RISE PORTION CONE,
C THEN POST-RISE FRUSTRUM.
X0=XLIM
Y0=YLIM
Z0=ZLIM
GOTO 150
220 IF (ICONE.EQ.1) RETURN
ICONE=1
C*** NOW POST RISE PORTION,
IF (PATHL.EQ.0.) GOTO 100
IF (IFLAG(1).NE.3.AND.IFLAG(2).NE.3) RETURN
IF (IFLAG(1).EQ.3.AND.IFLAG(2).EQ.3) RETURN
K=1
IF (IFLAG(2).EQ.3) K=2
J=1
IF (XL(2).LT.XL(1)) J=2
XL(J)=XINT(K)
YL(J)=YINT(K)
ZL(J)=ZINT(K)
X0=A
Y0=B
Z0=C
CALL XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)
PATHL=PATHL+SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+
*(ZINT(1)-ZINT(2))**2)
RETURN
END
WGG00710
WGG00720
WGG00730
WGG00740
WGG00750
WGG00760
WGG00770
WGG00780
WGG00790
WGG00800
WGG00810
WGG00820
WGG00830
WGG00840
WGG00850
WGG00860
WGG00870
WGG00880
WGG00890
WGG00900
WGG00910
WGG00920
WGG00930
WGG00940
WGG00950
WGG00960
WGG00970
WGG00980
WGG00990
WGG01000
WGG01010
WGG01020
WGG01030
WGG01040
WGG01050
WGG01060
WGG01070
WGG01080
WGG01090
WGG01100
WGG01110
WGG01120
WGG01130
WGG01140
WGG01150
WGG01160
WGG01170
WGG01180
WGG01190
WGG01200
WGG01210
WGG01220
WGG01230
WGG01240
WGG01250
WGG01260
WGG01270
WGG01280

```

```

FUNCTION QROOT(ISIGN,A,B,C)
C*****PURPOSE:
C   TO FIND ROOTS OF A QUADRATIC EQUATION.
   IF (A.EQ.0.) GOTO 2
   XX=1.0*ISIGN
   TEST= B*B - 4.0*A*C
   IF (TEST.LT.0.0) GO TO 1
   QROOT=(-1.0*B + XX*SQRT(TEST))/(2.0*A)
   GO TO 100
1  QROOT=0.0
   GOTO 100
2  IF (B.EQ.0.) GOTO 1
   QROOT=-C/B
100 RETURN
END

```

```

QRO00010
QRO00020
QRO00030
QRO00040
QRO00050
QRO00060
QRO00070
QRO00080
QRO00090
QRO00100
QRO00110
QRO00120
QRO00130
QRO00140
QRO00150

```



```

SUBROUTINE XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)  XYZ00010
DIMENSION XL(2),YL(2),ZL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)  XYZ00020
DIMENSION DIST(2),TEST(2)  XYZ00030
C*****SUBROUTINE XYZINT*****  XYZ00040
C*****PURPOSE:  XYZ00050
TO FIND THE X,Y,Z INTERCEPTS OF A TARGET-OBSERVER LINE OF SIGHT  XYZ00060
WITH A SMOKE CLOUD DESCRIBED BY:  XYZ00070
NCODE = 1 A HALF ELLIPTIC CONE WITH APEX AT THE ORIGIN  XYZ00080
AND LEADING EDGE TRUNCATION BY THE X=X0 PLANE  XYZ00090
AND BOTTOM EDGE TRUNCATION BY THE Z=0 PLANE.  XYZ00100
NCODE = 2 A QUARTER ELLIPSOID WITH APEX  XYZ00110
AT THE ORIGIN AND WITH LEADING EDGE TRUNCATED BY  XYZ00120
THE X=X0 AND BOTTOM EDGE TRUNCATED BY THE Z=0 PLANE.  XYZ00130
C*****INPUT:  XYZ00140
XL(2),YL(2),ZL(2) = X,Y,Z COORDINATES OF TWO POINTS THROUGH  XYZ00150
WHICH THE LOS PASSES (IE. TARGET AND OBSERVER COORDINATES).  XYZ00160
X0,Y0,Z0 = LENGTH OF SEMI-AXES OF ELLIPSOID.  XYZ00170
C*****OUTPUT:  XYZ00180
XINT(2),YINT(2),ZINT(2) = X,Y,Z COORDINATES OF THE INTERCEPTS  XYZ00190
OF THE LOS WITH THE ELLIPSOID.  XYZ00200
IFLAG(2) - INTERCEPT TYPE FOR EACH INTERCEPT COORD:  XYZ00210
= 0 NO INTERCEPT  XYZ00220
= 1 INTERIOR TO VOLUME  XYZ00230
= 2 ON CONICAL OR ELLIPTICAL SURFACE  XYZ00240
= 3 ON LEADING EDGE OF SURFACE  XYZ00250
C*****MATHEMATICAL APPROACH:  XYZ00260
THE EQUATION OF THE ELLIPSOID CAN BE WRITTEN AS:  XYZ00270
((X-X0)/X0)**2 + (Y/Y0)**2 + (Z/Z0)**2 = 1  XYZ00280
AND THE EQUATION OF THE LOS CAN BE WRITTEN AS:  XYZ00290
(X-XL1)/(XL2-XL1) = (Y-YL1)/(YL2-YL1) = (Z-ZL1)/(ZL2-ZL1)  XYZ00300
THE TWO EQUATIONS ARE COMBINED TO FORM A QUADRATIC EQUATION  XYZ00310
WHICH IS SOLVED TO GIVE THE INTERCEPTS.  XYZ00320
SIMILARLY FOR THE LOS EQUATIONS AND ELLIPTIC CONE :  XYZ00330
(Z/Z0)**2 + (Y/Y0)**2 - (X/X0)**2 = 0  XYZ00340
C*****SPECIAL NOTES  XYZ00350
(1) WHEN TWO OR MORE COORDINATES ARE THE SAME, SPECIAL CASES ARE  XYZ00360
FORMED WHICH MUST BE DEALT WITH SEPARATELY BECAUSE OF  XYZ00370
SINGULARITIES IN THE LOS EQUATION.  XYZ00380
(2) WHEN TARGET AND/OR OBSERVER ARE INSIDE THE CLOUD INTERCEPTS  XYZ00390
ARE TAKEN AS THE TARGET AND/OR OBSERVER COORDINATES.  XYZ00400
(3) PROPER ACCOUNT IS TAKEN FOR A LOS INTERCEPTING THE CLOUD  XYZ00410
LEADING EDGE BUT...  XYZ00420
(4) ALL COORDINATES MUST BE ABOVE THE Z=0 PLANE (IE. ABOVE THE  XYZ00430
SURFACE.)  XYZ00440
SUBROUTINE CALLED... GROOT  XYZ00450
C*****INITIALIZE PROGRAM VARIABLES  XYZ00460
I1=0  XYZ00470
I2=0  XYZ00480
INT=0  XYZ00490
LEAD=0  XYZ00500
ISURF=0  XYZ00510
K1=0  XYZ00520
K2=0  XYZ00530
K3=0  XYZ00540
K4=0  XYZ00550
TEST(1)=0.0  XYZ00560
TEST(2)=0.0  XYZ00570
TEST3=0.0  XYZ00580
TEST4=0.0  XYZ00590
DIST(1)=0.0  XYZ00600
DIST(2)=0.0  XYZ00610
DELX=XL(2)-XL(1)  XYZ00620
DELY=YL(2)-YL(1)  XYZ00630
DELZ=ZL(2)-ZL(1)  XYZ00640
IFLAG(1)=0  XYZ00650
IFLAG(2)=0  XYZ00660
C*** REJECT IMMEDIATELY IF BOTH TGT/OBS BELOW Z=0.  XYZ00670
IF (ZL(1).LT.0. AND.ZL(2).LT.0.) GOTO 800  XYZ00680
C*****DETERMINE SPECIAL CASES FOR LOS  XYZ00690
XYZ00700

```

```

ICASE=1
IF (ABS<DELX>.GT.(.01*ABS<DELY>).AND.
* ABS<DELX>.GT.ABS<.01*ABS<DELZ>)) GOTO 12
ICASE=2
IF (ABS<DELY>.GT.(.01*ABS<DELZ>)) GOTO 12
ICASE=3
IF (DELZ.GT.1.E-2.OR.DELZ.LT.-1.E-2) GOTO 12
ICASE=4
12 CONTINUE
C*****DEFAULT SPECIAL CASE OF OBS-TAR COINCIDENT
IF (ICASE.NE.4)GO TO 14
GO TO 800
14 CONTINUE
C*****SET UP TEST TO DETERMINE IF TARGET AND/OR OBSERVER ARE IN THE
C INTERIOR OF THE CLOUD
DO 1 I=1,2
IF (NCOOE.EQ.2) TEST(I)= ((XL(I)-X0)/X0)**2 + (YL(I)/Y0)**2 +
*(ZL(I)/Z0)**2-1.
IF (NCOOE.EQ.1) TEST(I)=(YL(I)/Y0)**2 + (ZL(I)/Z0)**2 -
*(XL(I)/X0)**2
1 CONTINUE
IF (TEST(1).GT.0.0)GO TO 2
IF (XL(1).LT.0..OR.ZL(1).LT.0.) GOTO 2
IF (XL(1).GT.X0) GOTO 2
IF (XL(1).EQ.X0) K1=1
IF (TEST(1).EQ.0..OR.ZL(1).EQ.0.) K3=1
I1=1
2 CONTINUE
IF (TEST(2).GT.0.0)GO TO 3
IF (XL(2).LT.0..OR.ZL(2).LT.0.) GOTO 3
IF (XL(2).GT.X0) GOTO 3
IF (XL(2).EQ.X0) K2=1
IF (TEST(2).EQ.0..OR.ZL(2).EQ.0.) K4=1
I2=1
C*****IF BOTH POINTS ARE IN THE CLOUD SET INTERCEPTS EQUAL TO THE
C TARGET-OBSERVER COORDINATES AND RETURN--OTHERWISE CONTINUE
3 IF (I1.EQ.0)GO TO 4
IF (I2.EQ.0)GO TO 4
DO 5 I=1,2
IFLAG(I)=1
XINT(I)=XL(I)
YINT(I)=YL(I)
5 ZINT(I)=ZL(I)
IF (K3.EQ.1) IFLAG(1)=2
IF (K4.EQ.1) IFLAG(2)=2
IF (K1.EQ.1) IFLAG(1)=3
IF (K2.EQ.1) IFLAG(2)=3
GO TO 999
C*****IF ONLY ONE POINT IS IN CLOUD KEEP TRACK OF IT FOR LATER
4 CONTINUE
IF (I1.EQ.0)GO TO 6
INT=1
6 IF (I2.EQ.0)GO TO 7
INT=2
7 CONTINUE
IF (K1.EQ.1) LEAD=1
IF (K2.EQ.1) LEAD=2
IF (K3.EQ.1) ISURF=1
IF (K4.EQ.1) ISURF=2
C*****SET UP LOS EQUATION DEPENDING UPON CASE
GO TO (10,20,30),ICASE
C*****CASE 1
10 SX=DELX/DELX
SY=DELY/DELX
SZ=DELZ/DELX
XI=XL(1)-SX*XL(1)
YI=YL(1)-SY*XL(1)
ZI=ZL(1)-SZ*XL(1)
GO TO 101
C*****CASE 2

```

```

XYZ000710
XYZ000720
XYZ000730
XYZ000740
XYZ000750
XYZ000760
XYZ000770
XYZ000780
XYZ000790
XYZ000800
XYZ000810
XYZ000820
XYZ000830
XYZ000840
XYZ000850
XYZ000860
XYZ000870
XYZ000880
XYZ000890
XYZ000900
XYZ000910
XYZ000920
XYZ000930
XYZ000940
XYZ000950
XYZ000960
XYZ000970
XYZ000980
XYZ000990
XYZ001000
XYZ001010
XYZ001020
XYZ001030
XYZ001040
XYZ001050
XYZ001060
XYZ001070
XYZ001080
XYZ001090
XYZ001100
XYZ001110
XYZ001120
XYZ001130
XYZ001140
XYZ001150
XYZ001160
XYZ001170
XYZ001180
XYZ001190
XYZ001200
XYZ001210
XYZ001220
XYZ001230
XYZ001240
XYZ001250
XYZ001260
XYZ001270
XYZ001280
XYZ001290
XYZ001300
XYZ001310
XYZ001320
XYZ001330
XYZ001340
XYZ001350
XYZ001360
XYZ001370
XYZ001380
XYZ001390
XYZ001400

```

```

20 SX=DELX/DELY
   SY=DELY/DELY
   SZ=DELZ/DELY
   XI=XL(1)-SX*YL(1)
   YI=YL(1)-SY*YL(1)
   ZI=ZL(1)-SZ*YL(1)
   GO TO 101
C*****CASE 3
30 SX=DELX/DELZ
   SY=DELY/DELZ
   SZ=DELZ/DELZ
   XI=XL(1)-SX*ZL(1)
   YI=YL(1)-SY*ZL(1)
   ZI=ZL(1)-SZ*ZL(1)
101 CONTINUE
C*****SET UP QUADRATIC COEFFICIENTS
   IF (NCODE.EQ.1) GOTO 60
   A=(SX/X0)**2 + (SY/Y0)**2 + (SZ/Z0)**2
   B=2.0*((XI/X0)*(SX/X0) + (YI/Y0)*(SY/Y0) + (ZI/Z0)*(SZ/Z0)
   *- (SX/X0))
   C=(XI/X0)**2 + (YI/Y0)**2 + (ZI/Z0)**2 - 2.0*XI/X0
   GOTO 61
60 A=(SY/Y0)**2 + (SZ/Z0)**2 - (SX/X0)**2
   B=2.0*((SY/Y0)*(SI/Y0) + (SZ/Z0)*(ZI/Z0) + (SX/X0)*(XI/X0))
   C=(YI/Y0)**2 + (ZI/Z0)**2 - (XI/X0)**2
61 CONTINUE
C*****DEFAULT ALL INTERCEPTS IF ROOTS ARE COMPLEX
   TEST0=B*B-4.0*A*C
   IF (TEST0.GE.0.0) GO TO 888
800 DO 13 I=1,2
   IFLAG(I)=0
   XINT(I)=0.0
   YINT(I)=0.0
   ZINT(I)=0.0
13 GO TO 999
C*****SOLVE QUADRATIC FOR X,Y OR Z DEPENDING ON CASE
888 GO TO (100,200,300),ICASE
100 XINT(1)=GROOT(+1,A,B,C)
   XINT(2)=GROOT(-1,A,B,C)
   DO 11 I=1,2
   YINT(I)=YI+SY*XINT(I)
   ZINT(I)=ZI+SZ*XINT(I)
   GOTO 400
200 YINT(1)=GROOT(+1,A,B,C)
   YINT(2)=GROOT(-1,A,B,C)
   DO 21 I=1,2
   XINT(I)=XI+SX*YINT(I)
   ZINT(I)=ZI+SZ*YINT(I)
   GO TO 400
300 ZINT(1)=GROOT(+1,A,B,C)
   ZINT(2)=GROOT(-1,A,B,C)
   DO 31 I=1,2
   XINT(I)=XI+SX*ZINT(I)
   YINT(I)=YI+SY*ZINT(I)
31 YINT(I)=YI+SY*ZINT(I)
C*** TEST FOR VALID INTERCEPTS
400 I1=0
   I2=0
   IFLAG(1)=2
   IFLAG(2)=2
   IF (ZINT(1).GE.0. .AND.XINT(1).GE.0. .AND.XINT(1).LE.X0) I1=1
   IF (ZINT(2).GE.0. .AND.XINT(2).GE.0. .AND.XINT(2).LE.X0) I2=1
   IF (I1.EQ.0. OR.I2.EQ.0) GOTO 450
   IF (XINT(1).EQ.XINT(2) .AND.YINT(1).EQ.YINT(2) .AND.ZINT(1).EQ.
   * ZINT(2)) I2=0
   IF (I1.EQ.1 .AND.I2.EQ.1) GOTO 600
C*** AT LEAST ONE INTERCEPT INVALID. FIRST COMPUTE POSSIBLE Z=0. INTCP
450 GOTO (460,470,480),ICASE
460 Z=0.
   IF (SZ.EQ.0.) GOTO 500
   X=-ZI/SZ

```

```

XYZ01410
XYZ01420
XYZ01430
XYZ01440
XYZ01450
XYZ01460
XYZ01470
XYZ01480
XYZ01490
XYZ01500
XYZ01510
XYZ01520
XYZ01530
XYZ01540
XYZ01550
XYZ01560
XYZ01570
XYZ01580
XYZ01590
XYZ01600
XYZ01610
XYZ01620
XYZ01630
XYZ01640
XYZ01650
XYZ01660
XYZ01670
XYZ01680
XYZ01690
XYZ01700
XYZ01710
XYZ01720
XYZ01730
XYZ01740
XYZ01750
XYZ01760
XYZ01770
XYZ01780
XYZ01790
XYZ01800
XYZ01810
XYZ01820
XYZ01830
XYZ01840
XYZ01850
XYZ01860
XYZ01870
XYZ01880
XYZ01890
XYZ01900
XYZ01910
XYZ01920
XYZ01930
XYZ01940
XYZ01950
XYZ01960
XYZ01970
XYZ01980
XYZ01990
XYZ02000
XYZ02010
XYZ02020
XYZ02030
XYZ02040
XYZ02050
XYZ02060
XYZ02070
XYZ02080
XYZ02090
XYZ02100

```


	IFLAG(I)=1	XYZ02810
	IF (LEAD.EQ.INT) IFLAG(I)=3	XYZ02820
	IF (ISURF.EQ.INT) IFLAG(I)=2	XYZ02830
	GOTO 700	XYZ02840
610	CONTINUE	XYZ02850
C***	REPLACE CLOSEST INTERCEPT OUTSIDE LOS RANGE	XYZ02860
	K1=1	XYZ02870
	IF (DIST(1).LT.DIST(2)) K1=2	XYZ02880
	XINT(K1)=XL(INT)	XYZ02890
	YINT(K1)=YL(INT)	XYZ02900
	ZINT(K1)=ZL(INT)	XYZ02910
	IFLAG(K1)=1	XYZ02920
700	IF (XINT(1).GE.(X0-.001).AND.XINT(1).LE.(X0+.001)) IFLAG(1)=3	XYZ02930
	IF (XINT(2).GE.(X0-.001).AND.XINT(2).LE.(X0+.001)) IFLAG(2)=3	XYZ02940
999	RETURN	XYZ02950
	END	XYZ02960

SUBROUTINE BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,
 *BRAT1,BRAT2,BRAT3,BRAT4,BRAT5)

THIS ROUTINE PROVIDES DEFAULT MUNITION CHARACTERISTIC VALUES TO
 SMOKE. SMOKE MUNITION TYPES (TYPM) ARE GIVEN IN THE COMMENTS
 OF THE MAIN ROUTINE IN SMOKE

DIMENSION B1(21),B2(21),B3(21),B4(21),B5(21)
 DIMENSION F(21),T(21),E(21),IT(21)

DATA B1 / .537, .631, .2218, .537, .2218, 1., 1., 1., 1., 1., 1., 0.,
 * 0., .521, 1.631, 1.808, .1204, .653, 1.731, 0., 1. /
 * DATA B2 / .476, -.4985, 3.915, .476, 3.915, 0., 0., 0., 0., 0., 0., 0.,
 * 2.106, .678, -2.556, 3.1012, -3.136, -2.852, 3.6832, 0. /
 * DATA B3 / 4.779, 6.745, -1.7368, 4.779, -1.7368, 0., 0., 0., 0., 0.,
 * 0., 0., -1.11, -5.907, 2.883, -2.2104, 15.309, 4.341, -5.3472, 0. /
 * DATA B4 / -5.472, -6.52, -2.3995, -5.472, -2.3995, 0., 0., 0., 0., 0.,
 * 0., 0., -.748, 4.012, -2.008, .206, -12.872, -3.108, 3.8348, 0. /
 * DATA B5 / 11*0, 2*1, 8*0 /
 DATA IT / 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 5, 5, 5, 4 /
 DATA T / 100, 70, 120, 100, 120, 900, 1, 1, 1, 1, 1, 600, /
 * 600, 240, 470, 390, 721, 260, 380, 750, 900. /
 DATA F / 5.46, 2.69, 1.65, 17.19, 7.50, 30., 0.76, 1.75, 8.14, 15.6,
 * 3.83, 8.14, 13.52, .463, .139, .234, 19.98, .128, .0243,
 * 19.4, 40.0 /
 * DATA E / 40., 40., 40., 40., 40., 24., 100., 100., 100., 100., 100.,
 * 60., 60., 66., 71., 67., 77., 53., 55., 51., 100. /

MAXS=21
 ITP=IFIX(TYPM+.0001)
 IF (ITP.EQ.0) RETURN
 IF (ITP.LE.MAXS) GOTO 10

IERR=1
 RETURN
 10 BRAT1=B1(ITP)
 BRAT2=B2(ITP)
 BRAT3=B3(ITP)
 BRAT4=B4(ITP)
 BRAT5=B5(ITP)
 ITYPE=IT(ITP)
 IF (MUNRD.NE.0) GOTO 20

XN=1.
 YF=0.
 TBURN=T(ITP)
 EFF=E(ITP)
 FW=F(ITP)
 RETURN
 20 IF (XN.EQ.0.) XN=1.
 IF (TBURN.EQ.0.) TBURN=T(ITP)
 IF (EFF.EQ.0.) EFF=E(ITP)
 IF (FW.EQ.0.) FW=F(ITP)
 RETURN
 END

BRA00010
 BRA00020
 BRA00030
 BRA00040
 BRA00050
 BRA00060
 BRA00070
 BRA00080
 BRA00090
 BRA00100
 BRA00110
 BRA00120
 BRA00130
 BRA00140
 BRA00150
 BRA00160
 BRA00170
 BRA00180
 BRA00190
 BRA00200
 BRA00210
 BRA00220
 BRA00230
 BRA00240
 BRA00250
 BRA00260
 BRA00270
 BRA00280
 BRA00290
 BRA00300
 BRA00310
 BRA00320
 BRA00330
 BRA00340
 BRA00350
 BRA00360
 BRA00370
 BRA00380
 BRA00390
 BRA00400
 BRA00410
 BRA00420
 BRA00430
 BRA00440
 BRA00450
 BRA00460
 BRA00470
 BRA00480
 BRA00490
 BRA00500
 BRA00510
 BRA00520

```

SUBROUTINE GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1, GPU00010
*ZPP2) GPU00020
C THIS ROUTINE COMPUTES THE CL CONTRIBUTION FROM AN EXOTHERMIC, GPU00030
C BUOYANTLY RISING SMOKE CLOUD OF GAUSSIAN DISTRIBUTION AND GPU00040
C UNIT CONCENTRATION. IT IS CENTERED ON THE LEADING EDGE AT GPU00050
C COORDINATES (A,B,C-2*SIGZ) WHERE GPU00060
C INPUTS A = CLOUD DOWNWIND DISTANCE GPU00070
C B = CLOUD BASE HALF-WIDTH (AT LEADING EDGE) GPU00080
C C = CLOUD HEIGHT (AT LEADING EDGE) GPU00090
C C2 = MOMENTUM RISE COEFFICIENT. (USUALLY BRIGGS OR LIMITED GPU00100
C RISE BRIGGS) GPU00110
C XPP1,...ZPP2 COORDINATES OF TARGET AND OBSERVER IN MUNITION GPU00120
C CENTERED COORDINATE SYSTEM WITH WIND VECTOR X-AXIS GPU00130
C OUTPUT CLGAUS = CL VALUE FOR UNIT MASS (METER**-2) GPU00140
C KCALL = SET TO 1 AFTER EVERY CHANGE IN TGT/OBS COORD. GPU00150
C DIMENSION AV(2),AP(2),BP(2),CP(2) GPU00160
C CLGAUS=0. GPU00170
C IF (KCALL.NE.0) GOTO 100 GPU00180
C KCALL=1 GPU00190
C ICASE=0 GPU00200
C*** COMPUTE LOS GENERALIZED COORDINATES. GPU00210
C IF (ZPP1.LT.0. .AND. ZPP2.LT.0.) RETURN GPU00220
C DELX=XPP2-XPP1 GPU00230
C DELY=YPP2-YPP1 GPU00240
C DELZ=ZPP2-ZPP1 GPU00250
C IF (ABS(DELX).LE..01*ABS(DELY) .OR. ABS(DELX).LE..01*ABS(DE LZ)) GPU00260
C 1 GOTO 10 GPU00270
C ICASE=1 GPU00280
C SX=DELX/DELX GPU00290
C SY=DELY/DELX GPU00300
C SZ=DELZ/DELX GPU00310
C XI=XPP1-SX*XPP1 GPU00320
C YI=YPP1-SY*XPP1 GPU00330
C ZI=ZPP1-SZ*XPP1 GPU00340
C AV(1)=XPP1 GPU00350
C AV(2)=XPP2 GPU00360
C GOTO 80 GPU00370
10 IF (ABS(DELZ).LE..01*ABS(DE LZ)) GOTO 20 GPU00380
C ICASE=2 GPU00390
C SX=DELX/DELY GPU00400
C SY=DELY/DELY GPU00410
C SZ=DELZ/DELY GPU00420
C XI=XPP1-SX*YPP1 GPU00430
C YI=YPP1-SY*YPP1 GPU00440
C ZI=ZPP1-SZ*YPP1 GPU00450
C AV(1)=YPP1 GPU00460
C AV(2)=YPP2 GPU00470
C GOTO 80 GPU00480
20 IF (ABS(DELZ).LT..001) RETURN GPU00490
C ICASE=3 GPU00500
C SX=DELX/DELZ GPU00510
C SY=DELY/DELZ GPU00520
C SZ=DELZ/DELZ GPU00530
C XI=XPP1-SX*ZPP1 GPU00540
C YI=YPP1-SY*ZPP1 GPU00550
C ZI=ZPP1-SZ*ZPP1 GPU00560
C AV(1)=ZPP1 GPU00570
C AV(2)=ZPP2 GPU00580
C GOTO 80 GPU00590
80 IF (ZPP1.LT.0.) AV(1)=-ZI/SZ GPU00600
C IF (ZPP2.LT.0.) AV(2)=-ZI/SZ GPU00610
C SMUL=SQRT(SX*SX+SY*SY+SZ*SZ) GPU00620
C IF (ICASE.EQ.0) RETURN GPU00630
C*** COMPUTE GAUSSIAN PARAMETERS, REAL AND REFLECTED IMAGE CLOUD TO GPU00640
C ACCOUNT FOR GROUND REFLECTED SMOKE GPU00650
C SIGZ=(2.73+C2*A)/2.15 GPU00660
C ZB=C-(2.73+C2*A) GPU00670
C IF (ZB.GE.C) RETURN GPU00680
C SIGY=B*SQRT(1.-(ZB/C)**2)/2.15 GPU00690
C IF (ZB.LT.0.) SIGY=B/2.15 GPU00700

```

```

SIGX=SIGY
ASIG=(SX/SIGX)**2 + (SY/SIGY)**2 + (SZ/SIGZ)**2
BMEAN=2.*(SX*(XI-A)/SIGX**2) + (SY*YI/SIGY**2)
BP(1)=BMEAN+2.*SZ*(ZI-ZB)/SIGZ**2
BP(2)=BMEAN+2.*SZ*(ZI+ZB)/SIGZ**2
CTOT=((XI-A)/SIGX)**2 + (YI/SIGY)**2
CP(1)=CTOT + ((ZI-ZB)/SIGZ)**2
CP(2)=CTOT + ((ZI+ZB)/SIGZ)**2
C*** CALCULATE FOR LOS INTEGRAL
CMUL=SMUL/(2.*3.14159*SIGX*SIGY*SIGZ*SQRT(ASIG))
DO 220 I=1,2
CEXU=.5*(CP(1)-(BP(1)**2)/(4.*ASIG))
IF (CEXU.GT.20.) GOTO 220
C*** INFINITE PATH LOS
CLU=EXP(-CEXU)
C*** CORRECTION FOR FINITE PATH
DO 210 J=1,2
AP1=(AV(J)+BP(I))/(2.*ASIG)*SQRT(ASIG/2.)
P1=ABS(AP1)
CP1=0.
IF (P1.LE.5.) CP1=0.5/(1.+P1*(.0705230784+P1*(.042282013+P1*(
*.0092705272+P1*(.0001520134+P1*(.0002765672+P1*.0000430638))))))
**16
IF (AP1.GE.0.) CP1=1.-CP1
AP(J)=CP1
210 CONTINUE
CLGAUS=CLGAUS+CLU*ABS(AP(2)-AP(1))
220 CONTINUE
CLGAUS=CMUL*CLGAUS
RETURN
END

```

```

GPU00710
GPU00720
GPU00730
GPU00740
GPU00750
GPU00760
GPU00770
GPU00780
GPU00790
GPU00800
GPU00810
GPU00820
GPU00830
GPU00840
GPU00850
GPU00860
GPU00870
GPU00880
GPU00890
GPU00900
GPU00910
GPU00920
GPU00930
GPU00940
GPU00950
GPU00960
GPU00970
GPU00980
GPU00990
GPU01000
GPU01010

```



```

SUBROUTINE LZTRAN(WAVE1, ICLMAT, LAZTRN, IERR)
THIS SUBROUTINE CALCULATES MOLECULAR ABSORPTION COEFFICIENTS AT
LASER FREQUENCIES. PH20 AND T ARE THE PARTIAL PRESSURE OF WATER
VAPOR AND TEMPERATURE, IN TORR AND DEGREES K RESPECTIVELY.
LID IS THE LASER LINE IDENTIFICATION AS DESCRIBED BELOW.
ABCOEF IS THE ABSORPTION COEFFICIENT RETURNED IN KM-1.
*****
INPUT
CARD 1  LNAME1, LNAME2, PH20, T, LZPATH FORMAT (2A4,3F10.3)
LNAME1  FIRST 4 CHARS OF LASER LINE  (A4)
LNAME2  SECOND 4 CHARS OF LASER LINE  (A4)
      ***IF LNAME NOT ENTERED WILL USE WAVELENGTH READ IN MAIN
PH20    WATER VAPOR PRESSURE  0. TO .35  (MB)  (F10.3)
T       AMBIENT AIR TEMPERATURE 260. TO .320 (C)  (F10.3)
LZPATH  PATHLENGTH IN KM F(F10.3)
      ***PH20, AND T NOT REQUIRED WHEN ICLIMATE(IN MAIN)=1 OR 2
*****
OUTPUT
LZTRAN  TRANSMISSION
NOTES
ABCOEF  RETURNS THE ABSORPTION COEFFICIENT (KM-1)
LNAME1  FIRST 4 CHARS OF LASER LINE ON NORMAL RETURN
      BLANK ON ERROR RETURN
LNAME2  SECOND 4 CHARS OF LASER LINE ON NORMAL RETURN
      BLANK ON ERROR RETURN
++ CALLED PROGRAMS ++
LZIDNM
*****
LASER LINE IDENTIFICATION
LID=1  ND:YAG LASER, 1.06 MICRONS
LID=2  CO2 LASER LINE P(20) 10.591 MICRONS
LID=101 TO 127 OF LASER, 3.521 TO 4.089 MICRONS
101 P3(12) * 107 P3(8) * 113 P3(5) * 119 P2(5) * 125 P1(5)
102 P3(11) * 108 P2(11) * 114 P2(8) * 120 P1(8) * 126 P1(4)
103 P3(10) * 109 P3(7) * 115 P2(7) * 121 P2(4) * 127 P1(3)
104 P2(13) * 110 P2(10) * 116 P1(10) * 122 P1(7) *
105 P3(9) * 111 P3(6) * 117 P2(6) * 123 P2(3) *
106 P2(12) * 112 P2(9) * 118 P1(9) * 124 P1(6) *
LID=201 TO 219 CO LASER, 4.908 TO 5.088 MICRONS
201 P6(12) * 205 P6(8) * 209 P5(12) * 213 P5(8) * 217 P4(9)
202 P6(11) * 206 P5(15) * 210 P5(11) * 214 P5(7) * 218 P4(8)
203 P6(10) * 207 P5(14) * 211 P5(10) * 215 P4(11) * 219 P4(7)
204 P6(9) * 208 P5(13) * 212 P5(9) * 216 P4(10) *
LID=301 TO 305 GA AS LASER, (GA.85 TO GA.950) LASER LINE NAMES
301 0.850 MICROMETERS * 304 0.925 MICROMETERS
302 0.875 MICROMETERS * 305 0.950 MICROMETERS
303 0.900 MICROMETERS *
INTEGER LNAME1, LNAME2, BLANK, LNAME3, LNAME4
REAL LAZTRN, LZPATH
DIMENSION ADF0(30), ADF1(30), ADF2(30), ADF3(30), ADF4(30)
1 ADF5(30)
DIMENSION AC00(20), AC01(20), AC02(20), AC03(20), AC04(20)
1 AC05(20)
DIMENSION AGA0(5), AGA1(5), AGA2(5), AGA3(5), AGA4(5),
1 AGA5(5), AGA6(5), AGA7(5), AGA8(5)
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
COMMON /CLYMAT/TEMP, PRESS, RH, AH, DP, VIS, CLDAMT, CLDHYT,
1 FOGPRB, WNDVEL, WNDDIR, IPASCT

```

```

LZT00010
LZT00020
LZT00030
LZT00040
LZT00050
LZT00060
LZT00070
LZT00080
LZT00090
LZT00100
LZT00110
LZT00120
LZT00130
LZT00140
LZT00150
LZT00160
LZT00170
LZT00180
LZT00190
LZT00200
LZT00210
LZT00220
LZT00230
LZT00240
LZT00250
LZT00260
LZT00270
LZT00280
LZT00290
LZT00300
LZT00310
LZT00320
LZT00330
LZT00340
LZT00350
LZT00360
LZT00370
LZT00380
LZT00390
LZT00400
LZT00410
LZT00420
LZT00430
LZT00440
LZT00450
LZT00460
LZT00470
LZT00480
LZT00490
LZT00500
LZT00510
LZT00520
LZT00530
LZT00540
LZT00550
LZT00560
LZT00570
LZT00580
LZT00590
LZT00600
LZT00610
LZT00620
LZT00630
LZT00640
LZT00650
LZT00660
LZT00670
LZT00680
LZT00690
LZT00700

```

```

COMMON /IDUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTL LZT00710
COMMON /GEOMET/PTS(15), IGEOSW LZT00720
THE POLYNOMIAL COEFFICIENTS ARE SELECTED BY THE LID. THE LZT00730
INDEX FOR THE COEFFICIENT ARRAYS FOR THE DF LASER IS LZT00740
I = LID - 100. NOW I IS IN THE RANGE 1..27 SINCE THERE ARE LZT00750
27 DF LASER LINES. WHEN THE POLYNOMIAL IS EVALUATED I IS LZT00760
USED TO INDEX THE ARRAYS ADF/0..5/ THUS SELECTING THE LZT00770
CORRECT COEFFICIENTS FOR THE LASER LINE SELECTED. LZT00780
COEFFICIENTS FOR THE OTHER LASER POLYNOMIALS ARE SELECTED LZT00790
IN THE SAME FASHION. LZT00800
POLYNOMIAL COEFFICIENTS FOR DF LASER LINES, (1..27) LZT00810
DATA ADF0/.1019,.08352,.04083,.03675,.02042,.01833, LZT00820
1 .04738,.03134,.07870,.05844,.1096E-2,.9353E- LZT00830
2 .2537E-2,.3254E-2,-.1103E-2,-.6471E-3,-.1423E- LZT00840
3 -.4664E-2,-.1221E-4,-.1698E-3,-.1172E-3,.6195E- LZT00850
4 .1272E-2,.5485E-2,.1651E-2,.6913E-2,-.4498E- LZT00860
5 2.3*0./ LZT00870
DATA ADF1/-.9718E-4,-.1160E-3,-.4892E-4,-.4230E-4,- LZT00880
1 .1750E-4,-.1524E-4,-.5589E-4,-.4642E-4,-.1218E- LZT00890
2 -.3683E-4,.1672E-4,-.1346E-4,-.4765E-6,-.8548E- LZT00900
3 .6089E-5,.8897E-5,.1507E-5,.4448E-4,.4540E- LZT00910
4 -.5569E-6,.1327E-6,-.1462E-4,-.7524E-6,-.1398E- LZT00920
5 -.7025E-6,-.1044E-4,.1816E-4,3*0./ LZT00930
DATA ADF2/.9666E-2,.7252E-2,.7050E-2,.7142E-2,.6320E- LZT00940
1 .6191E-2,.5400E-2,.5344E-2,.4064E-2,.4682E- LZT00950
2 .3734E-2,.5839E-2,.5075E-2,.2484E-2,.8190E- LZT00960
3 .6920E-2,.8779E-2,.7914E-2,.7094E-2,.5327E- LZT00970
4 .6692E-2,.9452E-2,.01025,.01367,.01279,.7844E- LZT00980
5 2.01201,3*0./ LZT00990
DATA ADF3/-.2655E-4,-.1805E-4,-.1879E-4,-.1937E-4,- LZT01000
1 .1704E-4,-.1669E-4,-.1412E-4,-.1417E-4,-.9076E- LZT01010
2 5.-.1165E-4,-.7371E-5,-.9113E-5,-.1290E-4,- LZT01020
3 .1070E-5,-.6746E-5,-.1573E-4,-.1683E-4,-.1883E- LZT01030
4 -.1774E-4,.3002E-4,-.1025E-4,-.2488E-4,-.2596E- LZT01040
5 -.2848E-4,-.2858E-4,-.4991E-5,-.2608E-4,3*0./ LZT01050
DATA ADF4/.7847E-4,.5729E-4,.5585E-4,.5606E-4,.4847E- LZT01060
1 .4668E-4,.4145E-4,.4218E-4,.3314E-4,.3642E- LZT01070
2 .3170E-4,.3682E-4,.3835E-4,.4798E-4,.3537E- LZT01080
3 .4635E-4,.4941E-4,.5601E-4,.5494E-4,-.1120E- LZT01090
4 .6567E-4,.9104E-4,.7398E-4,.8509E-4,.8746E- LZT01100
5 .1050E-3,.1060E-3,3*0./ LZT01110
DATA ADF5/-.2056E-6,-.1374E-6,-.1408E-6,-.1432E-6,- LZT01120
1 .1222E-6,-.1172E-6,-.1011E-6,-.1054E-6,-.7169E- LZT01130
2 7.-.8600E-7,-.6588E-7,-.8998E-7,-.9081E-7,- LZT01140
3 .1320E-6,-.1570E-7,-.9751E-7,-.8159E-7,-.1234E- LZT01150
4 6.-.1313E-6,-.1836E-6,-.1469E-6,-.2540E-6,-.1796E- LZT01160
5 6.-.1691E-6,-.1886E-6,-.2635E-6,-.2855E-6,3*0./ LZT01170
C POLYNOMIAL COEFFICIENTS FOR CO LASER LINES, (1..19) LZT01180
DATA AC00/-1.813E-3,-9.289E-4,1.153E-3,-1.985E-3,- LZT01190
1 4.523E-3,-1.205E-3,-2.225E-4,-4.061E-3,-4.522E- LZT01200
2 -2.267E-6,-5.917E-3,-1.423E-3,-3.640E-3,1.096E- LZT01210
3 6.455E-4,-3.922E-3,-1.873E-5,-1.055E-4,1.489E- LZT01220
4 2.0./ LZT01230
DATA AC01/3 426E-5,3.658E-6,-1.372E-6,7.229E-6,1.641E- LZT01240
1 5,6.428E-6,1.042E-7,1.435E-5,1.593E-4,-5.334E- LZT01250
2 7,1.498E-5,5.284E-6,1.806E-5,-3.651E-6,-1.303E- LZT01260
3 6,1.835E-5,4.755E-6,2.330E-6,6.196E-6,0./ LZT01270
DATA AC02/8 .813E-2,-1.020E-1,4.881E-2,6.872E-2,- LZT01280
1 6.244E-1,4.474E-2,1.226E-2,-1.462E-2,1.490E- LZT01290
2 1,1.428E-2,1.934,9.034E-3,-1.091E-1,1.284E-2,- LZT01300
3 2.131E-2,6.543E-3,2.824E-2,9.463E-3,-9.885E-2,0./ LZT01310
DATA AC03/8 .384E-4,1.211E-3,-4.687E-6,-7.765E- LZT01320
1 5,2.641E-3,-3.135E-5,1.620E-4,6.707E-4,2.581E- LZT01330
2 3,4.018E-4,-3.113E-3,9.692E-3,7.907E-4,1.070E- LZT01340
3 4,3.026E-4,1.216E-4,-1.859E-5,4.274E-5,6.241E- LZT01350
4 4,0./ LZT01360
DATA AC04/2.850E-4,-4.934E-5,-2.176E-5,1.253E- LZT01370
1 4,1.158E-3,1.926E-5,9.823E-5,-2.395E-4,-6.255E- LZT01380
2 5,-6.630E-5,4.851E-3,-5.183E-5,-6.993E-4,-8.196E- LZT01390
3 5,-2.239E-4,-1.120E-4,5.415E-5,-8.393E-5,-5.893E- LZT01400

```

```

4 DATA AC05/4.209E-6,3.426E-6,5.383E-7,6.252E-8,-4.149E-
1 6,2.374E-7,-8.637E-9,3.454E-6,1.526E-5,2.140E-6,-
2 4.282E-6,5.433E-7,4.231E-6,7.778E-7,1.695E-
3 6,8.830E-7,-1.548E-8,4.629E-7,3.241E-6,0./
C POLYNOMIAL COEFFICIENTS FOR CO2 LASER
DATA AC020,AC021,AC022,AC023,AC024,AC025,AC026,AC027,
1 AC028/.4488,-4.1864E-3,5.7903E-2,-3.6770E-
2 4,3.8521E-3,-4.7330E-6,1.0000E-5,6.0131E-7,-
3 1.7441E-8/
C POLYNOMIAL COEFFICIENTS FOR GA AS LASER, (1.,5)
DATA AGA0/7.947E-3,7.590E-3,1.010,0.6094,4.271/
DATA AGA1/-3.543E-5,-3.544E-5,-8.012E-3,-4.777E-3,-
1 3.371E-2/
DATA AGA2/1.740E-4,1.203E-4,8.736E-2,4.527E-2,.3364/
DATA AGA3/-5.955E-7,-5.093E-7,-2.135E-4,-1.104E-4,-
1 8.425E-4/
DATA AGA4/-1.282E-5,-9.236E-6,-3.964E-3,-2.154E-3,-
1 1.615E-2/
DATA AGA5/5.586E-8,4.720E-8,1.596E-5,8.512E-6,6.350E-
5/
1 DATA AGA6/5.124E-8,5.235E-8,1.600E-5,9.392E-6,6.705E-
5/
1 DATA AGA7/2.706E-10,4.379E-10,-1.396E-7,-6.982E-8,-
1 4.721E-7/
1 DATA AGA8/-4.963E-11,-5.728E-11,-1.125E-8,-5.657E-9,-
1 4.113E-8/
DATA BLANK/1H /
C CHANGE ACCURACY TO 3 DECIMAL PTS (PGM DATA LIMIT)
WAVEL=FLOAT(IFIX(1000.*WAVEL))/1000.
READ (IOIN,1100) LNAME1,LNAME3,LNAME2,LNAME4,PH20,T,LZPATH
IF(IGEOSW.NE.1)GO TO 99
LZPATH=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+
+(PTS(6)-PTS(3))**2)
99 CONTINUE
C CHANGE UNITS - MB TO TORR; C TO K
PH20=PH20/TORRMB
T=T+CDEGK
IF (ICLMAT.EQ.1) T=TEMP+CDEGK
IF (ICLMAT.EQ.1) PH20=6.11*10.**
1 (7.5*TEMP/(TEMP+237.3))*RH/(100.*TORRMB)
C PRINT HEADER WHEN THE WAVELENGTH CHANGES
IF (OLDWAV.NE.WAVEL) WRITE (IOOUT,1000)
OLDWAV=WAVEL
ABCOEF=0.
IF (WAVEL.EQ.0.0) GO TO 100
IF (WAVEL.LT.0.8.OR.WAVEL.GT.11.0) GO TO 900
100 CALL LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)
IF (LID.EQ.0) RETURN
P2=PH20*PH20
IF (T.GE.260.AND.T.LE.320.AND.PH20.GE.0.AND.
1 PH20.LE.35) GO TO 200
C PRINT WARNING THAT TEMP OR PRESSURE IS OUT OF RANGE FOR
C ACCURATE CALCULATIONS AND CONTINUE.
WRITE (IOOUT,1300)
200 IF (LID.GT.100) GO TO 500
IF (LID.LT.1.OR.LID.GT.2) GO TO 900
IF (LID-2) 300,400,900
C ND:YAG LASER. NO MOLECULAR ABSORPTION AT 1.06 MICRONS.
300 GO TO 800
C CO2 LASER LINE P(20)
400 T2=T*T
ABCOEF=AC020+AC021*T+AC022*PH20+AC023*T*PH20+AC024*P2+
1 AC025*T*P2+AC026*T2+AC027*T2*PH20+AC028*T2*P2
GO TO 800
500 IF (LID.GT.200) GO TO 600
C DF LASER. I IS THE LASER LINE INDEX
I=LID-100
IF (I.GT.27) GO TO 900
ABCOEF=ADF0(I)+ADF1(I)*T+ADF2(I)*PH20+ADF3(I)*T*PH20+
LZT01410
LZT01420
LZT01430
LZT01440
LZT01450
LZT01460
LZT01470
LZT01480
LZT01490
LZT01500
LZT01510
LZT01520
LZT01530
LZT01540
LZT01550
LZT01560
LZT01570
LZT01580
LZT01590
LZT01600
LZT01610
LZT01620
LZT01630
LZT01640
LZT01650
LZT01660
LZT01670
LZT01680
LZT01690
LZT01700
LZT01710
LZT01720
LZT01730
LZT01740
LZT01750
LZT01760
LZT01770
LZT01780
LZT01790
LZT01800
LZT01810
LZT01820
LZT01830
LZT01840
LZT01850
LZT01860
LZT01870
LZT01880
LZT01890
LZT01900
LZT01910
LZT01920
LZT01930
LZT01940
LZT01950
LZT01960
LZT01970
LZT01980
LZT01990
LZT02000
LZT02010
LZT02020
LZT02030
LZT02040
LZT02050
LZT02060
LZT02070
LZT02080
LZT02090
LZT02100

```

```

1      ADF4(I)*P2+ADF5(I)*T*P2
GO TO 800
600 IF (LID.GT.300) GO TO 700
C CO LASER. I IS THE LASER LINE INDEX.
I=LID-200
IF (I.GT.19) GO TO 900
ABCOEF=AC00(I)+AC01(I)*T+AC02(I)*PH20+AC03(I)*T*PH20+
1 AC04(I)*P2+AC05(I)*T*P2
GO TO 800
700 IF (LID.GT.400) GO TO 900
C GA AS LASER. I IS WAVELENGTH INDEX.
I=LID-300
IF (I.GT.5) GO TO 900
T2=T*T
ABCOEF=AGA0(I)+AGA1(I)*T+AGA2(I)*PH20+AGA3(I)*T*PH20+
1 AGA4(I)*P2+AGA5(I)*T*P2+AGA6(I)*T2+AGA7(I)*T2*
2 PH20+AGA8(I)*T2*P2
GO TO 800
C NORMAL RETURN
800 IF (ABCOEF.LT.0.) ABCOEF=0.
C COMPUTE TRANSMISSION
LAZTRN=EXP(-LZPATH*ABCOEF)
WRITE (IOOUT,1200) WAVEL,PH20,T,ABCOEF,LNAME1,LNAME3,
+LNAME2,LNAME4,LZPATH,LAZTRN
RETURN
C ERROR RETURN
900 WRITE (IOOUT,1400)
LNAME1=BLANK
LNAME3=BLANK
LNAME2=BLANK
LNAME4=BLANK
LAZTRN=1.
IERR=1
RETURN

C
1000 FORMAT (/,69X,10HABSORPTION,/,23X,11H WAVELENGTH,4X,
1 12H20 PRESSURE,4X,11HTEMPERATURE,4X,
2 11HCOEFFICIENT,6X,4HLINE,9X,10HPATHLENGTH,4X,
3 12HTRANSMISSION,/,24X,9H(MICRONS),8X,6H(TORR)
4 11X,5H(ABS),10X,6H(KM-1),24X,4H(KM),/)
1100 FORMAT (4(A2),3F10.3)
1200 FORMAT (1H,22X,F09.3,F15.3,F16.2,E16.3,7X,4(A2),5X,
1 E10.4,5X,E10.4)
1300 FORMAT (39H *** WARNING VALUE OF T OR PH20 OUT OF ,
1 10HRANGE *** /28H T RANGE = 260 TO 320 K ,
2 25HPH20 RANGE = 0 TO 35 TORR)
1400 FORMAT (40H *** ERROR WAVELENGTH OUT OF ACCEPTABLE
1 7HRANGE: ,26H .8 TO 11.0 MICRONS ***
2 37H CONTROL RETURNED TO MAIN FROM LZTRAN)
END

```

```

LZT02110
LZT02120
LZT02130
LZT02140
LZT02150
LZT02160
LZT02170
LZT02180
LZT02190
LZT02200
LZT02210
LZT02220
LZT02230
LZT02240
LZT02250
LZT02260
LZT02270
LZT02280
LZT02290
LZT02300
LZT02310
LZT02320
LZT02330
LZT02340
LZT02350
LZT02360
LZT02370
LZT02380
LZT02390
LZT02400
LZT02410
LZT02420
LZT02430
LZT02440
LZT02450
LZT02460
LZT02470
LZT02480
LZT02490
LZT02500
LZT02510
LZT02520
LZT02530
LZT02540
LZT02550
LZT02560
LZT02570
LZT02580

```

```

C*****
C          SUBROUTINE LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)
C*****
C          THIS SUBROUTINE CONVERTS THE WAVELENGTH IN MICRONS OR THE LASER
C          LINE NAME, IF WAVEL = 0, TO AN INTEGER LASER ID NUMBER WHICH IS
C          USED BY LZTRAN.
C          ++ WAVEL TO LID, WHEN WAVEL NOT = 0 ++
C          THE SUBROUTINE DOES A BINARY SEARCH OF THE ARRAY AWAVEL TO FIND
C          A MATCH, WAVEL = AWAVEL(K). WHEN A MATCH IS FOUND THE LASER ID
C          IS LOADED FROM THE ID ARRAY, LID = AID(K). THE LASER LINE NAME
C          IS ALSO LOADED INTO TWO VARIABLES, LNAME1 = INAME1(K) LNAME2 =
C          INAME2(K). IF AN EXACT MATCH IS NOT FOUND THE CLOSEST STANDARD
C          TO THE PARAMETER WAVEL IS USED. A WARNING IS PRINTED AND K IS
C          SET TO THE PROPER VALUE SO IT CAN BE USED TO INDEX THE ID AND
C          LINE ARRAYS.
C          ++ LASER LINE TO LID, WHEN WAVEL = 0 ++
C          WHEN THE WAVELENGTH PARAMETER IS ZERO THE CONVERSION IS DONE
C          FROM LASER LINE NAME TO LID. A SEQUENTIAL SEARCH OF THE LINE
C          NAME ARRAY IS PERFORMED. WHEN A MATCH IS FOUND K IS SET AND
C          WAVEL AND LID ARE LOADED FROM THE APPROPRIATE ARRAYS. WHEN
C          NO MATCH IS FOUND AN ERROR MESSAGE IS PRINTED AND LID IS SET
C          TO ZERO. LID IS USED TO NOTIFY LZTRAN THAT AN ERROR HAS
C          OCCURED AND NO CALCULATIONS SHOULD BE PERFORMED.
C          ++ PARAMETERS ++
C          WAVEL      LASER WAVELENGTH (MICRONS)
C          *** INPUTS IF WAVEL = 0.0 ***
C          LNAME1     FIRST 4 CHARS OF LASER LINE
C          LNAME2     SECOND 4 CHARS OF LASER LINE
C          ++ RESULTS ++
C          LID        LASER LINE IDENTIFIER
C          LNAME1     FIRST 4 CHARS OF LASER LINE
C          LNAME2     SECOND 4 CHARS OF LASER LINE
C*****
C          INTEGER AID(53)
C          COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUL
C          DIMENSION AWAVEL(53),INAME1(106),INAME2(53)
C          DATA AWAVEL/,85.,875.,9.,
C          1 925.,95.,1.06,3.521,3.55,3.581,3.612,3.636,3.645,
C          2 3.666,3.679,3.698,3.715,3.731,3.752,3.765,3.8,
C          3 3.82,3.837,3.854,3.875,3.89,3.915,3.927,3.956,
C          4 3.965,3.999,4.005,4.046,4.089,4.908,4.918,4.928,
C          5 4.938,4.948,4.972,4.982,4.992,5.002,5.012,5.022,
C          6 5.032,5.043,5.047,5.054,5.057,5.067,5.078,5.088,
C          7 10.591/,
C          DATA AID/301,302,303,304,305,1,127,126,125,124,123,
C          1 122,121,120,119,118,117,116,115,114,113,112,111,
C          2 110,109,108,107,106,105,104,103,102,101,219,218,
C          3 217,216,215,214,213,212,211,210,209,208,207,205,
C          4 206,204,203,202,201,2/,
C          DATA INAME1/2HGA,2H.8,2HGA,2H.8,2HGA,2H.9,2HGA,2H.9,2HGA,2H.9,
C          1 2HRU,2HBY,2HP1,2H<3,2HP1,2H<4,2HP1,2H<5,2HP1,2H<6,2HP2,2H<3,
C          2 2HP1,2H<7,2HP2,2H<4,2HP1,2H<8,2HP2,2H<5,2HP1,2H<9,2HP2,2H<6,
C          3 2HP1,2H<1,2HP2,2H<7,2HP2,2H<8,2HP3,2H<5,2HP2,2H<9,2HP3,2H<6,
C          4 2HP2,2H<1,2HP3,2H<7,2HP2,2H<1,2HP3,2H<8,2HP2,2H<1,2HP3,2H<9,
C          5 2HP2,2H<1,2HP3,2H<1,2HP3,2H<1,2HP3,2H<1,2HP4,2H<7,2HP4,2H<8,
C          6 2HP4,2H<9,2HP4,2H<1,2HP4,2H<1,2HP5,2H<7,2HP5,2H<8,2HP5,2H<9,
C          7 2HP5,2H<1,2HP5,2H<1,2HP5,2H<1,2HP5,2H<1,2HP5,2H<1,2HP6,2H<8,
C          8 2HP5,2H<1,2HP6,2H<9,2HP6,2H<1,2HP6,2H<1,2HP6,2H<1,2HP6,2H<1,2HP6,2H<1,
C          DATA INAME2/1H5,2H75,1H,2H25,1H5,1H,11*1H,2H0,5*
C          1 1H),2H0),1H),2H1),1H),2H2),1H),2H3),2H0),2H1),
C          2 2H2),3*1H),2H0),2H1),3*1H),2H0),2H1),2H2),2H3),
C*****
LZI00010
LZI00020
LZI00030
LZI00040
LZI00050
LZI00060
LZI00070
LZI00080
LZI00090
LZI00100
LZI00110
LZI00120
LZI00130
LZI00140
LZI00150
LZI00160
LZI00170
LZI00180
LZI00190
LZI00200
LZI00210
LZI00220
LZI00230
LZI00240
LZI00250
LZI00260
LZI00270
LZI00280
LZI00290
LZI00300
LZI00310
LZI00320
LZI00330
LZI00340
LZI00350
LZI00360
LZI00370
LZI00380
LZI00390
LZI00400
LZI00410
LZI00420
LZI00430
LZI00440
LZI00450
LZI00460
LZI00470
LZI00480
LZI00490
LZI00500
LZI00510
LZI00520
LZI00530
LZI00540
LZI00550
LZI00560
LZI00570
LZI00580
LZI00660
LZI00670
LZI00680

```

```

3      2H4),1H),2H5),1H),2H0),2H1),2H2),1H)/
DATA IFIRST,ILAST/1,53/
I=IFIRST
J=ILAST
TWAVEL=WAVEL
C CHECK FOR WAVELENGTH OR LASER LINE PASSED AS INPUT. IF
C WAVEL = 0 THEN DO A SEQUENTIAL SEARCH ON LINE NAME,
C LNAME1,LNAME2,
IF (WAVEL.NE.0.0) GO TO 200
C INPUT = LASER LINE NAME
C SEQUENTIAL SEARCH LOOP
DO 100 K=IFIRST,ILAST
KK=2*K-1
IF((LNAME1.EQ.INAME1(KK)).AND.(LNAME3.EQ.INAME1(KK+1))) GO TO 10
GO TO 100
10 IF (LNAME2.EQ.INAME2(K)) GO TO 600
100 CONTINUE
C ERROR, NO MATCH ON LASER NAME
LID=0
C PRINT ERROR MESSAGE
WRITE (IOOUT,900) LNAME1,LNAME3,LNAME2,LNAME4
GO TO 700
C INPUT = WAVELENGTH
C BINARY SEARCH LOOP
200 K=(I+J)/2
IF (WAVEL.LE.AWAVEL(K)) J=K-1
IF (WAVEL.GE.AWAVEL(K)) I=K+1
IF (I.LE.J) GO TO 200
C DID WAVELENGTH MATCH A STANDARD IN AWAVEL(*) ?
IF (I-1.GT.J) GO TO 600
C WAVELENGTH NOT EXACTLY EQUAL TO ONE OF THE STANDARDS IN ARRAY
C AWAVEL. CHANGE WAVELENGTH TO EQUAL THE STANDARD IT IS CLOSEST TO.
C THEN PRINT WARNING OF CHANGE.
IF (WAVEL.GT.AWAVEL(K).AND.K.EQ.ILAST) GO TO 500
IF (WAVEL.LT.AWAVEL(K).AND.K.EQ.IFIRST) GO TO 500
IF (WAVEL-AWAVEL(K)) 300,600,400
C WAVEL LT AWAVEL(K)
C CHECK IF CLOSER TO AWAVEL(K) OR AWAVEL(K-1)
300 DELTA1=WAVEL-AWAVEL(K-1)
DELTA2=AWAVEL(K)-WAVEL
IF (DELTA1.LT.DELTA2) K=K-1
GO TO 500
C WAVEL GT AWAVEL(K)
C CHECK IF CLOSER TO AWAVEL(K) OR AWAVEL(K+1)
400 DELTA1=WAVEL-AWAVEL(K)
DELTA2=AWAVEL(K+1)-WAVEL
IF (DELTA2.LT.DELTA1) K=K+1
C PRINT WARNING
500 WAVEL=AWAVEL(K)
WRITE (IOOUT,800) TWAVEL,WAVEL
C LOAD LASER ID NUMBER
600 LID=AID(K)
C LOAD LINE NAME
KK=2*K-1
LNAME1=INAME1(KK)
LNAME3=INAME1(KK+1)
LNAME2=INAME2(K)
C LOAD WAVELENGTH
WAVEL=AWAVEL(K)
700 RETURN
C
800 FORMAT (29H *** WARNING INPUT WAVELENGTH,F7.3,
1      11H CHANGED TO,F7.3,10H NEAREST STANDARD ,
2      14HWAVELENGTH ***)
900 FORMAT (24H *** ERROR LASER LINE #,4(A2),6H# NOT ,
1      9HVALID ***,/,26H CONTROL RETURNED TO MAIN ,
2      12HFROM LZTRAN.)
END

```

```

LZI00690
LZI00700
LZI00710
LZI00720
LZI00730
LZI00740
LZI00750
LZI00760
LZI00770
LZI00780
LZI00790
LZI00800
LZI00820
LZI00830
LZI00840
LZI00850
LZI00860
LZI00870
LZI00880
LZI00890
LZI00900
LZI00910
LZI00920
LZI00930
LZI00940
LZI00950
LZI00960
LZI00970
LZI00980
LZI00990
LZI01000
LZI01010
LZI01020
LZI01030
LZI01040
LZI01050
LZI01060
LZI01070
LZI01080
LZI01090
LZI01100
LZI01110
LZI01120
LZI01130
LZI01140
LZI01150
LZI01160
LZI01170
LZI01180
LZI01190
LZI01200
LZI01210
LZI01220
LZI01230
LZI01240
LZI01250
LZI01260
LZI01270
LZI01280
LZI01290
LZI01300
LZI01310
LZI01320

```

SUBROUTINE DRTRAN(WAVE1, ICLMAT, TRNLOS, IERR)

DRT00010

PURPOSE

DIRTRAN-2 EXPLOSION PRODUCED AND VEHICLE GENERATED DUST MODEL

DRT00020

INPUT, OUTPUT AND CALLING PROGRAM

DRT00030

INPUTS

DRT00040

VALUES IN ARGUMENT LIST

DRT00050

ICLMAT INTEGER VALUE USED TO INDICATE HOW METEOROLOGICAL DATA IS TO BE MADE AVAILABLE IF ICLMAT IS

DRT00060

0 - MET1 IDENTIFIER WITH THE APPROPRIATE PARAMETERS ARE TO BE READ IN

DRT00070

1 - NECESSARY METEOROLOGICAL DATA IS PASSED IN COMMON/CLYMAT/ AND MET1 IS NOT TO BE READ IN

DRT00080

DRT00090

DRT00100

DRT00110

DRT00120

DRT00130

DRT00140

DRT00150

DRT00160

DRT00170

WAVE1 WAVELENGTH IN MICROMETERS. USED TO DETERMINE NWL, WHERE NWL IS AN INTEGER INDEX FOR WAVELENGTH DETERMINED WITHIN THE CODE

DRT00180

DRT00190

DRT00200

DRT00210

DRT00220

DRT00230

DRT00240

DRT00250

DRT00260

DRT00270

DRT00280

DRT00290

DRT00300

DRT00310

DRT00320

DRT00330

DRT00340

DRT00350

DRT00360

DRT00370

DRT00380

DRT00390

DRT00400

DRT00410

DRT00420

DRT00430

DRT00440

DRT00450

DRT00460

DRT00470

DRT00480

DRT00490

DRT00500

DRT00510

DRT00520

DRT00530

DRT00540

DRT00550

DRT00560

DRT00570

DRT00580

DRT00590

DRT00600

DRT00610

DRT00620

DRT00630

DRT00640

DRT00650

DRT00660

DRT00670

DRT00680

DRT00690

INPUTS TO BE READ

EACH INPUT RECORD BEGINS WITH A 4 LETTER IDENTIFIER IN COLUMNS 1-4 FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, UP TO 9, 8 COLUMNS PER FIELD BEGINNING IN COLUMN 9.

THE INPUT FILE MAY CONTAIN SEVERAL SEQUENCES OF THE FOLLOWING RECORDS. EACH SEQUENCE SEPERATED BY A GO CARD. ONCE THE INITIAL SEQUENCE HAS BEEN READ IN AND THE MINIMUM REQUIREMENTS FOR EXECUTION OF THE DESIRED OPTION HAS BEEN SATISFIED, ANY FOLLOWING SEQUENCE MAY CONTAIN A SUBSET OF THE INITIAL RECORDS REDEFINING INPUT VARIABLES AS DESIRED OR MAY CONTAIN A COMPLETELY NEW SET OF RECORDS.

** EACH SET OF INPUTS MUST END WITH A DONE CARD**

RECORD 1
MET1

NATMOS INTEGER WITH VALUES 1 TO 6 CORRESPONDING TO PASQUILL CATEGORIES A TO F.

DRT00470

DRT00480

DRT00490

ZTMP THE HEIGHT AT WHICH A TEMPERATURE MEASUREMENT IS AVAILABLE. VALID RANGE 0.5 - 100.0 M.

DRT00500

DRT00510

DRT00520

TMPMES THE TEMPERATURE MEASURED IN DEGREES KELVIN TAKEN AT HEIGHT ZTMP. VALID RANGE 270.0 - 315.0.

DRT00530

DRT00540

DRT00550

ZWND THE HEIGHT AT WHICH A WIND SPEED MEASUREMENT IS AVAILABLE. VALID RANGE 0.5 - 100.0 M.

DRT00560

DRT00570

DRT00580

WDMES THE WIND SPEED IN METERS/SECOND MEASURED AT ZWND VALID RANGE .1 - 20.0 M/S

DRT00590

DRT00600

DRT00610

THWND THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES WITH THE USER'S POSITIVE X AXIS MEASURED IN DEGREES COUNTERCLOCKWISE, WHERE THE USERS POSITIVE X-AXIS POINTS EAST. THUS THWND IS THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES WITH THE EAST. VALID RANGE: -360.0 - 360.0 DEGREES. NOTE: THWND IS NOT NEEDED FOR OPTION 3

DRT00620

DRT00630

DRT00640

DRT00650

DRT00660

DRT00670

DRT00680

DRT00690

RECORD 2		DRT00700
MET2		DRT00710
ID	A FLAG TO INDICATE WHETHER THE INVERSION LAYER HEIGHT IS GROWING OR NOT. IF ID IS	DRT00720
	0. THE INVERSION LAYER HEIGHT IS RELATIVELY CONSTANT	DRT00730
	1. THE INVERSION LAYER HEIGHT IS GROWING	DRT00740
PHI	THE LATITUDE OF THE DETONATION SITE, VALID RANGE 1.0 - 90. DEGREES. THAT IS THE NORTHERN HEMISPHERE.	DRT00750
		DRT00760
		DRT00770
		DRT00780
		DRT00790
		DRT00800
RECORD 3		DRT00810
SOIL		DRT00820
		DRT00830
		DRT00840
NSOIL	INTEGER INDEX OF SOIL TYPE. NSOIL IS	DRT00850
	1. FOR SOIL-1, <DATA GRAF-II> EXPLOSIONS ONLY.	DRT00860
	2. FOR SOIL-2, <DATA DIRT-I> EXPLOSIONS ONLY.	DRT00870
	3. FOR SOIL-3, <DATA SMOKEWEEK-II> VEHICLES ONLY.	DRT00880
DSOD	DEPTH OF SOD IN METERS	DRT00890
	VALID RANGE: 0.0 - 1.0 M.	DRT00900
	NOTE: FOR VEHICLE MODEL IF DSOD>0.0 NO DUST IS GENERATED	DRT00910
		DRT00920
		DRT00930
		DRT00940
SILT	SILT CONTENT OF SOIL <PARTICLE DIAMETERS < 75 MICRONS> I.E. SILT=.15 INDICATES A SILT CONTENT OF 15%	DRT00950
	NOTE: THIS INPUT NEEDED ONLY FOR VEHICLE MODEL <IOPT=5>	DRT00960
		DRT00970
		DRT00980
		DRT00990
RECORD 4		DRT01000
CHAR		DRT01010
		DRT01020
NCHRG	CHARGE TYPE INDEX WITH FOLLOWING VALUES	DRT01030
	1. SURFACE - LIVE FIRE OR 30 DEGREE TILTED	DRT01040
	STATIC, TIP ON GROUND	DRT01050
	2. BARE CHARGE ON SURFACE	DRT01060
	3. 30 DEGREE TILTED TIP AT 0.3 METER DEPTH	DRT01070
	4. 30 DEGREE TILTED TIP AT 0.6 METER DEPTH	DRT01080
	5. HORIZONTAL PROJECTILE ON SURFACE	DRT01090
	DEFAULT VALUE IS 1 IF NCHRG IS NOT BETWEEN 1 AND 5.	DRT01100
CHWT	THE WEIGHT OF THE CHARGE IN KG-TNT.	DRT01110
	VALID RANGE: 0.1 - 100.0 KG-TNT.	DRT01120
		DRT01130
		DRT01140
DETDEP	THE DEPTH OF DETONATION IN METERS.	DRT01150
	VALID RANGE: 0.0 - 2.0 M.	DRT01160
		DRT01170
RECORD 5		DRT01180
EXPL		DRT01190
		DRT01200
NARY	TYPE OF CHARGE DISTRIBUTION <USED FOR PROPER INPUT AND OUTPUT FORMATS> IF THE VALUE OF NARY AND IOPT OF THE CARD ARE NOT COMPATIBLE CATASTROPHE COULD RESULT!	DRT01210
	NOTE: WHEN NARY IS	DRT01220
	1. IOPT MUST ALSO BE 1.	DRT01230
	2. IOPT MUST ALSO BE 2.	DRT01240
	3. IOPT MUST BE 4.	DRT01250
	4. IOPT MUST BE 4.	DRT01260
	1.-SIMULTANEOUS BURST, UNIFORMLY DISTRIBUTED CHARGES IN A PARALLELOGRAM.	DRT01270
	<SPECIAL CASES ARE ,SINGLE CHARGE ,RECTANGLE AND ZIG ZAG PATTERN>	DRT01280
		DRT01290
		DRT01300
		DRT01310
		DRT01320
	2.-SIMULTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES.	DRT01330
		DRT01340
	3.-SEQUENTIAL IN TIME AND RANDOM IN SPACE DISTRIBUTION OF CHARGES.	DRT01350
		DRT01360
		DRT01370
		DRT01380
	NOTE: WHEN NARY=2. EACH CHARGE LOCATION MUST BE SPECIFIED	DRT01390

	X-AXIS POINTS EAST, THUS VEHDIR IS THE ANGLE THE VELOCITY VECTOR MAKES WITH THE EAST. VALID RANGE: -360.0 - 360.0	DRT02100 DRT02110 DRT02120 DRT02130 DRT02140 DRT02150 DRT02160 DRT02170 DRT02180 DRT02190 DRT02200 DRT02210 DRT02220 DRT02230 DRT02240 DRT02250 DRT02260 DRT02270 DRT02280 DRT02290 DRT02300 DRT02310 DRT02320 DRT02330
VEHSPD	VEHICLE SPEED IN M/S	
VEHWID	VEHICLE WIDTH IN METERS	
VEHWHT	VEHICLE WEIGHT IN KGS.	
VEHTYP	TRACTION MECHANISM =0. VEHICLE HAS TIRES =1. VEHICLE IS TRACKED	
RECORD 8		
TRNC		
TRNCOR	A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE COORDINATES OF THE TRANSMITTER. THE COORDINATE SYSTEM MUST BE IN METERS. THE THIRD COORDINATE IS RESTRICTED TO BE BETWEEN .5 AND 10000.0 METERS (HEIGHT). VALID RANGE OF THE FIRST TWO COORDINATES: -10000.0 - 10000.0 M.	DRT02340 DRT02350 DRT02360 DRT02370 DRT02380 DRT02390 DRT02400 DRT02410 DRT02420 DRT02430 DRT02440 DRT02450 DRT02460
	** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE ** ARRAY TRNCOR NEED NOT BE SPECIFIED.	
TRNMIN	VALUE SUCH THAT A TRANSMITTANCE BELOW THIS VALUE CAN BE CONSIDERED ZERO. DEFAULT IS 1.E-05 VALID RANGE: 1.0 - 1.E-05	DRT02340 DRT02350 DRT02360 DRT02370 DRT02380 DRT02390 DRT02400 DRT02410 DRT02420 DRT02430 DRT02440 DRT02450 DRT02460
RECORD 9		
RECC		
RECCOR	A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE COORDINATES OF THE RECEIVER. (METERS) THE THIRD COORDINATE IS RESTRICTED TO BE BETWEEN .5 AND 10000.0 METERS. VALID RANGE OF THE FIRST TWO COORDINATES IS: -10000.0 - 10000.0 M.	DRT02470 DRT02480 DRT02490 DRT02500 DRT02510 DRT02520 DRT02530 DRT02540 DRT02550 DRT02560 DRT02570
	** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE ** ARRAY RECCOR NEED NOT BE SPECIFIED.	
RECORD 10		
OBSC		
OBSCOR	A SINGLY DIMENSIONED ARRAY CONTAINING THE X AND Y COORDINATES, RESP., OF THE OBSERVER. (METERS) VALID RANGE: -10000.0 - 10000.0	DRT02470 DRT02480 DRT02490 DRT02500 DRT02510 DRT02520 DRT02530 DRT02540 DRT02550 DRT02560 DRT02570
SPCHT	A SPECIFIED HEIGHT IN METERS AT WHICH THE WIDTH OF THE CLOUD AS VIEWED FROM POSITION OBSCOR IS DESIRED. MUST BE BETWEEN 1. AND 5. METERS.	DRT02550 DRT02560 DRT02570
	** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE ** ARRAY OBSCOR AND VARIABLE SPCHT NEED NOT BE SPECIFIED.	
RECORD 11		
TIMS		
TSTART	TIME AFTER DETONATION TO START TRANSMITTANCE AND/OR CLOUD DIMENSION CALCULATIONS VALID RANGE: .5 - 1000.0 SEC.	DRT02580 DRT02590 DRT02600 DRT02610 DRT02620 DRT02630 DRT02640 DRT02650 DRT02660 DRT02670 DRT02680 DRT02690 DRT02700
TEND	TIME AFTER DETONATION TO TERMINATE TRANSMITTANCE AND/OR CLOUD DIMENSIONS. VALID RANGE: .5-1000.0 SEC. (TEND MUST BE .GE. TSTART)	
TINC	TIME INCREMENT BETWEEN CALCULATIONS	

```

RECORD 12
GO ** THIS CARD INDICATES THAT THIS SEQUENCE OF INPUTS ARE
COMPLETE AND CALCULATIONS ARE TO BEGIN.

ILOPT OPTION TO BE USED

1. SIMULTANEDUS BURST ,UNIFORMLY DISTRIBUTED CHARGES IN A
PARALLELOGRAM

2. SIMUTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES

3. THE CODE IS TO PRECOMPUTE A SINGLE CLOUD AND STORE ON
AN EXTERNAL FILE FOR USE LATER

4. THE CODE IS TO USE A CLOUD THAT HAS BEEN PRECOMPUTED
<NO CLOUD DIMENSIONS ARE COMPUTED FOR THIS OPTION>

5. VEHICLE DUST MODEL

IFILE FORTRAN LOGICAL UNIT TO WHICH THE CODE IS TO WRITE FOR
OPTION 3 OTHERWISE IT NEED NOT BE SPECIFIED

RECORD 13
DONE ** THIS RECORD INDICATES THAT THE USER HAS COMPLETED HIS
DESIRED SEQUENCE OF INPUTS AND ALL CALCULATIONS ARE
TERMINATED

OUTPUTS

ZINV THE ESTIMATED INVERSION HEIGHT.

TRNLDS THE TRANSMITTANCE ALONG THE LINE OF SIGHT BETWEEN
THE TRANSMITTER AND THE RECEIVER.

IERR INTEGER ERROR CODE WHICH EQUALS 1 IF A FATAL ERROR
OCCURS AND 0 OTHERWISE

NERR INTEGER ERROR CODE WITH THE VALUES
0 NO ERRORS
4 NO TRANSMITTER AND RECEIVER OR OBSERVER
COORDINATES WERE SPECIFIED SO NO RESULTS WERE
CALCULATED.
7 THE CALCULATION OF ATMOSPHERIC PARAMETERS DID
NOT CONVERGE.

CNTRD A SINGLY DIMENSIONED ARRAY CONTAINING THE HORIZONTAL
COORDINATE AND THE VERTICAL COORDINATE OF THE
CENTROID OF THE CLOUD.

HEIGHT THE HEIGHT OF THE CLOUD IN METERS.

CENWTH THE WIDTH OF THE CLOUD IN METERS AT THE CENTROID
HEIGHT

SPCWTH THE WIDTH OF THE CLOUD IN METERS AT THE SPECIFIED
HEIGHT

NCPTS THE NUMBER OF POINTS DETERMINED ON THE EDGE OF THE
CLOUD.

CPTS A DOUBLY DIMENSIONED ARRAY CONTAINING THE COORDINATES
OF POINTS ON THE EDGE OF THE CLOUD. CPTS(I,J)
IS THE HORIZONTAL COORDINATE OF THE J-TH POINT
AND CPTS(2,J)IS THE VERTICAL COORDINATE OF THE
J-TH POINT. THE FIRST INDEX MUST BE DIMENSIONED
TO 2.

```

```

DRT02710
DRT02720
DRT02730
DRT02740
DRT02750
DRT02760
DRT02770
DRT02780
DRT02790
DRT02800
DRT02810
DRT02820
DRT02830
DRT02840
DRT02850
DRT02860
DRT02870
DRT02880
DRT02890
DRT02900
DRT02910
DRT02920
DRT02930
DRT02940
DRT02950
DRT02960
DRT02970
DRT02980
DRT02990
DRT03000
DRT03010
DRT03020
DRT03030
DRT03040
DRT03050
DRT03060
DRT03070
DRT03080
DRT03090
DRT03100
DRT03110
DRT03120
DRT03130
DRT03140
DRT03150
DRT03160
DRT03170
DRT03180
DRT03190
DRT03200
DRT03210
DRT03220
DRT03230
DRT03240
DRT03250
DRT03260
DRT03270
DRT03280
DRT03290
DRT03300
DRT03310
DRT03320
DRT03330
DRT03340
DRT03350
DRT03360
DRT03370
DRT03380
DRT03390
DRT03400

```

CCCCCCCCCCCCCCCC

```

SUBROUTINES CALLED
DUSTCL  CONTROLLING ROUTINE FOR THE CALCULATION OF CLOUD DIMENSIONS
        AND TRANSMITTANCES THROUGH DUST CLOUDS FOR OPTIONS 1 AND 2
        GIVEN METEOROLOGICAL DATA, SOIL AND EXPLOSIVE
        CHARACTERISTICS, AND WAVELENGTH.
COMPCL  CONTROLLING ROUTINE FOR PRECOMPUTING A SINGLE EXPLOSION
        (OPTION 3) GIVEN METEOROLOGICAL DATA, SOIL AND EXPLOSION
        CHARACTERISTICS. ALSO USES THIS PRECOMPUTED CLOUD AT SOME
        LATER RUNNING OF THE CODE (OPTION 4) TO ESTIMATE A
        TRANSMITTANCE GIVEN TRANSMITTER AND RECEIVER COORDINATES.
VEHCL   CONTROLLING ROUTINE FOR THE CALCULATION OF A TRANSMITTANCE
        THROUGH A VEHICLE GENERATED DUST CLOUD (OPTION 5) GIVEN
        METEOROLOGICAL DATA, SOIL CHARACTERISTICS, VEHICLE
        CHARACTERISTICS, AND WAVELENGTH.

```

DRT03410
DRT03420
DRT03430
DRT03440
DRT03450
DRT03460
DRT03470
DRT03480
DRT03490
DRT03500
DRT03510
DRT03520
DRT03530
DRT03540
DRT03550
DRT03560
DRT03570
DRT03580

```

*****
LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,          CLMRED,DHDT,ONCE
LOGICAL TEST,NEWVEH,NEWCOR
LOGICAL M1,M2,SL,CH,EX,TC,RC,OC, TM, VH
INTEGER VEHTYP
REAL M,N

```

DRTU3590
DRT03600
DRT03610
DRT03620
DRT03630

```

DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WDMES(2),TRNCOR(3)
DIMENSION SRCBAS(2),SIDE1(2),SIDE2(2),NCHS(2)
1 RECCOR(3),CPTS(2,6),CNTRD(2),OBSCOR(2)
DIMENSION RDIN(10),RKEY(12),V0(2),PAS(6)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /CLYMAT/ TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
1 FOGPRB,WINDVEL,WINDIR,IPASCT
COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ COOR(2,200),TSTAG(200),DMMY(401)
COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON /GEOMET/PTS(15),IGEOSW
COMMON/OPTION/IOPT,IFILE
DATA RKEY/4HMET1,4HMET2,4HSOIL,4HCHAR,4HEXPL,4HVEHC,4HTRNC,
1 4HRECC,4HOBSC,4HTIMS,4HGO ,4HDONE,
DATA PAS/4HA ,4HB ,4HC ,4HD ,4HE ,4HF /
DATA M1,M2,SL,CH,EX,TC,RC,OC, TM, VH/.FALSE.,.FALSE.,.FALSE.,.
1 .FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE./
DATA NEWATM,NEWSRC,NEWVEH,LOSTRN,EDGE,NEWTIM/.FALSE.,.FALSE.,
1 .FALSE.,.FALSE.,.FALSE.,.FALSE./
DATA NEWCOR/.FALSE./
DATA VISEXT,RTPI/.1,1.772454/
IERR=0
CLMRED=.FALSE.
DHDT=.FALSE.
ONCE=.FALSE.
TEST=.FALSE.
WRITE( IOOUT,800)
800 FORMAT(1H0,36X,42HDIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION,
1 15H CALCULATION,/,36X,60H*** NOTE -- ALL UNITS ARE MKS UNLESS
2 OTHERWISE SPECIFIED ***,/)
DO 5 K=1,200
TSTAG(K)=0.0
5 CONTINUE

```

DRT03640
DRT03650
DRT03660
DRT03670
DRT03680
DRT03690
DRT03700
DRT03710
DRT03720
DRT03730
DRT03740
DRT03750
DRT03760
DRT03770
DRT03780
DRT03790
DRT03800
DRT03810
DRT03820
DRT03830
DRT03840
DRT03850
DRT03860
DRT03870
DRT03880
DRT03890
DRT03900
DRT03910
DRT03920
DRT03930
DRT03940
DRT03950
DRT03960
DRT03970
DRT03980
DRT03990
DRT04000

```

DETERMINE INTEGER INDEX FOR WAVELENGTH
10 IF(WAVE1.LT.0.4)GO TO 29
IF(WAVE1.GT.0.7)GO TO 21
NWL=1
GO TO 30
21 IF(WAVE1.LT.0.8)GO TO 29
IF(WAVE1.GT.1.1)GO TO 22
NWL=2
GO TO 30

```

DRT04010
DRT04020
DRT04030
DRT04040
DRT04050
DRT04060
DRT04070
DRT04080
DRT04090

CCC

```

22 IF(WAVE1.LT.3.5)GO TO 29
IF(WAVE1.GT.4.0)GO TO 23
NWL=3
GO TO 30
23 IF(WAVE1.LT.8.5)GO TO 29
IF(WAVE1.GT.12.0)GO TO 24
NWL=4
GO TO 30
24 IF(WAVE1.LT.2100.)GO TO 29
IF(WAVE1.GT.3200.)GO TO 29
NWL=5
GO TO 30
29 WRITE(1000,802)
802 FORMAT(37X,38H*** DIRTRAN ERROR - WAVE1 OUT OF RANGE)
IERR=1
GO TO 999
30 CONTINUE
C
C C READ DATA AND STORE APPROPRIATELY
C
DO 300 II=1,15
IF(II.EQ.15)GO TO 900
READ(1010,700)(RDIN(J),J=1,10)
700 FORMAT(A4,4X,9F8.2)
IF(RDIN(1).EQ.RKEY(1))GO TO 50
IF(RDIN(1).EQ.RKEY(2))GO TO 70
IF(RDIN(1).EQ.RKEY(3))GO TO 90
IF(RDIN(1).EQ.RKEY(4))GO TO 110
IF(RDIN(1).EQ.RKEY(5))GO TO 130
IF(RDIN(1).EQ.RKEY(6))GO TO 150
IF(RDIN(1).EQ.RKEY(7))GO TO 170
IF(RDIN(1).EQ.RKEY(8))GO TO 190
IF(RDIN(1).EQ.RKEY(9))GO TO 210
IF(RDIN(1).EQ.RKEY(10))GO TO 230
IF(RDIN(1).EQ.RKEY(11))GO TO 310
IF(RDIN(1).EQ.RKEY(12))GO TO 999
WRITE(1000,804)
804 FORMAT(33X,52H***DIRTRAN-2 ERROR, INPUT DOES NOT CONFORM TO PROPER
1 14H CONVENTION**)
WRITE(1000,806)(RDIN(J),J=1,9)
806 FORMAT(26X,A4,4X,9F8.2)
GO TO 999
C
C C STORE AND PRINT OUT ATMOSPHERIC CONDITIONS
C
50 CONTINUE
M1=.TRUE.
NIO=1
IF(ICLMAT.EQ.1)GO TO 55
NATMOS=IFIX(RDIN(2))
ZTMP(1)=RDIN(3)
TMPMES(1)=RDIN(4)
ZWND(1)=RDIN(5)
WNDMES(1)=RDIN(6)
THWND=RDIN(7)
GO TO 60
C
C C C IPSCAT PASQUILL CATEGORY
C C WNDVEL WIND VELOCITY IN M/S MEASURED AT 2 M. ABOVE GROUND
C C WINDIR WIND DIRECTION IN DEGREES CLOCKWISE FROM TRUE NORTH
C C TEMP TEMPERATURE IN DEGREES C MEASURED AT 2 M. ABOVE GROUND
55 NATMOS=IPASCT
ZTMP(1)=2.
ZWND(1)=2.
WNDMES(1)=WNDVEL
TMPMES(1)=TEMP+273.0
THWND=270.0-WINDIR
60 CONTINUE
WRITE(1000,808)

```

```

DRT04100
DRT04110
DRT04120
DRT04130
DRT04140
DRT04150
DRT04160
DRT04170
DRT04180
DRT04190
DRT04200
DRT04210
DRT04220
DRT04230
DRT04240
DRT04250
DRT04260
DRT04270
DRT04280
DRT04290
DRT04300
DRT04310
DRT04320
DRT04330
DRT04340
DRT04350
DRT04360
DRT04370
DRT04380
DRT04390
DRT04400
DRT04410
DRT04420
DRT04430
DRT04440
DRT04450
DRT04460
DRT04470
DRT04480
DRT04490
DRT04500
DRT04510
DRT04520
DRT04530
DRT04540
DRT04550
DRT04560
DRT04570
DRT04580
DRT04590
DRT04600
DRT04610
DRT04620
DRT04630
DRT04640
DRT04650
DRT04660
DRT04670
DRT04680
DRT04690
DRT04700
DRT04710
DRT04720
DRT04730
DRT04740
DRT04750
DRT04760
DRT04770
DRT04780
DRT04790

```

808	FORMAT(1X)		DRT04800
	WRITE(IOOUT,810)PAS(NATMOS)		DRT04810
810	FORMAT(50X,28HPASQUILL CATEGORY	,A4)	DRT04820
	WRITE(IOOUT,812)(ZTMP(I),TMPMES(I),ZWND(I),WDMES(I),I=1,NIO)		DRT04830
812	FORMAT(36X,4H HT ,F8.2,7H TEMP ,F8.2,7H HT,F8.2,7H WIND ,		DRT04840
	F8.2)		DRT04850
	WRITE(IOOUT,814)THWND		DRT04860
814	FORMAT(51X,22H WIND DIRECTION	,F8.2)	DRT04870
	GO TO 300		DRT04880
70	CONTINUE		DRT04890
	M2=.TRUE.		DRT04900
	ID=IFIX(RDIN(2))		DRT04910
	IF(ID.NE.0)DHDT=.TRUE.		DRT04920
	PHI=RDIN(3)		DRT04930
	WRITE(IOOUT,819)PHI		DRT04940
	IF(DHDT)WRITE(IOOUT,816)		DRT04950
816	FORMAT(47X,37HTHE INVERSION LAYER HEIGHT IS GROWING)		DRT04960
	IF(.NOT.DHDT)WRITE(IOOUT,818)		DRT04970
818	FORMAT(47X,38HTHE INVERSION LAYER HEIGHT IS CONSTANT)		DRT04980
819	FORMAT(/,52X,20HLATITUDE	,F8.2)	DRT04990
	GO TO 300		DRT05000
C	STORE AND WRITE SOIL CHARACTERISTICS		DRT05010
C			DRT05020
C			DRT05030
90	CONTINUE		DRT05040
	SL=.TRUE.		DRT05050
	NSOIL=IFIX(RDIN(2))		DRT05060
	DSOD=RDIN(3)		DRT05070
	SILT=RDIN(4)		DRT05080
	IF(NSOIL.EQ.1)WRITE(IOOUT,821)		DRT05090
820	FORMAT(56X,15HSILT CONTENT	,F5.2)	DRT05100
821	FORMAT(/,63X,6HSOIL-1)		DRT05110
	IF(NSOIL.EQ.2)WRITE(IOOUT,822)		DRT05120
822	FORMAT(/,63X,6HSOIL-2)		DRT05130
	IF(NSOIL.EQ.3)WRITE(IOOUT,710)		DRT05140
710	FORMAT(/,23X,8H SOIL-3)		DRT05150
	IF(SILT.GT.1.E-06)WRITE(IOOUT,820)SILT		DRT05160
	WRITE(IOOUT,823)DSOD		DRT05170
823	FORMAT(53X,21H DEPTH OF SOD	,F5.2)	DRT05180
	IF(NSOIL.LT.1.OR.NSOIL.GT.2)NSOIL=2		DRT05190
	GO TO 300		DRT05200
C	STORE AND WRITE EXPLOSIVE CHARGE CHARACTERISTICS		DRT05210
C			DRT05220
C			DRT05230
110	CONTINUE		DRT05240
	CH=.TRUE.		DRT05250
	NCHRG=IFIX(RDIN(2))		DRT05260
	IF(NCHRG.LT.1.OR.NCHRG.GT.5)NCHRG=1		DRT05270
	CHWT=RDIN(3)		DRT05280
	DETDEP=RDIN(4)		DRT05290
	IF(NCHRG.EQ.1)WRITE(IOOUT,824)		DRT05300
824	FORMAT(/,35X,47HSURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC,,		DRT05310
	14H TIP ON GROUND)		DRT05320
	IF(NCHRG.EQ.2)WRITE(IOOUT,825)		DRT05330
825	FORMAT(/,55X,22HBARE CHARGE ON SURFACE)		DRT05340
	IF(NCHRG.EQ.3)WRITE(IOOUT,826)		DRT05350
826	FORMAT(/,46X,39H30 DEGREE TILTED TIP AT 0.3 METER DEPTH)		DRT05360
	IF(NCHRG.EQ.4)WRITE(IOOUT,827)		DRT05370
827	FORMAT(/,46X,39H30 DEGREE TILTED TIP AT 0.6 METER DEPTH)		DRT05380
	IF(NCHRG.EQ.5)WRITE(IOOUT,828)		DRT05390
828	FORMAT(/,50X,32HHORIZONTAL PROJECTILE ON SURFACE)		DRT05400
	WRITE(IOOUT,829)CHWT		DRT05410
829	FORMAT(45X,30HWEIGHT OF CHARGE	,F8.2,4H KG.)	DRT05420
	WRITE(IOOUT,830)DETDEP		DRT05430
830	FORMAT(47X,30HDETONATION DEPTH	,F8.2)	DRT05440
	GO TO 300		DRT05450
C	STORE AND WRITE OUT INFORMATION ABOUT THE DETONATION LOCATIONS		DRT05460
C			DRT05470
C			DRT05480
130	CONTINUE		DRT05490

	EX=.TRUE.	DRT05500
	NARY=IFIX(RDIN(2))	DRT05510
	NCHS(1)=IFIX(RDIN(3))	DRT05520
	NCHS(2)=IFIX(RDIN(4))	DRT05530
	SRCBAS(1)=RDIN(5)	DRT05540
	SRCBAS(2)=RDIN(6)	DRT05550
	SIDE1(1)=RDIN(7)	DRT05560
	SIDE1(2)=RDIN(8)	DRT05570
	SIDE2(1)=RDIN(9)	DRT05580
	SIDE2(2)=RDIN(10)	DRT05590
	IF(NARY.EQ.2)GO TO 133	DRT05600
	IF(NARY.EQ.3)GO TO 136	DRT05610
C	CHARGE DISTRIBUTION TYPE 1	DRT05620
	WRITE(IOOUT,831)	DRT05630
831	FORMAT(/,31X,42HSIMULTANEOUS BURST , UNIFORMLY DISTRIBUTED, +27H CHARGES IN A PARALLELOGRAM)	DRT05640
	NCH=NCHS(1)*NCHS(2)	DRT05650
	WRITE(IOOUT,832)NCH,(SRCBAS(I),I=1,2)	DRT05660
832	FORMAT(27X,26HTOTAL NUMBER OF CHARGES IS ,1X,13,1X, +27H WITH REFERENCE CHARGE AT (,F8.2,1H,,F8.2,1H))	DRT05670
	WRITE(IOOUT,834)NCHS(1),(SIDE1(I),I=1,2)	DRT05680
834	FORMAT(32X,13,1X,45HCHARGES WITH DIRECTION AND SPACING GIVEN BY (, +F8.2,1H,,F8.2,1H))	DRT05690
	WRITE(IOOUT,834)NCHS(2),(SIDE2(I),I=1,2)	DRT05700
	GO TO 300	DRT05710
C	CHARGE DISTRIBUTION TYPE 2	DRT05720
133	NCH=NCHS(1)	DRT05730
	DO 134 J=1,NCH	DRT05740
	READ(IOIN,701)(COOR(K,J),K=1,2)	DRT05750
701	FORMAT(8X,2F8.2)	DRT05760
134	CONTINUE	DRT05770
	WRITE(IOOUT,836)	DRT05780
836	FORMAT(/,42X,48HSIMULTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES)	DRT05790
	WRITE(IOOUT,838)NCHS(1)	DRT05800
838	FORMAT(51X,26HTOTAL NUMBER OF CHARGES IS ,1X,13)	DRT05810
	WRITE(IOOUT,840)	DRT05820
840	FORMAT(55X,22HDETONATION COORDINATES)	DRT05830
	DO 135 J=1,NCH	DRT05840
	WRITE(IOOUT,842)(COOR(I,J),I=1,2)	DRT05850
842	FORMAT(53X,2(3X,F8.2))	DRT05860
135	CONTINUE	DRT05870
	GO TO 300	DRT05880
C	CHARGE DISTRIBUTION TYPE 3	DRT05890
136	NCH=NCHS(1)	DRT05900
	DO 137 J=1,NCH	DRT05910
	READ(IOIN,702)(COOR(K,J),K=1,2),TSTAG(J)	DRT05920
702	FORMAT(8X,3F8.2)	DRT05930
137	CONTINUE	DRT05940
	WRITE(IOOUT,844)	DRT05950
844	FORMAT(/,30X,38HSEQUENTIAL IN TIME AND RANDOM IN SPACE, +24H DISTRIBUTION OF CHARGES)	DRT05960
	WRITE(IOOUT,838)NCH	DRT05970
	WRITE(IOOUT,846)	DRT05980
846	FORMAT(45X,25H DETONATION COORDINATES ,7X,10HBLAST TIME)	DRT05990
	DO 138 J=1,NCH	DRT06000
	WRITE(IOOUT,848)(COOR(I,J),I=1,2),TSTAG(J)	DRT06010
848	FORMAT(46X,F8.2,3X,F8.2,12X,F8.2)	DRT06020
138	CONTINUE	DRT06030
	GO TO 300	DRT06040
C	STORE AND PRINT OUT INFORMATION ABOUT VEHICLE	DRT06050
150	CONTINUE	DRT06060
	VH=.TRUE.	DRT06070
		DRT06080
		DRT06090
		DRT06100
		DRT06110
		DRT06120
		DRT06130
		DRT06140
		DRT06150
		DRT06160
		DRT06170
		DRT06180
		DRT06190

	V0(1)=RDIN(2)	DRT06200
	V0(2)=RDIN(3)	DRT06210
	VEHDIR=RDIN(4)	DRT06220
	VEHSPD=RDIN(5)	DRT06230
	VEHWID=RDIN(6)	DRT06240
	VEHWHT=RDIN(7)	DRT06250
	VEHTYP=IFIX(RDIN(8))	DRT06260
	WRITE(IOOUT,850)V0(1),V0(2)	DRT06270
850	FORMAT(//,44X,26HINITIAL VEHICLE POSITION (,F8.2,1H,,F8.2,1H))	DRT06280
	WRITE(IOOUT,852)VEHDIR	DRT06290
852	FORMAT(44X,19HVEHICLE DIRECTION ,F8.2,17H (CCW FROM EAST))	DRT06300
	WRITE(IOOUT,854)VEHSPD	DRT06310
854	FORMAT(50X,20HVEHICLE SPEED ,F8.2,4H M/S)	DRT06320
	WRITE(IOOUT,856)VEHWID	DRT06330
856	FORMAT(52X,20HVEHICLE WIDTH ,F8.2)	DRT06340
	WRITE(IOOUT,858)VEHWHT	DRT06350
858	FORMAT(52X,20HVEHICLE WEIGHT ,F8.2)	DRT06360
	IF(VEHTYP.EQ.0)WRITE(IOOUT,891)	DRT06370
891	FORMAT(58X,15HWHEELED VEHICLE)	DRT06380
	IF(VEHTYP.EQ.1)WRITE(IOOUT,892)	DRT06390
892	FORMAT(58X,15HTRACKED VEHICLE)	DRT06400
	GO TO 300	DRT06410
CCC	STORE TRANSMITTER COORDINATES AND TRANSMISSION TRHESHOLD	DRT06420
170	CONTINUE	DRT06430
	TC=.TRUE.	DRT06440
	NEWCOR=.TRUE.	DRT06450
	TRNCOR(1)=RDIN(2)	DRT06460
	TRNCOR(2)=RDIN(3)	DRT06470
	TRNCOR(3)=RDIN(4)	DRT06480
	TRNMIN=RDIN(5)	DRT06490
	IF(TRNMIN.LT.1.E-05)TRNMIN=1.E-05	DRT06500
	THRESH=-ALOG(TRNMIN)	DRT06510
	GO TO 300	DRT06520
CCC	STORE RECEIVER COORDINATES	DRT06530
190	CONTINUE	DRT06540
	RC=.TRUE.	DRT06550
	RECCOR(1)=RDIN(2)	DRT06560
	RECCOR(2)=RDIN(3)	DRT06570
	RECCOR(3)=RDIN(4)	DRT06580
	GO TO 300	DRT06590
CCC	STORE OBSERVER COORDINATES	DRT06600
210	CONTINUE	DRT06610
	OC=.TRUE.	DRT06620
	OBSCOR(1)=RDIN(2)	DRT06630
	OBSCOR(2)=RDIN(3)	DRT06640
	SPCHT=RDIN(4)	DRT06650
	GO TO 300	DRT06660
CCC	STORE TIME INTERVAL FOR CALCULATIONS	DRT06670
230	CONTINUE	DRT06680
	TM=.TRUE.	DRT06690
	TSTART=RDIN(2)	DRT06700
	TEND=RDIN(3)	DRT06710
	TINC=RDIN(4)	DRT06720
	IF (TINC.LE.0.0) TINC=1.	DRT06730
	IF(TEND.LT.TSTART)GO TO 903	DRT06740
	LIM=IFIX((TEND-TSTART)/TINC)+1	DRT06750
300	CONTINUE	DRT06760
310	CONTINUE	DRT06770
	IF(IGEOSW.NE.1) GO TO 333	DRT06780
	TRNCOR(1)=PTS(1)*1000.	DRT06790
	TRNCOR(2)=PTS(2)*1000.	DRT06800
	TRNCOR(3)=PTS(3)*1000.	DRT06810
		DRT06820
		DRT06830
		DRT06840
		DRT06850
		DRT06860
		DRT06870
		DRT06880

C	CHECK TO SEE IF MINIMUM INPUTS ARE AVAILABLE	DRT07600
C		DRT07610
C		DRT07620
	IF(.NOT.(M1.AND.M2.AND.SL.AND.CH.AND.EX.AND.(TC.AND.RC.OR.OC)).AND.	DRT07630
	1TM))GO TO 911	DRT07640
	CALL DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWWD,WNDMES,PHI,	DRT07650
1	THWD,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,	DRT07660
2	LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,	DRT07670
3	TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,	DRT07680
4	NERR,NCHS,SRCBAS,SIDE1,SIDE2,DHDT)	DRT07690
	NEWSRC=.FALSE.	DRT07700
	NEWATM=.FALSE.	DRT07710
330	IF(NERR.EQ.0)GO TO 335	DRT07720
	WRITE(IOOUT,857)NERR	DRT07730
857	FORMAT(55X,30H ***** DIRTRAN ERROR NUMBER ,12)	DRT07740
	GO TO 400	DRT07750
335	IF(ONCE.AND.(.NOT.NEWCOR))GO TO 340	DRT07760
	NEWCOR=.FALSE.	DRT07770
	IZINV=IFIX(ZINV)	DRT07780
	WRITE(IOOUT,859)IZINV	DRT07790
859	FORMAT(//,47X,30HESTIMATED INVERSION HEIGHT ,17)	DRT07800
		DRT07810
C	IF OBSERVER IS SPECIFIED, OUTPUT IS LABELED FOR EACH TIME.	DRT07820
C	IF ONLY TRANSMITTER AND RECEIVER ARE INPUT, OUTPUT IS TABULAR	DRT07830
		DRT07840
	IF(TC.AND.RC.AND.OC)GO TO 350	DRT07850
	IF(OC)GO TO 350	DRT07860
	IF(.NOT.(TC.AND.RC))GO TO 905	DRT07870
	WRITE(IOOUT,860)WAVE1	DRT07880
860	FORMAT(/,47X,18HWAVELENGTH ,F7.2,12H MICROMETERS)	DRT07890
	WRITE(IOOUT,862)(TRNCOR(I),I=1,3)	DRT07900
	WRITE(IOOUT,864)(RECCOR(I),I=1,3)	DRT07910
862	FORMAT(37X,28HTRANSMITTER COORDINATES ,3F10.2)	DRT07920
864	FORMAT(37X,28HRECEIVER COORDINATES ,3F10.2)	DRT07930
	WRITE(IOOUT,866)	DRT07940
866	FORMAT(52X,4HTIME,10X,13HTRANSMITTANCE)	DRT07950
340	CONTINUE	DRT07960
	WRITE(IOOUT,868)TIME,TRNLOS	DRT07970
868	FORMAT(52X,F8.2,10X,E10.5)	DRT07980
	ONCE=.TRUE.	DRT07990
	GO TO 400	DRT08000
350	WRITE(IOOUT,923)TIME	DRT08010
923	FORMAT(//,48X,28HTIME AFTER BLAST ,F7.2)	DRT08020
	IF(.NOT.(TC.AND.RC))GO TO 360	DRT08030
	WRITE(IOOUT,808)	DRT08040
	WRITE(IOOUT,860)WAVE1	DRT08050
	WRITE(IOOUT,862)(TRNCOR(I),I=1,3)	DRT08060
	WRITE(IOOUT,864)(RECCOR(I),I=1,3)	DRT08070
	WRITE(IOOUT,870)TRNLOS	DRT08080
870	FORMAT(42X,38HTRANSMITTANCE ALONG THE LINE OF SIGHT ,E10.3)	DRT08090
360	WRITE(IOOUT,808)	DRT08100
	WRITE(IOOUT,872)	DRT08110
872	FORMAT(57X,28HAERODYNAMIC CLOUD DIMENSIONS)	DRT08120
	WRITE(IOOUT,808)	DRT08130
	WRITE(IOOUT,874)(OBSCOR(I),I=1,2)	DRT08140
874	FORMAT(41X,28HOBSERVER COORDINATES ,2F10.2)	DRT08150
	WRITE(IOOUT,876)HEIGHT	DRT08160
876	FORMAT(39X,26HTHE HEIGHT OF THE CLOUD IS,10X,F10.2,7H METERS)	DRT08170
	WRITE(IOOUT,878)(CNTRD(I0),I0=1,2)	DRT08180
878	FORMAT(38X,28HTHE CENTROID COORDINATES ARE,8X,2F10.2)	DRT08190
	WRITE(IOOUT,880)CENWTH	DRT08200
880	FORMAT(38X,28HTHE WIDTH AT THE CENTROID IS,8X,F10.2,7H METERS)	DRT08210
	WRITE(IOOUT,882)SPCHT,SPCWTH	DRT08220
882	FORMAT(39X,12HTHE WIDTH AT,F8.2,11H METERS IS ,5X,F10.2,7H METERS)	DRT08230
	WRITE(IOOUT,884)NCPTS	DRT08240
884	FORMAT(46X,13,37H CONTOUR POINTS HAVE BEEN DETERMINED)	DRT08250
	WRITE(IOOUT,886)((CPTS(I0,IPT),I0=1,2),IPT=1,NCPTS)	DRT08260
886	FORMAT(60X,2(F10.3,2X))	DRT08270
400	CONTINUE	DRT08280
	GO TO 10	DRT08290

410 WRITE(1000,888)	DRT08300
888 FORMAT(/,38X,48H** THE CLOUD HAS BEEN PRECOMPUTED AND STORED ON,	DRT08310
1 5H FILE)	DRT08320
GO TO 10	DRT08330
900 WRITE(1000,901)	DRT08340
901 FORMAT(/,24X,48H*** DIRTRAN ERROR - MORE THAN 15 RECORDS OF DATA,	DRT08350
1 35H HAVE BEEN INPUT WITHOUT A GO CARD.)	DRT08360
IERR=1	DRT08370
GO TO 999	DRT08380
903 WRITE(1000,904)	DRT08390
904 FORMAT(/,39X,47H*** DIRTRAN ERROR - TIMES ARE NOT IN INCREASING,	DRT08400
+ 6H ORDER)	DRT08410
IERR=1	DRT08420
GO TO 999	DRT08430
905 WRITE(1000,906)	DRT08440
906 FORMAT(/,18X,46H*** DIRTRAN ERROR -NO TRANSMITTER AND RECEIVER,	DRT08450
+ 49H AND/OR OBSERVER COORDINATES HAVE BEEN SPECIFIED.)	DRT08460
IERR=1	DRT08470
GO TO 999	DRT08480
909 WRITE(1000,910)	DRT08490
910 FORMAT(/,25X,44H*** DIRTRAN ERROR - ONLY ONE DATA RECORD FOR,	DRT08500
1 38H CHARGE INFORMATION HAS BEEN SPECIFIED)	DRT08510
IERR=1	DRT08520
GO TO 999	DRT08530
911 WRITE(1000,912)	DRT08540
912 FORMAT(/,16X,49H*** DIRTRAN ERROR - MINIMUM AMOUNT OF INFORMATION,	DRT08550
1 26H REQUIRED IS NOT AVAILABLE./,10X,14H CHECK INPUTS)	DRT08560
IERR=1	DRT08570
GO TO 999	DRT08580
913 WRITE(1000,914)	DRT08590
914 FORMAT(/,20X,49H*** DIRTRAN ERROR - BOTH TRANSMITTER AND RECEIVER,	DRT08600
1 43H LOCATIONS MUST BE SPECIFIED, CHECK INPUTS)	DRT08610
IERR=1	DRT08620
GO TO 999	DRT08630
915 WRITE(1000,916)	DRT08640
916 FORMAT(/,30X,50H IOPT AND NARY DO NOT AGREE SEE THE ABOVE COMMENTS,	DRT08650
1 21H FOR CORRECT MATCHING)	DRT08660
IERR=1	DRT08670
999 RETURN	DRT08680
END	DRT08690


```

USTAR=UMES(1)/WDCAL(Z0,ZL,ZU(1))
TSTAR=TMES(1)*USTAR**2/1.568/ZL
IF(NATM-4)200,300,210
100 CONTINUE
C
C C C C
C USE ITERATIVE PROCEDURE TO CONVERGE ON BEST ATMOSPHERIC PROFILE
C TO MATCH DATA AT TWO HEIGHTS
DELTH=TMES(2)-TMES(1)+.0098*(ZT(2)-ZT(1))
NP=SIGN(1.,DELTH)
DELU=UMES(2)-UMES(1)
ZULOG=ALOG(ZU(2)/ZU(1))
ZTLOG=ALOG(ZT(2)/ZT(1))
USTAR=(UMES(2)-UMES(1))/ZULOG
TSTAR=DELTH/ZTLOG
ZL=.638*TMES(1)*USTAR**2/TSTAR
IF(ABS(ZL).GE.1000.)GO TO 300
DO 110 ITER=1,100
USTAR=DELU/(WDCAL(Z0,ZL,ZU(2))-WDCAL(Z0,ZL,ZU(1)))
TSTAR=DELTH/(TMPCAL(Z0,ZL,ZT(2))-TMPCAL(Z0,ZL,ZT(1)))
ZLP=ZL
ZL=.638*TMES(1)*USTAR**2/TSTAR
IF(ABS(ZL-ZLP)/ZLP).LT..01)GO TO 120
110 CONTINUE
ERR=.TRUE.
GO TO 999
120 CONTINUE
IF(ZL.GT.0.)Z0=1.E-04*ABS(ZL)
IF(ZL.LE.0.)Z0=1.E-03*ABS(ZL)
IF(NP)200,300,210
200 CONTINUE
C
C C C C
C UNSTABLE ATMOSPHERE
DXZ0=2.6
M=.079943
N=4./3.
DZ0=.7609*USTAR*ABS(ZL)**(1.-N)
U0=USTAR*14.2478/ABS(ZL)**M
GO TO 430
210 CONTINUE
C
C C C C
C STABLE ATMOSPHERE
DXZ0=3.3
N=.45644
M=.28414
DZ0=.059517*USTAR*ABS(ZL)**(1.-N)
U0=USTAR*36.6642/ABS(ZL)**M
GO TO 430
300 CONTINUE
C
C C C C
C NEUTRAL ATMOSPHERE
DZ0=.4*USTAR
DXZ0=2.8
NP=0
N=1.
M=1./7.
U0=45.92*USTAR/ABS(ZL)**M
GO TO 430
430 CONTINUE
C
C C C C
C COMMON CALCULATION TO UNSTABLE, NEUTRAL, AND STABLE ATMOSPHERES
DYX0=1.
IF(NATM.EQ.0)U0=(U0+UMES(2)/ZU(2)**M)/2.
C
C C C C
C ESTIMATE THE INVERSION HEIGHT AND COMPUTE THE NECESSARY
C PARAMETERS FOR THE WIND AND TEMPERATURE PROFILES BETWEEN
C ZSTAR AND ZINV WHEN DHDOT IS .FALSE..

```

```

ATMC1410
ATMC1420
ATMC1430
ATMC1440
ATMC1450
ATMC1460
ATMC1470
ATMC1480
ATMC1490
ATMC1500
ATMC1510
ATMC1520
ATMC1530
ATMC1540
ATMC1550
ATMC1560
ATMC1570
ATMC1580
ATMC1590
ATMC1600
ATMC1610
ATMC1620
ATMC1630
ATMC1640
ATMC1650
ATMC1660
ATMC1670
ATMC1680
ATMC1690
ATMC1700
ATMC1710
ATMC1720
ATMC1730
ATMC1740
ATMC1750
ATMC1760
ATMC1770
ATMC1780
ATMC1790
ATMC1800
ATMC1810
ATMC1820
ATMC1830
ATMC1840
ATMC1850
ATMC1860
ATMC1870
ATMC1880
ATMC1890
ATMC1900
ATMC1910
ATMC1920
ATMC1930
ATMC1940
ATMC1950
ATMC1960
ATMC1970
ATMC1980
ATMC1990
ATMC2000
ATMC2010
ATMC2020
ATMC2030
ATMC2040
ATMC2050
ATMC2060
ATMC2070
ATMC2080
ATMC2090
ATMC2100

```

```

C
APHI=PHI*PIRAD
FREQ=2.*OMEGA*SIN(APHI)
HC=K*USTAR/FREQ
IF(ZL.GT.0.0.AND.ZL.LT.1.E3)GO TO 500
ZINV=HC
GO TO 501
500 ZINV=.26*HC
501 ZSTAR=.13*ZINV
KM=DIFFUS(Z0,ZL,ZSTAR)
SP=USTAR*WNCAL(Z0,ZL,ZSTAR)
ALP=SQRT(FREQ/(2.*KM))
IF(DHOT)GO TO 813
ARG=ALP*ZSTAR
ARG1=BETA+ALP*ZINV
ARG2=ALP*(ZINV-ZSTAR)
C=SP*EXP(ARG)*SIN(ARG1)/SIN(ARG2)
PYF=C*EXP(-ARG)*COS(ARG)-SP*COS(BETA)
PXF=C*EXP(-ARG)*SIN(ARG)+SP*SIN(BETA)
UE=C*EXP(-ARG)*COS(ARG)-PYF
VE=-C*EXP(-ARG)*SIN(ARG)+PXF
UHAT=UE/SQRT(UE*UE+VE*VE)
VHAT=VE/SQRT(UE*UE+VE*VE)
CALL TEMP(ZSTAR,TA,DTADZ)
DTADH=0.0
TC3=(DTADH-DTADZ)/(2.*(ZINV-ZSTAR))
TC2=DTADZ-2.*TC3*ZSTAR
TC1=TA-TC2*ZSTAR-TC3*ZSTAR**2
GO TO 999
813 ZSTAR=1.E4
999 RETURN
END

```

```

ATMC2110
ATMC2120
ATMC2130
ATMC2140
ATMC2150
ATMC2160
ATMC2170
ATMC2180
ATMC2190
ATMC2200
ATMC2210
ATMC2220
ATMC2230
ATMC2240
ATMC2250
ATMC2260
ATMC2270
ATMC2280
ATMC2290
ATMC2300
ATMC2310
ATMC2320
ATMC2330
ATMC2340
ATMC2350
ATMC2360
ATMC2370
ATMC2380
ATMC2390
ATMC2400
ATMC2410
ATMC2420

```



```

SUBROUTINE AVRG(ZX, TIME, QTOT, XBAVG, SIG2X, SIG2Y)
ROUTINE FOR FINDING AVERAGES OF THE MOMENTS OF THE DISCS
INPUTS
  ZX - HEIGHT AT WHICH THE AVERAGES ARE DESIRED
  TIME - PRESENT TIME
  QTOT - SUM OF T00*QDSCS
  XBAVG- AVERAGE OF THE FIRST ORDER MOMENTS (ALONGWIND DISPLACEMENT
         IN THE WIND DIRECTION)
  SIG2X- AVERAGE OF THE SQUARE OF ONE OF THE SECOND ORDER MOMENTS
         (ALONGWIND SPREAD)
  SIG2Y- AVERAGE OF THE SQUARE OF ANOTHER SECOND ORDER MOMENT
         (CROSSWIND SPREAD)
*****
COMMON/DISCS/NDSCS, TDSC(20), XDSC(20), ZDSC(20), R2DSC(20), QDSC(20,3)
COMMON/PRTINF/R0, VGRAY(3), NPRTS
QTOT=0.0
QSIG2X=0.0
QSIG2Y=0.0
QXBAR=0.0
Z=ZX
DO 10 I=1, NDSCS
H=ZDSC(I)
ROH2=R2DSC(I)
TOF=TIME-TDSC(I)
CALL MOMENT(VGRAY(1), Z, H, TOF, Q, XBAR, SIGW2, SIGP2)
QT=QDSC(I,1)*Q
QTOT=QTOT+QT
QSIG2X=QSIG2X+(SIGW2+ROH2/2.)*QT
QSIG2Y=QSIG2Y+(SIGP2+ROH2/2.)*QT
QXBAR=QXBAR+(XBAR+XDSC(I))*QT
10 CONTINUE
XBAVG=QXBAR/QTOT
SIG2X=QSIG2X/QTOT
SIG2Y=QSIG2Y/QTOT
999 RETURN
END

```

```

AVRG0230
AVRG0010
AVRG0020
AVRG0030
AVRG0040
AVRG0050
AVRG0060
AVRG0070
AVRG0080
AVRG0090
AVRG0100
AVRG0110
AVRG0120
AVRG0130
AVRG0140
AVRG0150
AVRG0160
AVRG0170
AVRG0180
AVRG0190
AVRG0200
AVRG0210
AVRG0220
AVRG0240
AVRG0250
AVRG0260
AVRG0270
AVRG0280
AVRG0290
AVRG0300
AVRG0310
AVRG0320
AVRG0330
AVRG0340
AVRG0350
AVRG0360
AVRG0370
AVRG0380
AVRG0390
AVRG0400
AVRG0410
AVRG0420
AVRG0430
AVRG0440
AVRG0450
AVRG0460

```

```

SUBROUTINE CLIMB(FUNCT,GFUN,P1,FP1,NSERCH,NOCONT)
THIS MODULE IS A SUBROUTINE THAT FINDS A POINT ON A CONTOUR
BY FINDING THE GRADIENT VECTOR AT THAT POINT AND MARCHING ALONG
IT UNTIL IT FINDS ITSELF IN A REGION GREATER THAN THE CONTOUR LEVEL.
AT WHICH POINT IT MARCHES HORIZONTALLY, HALVING THE STEP SIZE
UNTIL THE CONTOUR IS REACHED WITHIN SPECIFIED RESOLUTION.
IN ADDITION IT WILL DETERMINE IF A CONTOUR EXISTS.
ARGUMENTS PASSED.
INPUT
  FUNCT-THE FUNCTION(X,Y) ALSO GIVEN IN EXTERNAL.
  P1-THE STARTING POINT.
OUTPUT
  P1  - THE POINT ON THE CONTOUR OR THE POINT AT WHICH
        THE FUNCTION REACHES A MAXIMUM BELOW THE CONTOUR
        LEVEL.
  FP1 - THE VALUE OF THE FUNCTION AT P
  NOCONT-THE ERROR FLAG.
        F-NO PROBLEM
        T-NO CONTOUR FOUND.
  ERR-ERROR FLAG RETURNED BY 'NTRST'
        F-NO ERROR
        T-ITERATION DIVERGED OR MAXIMUM SEARCH AREA EXCEEDED
IN ADDITION,IN COMMON ARE...
  YMIN-THE LOWER LIMIT ON Y.
  DELTA-THE STEP SIZE,MODIFIED IN THIS SUBROUTINE.
  CON-THE CONTOUR LEVEL.
  RES-THE RESOLUTION LENGTH
OTHER VARIABLES INCLUDE
  GRAD-THE GRADIENT VECTOR
  P0-THE CURRENT POINT ON THE GRADIENT.
  P1-THE POINT ON THE GRADIENT BEING TESTED
     TO SEE ABOUT CONTOUR EXISTENCE.
  FP0,FP1-THE FUNCTION VALUES OF P0 AND P1.
CALLED SUBROUTINES
  GRAD2-FINDS THE GRADIENT VECTOR OF A FUNCTION AT
        A POINT AND THE SLOPE THERE.
  UNIT-CALCULATES THE NORM AND MAGNITUDE OF A 2 VECTOR.
  VSUM-VECTOR SUM OF THE FORM C=A+SB WHERE S IS SCALAR
        MULTIPLIER OF B.
EXTERNAL FUNCT
LOGICAL NOCONT
DIMENSION GRAD(2),P0(2),P1(2)
COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETA,CON
COMMON/LIMIT/YMIN,FMIN
NOCONT=.FALSE.
ONEM=-1.0
IF (NSERCH.EQ.0)GO TO 7
DELTA=SIGN(DELTA,FLOAT(NSERCH))
FP1=FUNCT(P1(1),P1(2))
IF(FP1.LT.CON)GO TO 25
GO TO 22
3 CONTINUE
P0(1)=P1(1)
P0(2)=P1(2)
FP0=FP1
C ** FINDING THE UNIT GRADIENT AND THE NEXT POINT ALONG IT.
4 CALL GRAD2(P0,FUNCT,RES,GRAD,DFDS)

```

```

CLI00520
CLI00010
CLI00020
CLI00030
CLI00040
CLI00050
CLI00060
CLI00070
CLI00080
CLI00090
CLI00100
CLI00110
CLI00120
CLI00130
CLI00140
CLI00150
CLI00160
CLI00170
CLI00180
CLI00190
CLI00200
CLI00210
CLI00220
CLI00230
CLI00240
CLI00250
CLI00260
CLI00270
CLI00280
CLI00290
CLI00300
CLI00310
CLI00320
CLI00330
CLI00340
CLI00350
CLI00360
CLI00370
CLI00380
CLI00390
CLI00400
CLI00410
CLI00420
CLI00430
CLI00440
CLI00450
CLI00460
CLI00470
CLI00480
CLI00490
CLI00500
CLI00510
CLI00530
CLI00540
CLI00550
CLI00560
CLI00570
CLI00580
CLI00590
CLI00600
CLI00610
CLI00620
CLI00630
CLI00640
CLI00650
CLI00660
CLI00670
CLI00680
CLI00690
CLI00700

```

5	CALL VSUM(P0,GRAD,DELTA,P1)	CL100710
C **	IS THE POINT HEADING BELOW YMIN **	CL100720
	IF(P1<2).GE.YMIN)GO TO 7	CL100730
	P1<2)=YMIN	CL100740
	CALL VSUM(P1,P0,ONEM,GRAD)	CL100750
	CALL UNIT<GRAD,GRAD,DELTA)	CL100760
	IF<ABS<DELTA>,LT,RES)GO TO 25	CL100770
7	FP1=FUNCT<P1<1>,P1<2>)	CL100780
C **	HAS THE CONTOUR BEEN CROSSED **	CL100790
8	IF<FP1,GE,CON)GO TO 22	CL100800
	IF<FP1,GT,FP0)GO TO 3	CL100810
	DELTA=DELTA/2.	CL100820
	IF<ABS<DELTA>,LT,RES)GO TO 25	CL100830
	GO TO 5	CL100840
25	NOCONT=.TRUE.	CL100850
	GO TO 99	CL100860
22	CONTINUE	CL100870
C	BEGIN HORIZONTAL SEARCH	CL100880
	P0<2)=P1<2)	CL100890
31	P0<1)=P1<1)	CL100900
	FP0=FP1	CL100910
40	P1<1)=P0<1)+DELTA	CL100920
	FP1=FUNCT<P1<1>,P1<2>)	CL100930
	IF<ABS<DELTA>,LT,RES/2.)GO TO 99	CL100940
	IF<FP1,GE,CON)GO TO 31	CL100950
	DELTA=DELTA/2.	CL100960
	GO TO 40	CL100970
99	CONTINUE	CL100980
	RETURN	CL100990
	END	CL101000

SUBROUTINE CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS5,
1 ERR)

PURPOSE

CLDIM CALCULATES FIVE CONTOUR POINTS AND CLOUD DIMENSIONS AS SEEN FROM THE SPECIFIED OBSERVER POSITION. CLDIM REQUIRES CLOUD PARAMETERS FROM THE BUOYANT RISE STAGE OF CLOUD DEVELOPMENT WHICH ARE SUPPLIED IN COMMON STORAGE /BUOYCL/ AND /PRTINF/ AS WELL AS VIEWING GEOMETRY WHICH IS SUPPLIED IN COMMON /GEOM/. SPCHT IS REQUIRED INPUT IN THE ARGUMENTS. ALL OUTPUTS ARE ARGUMENTS.

INPUT

SPCHT THE SPECIFIED HEIGHT AT WHICH THE WIDTH OF THE CLOUD IS DESIRED. (METERS)

OUTPUT

CNTRD A SINGLY DIMENSIONED ARRAY OF LENGTH 2 WHICH CONTAINS THE HORIZONTAL AND VERTICAL COORDINATES, RESP., OF THE CLOUD CENTROID. (METERS)
HEIGHT THE HEIGHT OF THE CLOUD IN METERS
CENWTH THE WIDTH OF THE CLOUD AT THE CENTROID HEIGHT IN METERS
SPCWTH THE WIDTH OF THE CLOUD AT THE SPECIFIED HEIGHT (METERS)
NCPTS THE NUMBER OF CONTOUR POINTS (=6)
CPTS A DOUBLY DIMENSIONED ARRAY OF SIZE (2,N),N.GE.5, WHICH CONTAINS THE HORIZONTAL AND VERTICAL COORDINATES OF THE FIVE CONTOUR POINTS. (METERS)

REQUIRED SUBROUTINES

CLIMB DETERMINES IF THE CONTOUR EXISTS, IF SO FINDS A POINT ON THE CONTOUR.

CALLED BY DUSTCL

DIMENSION CNTRD(2),CPTS5(2,6),TOP(2)
LOGICAL HORIZ,NOCONT,SWITCH,CHANGE,ERR
REAL KZ,KX
COMMON /BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM
COMMON /PRTINF/R0,VGRAY(3),NPRTS
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTP1,SCRN(2)
COMMON /MODE/ HORIZ
COMMON /CLOCK/ T,TWIND
COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF
COMMON /WINDFRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
COMMON /TRAN/YTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON /SIG/SIG02,SIGC
COMMON /IQUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
1 QDSC(20,3)
COMMON /SPECS/ RES,STEP,TANT,CON
COMMON /CONST/PI,P12,PIRAD,TWOPI,TORRMB,CDEGK
EXTERNAL FUNCT,GFUN
DATA RES,TANT /.4, .1/
HORIZ=.TRUE.
ERR=.FALSE.
CON=ALOG(VISEXT)
CPTS5(2,1)=SPCHT
CPTS5(2,6)=SPCHT
U=U0*CPTS5(2,1)**UM

CLD00010
CLD00020
CLD00030
CLD00040
CLD00050
CLD00060
CLD00070
CLD00080
CLD00090
CLD00100
CLD00110
CLD00120
CLD00130
CLD00140
CLD00150
CLD00160
CLD00170
CLD00180
CLD00190
CLD00200
CLD00210
CLD00220
CLD00230
CLD00240
CLD00250
CLD00260
CLD00270
CLD00280
CLD00290
CLD00300
CLD00310
CLD00320
CLD00330
CLD00340
CLD00350
CLD00360
CLD00370
CLD00380
CLD00390
CLD00400
CLD00410
CLD00420
CLD00430
CLD00440
CLD00450
CLD00460
CLD00470
CLD00480
CLD00490
CLD00500
CLD00510
CLD00520
CLD00530
CLD00540
CLD00550
CLD00560
CLD00570
CLD00580
CLD00590
CLD00600
CLD00610
CLD00620
CLD00630
CLD00640
CLD00650
CLD00660
CLD00670
CLD00680
CLD00690
CLD00700

	CPTSS(1,1)=T*U*SINTH	CLD00710
	CPTSS(1,6)=CPTSS(1,1)	CLD00720
	NSERCH=-1	CLD00730
	STEP=20.	CLD00740
C	CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN ONE DIRECTION AT	CLD00750
C	THE HEIGHT OF THE OBSERVER.	CLD00760
	CALL CLIMB(FUNCT,GFUN,CPTSS,FP1,NSERCH,NOCONT)	CLD00770
	NSERCH=1	CLD00780
	STEP=20.	CLD00790
C	CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN THE OPPOSITE	CLD00800
C	DIRECTION AT THE HEIGHT OF THE OBSERVER.	CLD00810
	CALL CLIMB(FUNCT,GFUN,CPTSS(1,6),FP1,NSERCH,NOCONT)	CLD00820
	SPCWTB=CPTSS(1,6)-CPTSS(1,1)	CLD00830
	NCPTS=6	CLD00840
	IF(T.LE.TWIND)GO TO 50	CLD00850
	CNTRD(1)=(XTR+VTR*(T-TTR))*SINTH+CENDIF	CLD00860
	CNTRD(2)=ZTR	CLD00870
	SIGX2=SIG02+2.*KX*(T-TTR)	CLD00880
	SIGZ2=SIG02+2.*KZ*(T-TTR)	CLD00890
	SIGX=SQRT(SIGX2)	CLD00900
	SIGZ=SQRT(SIGZ2)	CLD00910
	BOT=1./(<2.*VISEXT)	CLD00920
	ARG=BOT*QPUFF(1)/PI/SIGX/SIGZ	CLD00930
	IF(ARG.LT.1.0)GO TO 998	CLD00940
	TOP(1)=CNTRD(1)	CLD00950
	TOP(2)=ZTR+SIGZ*SQRT(2.*ALOG(ARG))	CLD00960
	RAD=SIGX*SQRT(2.*ALOG(ARG))	CLD00970
	CENWTH=2.*(<RAD+PRJARY)	CLD00980
	HEIGHT=TOP(2)	CLD00990
	GO TO 100	CLD01000
50	CNTRD(1)=XCM*SCRN(1)+YCM*SCRN(2)+CENDIF	CLD01010
	CNTRD(2)=ZCM	CLD01020
	TOP(1)=XTOP*SCRN(1)+YTOP*SCRN(2)+CENDIF	CLD01030
	TOP(2)=ZCM+RSPH	CLD01040
	IF(TOP(2).GT.ZINV)TOP(2)=ZINV	CLD01050
	HEIGHT=TOP(2)	CLD01060
	CENWTH=2.*(<RSPH+PRJARY)	CLD01070
100	CPTSS(1,2)=CNTRD(1)-CENWTH/2.	CLD01080
	CPTSS(2,2)=CNTRD(2)	CLD01090
	CPTSS(1,3)=TOP(1)-PRJARY	CLD01100
	CPTSS(2,3)=TOP(2)	CLD01110
	CPTSS(1,4)=TOP(1)+PRJARY	CLD01120
	CPTSS(2,4)=TOP(2)	CLD01130
	CPTSS(1,5)=CNTRD(1)+CENWTH/2.	CLD01140
	CPTSS(2,5)=CNTRD(2)	CLD01150
	NCPTS=6	CLD01160
	GO TO 999	CLD01170
998	WRITE(1000,1000)	CLD01180
1000	FORMAT(50H *** UPPER PART OF CLOUD HAS DISSIPATED ***)	CLD01190
999	RETURN	CLD01200
	END	CLD01210
		CLD01220
		CLD01230
		CLD01240
		CLD01250

```

SUBROUTINE COMPC1<NEWATM,NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,
1          PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,
2          NWL,TRNCOR,RECCOR,TIME,DHDT,TRNLOS,NERR>
CONTROLING ROUTINE FOR PRECOMPUTING A SINGLE , EXPLOSION PRODUCED
DUST CLOUD AND STORING IT ON AN EXTERNAL FILE TO BE USED AT A
LATER RUNNING OF THE CODE FOR A RANDOM DISTRIBUTION IN SPACE AND TIME
OF CHARGES
INPUTS
FOR DETAILS SEE DRTRAN
OUTPUTS
TRNLOS -CALCULATE TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
*****
DIMENSION ZTEMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3),
1 RECCOR(3),TRNFRM(2,2),SIDE1(2),SIDE2(2),NCHS(2),ORIG(2),TRN(3),
2 REC(3),SRCBAS(2),PAS(6)
LOGICAL DHDT,NEWATM,NEWSRC,ERR,FLAG
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTP1,SCRN(2)
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON/VL/VLOAD
COMMON/OPTION/IOPT,IFILE
COMMON/CLOCK/FTIME,TWIND
COMMON/BUOYCL/Y(8),SPHNS(3),RTIM
COMMON/DISC/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ COOR(2,200),TSTAG(200),
+ ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
1 RB(3,25),Z2(3,25)
COMMON/WNDPRM/DXZ0,DYZ0,DZ0,U0,UM,ON,ZINV
COMMON/CARB/RCARB1,RCARB2
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DATA FLAG/.TRUE./
DATA PAS/4HA ,4HB ,4HC ,4HD ,4HE ,4HF /
IF<IOPT.EQ.4>GO TO 500
PRECOMPUTE INFORMATION AND STORE ON FORTRAN UNIT IFILE
THETA=THWND*PI/180.
TWIND=1.E5
TTR=1.E5
IF<.NOT.NEWATM>GO TO 10
CALL ATMCAL<NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,PHI,THETA,DHDT,ERR>
IF<.NOT.ERR>GO TO 10
NERR=7
GO TO 999
10 CONTINUE
IF<.NOT.NEWSRC>GO TO 20
CALL SOURCE<CHWT,NCHRG,DETDEP,NSOIL,DSOD>
20 CONTINUE
COMPUTE INITIAL LOADING
SUM=0.0
DO 25 I=1,NDSCS
SUM=SUM+QDSC(I,1)
25 CONTINUE
VLOAD=SPHNS(1)+SUM
CALL PRECL TO COMPUTE AND STORE THE QUADRATIC FITS NECESSARY
FOR THE CONE
CALL PRECL<NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,PHI,DHDT,
1 CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT>
GO TO 999
500 CONTINUE

```

```

COM00160
COM00170
COM00180
COM00010
COM00020
COM00030
COM00040
COM00050
COM00060
COM00070
COM00080
COM00090
COM00100
COM00110
COM00120
COM00130
COM00140
COM00150
COM00190
COM00200
COM00210
COM00220
COM00230
COM00240
COM00250
COM00260
COM00270
COM00280
COM00290
COM00300
COM00310
COM00320
COM00330
COM00340
COM00350
COM00360
COM00370
COM00380
COM00390
COM00400
COM00410
COM00420
COM00430
COM00440
COM00450
COM00460
COM00470
COM00480
COM00490
COM00500
COM00510
COM00520
COM00530
COM00540
COM00550
COM00560
COM00570
COM00580
COM00590
COM00600
COM00610
COM00620
COM00630
COM00640
COM00650
COM00660
COM00670
COM00680
COM00690

```

```

IF(.NOT.FLAG) GO TO 35
READ(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)
READ(IFILE)DHDT,PHI,CHWT,NCHRG,DETEP,NSOIL,DSOD,SILT,ZINV
READ(IFILE)YLOAD,RCARB1,RCARB2
READ(IFILE)ICOUNT
DO 30 J=1,ICOUNT
READ(IFILE) TIMES(J),(RT(I,J),RB(I,J),Z2(I,J),XC(I,J),
1 XC(I,J),I=1,3)
30 CONTINUE
WRITE(IOOUT,800)
800 FORMAT(/,5X,85H ATMOSPHERIC, CHARGE, AND SOIL CHARACTERISTICS USED
1 WHEN THE CLOUD WAS PRECOMPUTED.)
WRITE OUT ATMOSPHERIC INFORMATION
WRITE(IOOUT,808)
808 FORMAT(1X)
WRITE(IOOUT,810)PAS(NATMOS)
810 FORMAT(30H PASQUILL CATEGORY ,A4)
NIO=1
WRITE(IOOUT,812)(ZTEMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,NIO)
812 FORMAT(8H HT ,F8.2,7H TEMP ,F8.2,7H HT,F8.2,7H WIND ,
1 F8.2)
WRITE(IOOUT,814)THWND
814 FORMAT(22H WIND DIRECTION ,F8.2)
WRITE(IOOUT,819)PHI
IF(DHDT)WRITE(IOOUT,816)
816 FORMAT(40H THE INVERSION LAYER HEIGHT IS GROWING )
IF(.NOT.DHDT)WRITE(IOOUT,818)
818 FORMAT(42H THE INVERSION LAYER HEIGHT IS CONSTANT )
819 FORMAT(/,22H LATITUDE ,F8.2)
WRITE SOIL CHARACTERISTICS
IF(NSOIL.EQ.1)WRITE(IOOUT,821)
820 FORMAT(15H SILT CONTENT ,F5.2)
821 FORMAT(/,15H SOIL-1 )
IF(NSOIL.EQ.2)WRITE(IOOUT,822)
822 FORMAT(/,15H SOIL-2 )
WRITE(IOOUT,820)SILT
WRITE(IOOUT,823)DSOD
823 FORMAT(21H DEPTH OF SOD ,F5.2)
WRITE EXPLOSIVE CHARGE CHARACTERISTICS
IF(NCHRG.EQ.1)WRITE(IOOUT,824)
824 FORMAT(/,65H SURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC, TIP
1 ON GROUND )
IF(NCHRG.EQ.2)WRITE(IOOUT,825)
825 FORMAT(/,25H BARE CHARGE ON SURFACE )
IF(NCHRG.EQ.3)WRITE(IOOUT,826)
826 FORMAT(/,45H 30 DEGREE TILTED TIP AT 0.3 METER DEPTH )
IF(NCHRG.EQ.4)WRITE(IOOUT,827)
827 FORMAT(/,45H 30 DEGREE TILTED TIP AT 0.6 METER DEPTH )
IF(NCHRG.EQ.5)WRITE(IOOUT,828)
828 FORMAT(/,40H HORIZONTAL PROJECTILE ON SURFACE )
WRITE(IOOUT,829)CHWT
829 FORMAT(30H WEIGHT OF CHARGE ,F8.2,5H KG. )
WRITE(IOOUT,830)DETEP
830 FORMAT(30H DETONATION DEPTH ,F8.2)
FLAG=.FALSE.
COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER
DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN
THE WIND DIRECTION.
35 CONTINUE
IF(.NOT.NEWATM) GO TO 45
THETA=THWND*PI/180.
TRNFRM(1,1)=COS(THETA)

```

```

COM00700
COM00710
COM00720
COM00730
COM00740
COM00750
COM00760
COM00770
COM00780
COM00790
COM00800
COM00810
COM00820
COM00830
COM00840
COM00850
COM00860
COM00870
COM00880
COM00890
COM00900
COM00910
COM00920
COM00930
COM00940
COM00950
COM00960
COM00970
COM00980
COM00990
COM01000
COM01010
COM01020
COM01030
COM01040
COM01050
COM01060
COM01070
COM01080
COM01090
COM01100
COM01110
COM01120
COM01130
COM01140
COM01150
COM01160
COM01170
COM01180
COM01190
COM01200
COM01210
COM01220
COM01230
COM01240
COM01250
COM01260
COM01270
COM01280
COM01290
COM01300
COM01310
COM01320
COM01330
COM01340
COM01350
COM01360
COM01370

```

	TRNFRM(2,2)=TRNFRM(1,1)	COM01380
	TRNFRM(1,2)=SIN(THETA)	COM01390
	TRNFRM(2,1)=-TRNFRM(1,2)	COM01400
CCCC	MAKE THE ORIGIN OF THE LOCAL COORDINAT SYSTEM THE FIRST	COM01410
	CHARGE LOCATION THAT WAS INPUT BY THE USER	COM01420
	DO 40 I=1,2	COM01430
	SRCBAS(I)=COOR(I,1)	COM01440
	ORIG(I)=SRCBAS(I)	COM01450
40	CONTINUE	COM01460
CCCC	CALL SETUP TO COMPUTE THE ARRAY OF DIFFERENCE VECTORS	COM01470
	CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)	COM01480
CCCC	COMPUTE COORDINATES OF THE TRANSMITTER AND RECEIVER IN THE LOCAL	COM01490
	COORDINATE SYSTEM	COM01500
	TRN(3)=TRNCOR(3)	COM01510
	REC(3)=RECCOR(3)	COM01520
	DO 60 I=1,2	COM01530
	TRN(I)=0.0	COM01540
	REC(I)=0.0	COM01550
	DO 50 J=1,2	COM01560
	TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))	COM01570
	REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))	COM01580
50	CONTINUE	COM01590
60	CONTINUE	COM01600
CCCC4	CONTINUE	COM01610
	CALL PRETRN TO COMPUTE THE TRANSMITTANCE ALONG THE SPECIFIED LINE OF	COM01620
	SIGHT	COM01630
999	CALL PRETRN(TRN,REC,TIME,TRNLOS)	COM01640
	RETURN	COM01650
	END	COM01660
		COM01670
		COM01680
		COM01690
		COM01700
		COM01710
		COM01720
		COM01730


```

SUBROUTINE CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,
1XNORM,PLEN)
ROUTINE TO FIND THE LENGTH OF A NON-HORIZONTAL LINE THAT INTERSECTS
A CONE
INPUTS
U - UNIT VECTOR ALONG LINE CONNECTING THE TRANSMITTER AND
RECEIVER.
TR - TRANSMITTER COORDINATES
HTTOP- HEIGHT OF THE TOP OF THE CONE SHAPED PORTION OF THE CLOUD
HTBOT- HEIGHT OF THE BOTTOM OF THE CONE SHAPED PORTION OF THE
CLOUD
XCEN - X POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP
YCEN - Y POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP
RTOP - RADIUS OF THE CONE AT THE TOP
RBOT - RADIUS OF THE CONE AT THE BOTTOM
XB - X POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD
YB - Y POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD
OUTPUT
PLEN - LENGTH OF THE INTERSECTION OF CONE AND THE LINE OF SIGHT
FUNCTIONS AND SUBROUTINES
NONE
*****
DIMENSION U(3),TR(3)
IF(U(3).LT.0.0)GO TO 40
SET UP BOUNDS SO INTERSECTION OF LINE IS SUCH THAT HTBOT < Z < HTTOP
PMIN=(HTBOT-TR(3))/U(3)
PMAX=(HTTOP-TR(3))/U(3)
GO TO 50
40 PMIN=(HTTOP-TR(3))/U(3)
PMAX=(HTBOT-TR(3))/U(3)
50 P1=(HTTOP-HTBOT)/U(3)
P0=(HTBOT-TR(3))/U(3)
SET UP QUADRATIC TO BE SOLVED
DX1=U(1)*P1-XCEN+XB
DY1=U(2)*P1-YCEN+YB
DR=RTOP-RBOT
A=DX1**2+DY1**2-DR**2
DX0=TR(1)+U(1)*P0-XB
DY0=TR(2)+U(2)*P0-YB
B=2.*(DX1*DX0+DY1*DY0-DR*RBOT)
C=(DX0**2+DY0**2-RBOT**2)
RADIC1=(DX1*RBOT-DX0*DR)**2
RADIC2=(DY1*RBOT-DY0*DR)**2
RADIC3=(DX1*DY0-DY1*DX0)**2
DETERMINE PATH LENGTH IF THE LINE INTERSECTS THE CONE
IF(ABS(A).LT.1.E-20)GO TO 60
RADIC=RADIC1+RADIC2-RADIC3

```

```

CNL00380
CNL00390
CNL00010
CNL00020
CNL00030
CNL00040
CNL00050
CNL00060
CNL00070
CNL00080
CNL00090
CNL00100
CNL00110
CNL00120
CNL00130
CNL00140
CNL00150
CNL00160
CNL00170
CNL00180
CNL00190
CNL00200
CNL00210
CNL00220
CNL00230
CNL00240
CNL00250
CNL00260
CNL00270
CNL00280
CNL00290
CNL00300
CNL00310
CNL00320
CNL00330
CNL00340
CNL00350
CNL00360
CNL00370
CNL00400
CNL00410
CNL00420
CNL00430
CNL00440
CNL00450
CNL00460
CNL00470
CNL00480
CNL00490
CNL00500
CNL00510
CNL00520
CNL00530
CNL00540
CNL00550
CNL00560
CNL00570
CNL00580
CNL00590
CNL00600
CNL00610
CNL00620
CNL00630
CNL00640
CNL00650
CNL00660
CNL00670
CNL00680
CNL00690
CNL00700

```

	IF(RADIC.LT.0.0)GO TO 68	CNL00710
	ROOT1=-B/A/2.+SQRT(RADIC)/A	CNL00720
	ROOT2=-B/A/2.-SQRT(RADIC)/A	CNL00730
C	DETERMINE WHICH POINT IS CLOSEST AND WHICH IS FARTHEST FROM THE	CNL00740
C	TRANSMITTER	CNL00750
	TEMP1=P1*ROOT1+P0	CNL00760
	TEMP2=P1*ROOT2+P0	CNL00770
	IF (A.LT.0.0) GO TO 57	CNL00780
	IF(TEMP1.GT.TEMP2)GO TO 55	CNL00790
	PFAR=TEMP2	CNL00800
	PNEAR=TEMP1	CNL00810
	GO TO 67	CNL00820
55	PFAR=TEMP1	CNL00830
	PNEAR=TEMP2	CNL00840
	GO TO 67	CNL00850
57	IF (U<3).GT.0.0) GO TO 58	CNL00860
	PNEAR=-1.E20	CNL00870
	PFAR=AMIN1(TEMP1,TEMP2)	CNL00880
	GO TO 67	CNL00890
58	PNEAR=AMAX1(TEMP1,TEMP2)	CNL00900
	PFAR=1.E20	CNL00910
	GO TO 67	CNL00920
C	CONSIDER DEGENERATE CASE	CNL00930
C		CNL00940
60	IF(B.LT.1.E-20)GO TO 68	CNL00950
	IF(P1/B.LT.0.0)GO TO 65	CNL00960
	PFAR=P1*(-C/B)+P0	CNL00970
	PNEAR=0.0	CNL00980
	GO TO 67	CNL00990
65	PFAR=XNORM	CNL01000
	PNEAR=P1*(-C/B)+P0	CNL01010
67	PLEN=AMIN1(PFAR,XNORM,PMAX)-AMAX1(PNEAR,0.0,PMIN)	CNL01020
	IF(PLEN.LT.0.0)PLEN=0.0	CNL01030
	GO TO 999	CNL01040
68	PLEN=0.0	CNL01050
999	RETURN	CNL01060
	END	CNL01070
		CNL01080
		CNL01090
		CNL01100

```

SUBROUTINE CONVRT(T)
LOGICAL SWITCH, CHANGE
REAL KZ, KX, N, MDIF
COMMON/PRTINF/R0, VGRAV(3), NPRTS
COMMON/BUOYCL/RSPH, DELT, VZSPH, XCMSPH, YCMSPH, ZCMSPH, XTOP, YTOP,
1 SPHNS(3), RISTIM
COMMON/CLOCK/TIME, TWIND
COMMON/TRAN/VTR, KZ, KX, TTR, XTR, ZTR, QPUFF(3), SWITCH, CHANGE
COMMON/SIG/SIG02, SIGC
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUC
COMMON/GEOM/COSTH2, SINTH, SINTH2, VISEXT, RTP1, SCRK(2)
COMMON/LOAD/WAKAL, SPHAL
COMMON/WNDPRM/DXZ0, DYZ0, DZ0, U0, M, NDIF, ZINW
COMMON/STARS/USTAR, TSTAR, ZSTAR
COMMON/EKTEMP/Z0, ZL, T0, TC1, TC2, TC3
COMMON/WAKE/XDIF, YDIF, ZDIF, TDIF, TDZ, QLOC, QCOL, XBAVRG
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
*****
C *****
C PURPOSE
C TO CONVERT THE CURRENT BUOYANT CLOUD INTO A THREE DIMENSIONAL
C GAUSSIAN PUFF TO BE USED BY THE WIND DISPERSION MODEL.
C INPUT
C T TIME IN SECONDS AFTER DETONATION
C CALLED BY RISE
C FUNCTIONS AND SUBROUTINES NEEDED
C WNDCAL
C AMOUNT
C AVRG
C *****
C CHANGE=.TRUE.
C TWIND=T
C VTR=USTAR*WNDCAL(Z0, ZL, ZCMSPH)
C KZ=DIFFUS(Z0, ZL, ZCMSPH)
C KX=KZ*DXZ0
C TTR=T
C XTR=XCMSPH
C ZTR=ZCMSPH
C DO 10 IPRTS=1, NPRTS
C QPUFF(IPRTS)=SPHNS(IPRTS)
10 CONTINUE
C VOLSPH=(4./3.)*PI*RSPH**3
C CALL AMOUNT(VOLSPH, WAKAL, SPHAL)
C QPUFF(1)=SPHAL
60 SIG02=((2./3.)*RSPH)**2
C SIG0=SQRT(SIG02)
C ZX=5.0
61 CALL AVRGC(ZX, T, QTOT, XBAVRG, SIG2X, SIG2Y)
C SIGX=SQRT(SIG2X)
C SIGY=SQRT(SIG2Y)
C COMPUTE PARAMETERS NEEDED FOR LOCAL AND COLUMN DENISTY FOR THE WAKE
C SOLUTION
C SIGC=(SQRT(SIGX*SIGY)+SQRT(SIG02))/2.
C XDIF=XCMSPH-XBAVRG
C YDIF=YCMSPH
C ZDIF=ZCMSPH-5.0
C TDIF=SIGC**2/KX/2.
C TDZ=2.*KX
C TDZ=2.*KZ
C QLOC=WAKAL/4./PI/SQRT(2.)

```

998 QCOL=WAKAL/4./SQRT(PI)
RETURN
END

COV00700
COV00710
COV00720

```

FUNCTION CSPHER(X,Y,Z,T)
COMPUTE EITHER THE COLUMN DENSITY FOR A GIVEN LINE OF SIGHT OR
COMPUTE THE LOCAL CONCENTRATION AT X,Y,Z FOR THE SPHERE
INPUT
X,Y,Z COORDINATES IN METERS. IF LINE INTEGRAL IS DESIRED,
Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.
T THE TIME IN SECONDS AFTER DETONATION
OUTPUT
RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
THE LINE INTEGRAL OF CONCENTRATION (COLUMN DENSITY) IF HORIZ
IS .TRUE.
FUNCTIONS AND SUBROUTINES CALLED
NONE
*****
REAL M,N,KZ,KX
LOGICAL HORIZ, SWITCH, CHANGE, TEST
COMMON/PRTIME/ R0, VGRAV(3), NPRTS
COMMON /GEO/COSTH2, SINTH, SINTH2, VISEXT, RTP1, SCRN(2)
COMMON /MODE/ HORIZ
COMMON /WINDFRM/DXZ0, DYX0, DZ0, U0, M, N, ZINV
COMMON/DISCS/NDSCS, TDSC(20), XDSC(20), ZDSC(20), R2DSC(20),
1 QDSC(20,3)
COMMON/M05/DIFF(2,200), NCHTOT, PRSEP(200), NTOT, NARY, ITOT,
+ COOR(2,200), TSTAG(200), DMMY(401)
COMMON/TRAN/VTR, KZ, KX, TTR, XTR, ZTR, QPUFF(3), SWITCH, CHANGE
COMMON/SIG/SIG02, SIGC
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOJNIT, NDIRTU, NCLINT, KSTOR, NPLOTC
COMMON/LOAD/WAKAL, SPHAL
COMMON/ACL/CWINDS, CWINDC, CWINDW
COMMON/WAKE/XDIF, YDIF, ZDIF, TDIF, TDX, TDZ, QLOC, QCOL, XBAVRG
COMMON/LOS/TR(3), RE(3), U(3)
COMMON/CHARGE/NCHG
COMMON/TRANNY/THRESH, TEST, NWL, NSOIL
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
CSPHER=0.0
CWINDSC=0.0
CWINDC=0.0
IF(NARY.EQ.3)GO TO 999
IF((T-TTR).LT.1.E-20)GO TO 999
COMPUTE CONTRIBUTION FROM BUOYANT CLOUD AFTER SWITCHING TO THE
WIND MODEL USING A THREE DIMENSIONAL GAUSSIAN PUFF
SIGX2=SIG02+2.*KX*(T-TTR)
SIGZ2=SIG02+2.*KZ*(T-TTR)
SIGX=SQRT(SIGX2)
SIGZ=SQRT(SIGZ2)
ARG2=(Z-ZTR)**2/(2.*SIGZ2)
IF(ABS(ARG2).GT.30.)GO TO 25
TERM2=EXP(-ARG2)
GO TO 26
25 TERM2=0.0
26 ARG3=(Z+ZTR)**2/(2.*SIGZ2)
IF(ABS(ARG3).GT.30.)GO TO 27
TERM3=EXP(-ARG3)
GO TO 28
27 TERM3=0.0
28 IF(HORIZ)GO TO 50
COMPUTE CONCENTRATION AT X,Y,Z

```

```

CSP00230
CSP00010
CSP00020
CSP00030
CSP00040
CSP00050
CSP00060
CSP00070
CSP00080
CSP00090
CSP00100
CSP00110
CSP00120
CSP00130
CSP00140
CSP00150
CSP00160
CSP00170
CSP00180
CSP00190
CSP00200
CSP00210
CSP00220
CSP00240
CSP00250
CSP00260
CSP00270
CSP00280
CSP00290
CSP00300
CSP00310
CSP00320
CSP00330
CSP00340
CSP00350
CSP00360
CSP00370
CSP00380
CSP00390
CSP00400
CSP00410
CSP00420
CSP00430
CSP00440
CSP00450
CSP00460
CSP00470
CSP00480
CSP00490
CSP00500
CSP00510
CSP00520
CSP00530
CSP00540
CSP00550
CSP00560
CSP00570
CSP00580
CSP00590
CSP00600
CSP00610
CSP00620
CSP00630
CSP00640
CSP00650
CSP00660
CSP00670
CSP00680
CSP00690
CSP00700

```

CDC

```
XC=XTR+VTR*(T-TTR)+DIFF(1,NCHG)
YC=DIFF(2,NCHG)
ARG1=((X-XC)**2+(Y-YC)**2)/2./SIGX2
IF(ABS(ARG1).GT.30.)GO TO 999
TERM=SPHAL/((2.*PI)**(3./2.))/SIGZ/SIGX2
CWINDC=TERM*EXP(-ARG1)*(TERM2+TERM3)
GO TO 100
CSP00710
CSP00720
CSP00730
CSP00740
CSP00750
CSP00760
CSP00770
CSP00780
CSP00790
CSP00800
CSP00810
CSP00820
CSP00830
CSP00840
CSP00850
CSP00860
CSP00870
CSP00880
CSP00890
CSP00900
CSP00910
CSP00920
CSP00930
CSP00940
CSP00950

50 COMPUTE COLUMN DENSITY
50 TERM=SPHAL/2./PI/SIGX/SIGZ
DO 90 I=1,NCHTOT
XC=XTR+VTR*(T-TTR)+DIFF(1,I)
YC=DIFF(2,I)
XY=XC*SCRN(1)+YC*SCRN(2)
ARG1=(X-XY)**2/2./SIGX2
IF(ABS(ARG1).GT.30.)GO TO 90
CWINDSC=TERM*EXP(-ARG1)*(TERM2+TERM3)
CWINDC=CWINDC+CWINDSC
CALL TRNCHK(CWINDS,CWINDW,CWINDC)
IF(TEST)GO TO 999
90 CONTINUE
100 CSPHER=CWINDC
999 RETURN
END
```

```

FUNCTION CWAKE(X,Y,Z,T)
FUNCTION TO COMPUTE THE LOCAL CONCENTRATION OR COLUMN DENSITY AT
X,Y,Z AND TIME T AFTER THE BLAST FOR THE WAKE.

INPUT
X,Y,Z COORDINATES IN METERS. IF THE LINE INTEGRAL IS DESIRED
      THESE ARE NOT USED AND THE LINE IS SPECIFIED BY THE
      TRANSMITTER AND RECEIVER COORDINATES AND INFORMATION
      CALCULATED AT THE TIME THE BUOYANT FIREBALL CONVERTED
      TO THE WIND MODEL.
T THE TIME IN SECONDS AFTER DETONATION

OUTPUT
RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
THE LINE INTEGRAL OF CONCENTRATION (COLUMN DENSITY) IF
HORIZ IS .TRUE.

FUNCTIONS AND SUBROUTINES NEEDED
ERF COMPUTE THE ERROR FUNCTION
*****
REAL M,N,KZ,KX
LOGICAL HORIZ, SWITCH, CHANGE, TEST
COMMON/PRTINF/ R0,VGRAV(3),NPTS
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON/MODE/ HORIZ
COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
1 QDSC(20,3)
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ COOR(2,200),TSTAG(200),DMMY(401)
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON/IUNIT/IDIN,IODUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUC
COMMON/LOAD/WAKAL,SPHAL
COMMON/ACL/CWINDS,CWINDC,CWINDW
COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAYRG
COMMON/LOS/TR(3),RE(3),U(3)
COMMON/CHARGE/NCHG
COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
IF(NARY.EQ.3)GO TO 999
IF((T-TTR).LT.1.E-20)GO TO 999

COMPUTE CONTRIBUTION FROM THE WAKE AFTER SWITCHING TO THE WIND MODEL
FOR A SINGLE CHARGE
CWAKE=0.0
CWINDW=0.0
CWINDSW=0.0
SIGX2=TDX*(TDIF+(T-TTR))
SIGZ2=TDZ*(T-TTR)
IF(HORIZ)GO TO 210
XB=XBAYRG+DIFF(1,NCHG)
YB=DIFF(2,NCHG)

COMPUTE THE LOCAL CONCENTRATION
A=-((XDIF**2+YDIF**2)/2./SIGX2-(ZDIF**2/2./SIGZ2)
XX=X-VTR*(T-TTR)
B0=(XDIF*(XX-XB)+YDIF*(Y-YB))/SIGX2
B1=B0+(ZDIF*(Z-5.)/SIGZ2)

```

CWA00290
CWA00010
CWA00020
CWA00030
CWA00040
CWA00050
CWA00060
CWA00070
CWA00080
CWA00090
CWA00100
CWA00110
CWA00120
CWA00130
CWA00140
CWA00150
CWA00160
CWA00170
CWA00180
CWA00190
CWA00200
CWA00210
CWA00220
CWA00230
CWA00240
CWA00250
CWA00260
CWA00270
CWA00280
CWA00300
CWA00310
CWA00320
CWA00330
CWA00340
CWA00350
CWA00360
CWA00370
CWA00380
CWA00390
CWA00400
CWA00410
CWA00420
CWA00430
CWA00440
CWA00450
CWA00460
CWA00470
CWA00480
CWA00490
CWA00500
CWA00510
CWA00520
CWA00530
CWA00540
CWA00550
CWA00560
CWA00570
CWA00580
CWA00590
CWA00600
CWA00610
CWA00620
CWA00630
CWA00640
CWA00650
CWA00660
CWA00670
CWA00680
CWA00690
CWA00700

	B2=B0-(ZDIF*(Z+5.)/SIGZ2)	CWA00710
	C0=-(XX-XB)**2+(Y-YB)**2)/2./SIGX2	CWA00720
	C1=C0-(Z-5.)**2/2./SIGZ2)	CWA00730
	C2=C0-(Z+5.)**2/2./SIGZ2)	CWA00740
	RTMA=SQRT(-A)	CWA00750
	ARG1=C1-(B1**2/4./A)	CWA00760
	ARG2=-B1/2./RTMA	CWA00770
	ARG3=RTMA+ARG2	CWA00780
	ARG4=C2-(B2**2/4./A)	CWA00790
	ARG5=-B2/2./RTMA	CWA00800
	ARG6=RTMA+ARG5	CWA00810
	IF(ARG1.LT.-30.)GO TO 221	CWA00820
	TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))	CWA00830
	GO TO 222	CWA00840
221	TERM1=0.0	CWA00850
222	CONTINUE	CWA00860
	IF(ARG4.LT.-30.)GO TO 223	CWA00870
	TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))	CWA00880
	GO TO 224	CWA00890
223	TERM2=0.0	CWA00900
224	CWAKE=QLOC/SIGX2/SQRT(SIGZ2)/RTMA*(TERM1+TERM2)	CWA00910
	GO TO 999	CWA00920
CCC	COMPUTE COLUMN DENSITY	CWA00930
210	DO 245 J=1,NCHTOT	CWA00940
	XB=XBAVRG+DIFF(1,J)	CWA00950
	YB=DIFF(2,J)	CWA00960
230	A=-(XDIF*U(2)-YDIF*U(1))**2/2./SIGX2-(ZDIF)**2/2./SIGZ2	CWA00970
	TRR=TR(1)-VTR*(T-TTR)	CWA00980
	B0=(YDIF*U(1)-XDIF*U(2))*(TR(2)-YB)*U(1)-(TRR-XB)*U(2))/SIGX2	CWA00990
	B1=B0+ZDIF*(TR(3)-5.)/SIGZ2	CWA01000
	B2=B0-ZDIF*(TR(3)+5.)/SIGZ2	CWA01010
	C0=-(TR(2)-YB)*U(1)-(TRR-XB)*U(2))**2/2./SIGX2	CWA01020
	C1=C0-(TR(3)-5.)**2/2./SIGZ2	CWA01030
	C2=C0-(TR(3)+5.)**2/2./SIGZ2	CWA01040
	RTMA=SQRT(-A)	CWA01050
	ARG1=C1-B1**2/4./A	CWA01060
	ARG2=-B1/2./RTMA	CWA01070
	ARG3=RTMA+ARG2	CWA01080
	ARG4=C2-B2**2/4./A	CWA01090
	ARG5=-B2/2./RTMA	CWA01100
	ARG6=RTMA+ARG5	CWA01110
	IF(ARG1.GT.30.)ARG1=30.	CWA01120
	IF(ARG1.LT.-30.)GO TO 231	CWA01130
	TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))	CWA01140
	GO TO 232	CWA01150
231	TERM1=0.0	CWA01160
232	CONTINUE	CWA01170
	IF(ARG4.LT.-30.)GO TO 233	CWA01180
	TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))	CWA01190
	GO TO 234	CWA01200
233	TERM2=0.0	CWA01210
234	ARG=SIGX2*SIGZ2	CWA01220
	CWNSW=(QCOL/SQRT(ARG)/RTMA)*(TERM1+TERM2)	CWA01230
240	CONTINUE	CWA01240
	CWINDW=CWINDW+CWNSW	CWA01250
	CALL TRNCHK(CWINDS,CWINDW,CWINDC)	CWA01260
	IF(TEST)GO TO 999	CWA01270
245	CONTINUE	CWA01280
	CWAKE=CWINDW	CWA01290
999	RETURN	CWA01300
	END	CWA01310
		CWA01320
		CWA01330


```

C          WEIGHTED MASS OF PARTICLES OF THE J SIZE RANGE IN THE
C          I DISC.
C          SUM THE CONTRIBUTIONS OF THE DISC SOURCES TO THE
C          OPTICALLY WEIGHTED CONCENTRATION AT (X,Y,Z,T)
C          CWIND=0.
C          CWNDS=0.0
C          CWINDS=0.0
C          CWINDC=0.0
C          CWINDM=0.0
C          IF(HORIZ)GO TO 110
C          COMPUTE CONCENTRATION AT X,Y,Z (FOR SIMULTANEOUS BURST)
C          DO 100 I=1,NDSCS
C          REF0(1)=XDSC(I)
C          REF0(2)=0.0
C          ROH2=R2DSC(I)
C          H=ZDSC(I)
C          TOF=T-TDSC(I)
C          DO 90 J=1,NPRTS
C          DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z
C          CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)
C          IF(Q.GT.1.E-10)GO TO 50
C          CWNDS=0.0
C          GO TO 100
C          50 CONTINUE
C          RX2=ROH2+2.*SIGW2
C          RY2=ROH2+2.*SIGP2
C          DO 114 NA=1,2
C          REF(NA)=REF0(NA)+DIFF(NA,NCHG)
C          114 CONTINUE
C          ARG=-((X-REF(1))-XBAR)**2/RX2
C          IF(ABS(ARG).GT.30.)GO TO 100
C          CWNDS=(Q/RTPI/SQRT(RX2))*EXP(ARG)
C          ARG=-((Y-REF(2))-XBAR)**2/RY2
C          IF(ABS(ARG).GT.30.)GO TO 100
C          CY=EXP(ARG)/RTPI/SQRT(RY2)
C          CWNDS=CWNDS+(QDSC(I,J)*CWNDS*CY)
C          CWINDS=CWINDS+CWNDS
C          90 CONTINUE
C          100 CONTINUE
C          CWINDC=CWINDC
C          GO TO 999
C          110 DO 220 ICHG=1,NTOT
C          IF(T.LT.TSTAG(ICHG))GO TO 220
C          DO 211 I=1,NDSCS
C          TOF=T-TDSC(I)-TSTAG(ICHG)
C          REF0(1)=XDSC(I)
C          REF0(2)=0.0
C          ROH2=R2DSC(I)
C          H=ZDSC(I)
C          IF(HORIZ) REF0(1)=REF0(1)*SINTH
C          DO 210 J=1,NPRTS
C          CWNDS=0.0
C          DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z
C          CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)
C          IF(Q.GT.1.E-10) GO TO 113
C          IF Q IS TOO SMALL, ITS CONTRIBUTION IS IGNORED
C          CWNDS=0.
C          GO TO 210
C          113 CONTINUE
C          CWI00710
C          CWI00720
C          CWI00730
C          CWI00740
C          CWI00750
C          CWI00760
C          CWI00770
C          CWI00780
C          CWI00790
C          CWI00800
C          CWI00810
C          CWI00820
C          CWI00830
C          CWI00840
C          CWI00850
C          CWI00860
C          CWI00870
C          CWI00880
C          CWI00890
C          CWI00900
C          CWI00910
C          CWI00920
C          CWI00930
C          CWI00940
C          CWI00950
C          CWI00960
C          CWI00970
C          CWI00980
C          CWI00990
C          CWI01000
C          CWI01010
C          CWI01020
C          CWI01030
C          CWI01040
C          CWI01050
C          CWI01060
C          CWI01070
C          CWI01080
C          CWI01090
C          CWI01100
C          CWI01110
C          CWI01120
C          CWI01130
C          CWI01140
C          CWI01150
C          CWI01160
C          CWI01170
C          CWI01180
C          CWI01190
C          CWI01200
C          CWI01210
C          CWI01220
C          CWI01230
C          CWI01240
C          CWI01250
C          CWI01260
C          CWI01270
C          CWI01280
C          CWI01290
C          CWI01300
C          CWI01310
C          CWI01320
C          CWI01330
C          CWI01340
C          CWI01350
C          CWI01360
C          CWI01370
C          CWI01380
C          CWI01390
C          CWI01400

```

	DO 200 MA=1,NCHTOT	CWI01410
	INDX=MA	CWI01420
	IF(NARY.EQ.3)INDX=ICMG	CWI01430
	RX2=ROH2+2.*SIGW2	CWI01440
	RY2=ROH2+2.*SIGP2	CWI01450
CCC	COMPUTE CONCENTRATION ALONG LINE OF SIGHT SPECIFIED BY X,Z	CWI01460
		CWI01470
		CWI01480
120	CONTINUE	CWI01490
	REF(1)=REF0(1)+PRSEP(INDX)	CWI01500
	REFF2=RX2*SINTH2+RY2*COSTH2	CWI01510
	ARG=-((X-REF(1)-XBAR*SINTH)**2/REFF2	CWI01520
	IF(ABS(ARG).GT.30.)GO TO 150	CWI01530
	CWINDSC=EXP(ARG) /SQRT(REFF2)/RTPI	CWI01540
	CWINDSC=CWINDSC*Q*GDSC(I,J)	CWI01550
150	CONTINUE	CWI01560
	CWINDS=CWINDS+CWINDSC	CWI01570
	IF(SKIP)GO TO 190	CWI01580
	CALL TRNCHK(CWINDS,CWINDW,CWINDC)	CWI01590
	IF(TEST)GO TO 999	CWI01600
190	CONTINUE	CWI01610
200	CONTINUE	CWI01620
210	CONTINUE	CWI01630
211	CONTINUE	CWI01640
220	CONTINUE	CWI01650
	CWIND=CWINDS	CWI01660
999	RETURN	CWI01670
	END	CWI01680

```

SUBROUTINE DIFEQ(N,T,Y,YP)
REAL KM,KZ,KX
LOGICAL SWITCH,CHANGE
DIMENSION Y(N),YP(N)
COMMON/PRTINF/ROCL,VGRAV(3),NPRTS
COMMON/WNDPRM/DXZO,DYXO,DZO,UO,UM,ON,ZINV
COMMON /ARRAY/OVLAP,AREA,PERIM,PRJARY,CENDIF
COMMON/MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ DMM(600),DMMY(401)
COMMON /BURST/ ACCEL,TBURST
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/EKWIND/ALP,C,PYF,PIX,UHAT,VHAT
COMMON/EKTEMP/ZO,ZL,TO,TC1,TC2,TC3
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
DATA ALPHA/.25/
*****

PURPOSE

DIFEQ CONTAINS THE PARTIAL DIFFERENTIAL EQUATIONS FOR THE
RISE OF A BUOYANT CLOUD WHICH ARE USED BY SUBROUTINE RKM.

INPUT

N THE NUMBER OF DEPENDENT VARIABLES
T THE INDEPENDENT VARIABLE, I.E. TIME
Y(1) RADIUS OF CLOUD
Y(2) CLOUD TEMPERATURE MINUS SURROUNDING TEMPERATURE
Y(3) VERTICAL VELOCITY OF CLOUD
Y(4) X-COORDINATE OF CENTER OF MASS FOR THE CLOUD
Y(5) Y-COORDINATE OF CENTER OF MASS FOR THE CLOUD
Y(6) THE HEIGHT OF THE CLOUD C.O.M.
Y(7) X-COORDINATE OF TOP OF CLOUD
Y(8) Y-COORDINATE OF TOP OF CLOUD

OUTPUT

YP AN ARRAY CONTAINING COMPUTED DERIVATIVES OF THE DEPENDENT
VARIABLES WITH RESPECT TO THE INDEPENDENT VARIABLE.

REQUIRED FUNCTIONS

TEMP CALCULATES AMBIENT ATMOSPHERIC TEMPERATURE AND THE
TEMPERATURE GRADIENT AT CLOUD HEIGHT.

WIN CALCULATES THE WIND SPEED IN THE X AND Y DIRECTION AT
CLOUD HEIGHT.

DIFFUS COMPUTES THE DIFFUSIVITY AT A SPECIFIED HEIGHT.

CALLED BY RKM
*****
IF(T.LT.TBURST)GO TO 200

IF(Y(6).GT.ZSTAR)GO TO 5
CALL TEMP(Y(6),TA,DTADZ)
GO TO 6
5 TA=TC1+TC2*Y(6)+TC3*Y(6)**2
DTADZ=TC2+2.*TC3*Y(6)
6 CALL WIND(Y(6),XWCM,YWCM)
TOP=Y(6)+Y(1)
CALL WIND(TOP,XWTOP,YWTOP)

```

```

DFQ00010
DFQ00020
DFQ00030
DFQ00040
DFQ00050
DFQ00060
DFQ00070
DFQ00080
DFQ00090
DFQ00100
DFQ00110
DFQ00120
DFQ00130
DFQ00140
DFQ00150
DFQ00160
DFQ00170
DFQ00180
DFQ00190
DFQ00200
DFQ00210
DFQ00220
DFQ00230
DFQ00240
DFQ00250
DFQ00260
DFQ00270
DFQ00280
DFQ00290
DFQ00300
DFQ00310
DFQ00320
DFQ00330
DFQ00340
DFQ00350
DFQ00360
DFQ00370
DFQ00380
DFQ00390
DFQ00400
DFQ00410
DFQ00420
DFQ00430
DFQ00440
DFQ00450
DFQ00460
DFQ00470
DFQ00480
DFQ00490
DFQ00500
DFQ00510
DFQ00520
DFQ00530
DFQ00540
DFQ00550
DFQ00560
DFQ00570
DFQ00580
DFQ00590
DFQ00600
DFQ00610
DFQ00620
DFQ00630
DFQ00640
DFQ00650
DFQ00660
DFQ00670
DFQ00680
DFQ00690
DFQ00700

```

CCCCCCCCCCCC
 CCCCCCCCCC
 CCCCCC
 CCCC
 CCCC

```

TDIF=TA-T0
TA THE AMBIENT ATMOSPHERIC TEMPERATURE AT CLOUD HEIGHT
DTADZ THE TEMPERATURE GRADIENT AT CLOUD HEIGHT
XWCM THE WIND SPEED IN THE X DIRECTION AT CLOUD C.O.M.
YXCM THE WIND SPEED IN THE Y DIRECTION AT CLOUD C.O.M.
XWTOP THE WIND SPEED IN THE X DIRECTION AT THE TOP OF THE CLOUD
YWTOP THE WIND SPEED IN THE Y DIRECTION AT THE TOP OF THE CLOUD
TR THE RATIO OF CLOUD TEMPERATURE TO AMBIENT TEMPERATURE

TR=Y(2)/TA
CALCULATE ARVOL, THE SURFACE AREA TO VOLUME RATIO
ARVOL=3./Y(1)
DEFINITION OF DIFFERENTIAL EQUATIONS
YP(1)=ALPHAK*ABS(Y(3))
ZZ1=Y(6)
KM=DIFFUS(Z0,ZL,ZZ1)
GROWTH=KM/Y(1)
IF(Y(1).LT.GROWTH)YP(1)=GROWTH
YP(2)=-(.+TR)*ARVOL*Y(2)*YP(1)-Y(3)*(DTADZ)
YP(3)=9.8*TR-1.4*ARVOL*Y(3)*YP(1)
IF(Y(1)+Y(6).GT.ZINV)YP(3)=0.0
YP(4)=XWCM
YP(5)=YWCM
YP(6)=Y(3)
IF(Y(1)+Y(6).GT.ZINV)YP(6)=0.0
YP(7)=XWTOP
YP(8)=YWTOP
GO TO 999
200 CONTINUE
DO 210 I=1,N
YP(I)=0.
210 CONTINUE
YP(3)=ACCEL
YP(6)=Y(3)
999 RETURN
END

```

DFQ00710
 DFQ00720
 DFQ00730
 DFQ00740
 DFQ00750
 DFQ00760
 DFQ00770
 DFQ00780
 DFQ00790
 DFQ00800
 DFQ00810
 DFQ00820
 DFQ00830
 DFQ00840
 DFQ00850
 DFQ00860
 DFQ00870
 DFQ00880
 DFQ00890
 DFQ00900
 DFQ00910
 DFQ00920
 DFQ00930
 DFQ00940
 DFQ00950
 DFQ00960
 DFQ00970
 DFQ00980
 DFQ00990
 DFQ01000
 DFQ01010
 DFQ01020
 DFQ01030
 DFQ01040
 DFQ01050
 DFQ01060
 DFQ01070
 DFQ01080
 DFQ01090
 DFQ01100
 DFQ01110

```
FUNCTION DIFFUS(Z0,ZL,Z)
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
```

```
DIF00010
DIF00020
DIF00030
DIF00040
DIF00050
DIF00060
DIF00070
DIF00080
DIF00090
DIF00100
DIF00110
DIF00120
DIF00130
DIF00140
DIF00150
DIF00160
DIF00170
DIF00180
DIF00190
DIF00200
DIF00210
DIF00220
DIF00230
DIF00240
DIF00250
DIF00260
DIF00270
DIF00280
DIF00290
DIF00300
DIF00310
DIF00320
DIF00330
DIF00340
DIF00350
DIF00360
DIF00370
DIF00380
DIF00390
DIF00400
DIF00410
DIF00420
DIF00430
DIF00440
DIF00450
DIF00460
DIF00470
DIF00480
DIF00490
DIF00500
DIF00510
DIF00520
DIF00530
DIF00540
```

```
*****
PURPOSE
```

```
TO CALCULATE THE DIFFUSIVITY AT A GIVEN HEIGHT
```

```
INPUTS
```

```
Z0 FRICTION HEIGHT IN METERS.
ZL MONIN OBUKHOV LENGTH IN METERS.
Z HEIGHT AT WHICH DIFFUSIVITY IS DESIRED.
```

```
CALLED BY ATMCAL, RISE AND DIFEQ
```

```
SUBROUTINES AND FUNCTIONS NEEDED
```

```
NONE
```

```
*****
```

```
ZZ=Z
IF(Z.GT.ZSTAR)Z=ZSTAR
```

```
NEUTRAL CASE
```

```
IF(ABS(ZL).LT.1.E3)GO TO 100
DIFFUS=.4*USTAR*Z
GO TO 999
```

```
100 IF(ZL.GT.0.0)GO TO 200
```

```
UNSTABLE CASE
```

```
S=Z/ZL
IF(S.LT.-2.)GO TO 110
DIFFUS=.4*ABS(ZL*USTAR*S*(1.-16.*S)**(1./4.))
GO TO 999
110 DIFFUS=.4*ABS(ZL*(3./AW)*(-1.*S)**(4./3.))*USTAR
GO TO 999
```

```
STABLE CASE
```

```
200 S=Z/ZL
SO=Z0/ZL
IF(S.GT.1.5)GO TO 210
DIFFUS=.4*ZL*USTAR*ABS(1./((1./((S0+S)+7.)))
GO TO 999
210 DIFFUS=.4*ZL*USTAR*ABS(1./BW)
```

```
Z=ZZ
999 RETURN
END
```

C
C
C

```
FUNCTION DOTPRD(A,B)
DIMENSION A(2),B(2)
DOTPRD IS THE SCALAR PRODUCT OF A AND B
DOTPRD=A(1)*B(1)+A(2)*B(2)
RETURN
END
```

```
DOT00010
DOT00020
DOT00030
DOT00040
DOT00050
DOT00060
DOT00070
DOT00080
```

```

FUNCTION DTERPI(NDIM,XI,XVAL,VAL,VMIN,WORK)
*****
PURPOSE
PERFORMS AN N-DIMENSIONAL LINEAR INTERPOLATION

INPUT
NDIM - THE NUMBER OF DIMENSIONS. (<- DONT RECALCUALTE WEIGHTS)
XI - THE POINT IN THE HYPERSPACE AT WHICH THE INTERPOLATED
VALUE IS DESIRED. XI MUST BE A VECTOR OF ATLEAST NDIM
IN LENGTH.
XVAL - THE COORDINATE VALUES AT THE CORNERS OF THE HYPERCUBE.
THE VECTOR MUST BE SET UP LIKE A TWO-DIMENSIONAL ARRAY
(2 X NDIM), WHERE THE FIRST SUBSCRIPT REFERS TO THE
HYPERCUBE COORDINATES IN THE SECOND SUBSCRIPTS
DIRECTION.
VAL - THE FUNCTIONAL VALUES AT THE CORNERS OF THE HYPERCUBE
SURROUNDING XI. THIS VECTOR MUST BE FILLED EQUIVALENT
TO AN NDIM ARRAY WITH EACH DIMENSION AS 2. THE SIZE
OF VAL SHOULD BE ATLEAST 2**NDIM.
VMIN - A MINIMUM VALUE OF VAL FOR WHICH THE INTERPOLATION
WILL USE A CORNER VALUE.
WORK - A WORK VECTOR OF ATLEAST NDIM*2. USE TO STORE COOR-
DINATE WEIGHTS.

OUTPUT
RETURNS INTERPOLATED VALUE OF VAL AT XI
CALLED BY MOMENT
*****
DIMENSION XI(4),XVAL(8),VAL(16),WORK(8)

SET UP THE COORDINATE WEIGHTS
NDI=IABS(NDIM)
IF(NDIM.LT.0) GO TO 1
DO 100 I=1,NDI
I2=I*2
I1=I2-1
WORK(I2)=(XI(I)-XVAL(I1))/(XVAL(I2)-XVAL(I1))
WORK(I1)=1. - WORK(I2)
100 CONTINUE

INTERPOLATE - USE BINARY COUNTER FOR COORDINATE LOCATION
1
DTERPI=0.
SUM=0.
ND=2**NDI
DO 201 I=1,ND
IF(VAL(I).LT.VMIN) GO TO 201
L=I-1
WEIGHT=1.
DO 200 J=1,NDI
N=MOD(L,2)+J*2-1
WEIGHT=WEIGHT*WORK(N)
L=L/2
200 CONTINUE
SUM=SUM+WEIGHT
DTERPI=DTERPI+WEIGHT*VAL(I)
201 CONTINUE
IF(SUM.EQ.0.) GO TO 202
DTERPI=DTERPI/SUM
RETURN

```

DTI00010
DTI00020
DTI00030
DTI00040
DTI00050
DTI00060
DTI00070
DTI00080
DTI00090
DTI00100
DTI00110
DTI00120
DTI00130
DTI00140
DTI00150
DTI00160
DTI00170
DTI00180
DTI00190
DTI00200
DTI00210
DTI00220
DTI00230
DTI00240
DTI00250
DTI00260
DTI00270
DTI00280
DTI00290
DTI00300
DTI00310
DTI00320
DTI00330
DTI00340
DTI00350
DTI00360
DTI00370
DTI00380
DTI00390
DTI00400
DTI00410
DTI00420
DTI00430
DTI00440
DTI00450
DTI00460
DTI00470
DTI00480
DTI00490
DTI00500
DTI00510
DTI00520
DTI00530
DTI00540
DTI00550
DTI00560
DTI00570
DTI00580
DTI00590
DTI00600
DTI00610
DTI00620
DTI00630
DTI00640
DTI00650
DTI00660
DTI00670
DTI00680
DTI00690
DTI00700

C
202 STOP
END

DTI00710
DTI00720
DTI00730

```

SUBROUTINE DTERPS(II,X,VAL,NZ)
DIMENSION X(81,4,3),VAL(16),II(4)
*****
PURPOSE
  TO SET UP A ONE DIMENSIONAL ARRAY OF THE VALUES CORRESPONDING
  TO THE CORNERS OF THE CUBE WITHIN A TABULATED ARRAY WITH
  LOWEST CORNER INDICES GIVEN
INPUT
  II  SINGLY DIMENSIONED ARRAY CONTAINING THE INDICES OF THE
      LOWEST CORNER OF THE CUBE
  X   A TRIPLY DIMENSIONED ARRAY CONTAINING THE TABULATED
      VALUES TO BE SET UP. THE FIRST INDEX IS THE COLLAPSED
      INDEX FOR THE FIRST TWO INDICES OF A FOUR-DIMENSIONAL
      ARRAY
  NZ  THE RANGE OF THE FIRST INDEX OF THE FOUR-DIMENSIONAL
      ARRAY
OUTPUT
  VAL  SINGLY DIMENSIONED ARRAY CONTAINING THE VALUES OF X
       FOR THE 16 CORNER POINTS OF THE CUBE
CALLED BY MOMENT
*****
M=0
DO 104 L=1,2
LX=L + II(4) - 1
DO 103 K=1,2
KX=K + II(3) - 1
DO 102 JI=1,2
JIX=(JI + II(2) - 2)*NZ
DO 101 IJ=1,2
IJX=JIX + IJ + II(1) -1
M=M+1
VAL(M)=X(IJX,KX,LX)
101 CONTINUE
102 CONTINUE
103 CONTINUE
104 CONTINUE
RETURN
END
DTS00010
DTS00020
DTS00030
DTS00040
DTS00050
DTS00060
DTS00070
DTS00080
DTS00090
DTS00100
DTS00110
DTS00120
DTS00130
DTS00140
DTS00150
DTS00160
DTS00170
DTS00180
DTS00190
DTS00200
DTS00210
DTS00220
DTS00230
DTS00240
DTS00250
DTS00260
DTS00270
DTS00280
DTS00290
DTS00300
DTS00310
DTS00320
DTS00330
DTS00340
DTS00350
DTS00360
DTS00370
DTS00380
DTS00390
DTS00400
DTS00410
DTS00420
DTS00430
DTS00440
DTS00450
DTS00460
DTS00470
DTS00480

```

```

SUBROUTINE DUSTCL(NEWATM,NATHOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,      DUS00010
1 THWND,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,                  DUS00020
2 LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,              DUS00030
3 TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCPTH,NCPTS,CPTS,NERR,     DUS00040
4 NCHS,SRCBAS,SIDE1,SIDE2,DHDT)                               DUS00050
LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,HORIZ,ERR           DUS00060
LOGICAL SWITCH,CHANGE,DHDT                                   DUS00070
DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3)      DUS00080
1 ,RECCOR(3),CPTS(2,6),ORIG(2),TRNFRM(2,2),TRN(3),REC(3)    DUS00090
2 ,CNTRD(2),OBSCOR(2),DIR(2)                                DUS00100
3 ,SRCBAS(2),SIDE1(2),SIDE2(2),TEMP(2),NCHS(2)              DUS00110
REAL KZ,KX                                                    DUS00120
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)          DUS00130
COMMON /MODE/ HORIZ                                          DUS00140
COMMON /WINDPRM/ DXZO,DYXO,DZO,UO,UM,DN,ZINV                 DUS00150
COMMON /CLOCK/ FTIME,TWIND                                    DUS00160
COMMON /MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,    DUS00170
+ COOR(2,200),TSTAG(200),DMY(401)                            DUS00180
COMMON /ARRAY/OVLAP,AREA,PERIM,PRJARY,CENDIF                 DUS00190
COMMON /IUNIT/IOIN,IOOUT,IPHUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU DUS00200
COMMON /CARB/RCARB1,RCARB2                                    DUS00210
COMMON /ACL/CWINDS,CWINDC,CWINDW                              DUS00220
COMMON /TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE    DUS00230
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK               DUS00240
DATA ONEM/-1./                                               DUS00250
*****                                                       DUS00260
*****                                                       DUS00270
PURPOSE                                                       DUS00280
    DUSTCL CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES DUS00290
    THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE, DUS00300
    EXPLOSIVE CHARACTERISTICS, AND WAVELENGTH.                 DUS00310
    SEE COMMENTS IN DRTRAN FOR DETAILS.                        DUS00320
    SUBROUTINES CALLED                                         DUS00330
    ATMCAL ACCEPTS METEOROLOGICAL DATA AS ARGUMENTS AND COMPUTES DUS00340
    NECESSARY PARAMETERS IN COMMON /WINDPRM/, /EKWIND/        DUS00350
    /EKTEMP, /STARS/                                           DUS00360
    SOURCE ACCEPTS SOIL, CHARGE, AND WAVELENGTH SPECIFICATIONS DUS00370
    AS INPUT AND COMPUTES NECESSARY PARAMETERS AND INITIAL    DUS00380
    VALUES IN COMMON /PRTINF/, /BUOYCL/ AND /CARB/          DUS00390
    SETUP ACCEPTS THE USER DEFINED COORDINATES OF THE CHARGES DUS00400
    AND CONVERTS THEM TO THE INTERNAL (LOCAL) COORDINATE      DUS00410
    SYSTEM. ALSO COMPUTES THE AREA AND PERIMETER OF THE       DUS00420
    BOUNDING PARALLELOGRAM AND OVERLAP DISTANCE OF THE        DUS00430
    CHARGES AND RETURNS THEM IN COMMON /ARRAY/ AND            DUS00440
    /SEPRN/.                                                    DUS00450
    RISE GIVEN CLOUD DIMENSIONS DURING BUOYANT RISE DEVELOPMENT DUS00460
    OF CLOUD, RISE CALCULATES THE DIMENSIONS AT A LATER      DUS00470
    TIME                                                         DUS00480
    CLDIM DETERMINES THE OUTPUT VARIABLES DESCRIBING THE CLOUD DUS00490
    DIMENSIONS.                                                 DUS00500
    TRNCAL CONTROLLING ROUTINE FO THE CALCULATION OF TRANSMITTANCES. DUS00510
    *****                                                    DUS00520
    *****                                                    DUS00530
    IF(LOSTRN.OR.EDGE)GO TO 101                                DUS00540
    NERR=4                                                       DUS00550
    GO TO 999                                                    DUS00560
    101 IF(.NOT.NEWATM)GO TO 200                                 DUS00570
    THETA=THWND*PIRAD                                           DUS00580
    CALL ATMCAL(NATHOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,THETA,DHDT,ERR) DUS00590
    99999 IF(.NOT.ERR)GO TO 155                                 DUS00600
    *****                                                    DUS00610
    *****                                                    DUS00620
    *****                                                    DUS00630
    *****                                                    DUS00640
    *****                                                    DUS00650
    *****                                                    DUS00660
    *****                                                    DUS00670
    *****                                                    DUS00680
    *****                                                    DUS00690
    *****                                                    DUS00700

```

	NERR=7	DUS00710
	GO TO 999	DUS00720
155	CONTINUE	DUS00730
		DUS00740
CCCC	COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT USER	DUS00750
	DEFINED COORDINATES INTO LOCAL COORDINATES WITH X AXIS IN	DUS00760
	THE WIND DIRECTION.	DUS00770
		DUS00780
	THETAX=THWIND*PIRAD	DUS00790
	TRNFRM(1,1)=COS(THETAX)	DUS00800
	TRNFRM(2,2)=TRNFRM(1,1)	DUS00810
	TRNFRM(1,2)=SIN(THETAX)	DUS00820
200	TRNFRM(2,1)=-TRNFRM(1,2)	DUS00830
	CONTINUE	DUS00840
	IF(.NOT.NEWSRC) GO TO 300	DUS00850
	TWIND=1.E5	DUS00860
	TTR=1.E5	DUS00870
	TPRES=0.	DUS00880
	DEL=.001	DUS00890
	DO 250 I=1,2	DUS00900
	IF(NARY.GT.1)SRCBAS(I)=COOR(I,1)	DUS00910
	ORIG(I)=SRCBAS(I)	DUS00920
250	CONTINUE	DUS00930
	CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)	DUS00940
	CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)	DUS00950
300	CONTINUE	DUS00960
	IF(.NOT.LOSTRN) GO TO 400	DUS00970
		DUS00980
CCCC	CONVERT TRNCOR AND RECCOR TO LOCAL COORDINATES WITH ORIGIN AT	DUS00990
	SRCBAS AND X AXIS IN WIND DIRECTION.	DUS01000
		DUS01010
	TRN(3)=TRNCOR(3)	DUS01020
	REC(3)=RECCOR(3)	DUS01030
	DO 320 I=1,2	DUS01040
	TRN(I)=0.	DUS01050
	REC(I)=0.	DUS01060
	DO 310 J=1,2	DUS01070
	TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))	DUS01080
	REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))	DUS01090
310	CONTINUE	DUS01100
320	CONTINUE	DUS01110
400	CONTINUE	DUS01120
	IF(.NOT.EDGE) GO TO 500	DUS01130
		DUS01140
CCCC	COMPUTE A UNIT VECTOR IN THE DIRECTION OF THE OBSERVERS LINE	DUS01150
	OF SIGHT AND A UNIT VECTOR PERPENDICULAR TO THE LINE OF SIGHT	DUS01160
		DUS01170
	CALL VSUM(ORIG,OBSCOR,ONEM,DIR)	DUS01180
	CALL UNIT(DIR,DIR,RANGE)	DUS01190
	COSTH=0.	DUS01200
	SINTH=0.	DUS01210
	DO 410 J=1,2	DUS01220
	COSTH=COSTH+TRNFRM(1,J)*DIR(J)	DUS01230
	SINTH=SINTH+TRNFRM(2,J)*DIR(J)	DUS01240
410	CONTINUE	DUS01250
	SINTH2=SINTH*SINTH	DUS01260
	COSTH2=COSTH**2	DUS01270
	SCRN(1)=SINTH	DUS01280
	SCRN(2)=-COSTH	DUS01290
		DUS01300
CCCC	COMPUTE THE PROJECTION OF EACH DIFFERENCE VECTOR DIFF ONTO THE	DUS01310
	VECTOR PERPENDICULAR TO THE LINE OF SIGHT,(DIFF(1,J),DIFF(2,J))	DUS01320
	IS THE VECTOR FROM THE REFERENCE CHARGE TO THE JTH CHARGE	DUS01330
	LOCATION IN THE INTERNAL COORDINATE SYSTEM.	DUS01340
		DUS01350
	PARY1=0.0	DUS01360
	PARY2=0.0	DUS01370
	DO 420 J=1,NCHTOT	DUS01380
	DO 415 I=1,2	DUS01390
	TEMP(I)=DIFF(I,J)	DUS01400

415	CONTINUE	DUS01410
	PRSEP(J)=DOTPRD(TEMP,SCRN)	DUS01420
	X=PRSEP(J)	DUS01430
	IF(X.LT.0.0)GO TO 416	DUS01440
	IF(X.GT.PARY1)PARY1=X	DUS01450
	GO TO 420	DUS01460
416	IF(X.LT.PARY2)PARY2=X	DUS01470
420	CONTINUE	DUS01480
	PRJARY=(PARY1-PARY2)/2.	DUS01490
	CENDIF=(PARY1+PARY2)/2.	DUS01500
500	CONTINUE	DUS01510
	IF(NARY.EQ.3)GO TO 600	DUS01520
	IF(NEWTIM)CALL RISE(TPRES,TIME,DEL)	DUS01530
600	IF(.NOT.EDGE)GO TO 650	DUS01540
	ETIME=TIME	DUS01550
	CALL CLDIM(CNTRD,HEIGHT,CENWTH,SPCWT,SPCWTH,NCPTS,CPTS,ERR)	DUS01560
	IF(.NOT.ERR)GO TO 650	DUS01570
	NERR=6	DUS01580
	GO TO 999	DUS01590
650	CONTINUE	DUS01600
	IF(.NOT.LOSTRN)GO TO 999	DUS01610
C	CALL TRNCAL(TRN,REC,TIME,TRNLOS)	DUS01620
999	RETURN	DUS01630
	END	DUS01640
		DUS01650

```

FUNCTION ERF(X)
CALCULATES THE ERROR FUNCTION

INPUT
  X VALUE AT WHICH ERROR FUNCTION IS DESIRED
FUNCTIONS AND SUBROUTINES NEEDED
  NONE
*****
DIMENSION P(3,5),Q(3,5)
DATA RPI/3641896/
DATA P/2.138533E+01,7.373888E+00,-4.257996E-02,
+ 1.722276E+00,6.865018E+00,-1.960690E-01,
+ 3.166529E-01,3.031799E+00,-5.168823E-02,
+ 0.,5.631696E-01,0.,
+ 0.,4.318779E-05,0.,
DATA Q/1.895226E+01,7.373961E+00,1.509421E-01,
+ 7.843746E+00,1.518491E+01,9.214524E-01,
+ 1.000000E+00,1.279553E+01,1.000000E+00,
+ 0.,5.354217E+00,0.,
+ 0.,1.000000E+00,0.
AX=ABS(X)
ERFC=0.0
IF(AX.GT.11.0) GO TO 300
X2=AX*AX
I=2
IF(AX.LT.0.5) I=1
IF(AX.GT.4.) I=3
IF(I-2) 10,20,30
10 N=3
Z=X2
GO TO 40
20 N=5
Z=AX
GO TO 40
30 N=3
Z=1./X2
40 SP=P(I,N)
SQ=Q(I,N)
N1=N-1
DO 50 K=1,N1
J=N-K
SP=SP*Z+P(I,J)
50 SQ=SQ*Z+Q(I,J)
IF(I-2) 60,70,80
60 ERFC=1.0-X*SP/SQ
ERF=1.-ERFC
RETURN
70 ERFC=EXP(-X2)*SP/SQ
GO TO 300
80 ERFC=EXP(-X2)/AX*(RPI+SP/(SQ*X2))
300 IF(X.LT.0.0) ERFC=2.0-ERFC
ERF=1.-ERFC
RETURN
END

```

```

ERFF 00120
ERFF 00010
ERFF 00020
ERFF 00030
ERFF 00040
ERFF 00050
ERFF 00060
ERFF 00070
ERFF 00080
ERFF 00090
ERFF 00100
ERFF 00110
ERFF 00130
ERFF 00140
ERFF 00150
ERFF 00160
ERFF 00170
ERFF 00180
ERFF 00190
ERFF 00200
ERFF 00210
ERFF 00220
ERFF 00230
ERFF 00240
ERFF 00250
ERFF 00260
ERFF 00270
ERFF 00280
ERFF 00290
ERFF 00300
ERFF 00310
ERFF 00320
ERFF 00330
ERFF 00340
ERFF 00350
ERFF 00360
ERFF 00370
ERFF 00380
ERFF 00390
ERFF 00400
ERFF 00410
ERFF 00420
ERFF 00430
ERFF 00440
ERFF 00450
ERFF 00460
ERFF 00470
ERFF 00480
ERFF 00490
ERFF 00500
ERFF 00510
ERFF 00520
ERFF 00530
ERFF 00540
ERFF 00550
ERFF 00560
ERFF 00570
ERFF 00580

```

```

SUBROUTINE FIT(X,F,A,B,C)
QUADRATIC FIT TO THREE POINTS USING NEWTON'S FUNDAMENTAL FORMULA
INPUTS
  X - 3 VALUES OF THE INDEPENDENT VARIABLE
  F - 3 FUNCTION VALUES CORRESPONDING TO THE X VALUES
OUTPUTS
  A - COEFFICIENT OF THE X**2 TERM
  B - COEFFICIENT OF THE X TERM
  C - CONSTANT TERM
*****
DIMENSION X(3),F(3)
H=X(2)-X(1)
DF1=(F(2)-F(1))/H
DF2=(F(3)-2.*F(2)+F(1))/(2.*H**2)
A=DF2
B=DF1-DF2*(X(2)+X(1))
C=F(1)+X(1)*(X(2)*DF2-DF1)
RETURN
END

```

```

FIT00190
FIT00010
FIT00020
FIT00030
FIT00040
FIT00050
FIT00060
FIT00070
FIT00080
FIT00090
FIT00100
FIT00110
FIT00120
FIT00130
FIT00140
FIT00150
FIT00160
FIT00170
FIT00180
FIT00200
FIT00210
FIT00220
FIT00230
FIT00240
FIT00250
FIT00260
FIT00270
FIT00280

```

```

FUNCTION FUNCT(X,Z)
LOGICAL HORIZ,SKIP
COMMON /CLOCK/ TIME,TWIND
COMMON /MODE/HORIZ
COMMON /SKIPIT/SKIP

```

PURPOSE

TO SUPPLY A TRANSMITTANCE FUNCTION FOR THE CONTOUR TRACING ROUTINE IN ORDER TO DETERMINE THE CLOUD EDGE.

INPUT

X THE HORIZONTAL COORDINATE IN METERS
Z THE VERTICAL COORDINATE IN METERS

OUTPUT

RETURNS THE LOG OF THE OPTICALLY WEIGHTED CL VALUE (AT VISIBLE WAVELENGTHS) FOR THE LINE OF SIGHT SPECIFIED BY X,Z

FUNCTIONS CALLED

CWIND

CALLED BY GFUN, CLIMB, GRAD2

```

HORIZ=.TRUE.
SKIP=.TRUE.
Y=0.
EXT1=0.0
EXT2=0.0
IF(Z.LE.0.)GO TO 100
EXT1=CWIND(X,Y,Z,TIME)
IF(TIME.LE.TWIND)GO TO 10
EXT2=CSPHER(X,Y,Z,TIME)
10 EXT=EXT1+EXT2
IF(EXT.LE.1.E-30)GO TO 100
FUNCT=ALOG(EXT)
GO TO 999
100 FUNCT=-30.
999 CONTINUE
RETURN
END

```

```

FUC00010
FUC00020
FUC00030
FUC00040
FUC00050
FUC00060
FUC00070
FUC00080
FUC00090
FUC00100
FUC00110
FUC00120
FUC00130
FUC00140
FUC00150
FUC00160
FUC00170
FUC00180
FUC00190
FUC00200
FUC00210
FUC00220
FUC00230
FUC00240
FUC00250
FUC00260
FUC00270
FUC00280
FUC00290
FUC00300
FUC00310
FUC00320
FUC00330
FUC00340
FUC00350
FUC00360
FUC00370
FUC00380
FUC00390
FUC00400
FUC00410
FUC00420
FUC00430
FUC00440
FUC00450
FUC00460
FUC00470

```



```

SUBROUTINE GAMMA(XX,GX,IER)
SUBROUTINE GAMMA
PURPOSE
  COMPUTES THE GAMMA FUNCTION FOR A GIVEN ARGUMENT
USAGE
  CALL GAMMA(XX,GX,IER)
DESCRIPTION OF PARAMETERS
  XX -THE ARGUMENT FOR THE GAMMA FUNCTION
  GX -THE RESULTANT GAMMA FUNCTION
  IER -THE RESULTANT ERROR CODE WHERE
      IER=0 NO ERROR
      IER=1 XX IS WITHIN .000001 OF BEING A NEGATIVE INTEGER
      IER=2 XX GT 57, OVERFLOW, GX SET TO 1.E32
COMMENTS
  NONE
SUBROUTINES AND FUNCTIONS
  NONE
METHOD
  THE RECURSION RELATION AND POLYNOMIAL APPROXIMATION
  BY C. HASTINGS, JR., 'APPROXIMATIONS FOR DIGITAL COMPUTERS',
  PRINCETON UNIVERSITY PRESS, 1955
.....
  4 IF(XX-57.) 6,6,4
    IER=2
    GX=1.E32
    RETURN
  6 X=XX
    ERR=1.0E-6
    IER=0
    GX=1.0
    IF(X-2.0) 50,50,15
  10 IF(X-2.0) 110,110,15
  15 X=X-1.0
    GX=GX*X
    GO TO 10
  50 IF(X-1.0) 60,120,110
    SEE IF X IS NEAR NEGATIVE INTEGER OR ZERO
  60 IF(X-ERR) 62,62,80
  62 Y=FLOAT(INT(X))-X
    IF(ABS(Y)-ERR) 130,130,70
    X NOT NEAR A NEGATIVE INTEGER OR ZERO
  70 IF(X-1.0) 80,80,110
  80 GX=GX/X
    X=X+1.0
    GO TO 70
  110 Y=X-1.0
    CY=1.0+Y*(-0.5771017+Y*(0.9858540+Y*(-0.8764218+Y*(0.8328212+
    1 Y*(-0.5684729+Y*(0.2548205+Y*(-0.05149930))))))
    GX=GX*CY
  120 RETURN
  130 IER=1
    RETURN
END

```

CCCC

FUNCTION GFUNK(S)

GFUN IS THE RESTRICTION OF THE TWO DIMENSIONAL FUNCTION, F, TO
A LINE. I.E. FORM $G(S)=F(X,Y)$, WHERE $(X,Y)=BASE+S*DIR$.

EXTERNAL FUNCT

DIMENSION

P(2)

COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETA,CON

CALL VSUM(BASE,DIR,S,P)

GFUN=FUNCT(P(1),P(2))

RETURN

END

GFU00010

GFU00020

GFU00030

GFU00040

GFU00050

GFU00060

GFU00070

GFU00080

GFU00090

GFU00100

GFU00110

GFU00120


```

SUBROUTINE GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)
ROUTINE TO EVALUATE THE INTEGRAND FOR THE TRAPEZOIDAL INTEGRATION
FOR FINDING THE OPTICALLY WEIGHTED CONCENTRATION ALONG THE LINE OF
SIGHT WHERE THE DUST IS GENERATED BY A VEHICLE.
INPUTS
  U      - A UNIT VECTOR ALONG THE LINE OF SIGHT
  TR     - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE
          SYSTEM
  XNORM  - DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER
  TIME   - PRESENT TIME AT WHICH A TRANSMITTANCE IS WANTED
  TIVEN- TIME THAT THE VEHICLE HAS TRAVELED
  VDIR   - VECTOR CONTAINING DESCRIBING THE VEHICLE DIRECTION AND
          SPEED
OUTPUT
  VALUE  - VALUE OF THE INTEGRAND
FUNCTIONS AND SUBROUTINES NEEDED
  CONLEN - TO FIND THE LENGTH OF THE INTERSECTION OF THE LINE OF
          SIGHT AND THE TILTED CYLINDER
C*****
  DIMENSION U(3),TR(3),VDIR(2),VP(2)
  COMMON/M05/DMMY(604),DMM(600),
  +      ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
  +      RB(3,25),Z2(3,25)
  COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
  COMMON/VL/VLOAD
  COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
  FIND THE VEHICLE POSITION AT TIME TIVEH
  VP(1)=TIVEH*VDIR(1)
  VP(2)=TIVEH*VDIR(2)
  TOF=TIME-TIVEH
  DO 10 I=1,ICOUNT
  IND=I
  IF(TOF.LT.TIMES(I))GO TO 20
  10 CONTINUE
  IF TOF (TIME OF FLIGHT) IS GREATER THAN TABULATED VALUES IT IS
  ASSUMED THE THE CLOUD HAS DISSIPATED
  XC=X1*Z+X0 IS THE LINE THRUH THE CENTER OF THE CYLINDER
  GO TO 50
  20 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)
  X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)
  RAD=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND)
  HRTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)
  HTBOT=0.0
  RTOP=RAD
  RBOT=RAD
  XCEN=VP(1)+(X1*HRTOP+X0)
  YCEN=VP(2)
  XB=VP(1)+(X1*HTBOT+X0)
  YB=VP(2)
  IF(U(3).LT.1.E-06)GO TO 30
  COMPUTE INTERSECTION LENGTH FOR NON HORIZONTAL LINES OF SIGHT

```

```

GRD00320
GRD00010
GRD00020
GRD00030
GRD00040
GRD00050
GRD00060
GRD00070
GRD00080
GRD00090
GRD00100
GRD00110
GRD00120
GRD00130
GRD00140
GRD00150
GRD00160
GRD00170
GRD00180
GRD00190
GRD00200
GRD00210
GRD00220
GRD00230
GRD00240
GRD00250
GRD00260
GRD00270
GRD00280
GRD00290
GRD00300
GRD00310
GRD00330
GRD00340
GRD00350
GRD00360
GRD00370
GRD00380
GRD00390
GRD00400
GRD00410
GRD00420
GRD00430
GRD00440
GRD00450
GRD00460
GRD00470
GRD00480
GRD00490
GRD00500
GRD00510
GRD00520
GRD00530
GRD00540
GRD00550
GRD00560
GRD00570
GRD00580
GRD00590
GRD00600
GRD00610
GRD00620
GRD00630
GRD00640
GRD00650
GRD00660
GRD00670
GRD00680
GRD00690
GRD00700

```

	CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,XNORM,PLEN)	GRD00710
	GO TO 40	GRD00720
CCC	DETERMINE LENGTH OF INTERSECTION FOR A HORIZONTAL LINE OF SIGHT	GRD00730
	30 IF(HTTOP,LT,TR(3))GO TO 35	GRD00740
	A=U(1)**2+U(2)**2	GRD00750
	B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)	GRD00760
	C=(TR(1)-XCEN)**2+(TR(2)-YCEN)**2-RAD**2	GRD00770
	X=B**2-A*C	GRD00780
	IF(X.LT.0.0)GO TO 35	GRD00790
	P1=(-B+SQRT(X))/A	GRD00800
	P2=(-B-SQRT(X))/A	GRD00810
	IF(P1.GT.XNORM.AND.P2.GT.XNORM)GO TO 35	GRD00820
	IF(P2.LT.0.0.AND.P1.LT.0.0)GO TO 35	GRD00830
	PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0)	GRD00840
	GO TO 40	GRD00850
35	PLEN=0.0	GRD00860
40	VOL=PI*HTTOP*(RAD**2)	GRD00870
	VALUE=VLOAD*PLEN/VOL	GRD00880
	GO TO 999	GRD00890
50	VALUE=0.0	GRD00900
999	RETURN	GRD00910
	END	GRD00920
		GRD00930
		GRD00940

```

SUBROUTINE GREEN(Z,Z1,T,ALPHA,T0,IER)
*****
PURPOSE
  TO COMPUTE THE GENERALIZED GREENS FUNCTION
  USES GREEN1
  SEE GREEN1 FOR ARGUMENT LIST
*****
REAL N,M
COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
IF(N.EQ.1.) GO TO 2
X2=2,-N
AT=ALPHA*T
T0=0.
IF(AT.GE.Z1) RETURN
CALL GREEN1((Z+AT)**X2,Z1**X2,X2*X2*T,(N-1.)/X2,T1,IER)
T1=T1*X2*Z1**(1.-N)
U=1.
T2=0.
IF(ABS(ALPHA).LT.1.E-4) GO TO 1
ZMZN=Z-Z1+AT
X2=N+1
AN1=ALPHA*X2
ZMZN=Z1**X2 - (Z1-AT)**X2
ARG=(-AN1*ZMZN)/(4.*ZMZN)
IF(ARG.LT.-70.) GO TO 3
T2=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
3 IF(T1.LT.1.E-30 .AND. T2.LT.1.E-30) RETURN

CALCULATION OF MIXING RATIO, U, BY N=1 ANALOGY
CALL GREEN1(Z+AT,Z1,T,0.,T1U,IER)
X2=2.
AN1=ALPHA*X2
ZMZN=Z1**X2 - (Z1-AT)**X2
T2U=0.
ARG=(-AN1*ZMZN)/(4.*ZMZN)
IF(ARG.LT.-70.) GO TO 4
T2U=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
4 IF(T1U.LT.1.E-30 .AND. T2U.LT.1.E-30) GO TO 1
CALL GREEN1(Z,Z1,T,ALPHA,G,IER)
U=(G-T2U)/(T1U-T2U)
1 IF(U.LT.0.) U=0.
IF(U.GT.1.) U=1.

COMBINE LIMITING SOLUTIONS WITH DETERMINED MIXING RATIO
T0=U*T1 + (1.-U)*T2
RETURN
2 CALL GREEN1(Z,Z1,T,ALPHA,T0,IER)
RETURN
END

```

```

GRE00010
GRE00020
GRE00030
GRE00040
GRE00050
GRE00060
GRE00070
GRE00080
GRE00090
GRE00100
GRE00110
GRE00120
GRE00130
GRE00140
GRE00150
GRE00160
GRE00170
GRE00180
GRE00190
GRE00200
GRE00210
GRE00220
GRE00230
GRE00240
GRE00250
GRE00260
GRE00270
GRE00280
GRE00290
GRE00300
GRE00310
GRE00320
GRE00330
GRE00340
GRE00350
GRE00360
GRE00370
GRE00380
GRE00390
GRE00400
GRE00410
GRE00420
GRE00430
GRE00440
GRE00450
GRE00460
GRE00470
GRE00480
GRE00490
GRE00500
GRE00510
GRE00520
GRE00530
GRE00540
GRE00550
GRE00560
GRE00570
GRE00580
GRE00590

```

```

SUBROUTINE GREEN1(Z,Z1,T,NU,BI,IER)
SUBROUTINE GREEN1
PURPOSE
  COMPUTE THE I BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
  AND MULTIPLY BY AN APPROPRIATE POWER OF THE ARGUMENT
  AND AN EXPONENTIAL IN ORDER TO CALCULATE THE GREENS
  FUNCTION FOR THE WIND DIFFUSION EQUATION
USAGE
  CALL GREEN1(Z,Z1,T,NU,BI,IER)
DESCRIPTION OF PARAMETERS
  Z,Z1,T -THE ARGUMENTS OF THE FUNCTION DESIRED
  NU -THE ORDER OF THE I BESSEL FUNCTION
  BI -THE RESULTANT BESSEL FUNCTION
  IER -RESULTANT ERROR CODE WHERE
  IER=1 EXPONENTIAL UNDERFLOW (NON-FATAL), BI SET TO 0.0
  IER=0 NO ERROR
  IERR=1 NU NEAR NEGATIVE INTEGER
  IERR=2 OVERFLOW IN GAMMA
  IER=3 UNDERFLOW, BI .LT. 1.E-32, BI SET TO 0.0
  IER=4 OVERFLOW, X .GT. 90 WHERE X .GT. N
  IER=5 X IS NEGATIVE
REMARKS
  NU IS A REAL NUMBER
  N AND X MUST BE .GE. ZERO
  THIS SUBROUTINE IS A MODIFICATION OF BESI WHICH COMPUTES THE
  I BESSEL FUNCTION FOR INTEGER ORDERS. THE CHANGE REQUIRES
  USE OF THE GAMMA FUNCTION FOR COMPUTING THE FIRST TERM OF THE
  SERIES. THE SUCCESSIVE TERMS ARE CALCULATED WITH THE SAME
  RECURSION FORMULA AND THE ASYMPTOTIC APPROXIMATION IS ALSO
  UNCHANGED. BESI IS IN THE IBM SYSTEM/360 SCIENTIFIC
  SUBROUTINE PACKAGE. MODIFICATIONS MADE BY D. DVORE, AERODYNE
  RESEARCH INC. JANUARY 15,1979.
SUBROUTINES AND FUNCTIONS REQUIRED
  GAMMA WHICH COMPUTES THE GAMMA FUNCTION
METHOD
  COMPUTES I BESSEL FUNCTION USING SERIES OR ASYMPTOTIC
  APPROXIMATION DEPENDING ON THE RANGE OF THE ARGUMENT.
CALLED BY MOMENT
.....
REAL NU
X=2.*SQRT(Z*Z1)/T
CHECK FOR ERRORS IN NU AND X AND EXIT IF ANY ARE PRESENT
IER=0
BI=1.0
IF(NU)10,15,10
10 IF(X)160,20,20
15 IF(X)160,17,20
17 ARG=-(Z+Z1)/T
IF(ARG .LT. -90.) GO TO 170
BI=EXP(ARG)/T
RETURN
DEFINE TOLERANCE
20 TOL=1 E-3
IF ARGUMENT GT 12 AND GT NU, USE ASYMPTOTIC FORM

```

	IF(X-12.)40,40,30	GRV00710
30	IF(X-ABS(NU))40,40,110	GRV00720
	COMPUTE FIRST TERM OF SERIES AND SET INITIAL VALUE OF THE SUM	GRV00730
40	XX=X/2.	GRV00740
	N=INT(NU)	GRV00750
	FN=N	GRV00760
	R=NU-FN	GRV00770
	CALL GAMMA(1,+NU,GR,IER)	GRV00780
	IF(IER.EQ.0)GO TO 60	GRV00790
50	BI=0.0	GRV00800
	RETURN	GRV00810
60	TERM=1./GR	GRV00820
70	BI=TERM	GRV00830
	XX=XX*XX	GRV00840
	COMPUTE TERMS, STOPPING WHEN ABS(TERM) LE ABS(SUM OF TERMS)*TOLERA	GRV00850
	DO 90 K=1,1000	GRV00860
	IF(ABS(TERM)-ABS(BI*TOL))95,95,80	GRV00870
80	FK=K	GRV00880
	FK=FK*(NU+FK)	GRV00890
	TERM=TERM*(XX/FK)	GRV00900
90	BI=BI+TERM	GRV00910
95	ARG=-((Z+Z1)/T)	GRV00920
	IF(ARG.LT.-80.)GO TO 170	GRV00930
	BI=BI*(Z1/T)**NU*EXP(ARG)/T	GRV00940
	RETURN BI AS ANSWER	GRV00950
100	RETURN	GRV01000
	X GT 12 AND X GT NU, SO USE ASYMPTOTIC APPROXIMATION	GRV01010
110	FN=4.*NU*NU	GRV01020
115	XX=1./(8.*X)	GRV01030
	TERM=1.	GRV01040
	BI=1.	GRV01050
	DO 130 K=1,30	GRV01060
	IF(ABS(TERM)-ABS(BI*TOL))140,140,120	GRV01070
120	FK=(2*K-1)**2	GRV01080
	TERM=TERM*XX*(FK-FN)/FLOAT(K)	GRV01090
130	BI=BI+TERM	GRV01100
	SIGNIFICANCE LOST AFTER 30 TERMS, TRY SERIES	GRV01110
	GO TO 40	GRV01120
140	PI=3.141592653	GRV01130
	ARG=X-((Z+Z1)/T)	GRV01140
	IF(ARG.LT.-80.)GO TO 170	GRV01150
	BI=BI*(Z1/Z)**(NU/2.)*EXP(ARG)/SQRT(2.*PI*X)/T	GRV01160
	GO TO 100	GRV01170
160	IER=5	GRV01180
	GO TO 100	GRV01190
170	BI=0.0	GRV01200
	GO TO 50	GRV01210
	END	GRV01220
		GRV01230
		GRV01240
		GRV01250
		GRV01260
		GRV01270
		GRV01280


```

SUBROUTINE MOMENT(VGRAV,ZIN,H,TIN,Q,XBAR,SIGW2,SIGP2)
REAL M,N,NM
DIMENSION AL(9),Z(9),T(9),XB(81,4,3),SW(81,4,3),SP(81,4,3),NM(9)
DIMENSION VAL(16),XVAL(8),W(8),XI(4),IB(4),NTC(4),II(4),X(9,4)
LOGICAL FIRST
COMMON /WINDPRM/DXZO,DYXO,DZO,UO,M,N,ZINV
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
EQUIVALENCE (Z(1),X(1,1)),(T(1),X(1,2)),(AL(1),X(1,3))
EQUIVALENCE (NM(1),X(1,4))
DATA FIRST/.TRUE./,IB/4,4,1,2/,ITC/11/,HREF/1./
*****

PURPOSE
    TO CONVERT PARAMETERS TO NONDIMENSIONAL FORM AND THEN COMPUTE
    THE ZERO ORDER MOMENT AND INTERPOLATE FROM TABULATED VALUES OF
    THE HIGHER ORDER MOMENTS

INPUT
    VGRAV  THE GRAVITATIONAL SETTLING VELOCITIES OF THE PARTICLE
           IN METERS / SEC
    ZIN    THE HEIGHT (METERS) AT WHICH THE MOMENTS ARE DESIRED
    H      THE HEIGHT OF RELEASE OF THE PARTICLES IN METERS
    TIN    THE TIME IN SECONDS AFTER RELEASE

OUTPUT
    Q      THE VERTICAL CONCENTRATION OF PARTICLES AT HEIGHT Z
    XBAR   THE DISPLACEMENT (METERS) IN THE X (IE WIND) DIRECTION
           OF THE CENTER OF MASS OF PARTICLES AT HEIGHT Z
    SIGW2  THE SQUARE OF THE STANDARD DEVIATION OF THE WINDWARD
           DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
    SIGP2  THE SQUARE OF THE STANDARD DEVIATION OF THE CROSS-WIND
           DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2

SUBROUTINES CALLED
    DTERPS PUTS THE NEEDED VALUES OF THE TABULATED MOMENTS
           INTO A ONE-DIMENSIONAL ARRAY
    DTERPI A FUNCTION WHICH RETURNS THE INTERPOLATED VALUE
           FOR GIVEN ARGUMENTS AND ARRAYS
    GREEN  CALCULATES THE GREENS FUNCTION WHICH IS THE
           0-ORDER MOMENT

CALLED BY CWIND
*****
IF(.NOT.FIRST)GO TO 5

READ IN THE TABLE OF MOMENTS ON THE FIRST CALL OF MOMENT
    Z      LOG OF NON-DIMENSIONAL HEIGHTS AT WHICH MOMENTS ARE TABULATED
    T      LOG OF NON-DIMENSIONAL TIMES AT WHICH MOMENTS ARE TABULATED
    AL     NON-DIMENSIONAL SETTLING VELOCITIES AT WHICH MOMENTS ARE
           TABULATED
    NM     DIFFUSIVITY POWER LAW EXPONENTS AT WHICH MOMENTS ARE
           TABULATED

    XB     TABULATED VALUES OF LOGS OF FIRST ORDER MOMENTS (RELATED
           TO MEAN HORIZONTAL DISPLACEMENT)
    SW     TABULATED VALUES OF LOGS OF WIND SHEAR COMPONENT OF SECOND
           ORDER MOMENT (CONTRIBUTES TO VARIANCE IN WIND DIRECTION)
    SP     TABULATED VALUES OF LOGS OF SECOND ORDER MOMENT COMMON TO
           WIND AND CROSS-WIND VARIANCES

READ(NDIRTU,1) NZ,NT,NA,NN
1 FORMAT(4I3)

```

```

MOM00010
MOM00020
MOM00030
MOM00040
MOM00050
MOM00060
MOM00070
MOM00080
MOM00090
MOM00100
MOM00110
MOM00120
MOM00130
MOM00140
MOM00150
MOM00160
MOM00170
MOM00180
MOM00190
MOM00200
MOM00210
MOM00220
MOM00230
MOM00240
MOM00250
MOM00260
MOM00270
MOM00280
MOM00290
MOM00300
MOM00310
MOM00320
MOM00330
MOM00340
MOM00350
MOM00360
MOM00370
MOM00380
MOM00390
MOM00400
MOM00410
MOM00420
MOM00430
MOM00440
MOM00450
MOM00460
MOM00470
MOM00480
MOM00490
MOM00500
MOM00510
MOM00520
MOM00530
MOM00540
MOM00550
MOM00560
MOM00570
MOM00580
MOM00590
MOM00600
MOM00610
MOM00620
MOM00630
MOM00640
MOM00650
MOM00660
MOM00670
MOM00680
MOM00690
MOM00700

```

	NTC(1)=NZ-1	MOM00710
	NTC(2)=NT-1	MOM00720
	NTC(3)=NA-1	MOM00730
	NTC(4)=NN-1	MOM00740
	READ(NDIRTU,2) (Z(I),I=1,NZ)	MOM00750
2	FORMAT(6E13,5)	MOM00760
	READ(NDIRTU,2) (T(I),I=1,NT)	MOM00770
	READ(NDIRTU,2) (AL(I),I=1,NA)	MOM00780
	READ(NDIRTU,2) (NM(I),I=1,NN)	MOM00790
	NZT=NZ*NT	MOM00800
	DO 3 L=1,NN	MOM00810
	READ(NDIRTU,2) ((XB(IJ,K,L),IJ=1,NZT),K=1,NA)	MOM00820
	READ(NDIRTU,2) ((SW(IJ,K,L),IJ=1,NZT),K=1,NA)	MOM00830
	READ(NDIRTU,2) ((SP(IJ,K,L),IJ=1,NZT),K=1,NA)	MOM00840
3	CONTINUE	MOM00850
	FIRST=.FALSE.	MOM00860
	REWIND NDIRTU	MOM00870
5	CONTINUE	MOM00880
	CONVERT INPUT PARAMETERS TO NONDIMENSIONAL FORM	MOM00890
	SCLU=DZ0*H**(N-1.)	MOM00900
	XI(1)=ZIN/H	MOM00910
	XI(2)=SCLU*TIN/H	MOM00920
	XI(3)=VGRAY/SCLU	MOM00930
	XI(4)=N	MOM00940
	CALL GREEN(XI(1),HREF,XI(2),XI(3),Q,IER)	MOM00950
	Q=Q/H	MOM00960
	IF(Q.LE.1.E-10) GO TO 999	MOM00970
	TAKE LOGS FOR LOGARITHMIC INTERPOLATION	MOM00980
	XI(1)=ALOG(XI(1))	MOM00990
	XI(2)=ALOG(XI(2))	MOM01000
	DETERMINE INDICES OF LOWEST CORNER POINT OF THE CUBE TO	MOM01010
	BE USED IN INTERPOLATION MAKING SURE THAT ENOUGH CORNER POINTS	MOM01020
	OF THE CUBE HAVE TABULATED VALUES	MOM01030
	DO 100 I=1,4	MOM01040
	II(I)=IB(I)	MOM01050
100	CONTINUE	MOM01060
	DO 101 III=1,4	MOM01070
	I=5-III	MOM01080
6	IA=II(I)	MOM01090
	IF(XI(I).GE.X(IA,I).AND.XI(I).LE.X(IA+1,I)) GO TO 101	MOM01100
	IF(XI(I).LT.X(IA,I).AND.IA.EQ.1) GO TO 101	MOM01110
	IF(XI(I).GT.X(IA,I).AND.IA.EQ.NTC(I)) GO TO 101	MOM01120
	ISAV=II(I)	MOM01130
	II(I)=IA + IFIX(SIGN(1.,XI(I)-X(IA,I)))	MOM01140
	IT=0	MOM01150
	DO 102 JI=1,2	MOM01160
	JIX=JI + II(1) - 1	MOM01170
	DO 102 IJ=1,2	MOM01180
	IJX=JIX + (IJ + II(2) - 2)*NZ	MOM01190
	DO 102 K=1,2	MOM01200
	KX=K-1 + II(3)	MOM01210
	DO 102 L=1,2	MOM01220
	LX=L-1 + II(4)	MOM01230
	IF(XB(IJX,KX,LX).GT.-100.) IT=IT+1	MOM01240
102	CONTINUE	MOM01250
	IF(IT.GT.ITC) GO TO 6	MOM01260
	II(I)=ISAV	MOM01270
101	CONTINUE	MOM01280
	PERFORM THE INTERPOLATION WITH DETERMINED CUBE OF POINTS	MOM01290
	DO 103 I=1,4	MOM01300
	I2=I*2	MOM01310
	I1=I2-1	MOM01320
		MOM01330
		MOM01340
		MOM01350
		MOM01360
		MOM01370
		MOM01380
		MOM01390
		MOM01400

	IA=II(I)	MOM01410
	XVAL(I1)=X(IA,I)	MOM01420
	XVAL(I2)=X(IA+1,I)	MOM01430
103	CONTINUE	MOM01440
	CALL DTERPS(II,XB,VAL,NZ)	MOM01450
	XBAR=DTERPI(4,XI,XVAL,VAL,-100.,W)	MOM01460
	CALL DTERPS(II,SW,VAL,NZ)	MOM01470
	SIGW2=DTERPI(-4,XI,XVAL,VAL,-100.,W)	MOM01480
	CALL DTERPS(II,SP,VAL,NZ)	MOM01490
	SIGP2=DTERPI(-4,XI,XVAL,VAL,-100.,W)	MOM01500
C		MOM01510
C	CONVERT THE LOG OF THE NONDIMENSIONAL VALUES INTERPOLATED	MOM01520
C	TO THE USUAL DIMENSIONAL FORM	MOM01530
	SCL=U0*H**(M+1.)/SCLU	MOM01540
	XBAR=SCL*EXP(XBAR)	MOM01550
	SIGW2=SCL*SCL*EXP(SIGW2)	MOM01560
	SIGP2=2.*DX20*H*H*EXP(SIGP2)	MOM01570
	SIGW2=SIGW2+SIGP2	MOM01580
	SIGP2=DYX0*SIGP2	MOM01590
999	RETURN	MOM01600
	END	MOM01610
		MOM01620

SUBROUTINE PATH(T,U,XCEN,YCEN,RAD,PLEN)

THIS SUBROUTINE COMPUTES THE PATH LENGTH THROUGH THE SPHERE OR WAKE FOR A HORIZONTAL PATH.

INPUTS

- T - TRANSMITTER COORDINATE IN THE LOCAL COORDINATE SYSTEM
- U - UNIT VECTOR ALONG THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
- XCEN - X COORDINATE OF THE CENTER OF THE CIRCLE
- YCEN - Y COORDINATE OF THE CENTER OF THE CIRCLE
- RAD - RADIUS AT THE DESIRED HEIGHT

OUTPUT

- PLEN - LENGTH OF THE INTERSECTION OF THE CONE AT HEIGHT T(3) AND THE LINE OF SIGHT

FUNCTIONS AND SUBROUTINES NEEDED

NONE

PATH0290
 PATH0010
 PATH0020
 PATH0030
 PATH0040
 PATH0050
 PATH0060
 PATH0070
 PATH0080
 PATH0090
 PATH0100
 PATH0110
 PATH0120
 PATH0130
 PATH0140
 PATH0150
 PATH0160
 PATH0170
 PATH0180
 PATH0190
 PATH0200
 PATH0210
 PATH0220
 PATH0230
 PATH0240
 PATH0250
 PATH0260
 PATH0270
 PATH0280
 PATH0300
 PATH0310

PATH0400

```

*****
DIMENSION T(3),U(3)
A=U(1)**2+U(2)**2
PLEN=0.
X=RAD**2*A-(U(2)*(T(1)-XCEN)-U(1)*(T(2)-YCEN))**2
IF (X.GT.0.) PLEN=2.*SQRT(X)/A
RETURN
END
  
```

```
SUBROUTINE PERP(A,B)
DIMENSION A(2),B(2)
C *** B IS ROTATED 90 DEGREES COUNTERCLOCKWISE FROM A
B(1)=-A(2)
B(2)=A(1)
RETURN
END
```

```
PER00010
PER00020
PER00030
PER00040
PER00050
PER00060
PER00070
```



```

ZREF=2.0
ROH2=SUM2/NDSCS
10 ICOUNT=ICOUNT+1
DO 50 I=1,3
T(I)=T(3)+FLOAT(I-1)*TINC
CCC
FIND THE AVERAGE OF THE MOMENTS AT HEIGHT ZREF
CALL AVRQ(ZREF,T(I),QTOT,XBAVRG,SIG2X,SIG2Y)
IF(QTOT.LT.1.E-10)GO TO 15
SIGX=SQRT(SIG2X+ROH2/2.)
SIGY=SQRT(SIG2Y+ROH2/2.)
FRB(I)=SQRT(SIGX*SIGY)*1.5
XB(I)=XBAVRG
GO TO 20
15 CALL WINK(ZREF,UX,V)
XB(I)=UX*T(I)
FRB(I)=0.0
20 IF(T(I).GT.TWIND)GO TO 30
CALL RISE(TPRES,T(I),DEL)
FZ2(I)=ZCM+(2./3.)*RSPH
IF(FZ2(I).GT.ZINV)FZ2(I)=ZINV
FRT(I)=RSPH
XB(2)=XCM
GO TO 40
30 XB(2)=XTR+VTR*(T(I)-TTR)
SIGZ2=SIG02+2.*KX*(T(I)-TTR)
SIGZ2=SIG02+2.*KZ*(T(I)-TTR)
SIGX=SQRT(SIGX2)
SIGZ=SQRT(SIGZ2)
SIG=SQRT(SIGX*SIGZ)
FZ2(I)=ZTR+SIG
FRT(I)=1.5*SIG
40 FXC1(I)=(XB(2)-XB(1))/((FZ2(I)-ZREF)
FXC0(I)=XB(1)-FXC1(I)*ZREF
50 CONTINUE
CCC
COMPUT AND STORE QUADRATIC FITS
TIMES(ICOUNT)=T(3)
CCC
FIT AND STORE RADIUS AT TOP
CALL FIT(T,FRT,A,B,C)
RT(1,ICOUNT)=A
RT(2,ICOUNT)=B
RT(3,ICOUNT)=C
CCC
FIT AND STORE RADIUS AT BOTTOM
CALL FIT(T,FRB,A,B,C)
RB(1,ICOUNT)=A
RB(2,ICOUNT)=B
RB(3,ICOUNT)=C
CCC
FIT AND STORE HEIGHT OF CLOUD
CALL FIT(T,FZ2,A,B,C)
Z2(1,ICOUNT)=A
Z2(2,ICOUNT)=B
Z2(3,ICOUNT)=C
CCC
FIT AND STORE XC1
CALL FIT(T,FXC1,A,B,C)
XC1(1,ICOUNT)=A
XC1(2,ICOUNT)=B
XC1(3,ICOUNT)=C
CCC
FIT AND STORE XC0
PRE00710
PRE00720
PRE00730
PRE00740
PRE00750
PRE00760
PRE00770
PRE00780
PRE00790
PRE00800
PRE00810
PRE00820
PRE00830
PRE00840
PRE00850
PRE00860
PRE00870
PRE00880
PRE00890
PRE00900
PRE00910
PRE00920
PRE00930
PRE00940
PRE00950
PRE00960
PRE00970
PRE00980
PRE00990
PRE01000
PRE01010
PRE01020
PRE01030
PRE01040
PRE01050
PRE01060
PRE01070
PRE01080
PRE01090
PRE01100
PRE01110
PRE01120
PRE01130
PRE01140
PRE01150
PRE01160
PRE01170
PRE01180
PRE01190
PRE01200
PRE01210
PRE01220
PRE01230
PRE01240
PRE01250
PRE01260
PRE01270
PRE01280
PRE01290
PRE01300
PRE01310
PRE01320
PRE01330
PRE01340
PRE01350
PRE01360
PRE01370
PRE01380
PRE01390
PRE01400

```

C	CALL FIT(T,FXC0,A,B,C)	PRE 01410
	XC0(1,ICOUNT)=A	PRE 01420
	XC0(2,ICOUNT)=B	PRE 01430
	XC0(3,ICOUNT)=C	PRE 01440
C C C	CHECK TO SEE IF PRECOMPUTE CAN BE STOPPED	PRE 01450
	IF(FRT(3).GE.FRB(3))PLEN=2.*FRT(3)	PRE 01460
	IF(FRB(3).GE.FRT(3))PLEN=2.*FRB(3)	PRE 01470
	VOL=(PI/3.)*(FZ2(3)-ZREF)*(FRT(3)**2+FRT(3)*FRB(3)+FRB(3)**2)	PRE 01480
	NWL=1	PRE 01490
	ACL=(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL))*VLOAD*PLEN/VOL	PRE 01500
	TINC=1.2*TINC	PRE 01510
	IF(ICOUNT.LT.25.AND.ACL.GT..001)GO TO 10	PRE 01520
C C C	WRITE PRECOMPUTED CLOUD INFORMATION ONTO FORTRAN UNIT IFILE WITH A BINARY WRITE	PRE 01530
	WRITE(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)	PRE 01540
	WRITE(IFILE)DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV	PRE 01550
	WRITE(IFILE)VLOAD,RCARB1,RCARB2	PRE 01560
	WRITE(IFILE)ICOUNT	PRE 01570
	DO 1000 J=1,ICOUNT	PRE 01580
	WRITE(IFILE) TIMES(J),(RT(I,J),RB(I,J),Z2(I,J),XC0(I,J),	PRE 01590
	XC1(I,J),I=1,3)	PRE 01600
1000	CONTINUE	PRE 01610
	REWIND IFILE	PRE 01620
999	RETURN	PRE 01630
	END	PRE 01640
		PRE 01650
		PRE 01660
		PRE 01670
		PRE 01680
		PRE 01690
		PRE 01700


```

SUBROUTINE PRETRN(TRN,REC,TIME,TRNLOS)
COMPUTE THE TRANSMITTANCE FOR THE RANDOM IN SPACE AND TIME
DISTRIBUTION OF CHARGES
INPUTS
  TRN  -TRANSMITTER COORDINATES IN LOCAL COORDINATE SYSTEM
  REC  -RECEIVER COORDINATES IN LOCAL COORDINATE SYSTEM
  TIME -TIME AT WHICH TRANSMITTANCE IS DESIRED
  NWL  -INTEGER INDEX FOR WAVELENGTH
  NSOIL -SOIL TYPE
OUTPUTS
  TRNLOS -TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
*****
LOGICAL TEST
DIMENSION TRN(3),REC(3),TR(3),RE(3),OWF(5,2),U(3),OWFC(5)
COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),MTOT,NARY,ITOT,
+      COOR(2,200),TSTAG(200),
+      ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
+      RB(3,25),Z2(3,25)
COMMON/VL/VLOAD
COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
COMMON/CARB/RCARB1,RCARB2
COMMON /CONST/PI,P12,PIRAD,TWOPI,TORRMB,CDEGK
DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
DATA OWFC/1.,.95,.5,.2,1.E-03/
PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
  TEST=.FALSE.
  XNORM=0.0
  DO 10 I=1,3
  RE(I)=REC(I)
  TR(I)=TRN(I)
  U(I)=RE(I)-TR(I)
  XNORM=XNORM+U(I)**2
10 CONTINUE
  XNORM=SQRT(XNORM)
  U(1)=U(1)/XNORM
  U(2)=U(2)/XNORM
  U(3)=U(3)/XNORM
COMPUTE THE CONTRIBUTION FROM EACH CHARGE TO THE OPTICALLY
WEIGHTED CONCENTRATION ALONG THE LINE OF SIGHT
  SUM=0.0
  DO 100 I=1,ITOT
  IF(TIME.LT.TSTAG(I))GO TO 100
  TOF=TIME-TSTAG(I)
  IF(TOF.GT.TIMES(ICOUNT))GO TO 100
  DO 20 J=1,ICOUNT
  IND=J
  IF(TOF.LE.TIMES(J))GO TO 30
20 CONTINUE
DETERMINE NECESSARY PARAMETERS DESCRIBING THE CONICAL SHAPE SO THAT
THE LENGTH OF INTERSECTION OF THE LINE OF SIGHT AND CONE CAN BE
DETERMINED
30 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)
  X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)
  HTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)

```

```

PRT00220
PRT00010
PRT00020
PRT00030
PRT00040
PRT00050
PRT00060
PRT00070
PRT00080
PRT00090
PRT00100
PRT00110
PRT00120
PRT00130
PRT00140
PRT00150
PRT00160
PRT00170
PRT00180
PRT00190
PRT00200
PRT00210
PRT00230
PRT00240
PRT00250
PRT00260
PRT00270
PRT00280
PRT00290
PRT00300
PRT00310
PRT00320
PRT00330
PRT00340
PRT00350
PRT00360
PRT00370
PRT00380
PRT00390
PRT00400
PRT00410
PRT00420
PRT00430
PRT00440
PRT00450
PRT00460
PRT00470
PRT00480
PRT00490
PRT00500
PRT00510
PRT00520
PRT00530
PRT00540
PRT00550
PRT00560
PRT00570
PRT00580
PRT00590
PRT00600
PRT00610
PRT00620
PRT00630
PRT00640
PRT00650
PRT00660
PRT00670
PRT00680
PRT00690
PRT00700

```

	RTOP=TOF*(RT<1,IND>*TOF+RT<2,IND>)+RT<3,IND>	PRT00710
	HTBOT=0.0	PRT00720
	RBOT=TOF*(RB<1,IND>*TOF+RB<2,IND>)+RB<3,IND>	PRT00730
	XCEN=DIFF<1,I>+(X1*HTTOP+X0)	PRT00740
	YCEN=DIFF<2,I>	PRT00750
	XB=DIFF<1,I>+(X1*HTBOT+X0)	PRT00760
	YB=DIFF<2,I>	PRT00770
	IF<ABS<U<3>>.LT.1.E-06>GO TO 40	PRT00780
C	COMPUTE THE INTERSECTION LENGTH FOR A NON-HORIZONTAL LINE OF SIGHT	PRT00790
C	CALL CONLEN<U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,	PRT00800
	1 XNORM,PLEN>	PRT00810
	GO TO 50	PRT00820
C	COMPUTE THE INTERSECTION LENGTH FOR A HORIZONTAL LINE OF SIGHT	PRT00830
C	40 IF<HTTOP.LT,TR<3>>GO TO 45	PRT00840
	A=U<1>***2+U<2>***2	PRT00850
	B=U<1>*(TR<1>-XCEN)+U<2>*(TR<2>-YCEN)	PRT00860
C	DETERMINE THE RADIUS,X,AND Y POSITIONS OF THE CONE AT THE	PRT00870
C	TRANSMITTER HEIGHT	PRT00880
	ZETA=TR<3>/HTTOP	PRT00890
	RAD=ZETA*RTOP+(1.-ZETA)*RBOT	PRT00900
	XCEN=ZETA*XCEN+(1.-ZETA)*XB	PRT00910
	YCEN=ZETA*YCEN+(1.-ZETA)*YB	PRT00920
	A=U<1>***2+U<2>***2	PRT00930
	B=U<1>*(TR<1>-XCEN)+U<2>*(TR<2>-YCEN)	PRT00940
	C=(TR<1>-XCEN)**2+(TR<2>-YCEN)**2-RAD**2	PRT00950
	X=B**2-A*C	PRT00960
	IF<X.LT.0.0>GO TO 45	PRT00970
	P1=(-B+SQRT(X))/A	PRT00980
	P2=(-B-SQRT(X))/A	PRT00990
	IF<P1.GT.XNORM.AND.P2.GT.XNORM>GO TO 45	PRT01000
	IF<P2.LT.0.0.AND.P1.LT.0.0>GO TO 45	PRT01010
	PLEN=AMIN1<P1,XNORM>-AMAX1<P2,0.0>	PRT01020
	GO TO 50	PRT01030
45	PLEN=0.0	PRT01040
50	VOL=(PI/3.)*<HTTOP-HTBOT>*(RTOP**2+RTOP*RBOT+RBOT**2)	PRT01050
	CONT=VLOAD*PLEN/VOL	PRT01060
	SUM=SUM+CONT	PRT01070
	ACLSKT=0.0	PRT01080
	ACLSPH=0.0	PRT01090
	CALL TRNCHK<ACLSKT,SUM,ACLSPH>	PRT01100
	IF<TEST>GO TO 998	PRT01110
100	CONTINUE	PRT01120
	TRNLOS=EXP<-SUM*(RCARB1*OWF<NWL,NSOIL>+RCARB2*OWFC<NWL>>>	PRT01130
	GO TO 999	PRT01140
998	TRNLOS=0.0	PRT01150
999	RETURN	PRT01160
	END	PRT01170
		PRT01180
		PRT01190
		PRT01200
		PRT01210
		PRT01220
		PRT01230


```

C COMPUTE R , Z2 , XC1 , XC0 AT THREE CONSECUTIVE TIMES WITH SPACING PRV00710
C TINC AND THEN CALL FIT , WHICH CALCULATES A QUADRATIC FIT TO THESE PRV00720
C POINTS AND STORE THEM IN COMMON /QUADFT/ PRV00730
C PRV00740
C TINC=1.0 PRV00750
C ICOUNT=0 PRV00760
C T(3)=0.0 PRV00770
10 ICOUNT=ICOUNT+1 PRV00780
DO 20 I=1,3 PRV00790
T(I)=T(3)+FLOAT(I-1)*TINC PRV00800
ZREF=ZT0+SQRT(T(I))*RT2DZ PRV00810
TOF=T(I)-TDSC(NDSCS) PRV00820
CALL MOMENT(VGRAV,ZREF,ZDSC(I),TOF,Q,XBAR,SIGW2,SIGP2) PRV00830
C COMPUTE R THE RADIUS OF THE CLOUD PRV00840
C PRV00850
C ROH2=R2DSC(I) PRV00860
C SIGW=SQRT(SIGW2+ROH2/2.) PRV00870
C SIGP=SQRT(SIGP2+ROH2/2.) PRV00880
C FR(I)=1.5*SQRT(SIGW*SIGP) PRV00890
C ACL=CWIND(X,Y,ZREF,T(I))*OWF(NWL,NSOIL) PRV00900
C COMPUTE Z2 APPROXIMATE HEIGHT OF THE CLOUD PRV00910
C PRV00920
C FZ2(I)=(2.*QDSC(1,I))/PI/FR(I)/ACL PRV00930
C Z22=FZ2(I) PRV00940
C PRV00950
C COMPUTE THE X POSITION OF THE CLOUD AT A HEIGHT OF Z2 AND A HEIGHT PRV00960
C OF 1 METER. PRV00970
C CALL MOMENT(VGRAV,Z22,ZDSC(I),TOF,Q,XBAR,SIGW2,SIGP2) PRV00980
C XB(2)=XBAR+XDSC(1) PRV00990
C Z1=1.0 PRV01000
C CALL MOMENT(VGRAV,Z1,ZDSC(I),TOF,Q,XBAR,SIGW2,SIGP2) PRV01010
C XB(1)=XBAR+XDSC(1) PRV01020
C FXC(I)=(XB(2)-XB(1))/((Z22-Z1)) PRV01030
C FXC0(I)=XB(1)-FXC(I) PRV01040
20 CONTINUE PRV01050
C COMPUTE AND STORE THE QUADRATIC FITS PRV01060
C PRV01070
C TIMES(ICOUNT)=T(3) PRV01080
C PRV01090
C FIT AND STORE THE CLOUD RADIUS PRV01100
C PRV01110
C CALL FIT(T,FR,A,B,C) PRV01120
C RT(1,ICOUNT)=A PRV01130
C RT(2,ICOUNT)=B PRV01140
C RT(3,ICOUNT)=C PRV01150
C FIT AND STORE Z2, APPROXIMATE CLOUD HEIGHT PRV01160
C PRV01170
C CALL FIT(T,FZ2,A,B,C) PRV01180
C Z2(1,ICOUNT)=A PRV01190
C Z2(2,ICOUNT)=B PRV01200
C Z2(3,ICOUNT)=C PRV01210
C FIT AND STORE XC1 PRV01220
C PRV01230
C CALL FIT(T,FXC1,A,B,C) PRV01240
C XC1(1,ICOUNT)=A PRV01250
C XC1(2,ICOUNT)=B PRV01260
C XC1(3,ICOUNT)=C PRV01270
C FIT AND STORE XC0 PRV01280
C PRV01290
C CALL FIT(T,FXC0,A,B,C) PRV01300
C XC0(1,ICOUNT)=A PRV01310
C XC0(2,ICOUNT)=B PRV01320
C XC0(3,ICOUNT)=C PRV01330
C PRV01340
C PRV01350
C PRV01360
C PRV01370
C PRV01380
C PRV01390
C PRV01400

```

```
TINC=1.2*TINC  
IF(ICOUNT.LT.20.AND.T(3).LT.TMAX) GO TO 10  
RETURN  
END
```

```
PRV01410  
PRV01420  
PRV01430  
PRV01440
```

```

SUBROUTINE RISE(TPRES,TNEXT,DEL)
REAL M,NDIF,KZ,KX
LOGICAL SWITCH,CHANGE
DIMENSION WK(12,6)
COMMON/BUOYCL/ Y(8),SPHNS(3),RISTIM
COMMON/WNDPRM/ DXZ0,DYX0,DZ0,U0,M,NDIF,ZINV
COMMON/CLOCK/ TIME,TWIND
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/EKTEMP/20,ZL,T0,TC1,TC2,TC3
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
1QDSC(20,3)
COMMON/BURST/ACCEL,TBURST
DATA HMIN,ACCURC,WK,N,ND/.001,.001,72*0.,.8,12/
*****

PURPOSE
THIS ROUTINE CALLS A RUNGA KUTTA ROUTINE TO INTEGRATE IN TIME
THE EQUATIONS FOR THE RISE OF A BUOYANT CLOUD BEGINNING AT TPRES
ENDING AT TNEXT UNLESS THE CONDITION FOR SWITCHING TO THE WIND
DISPERSION MODEL IS ENCOUNTERED IN WHICH CONVRT IS CALLED.
SEE SUBROUTINE DIFEQ FOR THE DEFINITIONS OF Y(I).

ARGUMENTS
TPRES AS INPUT TPRES IS THE INITIAL TIME OF THIS SEGMENT OF
INTEGRATION AND IS RETURNED WITH THE VALUE OF THE LAST
SUCCESSFUL INTEGRATION STEP.
TNEXT THE ENDPOINT OF THE TIME INTERVAL WHICH IS INPUT.

REQUIRED SUBROUTINES
RKM A RUNGA-KUTTA-MERSON INTEGRATION ROUTINE
CONVRT A SUBROUTINE WHICH CONVERTS THE CURRENT BUOYANT
DUST CLOUD TO A NUMBER OF DISC SOURCES FOR THE
WIND DISPERSION MODEL. A GAP TIME DURING WHICH THE
BUOYANT MODEL IS CONTINUED IS COMPUTED.
WINDCAL COMPUTES SCALED WIND SPEED AT A SPECIFIED HEIGHT
DIFFUS COMPUTES DIFFUSIVITY AT A SPECIFIED HEIGHT

CALLED BY DUSTCL
*****
IF(TNEXT.GT.TWIND)GO TO 999
SWITCH=.FALSE.
CHANGE=.FALSE.
T2=TPRES

PERFORM INTEGRATION IN SEGMENTS OF TIME
10 DO 20 NT=1,300
T1=T2
T2=1.2*T1
IF(T2.LE.0.)T2=.5
IF(T2.GT.TNEXT)T2=TNEXT
IF(DEL.LT.HMIN)DEL=HMIN
CALL RKM(N,T1,T2,Y,HMIN,DEL,ACCURC,WK,ND)

CHECK TO SEE IF CLOUD GROWTH IS DOMINATED BY WIND DIFFUSION
OVER BUOYANT RISE BY COMPARING WIND DIFFUSIVITY, DIFW, TO
THE EFFECTIVE BUOYANT DIFFUSIVITY, DIFB AND IF THE HEIGHT
OF THE CENTER OF MASS IS LESS THAN ZSTAR SWITCH TO THE

```

```

RIS00010
RIS00020
RIS00030
RIS00040
RIS00050
RIS00060
RIS00070
RIS00080
RIS00090
RIS00100
RIS00110
RIS00120
RIS00130
RIS00140
RIS00150
RIS00160
RIS00170
RIS00180
RIS00190
RIS00200
RIS00210
RIS00220
RIS00230
RIS00240
RIS00250
RIS00260
RIS00270
RIS00280
RIS00290
RIS00300
RIS00310
RIS00320
RIS00330
RIS00340
RIS00350
RIS00360
RIS00370
RIS00380
RIS00390
RIS00400
RIS00410
RIS00420
RIS00430
RIS00440
RIS00450
RIS00460
RIS00470
RIS00480
RIS00490
RIS00500
RIS00510
RIS00520
RIS00530
RIS00540
RIS00550
RIS00560
RIS00570
RIS00580
RIS00590
RIS00600
RIS00610
RIS00620
RIS00630
RIS00640
RIS00650
RIS00660
RIS00670
RIS00680
RIS00690
RIS00700

```

C
C

WIND MODEL.

```
5 IF(T2.LT.TBURST)GO TO 15
  DIFB=ABS(.1111*Y(1)*Y(3))
  TOP=Y(6)+Y(1)
  DIFW=DIFFUS(Z0,ZL, TOP)
  IF(DIFB.GT.DIFW)GO TO 15
  IF(TOP.GT.ZSTAR)SWITCH=.TRUE.
  IF(TOP.GT.ZSTAR)GO TO 15
  VTR=USTAR*WINDCAL(Z0,ZL,Y(6))
  KZ=DIFFUS(Z0,ZL,Y(6))
  KX=DXZ0*KZ
  TTR=T2
  XTR=Y(4)
  ZTR=Y(6)
  CHANGE=.TRUE.
  CALL CONVRT(T2)
  GO TO 200
15 CONTINUE
  IF(T2.GE.TNEXT)GO TO 200
  IF(T2.GT.300.)GO TO 99
20 CONTINUE
99 WRITE(IDOUT,98)
98 FORMAT(54H *** DIRTRAN ERROR - 5 MINUTE CUT-OFF ON BUOYANT RISE )
  STOP
200 TPRES=T2
  RISTIM=TPRES
999 RETURN
  END
```

R1600710
R1600720
R1600730
R1600740
R1600750
R1600760
R1600770
R1600780
R1600790
R1600800
R1600810
R1600820
R1600830
R1600840
R1600850
R1600860
R1600870
R1600880
R1600890
R1600900
R1600910
R1600920
R1600930
R1600940
R1600950
R1600960
R1600970
R1600980
R1600990

```

SUBROUTINE RKM(N,XL,XU,Y,HMIN,DEL,ACCURC,WK,ND)
NUMERICAL INTEGRATION ROUTINE FOR SYSTEMS OF ODE'S
  USING THE RUNGE-KUTTA-MERSON TECHNIQUE

INPUT PARAMETERS
  N - NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS
  XL - INITIAL ABSCISSA OF THE INTERVAL
  XU - THE FINAL ABSCISSA OF THE INTEGRATION INTERVAL
  Y - A SINGLY DIMENSIONED ARRAY OF LENGTH N. WHEN
      RKM IS CALLED IT MUST CONTAIN THE VALUES OF
      THE DEPENDENT VARIABLES AT XL. UPON RETURN
      TO THE CALLING PROGRAM Y CONTAINS THE VALUES
      OF THE DEPENDENT VARIABLES AT XU.
  HMIN - THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE
      INTEGRATION
  DEL - THE INITIAL ESTIMATE OF THE STEP SIZE AND UPON
      RETURN TO THE CALLING PROGRAM DEL CONTAINS THE
      FINAL STEP SIZE USED. THIS VALUE SHOULD BE USED
      IN THE NEXT CALL TO PRODUCE AN EFFICIENT INTEGRATION.
      DEL IS RETURNED WITH THE VALUE ZERO IF IT HAS
      BEEN HALVED BELOW HMIN.
  ACCURC - PREASSIGNED ACCURACY WHICH IS ALSO USED IN ADJUSTING
      THE STEP SIZE.
  WK - AT LEAST A BLOCK OF N BY 6 FLOATING POINT LOCATIONS
      USED FOR A WORK ARRAY.
  ND - THE DIMENSION OF ARRAYS Y AND WK.

IT IS REQUIRED THAT THE USER OF RKM WRITE A SUBROUTINE
DEFINING THE DIFFERENTIAL EQUATIONS. THE SUBROUTINE
STATEMENT SHOULD LOOK LIKE - SUBROUTINE DIFEQ(N,X,Y,YP) .

WHERE
  N - THE NUMBER OF EQUATIONS
  X - THE INDEPENDENT VARIABLE
  Y - SINGLY DIMENSIONED ARRAY OF DEPENDENT VARIABLES
  YP - SINGLY DIMENSIONED ARRAY OF THE RATES OF Y AT X
      YP(I) = D Y(I)/DX

DIMENSION Y(ND),WK(ND,6)
LOGICAL FIRST,QUIT

SET UP NEEDED VARIABLES UPON ENTRY
  XN=XL
  H=DEL
  FIRST=.TRUE.
  QUIT=.FALSE.

CHECK IF XN IS CLOSE TO XU
20 IF(XN+H .LT. XU) GO TO 30
  DEL=H
  H=XU-XN
  QUIT=.TRUE.
  IF(FIRST) DEL=H

MAKE FIRST CALL TO DIFEQ AT THE BEGINNING OF INTERVAL
30 CALL DIFEQ(N,XN,Y,WK(1,1))

PERFORM THE RUNGE-KUTTA-MERSON ALGORITHM
40 H3=H/3.
  DO 50 I=1,N

```

```

RKM00010
RKM00020
RKM00030
RKM00040
RKM00050
RKM00060
RKM00070
RKM00080
RKM00090
RKM00100
RKM00110
RKM00120
RKM00130
RKM00140
RKM00150
RKM00160
RKM00170
RKM00180
RKM00190
RKM00200
RKM00210
RKM00220
RKM00230
RKM00240
RKM00250
RKM00260
RKM00270
RKM00280
RKM00290
RKM00300
RKM00310
RKM00320
RKM00330
RKM00340
RKM00350
RKM00360
RKM00370
RKM00380
RKM00390
RKM00400
RKM00410
RKM00420
RKM00430
RKM00440
RKM00450
RKM00460
RKM00470
RKM00480
RKM00490
RKM00500
RKM00510
RKM00520
RKM00530
RKM00540
RKM00550
RKM00560
RKM00570
RKM00580
RKM00590
RKM00600
RKM00610
RKM00620
RKM00630
RKM00640
RKM00650
RKM00660
RKM00670
RKM00680
RKM00690
RKM00700

```


	WK(I,3)=H3*WK(I,1)	RKM00710
50	WK(I,6)=Y(I)+WK(I,3)	RKM00720
	CALL DIFEQ(N,XN+H3,WK(I,6),WK(I,2))	RKM00730
	DO 60 I=1,N	RKM00740
60	WK(I,6)=Y(I)+(WK(I,3)+H3*WK(I,2))/2.	RKM00750
	CALL DIFEQ(N,XN+H3,WK(I,6),WK(I,2))	RKM00760
	DO 70 I=1,N	RKM00770
	WK(I,4)=H3*WK(I,2)	RKM00780
70	WK(I,6)=Y(I)+(3.*WK(I,3)+9.*WK(I,4))/8.	RKM00790
	CALL DIFEQ(N,XN+H/2.,WK(I,6),WK(I,2))	RKM00800
	DO 80 I=1,N	RKM00810
	WK(I,5)=H3*WK(I,2)	RKM00820
80	WK(I,6)=Y(I)+(3.*WK(I,3)-9.*WK(I,4)+12.*WK(I,5))/2.	RKM00830
	CALL DIFEQ(N,XN+H,WK(I,6),WK(I,2))	RKM00840
CCCC	FIND THE LARGEST RELATIVE ERROR	RKM00850
	TEST=0.	RKM00860
	DO 90 I=1,N	RKM00870
	YX=Y(I)	RKM00880
	IF(YX.EQ. 0.) YX=ACCURC	RKM00890
	E=((WK(I,3)-9.*WK(I,4))/2.+4.*WK(I,5)-H3*WK(I,2))/2.)/5.)/YX	RKM00900
90	TEST=AMAX1(TEST,ABS(E))	RKM00910
	FIRST=.FALSE.	RKM00920
	IF(TEST.LT. ACCURC) GO TO 100	RKM00930
CCCC	IF THE LARGEST ERROR IS GREATER THAN ACCURC HALF THE STEP SIZE AND TRY AGAIN.	RKM00940
	H=H/2.	RKM00950
	IF(H.LT. HMIN) GO TO 10	RKM00960
	QUIT=.FALSE.	RKM00970
	GO TO 40	RKM00980
CCCC	TRUNCATION ERROR LESS THAN ACCURC, RESET THE Y ARRAY TO SET UP FOR THE NEXT INTERVAL	RKM00990
100	XN=XN+H	RKM01000
	DO 110 I=1,N	RKM01010
110	Y(I)=Y(I)+(WK(I,3)+4.*WK(I,5)+H3*WK(I,2))/2.	RKM01020
CCCC	CHECK FOR STEP SIZE DOUBLING. DOUBLE IF LARGEST RELATIVE ERROR IS 32 TIMES LESS THAN ACCURC.	RKM01030
	IF(.NOT.(TEST.GE. ACCURC/32. .OR. QUIT)) H=H+H	RKM01040
	IF(.NOT. QUIT) GO TO 20	RKM01050
	RETURN	RKM01060
CCCC	THE VALUE OF H (DEL) IS LESS THAN THE SPECIFIED MINIMUM. REPORT THIS AND ERROR OUT.	RKM01070
10	CONTINUE	RKM01080
1000	FORMAT('1 H BELOW HMIN'/'0 INTEGRATION ABORTED')	RKM01090
	DEL=0.	RKM01100
	RETURN	RKM01110
	END	RKM01120
		RKM01130
		RKM01140
		RKM01150
		RKM01160
		RKM01170
		RKM01180
		RKM01190
		RKM01200
		RKM01210
		RKM01220
		RKM01230
		RKM01240
		RKM01250
		RKM01260

```

SUBROUTINE SETUP(NCHS, SRCBAS, SIDE1, SIDE2, TRNFRM)          SET 00010
DIMENSION NCHS(2), SRCBAS(2), SIDE1(2), SIDE2(2), TRNFRM(2,2), REF(2) SET 00020
COMMON /ARRAY/OVRLAP, AREA, PERIM, PRJARY, CENDIF           SET 00030
COMMON/MOS/DIFF(2,200), NCHTOT, PRSEP(200), NTOT, NARY, ITOT, SET 00040
+ COOR(2,200), TSTAG(200), DMMY(401)                       SET 00050
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU SET 00060
*****                                                    SET 00070
C*****                                                    SET 00080
C*****                                                    SET 00090
C*****                                                    SET 00100
PURPOSE                                                    SET 00110
    TO CONVERT THE USER DEFINED COORDINATES OF THE CHARGES TO THE SET 00120
    LOCAL COORDINATE SYSTEM.                               SET 00130
C*****                                                    SET 00140
INPUTS                                                    SET 00150
    NCHS - SINGLY DIMENSIONED ARRAY CONTAINING THE NUMBER OF SET 00160
    CHARGES.                                               SET 00170
    SRCBAS - A REFERENCE CHARGE IN THE USER DEFINED COORDINATES. SET 00180
    SIDE1, SIDE2 - VECTORS DESCRIBING THE BOUNDING PARRALLELOGRAM. SET 00190
    TRNFRM - COORDINATE SYSTEM TRANSFORMATION MATRIX.     SET 00200
C*****                                                    SET 00210
OUTPUTS RETURNED IN COMMON /ARRAY/ AND /SEPRTN/          SET 00220
    DIFF- DOUBLY DIMENSIONED ARRAY CONTAINING THE CHARGE COORDINATES SET 00230
    IN THE LOCAL COORDINATE SYSTEM.                       SET 00240
    ITOT- TOTAL NUMBER OF CHARGES.                        SET 00250
    NCHTOT- WHEN NARY=1 OR 2 THE TOTAL NUMBER OF CHARGES. WHEN SET 00260
    WHEN NARY=3 IS SET =1.                                SET 00270
    NTOT- WHEN NARY=1 OR 2 IS SET =1 AND WHEN NARY=3 IS THE TOTAL SET 00280
    NUMBER OF CHARGES.                                    SET 00290
C*****                                                    SET 00300
SUBROUTINES AND FUNCTIONS                                 SET 00310
    UNIT COMPUTES THE UNIT VECTOR OF A GIVEN VECTOR      SET 00320
C*****                                                    SET 00330
C*****                                                    SET 00340
C*****                                                    SET 00350
C*****                                                    SET 00360
C*****                                                    SET 00370
C*****                                                    SET 00380
C*****                                                    SET 00390
C*****                                                    SET 00400
C*****                                                    SET 00410
C*****                                                    SET 00420
C*****                                                    SET 00430
C*****                                                    SET 00440
C*****                                                    SET 00450
C*****                                                    SET 00460
C*****                                                    SET 00470
C*****                                                    SET 00480
C*****                                                    SET 00490
C*****                                                    SET 00500
C*****                                                    SET 00510
C*****                                                    SET 00520
C*****                                                    SET 00530
C*****                                                    SET 00540
C*****                                                    SET 00550
C*****                                                    SET 00560
C*****                                                    SET 00570
C*****                                                    SET 00580
C*****                                                    SET 00590
C*****                                                    SET 00600
C*****                                                    SET 00610
C*****                                                    SET 00620
C*****                                                    SET 00630
C*****                                                    SET 00640
C*****                                                    SET 00650
C*****                                                    SET 00660
C*****                                                    SET 00670
C*****                                                    SET 00680
C*****                                                    SET 00690
C*****                                                    SET 00700

```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(1)
UNCLASSIFIED FEB 82 R G STEINHOFF ERADCOM/ASL-TR-0107-V2-SU NL

3-6
7
8

The table consists of a grid with 12 columns and 12 rows. The top-left cell is white and contains the text '3-6', '7', and '8' stacked vertically. The remaining 144 cells in the grid are blacked out.

	DIFF(I,J)=0.0	SET00710
30	CONTINUE	SET00720
40	CONTINUE	SET00730
	IF(NARY.GT.1)GO TO 90	SET00740
	NM=0	SET00750
	NC1=NCHS(1)	SET00760
	NC2=NCHS(2)	SET00770
CCCC	COMPUTE LOCATIONS OF CHARGES FOR INTERNAL COORDINATE SYSTEM FOR UNIFORMLY DISTRIBUTED CHARGES.	SET00780
	DO 80 M=1,NC2	SET00790
	DO 70 N=1,NC1	SET00800
	NM=NK+1	SET00810
	DO 60 I=1,2	SET00820
	DO 50 J=1,2	SET00830
	DIFF(I,NM)=DIFF(I,NM)+FLOAT(N-1)*TRNFRM(I,J)*SIDE1(J)	SET00840
	1 +FLOAT(M-1)*TRNFRM(I,J)*SIDE2(J)	SET00850
50	CONTINUE	SET00860
60	CONTINUE	SET00870
70	CONTINUE	SET00880
80	CONTINUE	SET00890
	GO TO 999	SET00900
90	CONTINUE	SET00910
CCCC	TRANSFORM CHARGE LOCATIONS TO LOCAL COORDINATE SYSTEM FOR RANDOM CHARGES.	SET00920
	NC1=NCHS(1)	SET00930
	DO 120 M=1,NC1	SET00940
	DO 110 I=1,2	SET00950
	DO 100 J=1,2	SET00960
	DIFF(I,M)=DIFF(I,M)+TRNFRM(I,J)*COOR(J,M)	SET00970
100	CONTINUE	SET00980
	DIFF(I,M)=DIFF(I,M)-REF(I)	SET00990
110	CONTINUE	SET01000
120	CONTINUE	SET01010
	GO TO 999	SET01020
998	WRITE(1000,778)	SET01030
778	FORMAT(5X,23H *** NARY OUT OF RANGE)	SET01040
999	RETURN	SET01050
	END	SET01060
		SET01070
		SET01080
		SET01090
		SET01100
		SET01110
		SET01120

```

SUBROUTINE SOURCE(W,NCHRG,DD,NSOIL,DSOD)
*****
PURPOSE
TO CALCULATE EXPLOSIVE DUST SOURCE TERM FOR THE
DIRTRAN CODE

INPUT
W      THE WEIGHT OF THE CHARGE IN KG-TNT
DD     DETONATION DEPTH IN METERS
NSOIL  INTEGER SOIL INDEX
DSOD   DEPTH OF SOD IN METERS

OUTPUT (RETURNED IN COMMON /PRTINF/ , /BUOYCL/ AND /CARB/
R0     INITIAL CLOUD RADIUS IN METERS
VGRAV  SINGLY DIMENSIONED ARRAY CONTAINING OPTICALLY WEIGHTED
        AVERAGE SETTLING VELOCITIES FOR EACH SIZE RANGE IN
        THE PARTICLE DISTRIBUTION (METERS/SEC)
NPRTS  THE NUMBER OF SIZE RANGES IN THE PARTITIONING OF THE
        PARTICLE SIZE SPECTRUM
RSPH   THE INITIAL RADIUS OF THE CLOUD IN METERS
DELT   THE INITIAL DIFFERENCE IN TEMPERATURE BETWEEN THE CLOUD
        AND SURROUNDINGS (DEGREES KELVIN)
VZSPH  THE INITIAL VERTICAL VELOCITY OF THE CLOUD (M/S)
XCMSPH INITIAL HORIZONTAL POSITION OF THE CLOUD (METERS)
YCMSPH INITIAL Y POSITION OF THE CLOUD (METERS)
ZCMSPH INITIAL HEIGHT OF THE CLOUD (METERS)
XTOP   INITIAL X POSITION OF THE TOP OF THE CLOUD (METERS)
YTOP   INITIAL Y POSITION OF THE TOP OF THE CLOUD (METERS)
RISTIM TIME LAPSED SINCE DETONATION IN SECONDS

RCARB1 PORTION OF BUOYANT CLOUD WHICH IS DIRT PARTICLES
RCARB2 PORTION OF BUOYANT CLOUD WHICH IS CARBON PARTICLES

CALLED BY DUSTCL
SUBROUTINES AND FUNCTIONS
NONE
*****
LOGICAL HORIZ,ONCE
DIMENSION CR(5,7),CD(5,7),DWML(3,4),DWSV(3,4),PRTTN(4)
DIMENSION S(3),BURHTR(5),WTRAT(5)
COMMON/PRTINF/ R0,VGRAV(3),NPRTS
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUS
COMMON/BUOYCL/ RSPH,DELT,VZSPH,XCMSPH,YCMSPH,ZCMSPH,XTOP,YTOP,
* SPHNS(3),RISTIM
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON /WINDPRM/ DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
COMMON /BURST/ ACCEL,TBURST
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON /MODE/ HORIZ
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
1 QDSC(20,3)
COMMON/CARB/RCARB1,RCARB2
COMMON/NTAL/TNOT,VOLSPH,TNO,CBLEED

CR IS THE CRATER RADIUS INDEXED BY COEFFICIENT AND SOIL TYPE
DATA CR/.271,-.684,.39,.886,0.,.271,-.684,.39,.886,0.,
1 .386,-.849,.367,.993,0.,.503,-.954,.45,1.19,0.,
2 .629,-1.08,.264,1.12,0.,.629,-1.08,.264,1.12,0.,

```

```

SORC0010
SORC0020
SORC0030
SORC0040
SORC0050
SORC0060
SORC0070
SORC0080
SORC0090
SORC0100
SORC0110
SORC0120
SORC0130
SORC0140
SORC0150
SORC0160
SORC0170
SORC0180
SORC0190
SORC0200
SORC0210
SORC0220
SORC0230
SORC0240
SORC0250
SORC0260
SORC0270
SORC0280
SORC0290
SORC0300
SORC0310
SORC0320
SORC0330
SORC0340
SORC0350
SORC0360
SORC0370
SORC0380
SORC0390
SORC0400
SORC0410
SORC0420
SORC0430
SORC0440
SORC0450
SORC0460
SORC0470
SORC0480
SORC0490
SORC0500
SORC0510
SORC0520
SORC0530
SORC0540
SORC0550
SORC0560
SORC0570
SORC0580
SORC0590
SORC0600
SORC0610
SORC0620
SORC0630
SORC0640
SORC0650
SORC0660
SORC0670
SORC0680
SORC0690
SORC0700

```

```

3 .806,-1.28,-.178,.852,0./
CD IS THE CRATER DEPTH INDEXED BY COEFFICIENT AND SOIL TYPE
DATA CD/.113,-.477,.27,1.84,1.05,.134,-.571,.343,2.24,1.31,
1 .189,-.84,.447,3.3,2.1,.251,-1.17,.494,4.72,3.34,
2 .189,-.84,.447,3.3,2.1,.331,-1.49,.579,4.92,3.13,
3 .449,-1.82,.322,4.11,2.02/
OWML IS THE OPTICALLY WEIGHTED MASS LOADING COEFFICIENT INDEXED BY
BIN SIZE AND SOIL TYPE
DATA OWML/2.88E3,2*0.,3.08E3,8*0./
OWSV IS THE OPTICALLY WEIGHTED PARTICLE SETTLING VELOCITY (CM/SEC)
INDEXED BY BIN SIZE AND SOIL TYPE
DATA OWSV/12*0./
PRTTN IS THE PARTITIONING RATIO INDEXED ON SOIL TYPE
DATA PRTTN/4*.8/
BURHTR IS THE RATIO OF BURST HEIGHT TO INITIAL RADIUS AND WTRAT
IS THE FRACTION OF THE TOTAL WEIGHT WHICH IS EFFECTIVE IN THE CLOUD
DATA BURHTR/0.,4.,2.,4.,3./,WTRAT/.6,1.,.8,1.,.7/

RISTM=0.
XCMSPH=0.
YCMSPH=0.
XTOP=0.
YTOP=0.
TNO=T0
NPRTS=1
C SCARB IS THE OPTICALLY WEIGHTED CARBON PARTICLE LOADING COEFFICIENT
SCARB=270.*W
W3=(W*WTRAT(NCHRG))*3.333333
R0=2.0*W3
TAMB=T0+TMPCAL(Z0,ZL,R0)*TSTAR
DELT=.57*TAMB
RSPH=R0
ZCMSPH=R0
BURHT=BURHTR(NCHRG)*R0
BURVZ=1.3*SQRT(R0)
TBURST=.15*R0
VZSPH=2.*BURHT/TBURST-BURVZ
ACCEL=(BURVZ-VZSPH)/TBURST
VOLSPH=(4./3.)*3.141593*R0**3
TNOT=T0+DELT
CLAM=DD/W3
CALCULATE CRATER RADIUS AND DEPTH
ONCE=.FALSE.
IF(NSOIL.EQ.1)IDX=4
IF(NSOIL.EQ.2)IDX=3
GO TO 70
60 IF(NSOIL.EQ.1)IDX=6
IF(NSOIL.EQ.2)IDX=4
70 CONTINUE
RC=CR(I,IDX)
DC=CD(I,IDX)
IF (CLAM.LT.1.E-30) GO TO 98
TERM=1.
DO 100 I=2,5
TERM=TERM*CLAM
RC=RC + CR(I,IDX)*TERM
DC=DC + CD(I,IDX)*TERM

```

```

SORC0710
SORC0720
SORC0730
SORC0740
SORC0750
SORC0760
SORC0770
SORC0780
SORC0790
SORC0800
SORC0810
SORC0820
SORC0830
SORC0840
SORC0850
SORC0860
SORC0870
SORC0880
SORC0890
SORC0900
SORC0910
SORC0920
SORC0930
SORC0940
SORC0950
SORC0960
SORC0970
SORC0980
SORC0990
SORC1000
SORC1010
SORC1020
SORC1030
SORC1040
SORC1050
SORC1060
SORC1070
SORC1080
SORC1090
SORC1100
SORC1110
SORC1120
SORC1130
SORC1140
SORC1150
SORC1160
SORC1170
SORC1180
SORC1190
SORC1200
SORC1210
SORC1220
SORC1230
SORC1240
SORC1250
SORC1260
SORC1270
SORC1280
SORC1290
SORC1300
SORC1310
SORC1320
SORC1330
SORC1340
SORC1350
SORC1360
SORC1370
SORC1380
SORC1390
SORC1400

```

100	CONTINUE	SORC1410
99	CONTINUE	SORC1420
	RC=RC*W3	SORC1430
	DC=DC*(W*WTRAT(NCHRG))**.3	SORC1440
C	GET CRATER VOLUME	SORC1450
C	DSDC=DSOD/DC	SORC1460
	VC=(2.*3.141592/3.)*RC*RC*DC*(1.-1.5*DSDC*(1.-DSDC*DSDC/3.))	SORC1470
	IF(DSOD.GE.DC)VC=0.0	SORC1480
	IF(ONCE)GO TO 110	SORC1490
	ONCE=.TRUE.	SORC1500
	IF(NSOIL.EQ.1)VC1=.5*VC	SORC1510
	IF(NSOIL.EQ.2)VC1=.25*VC	SORC1520
	GO TO 60	SORC1530
110	IF(NSOIL.EQ.1)VC=VC1+.5*VC	SORC1540
	IF(NSOIL.EQ.2)VC=VC1+.75*VC	SORC1550
C	CALCULATE OPTICALLY WEIGHTED PARAMETERS	SORC1560
C	NDSCS=MIN0(10,IFIX(5.*W3/1.8))	SORC1570
	CBLEED=0.	SORC1580
	DO 101 L=1,NPRTS	SORC1590
	S(L)=QWML(L,NSOIL) * VC	SORC1600
	VGRAY(L)=QMSV(L,NSOIL)	SORC1610
	SPHNS(L)=PRTTN(NSOIL) * S(L)	SORC1620
	QDSC(1,L)=(1.-PRTTN(NSOIL)) * S(L)/FLOAT(NDSCS)	SORC1630
	CBLEED=CBLEED+S(L)	SORC1640
101	CONTINUE	SORC1650
	CBLEED=CBLEED*.03/W3**3	SORC1660
	RCARB=SCARB/SPHNS(1)	SORC1670
	RCARB1=1./($1.+RCARB$)	SORC1680
	RCARB2=RCARB/($1.+RCARB$)	SORC1690
	SPHNS(1)=SPHNS(1)+SCARB	SORC1700
	DELH=2.*R0/FLOAT(NDSCS)	SORC1710
	Z=-DELH/2.	SORC1720
	DO 200 I=1,NDSCS	SORC1730
	Z=Z+DELH	SORC1740
	ZDSC(I)=2	SORC1750
	DO 201 J=1,NPRTS	SORC1760
	QDSC(1,J)=QDSC(1,J)	SORC1770
201	CONTINUE	SORC1780
	CON=ALOG(QDSC(1,1))/VISEXT/DELH/($2.*R0$)/3.14159)	SORC1790
	IF(CON.GT.1.)GO TO 210	SORC1800
	D=1	SORC1810
	GO TO 230	SORC1820
210	D=CON	SORC1830
	DO 220 IT=1,5	SORC1840
	D=$(CON-1.+ALOG(D))*D/(D-1)$	SORC1850
220	CONTINUE	SORC1860
230	R2DSC(I)=4.*R0*R0/D	SORC1870
	TDSC(I)=-DELH*DELH/D/($DZ0*Z**DN$)/4.	SORC1880
	SIGZ=DELH*DELH/D	SORC1890
	XDSC(I)=U0*Z**UM * TDSC(I)	SORC1900
200	CONTINUE	SORC1910
999	RETURN	SORC1920
	END	SORC1930
		SORC1940
		SORC1950
		SORC1960
		SORC1970

```

SUBROUTINE TEMP(Z,TA,DTADZ)
PURPOSE
  TO COMPUTE THE AMBIENT ATMOSPHERIC POTENTIAL TEMPERATURE AND
  GRADIENT AT A GIVEN HEIGHT.
INPUTS
  Z   HEIGHT AT WHICH AMBIENT TEMPERATURE AND TEMPERATURE
      GRADIENT ARE DESIRED.
OUTPUTS
  TA   AMBIENT POTENTIAL TEMPERATURE
  DTADZ TEMPERATURE GRADIENT
SUBROUTINES AND FUNCTIONS NEEDED
  TMPCAL  COMPUTES SCALED TEMPERATURE AT A GIVEN HEIGHT
  CALLED BY DIFEQ, ATMCAL
*****
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/EKWIND/ALP,C,PFY,PFX,UHAT,VHAT
COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
S=Z/ZL
TA=TSTAR*TMPCAL(Z0,ZL,Z)+T0
IF(ABS(ZL).LE.1.E03)GO TO 10
CCC
NEUTRAL CASE
  DTADZ=TSTAR/(Z0+Z)
  GO TO 999
10 IF(ZL.GT.0.0)GO TO 15
CCC
UNSTABLE CASE
  DTADZ=(TSTAR/Z)*(1.-16.*(S)**(-1./2.))
  IF(S.LT.-2.0)DTADZ=(TSTAR/ZL)*(AT/3.*(-ZL/Z)**(4./3.))
  GO TO 999
CCC
STABLE CASE
15 DTADZ=(TSTAR/ZL)*(ZL/(Z0+Z)+11.)
  IF(S.GT.1.5)DTADZ=BT*TSTAR/ZL
999 RETURN
END

```

```

TEMP 0240
TEMP 0010
TEMP 0020
TEMP 0030
TEMP 0040
TEMP 0050
TEMP 0060
TEMP 0070
TEMP 0080
TEMP 0090
TEMP 0100
TEMP 0110
TEMP 0120
TEMP 0130
TEMP 0140
TEMP 0150
TEMP 0160
TEMP 0170
TEMP 0180
TEMP 0190
TEMP 0200
TEMP 0210
TEMP 0220
TEMP 0230
TEMP 0250
TEMP 0260
TEMP 0270
TEMP 0280
TEMP 0290
TEMP 0300
TEMP 0310
TEMP 0320
TEMP 0330
TEMP 0340
TEMP 0350
TEMP 0360
TEMP 0370
TEMP 0380
TEMP 0390
TEMP 0400
TEMP 0410
TEMP 0420
TEMP 0430
TEMP 0440
TEMP 0450
TEMP 0460
TEMP 0470
TEMP 0480
TEMP 0490
TEMP 0500
TEMP 0510

```


C

```
S2=1.5  
BT=1./(<math>Z0/ZL+S2</math>)+11.  
DT=-1.*BT*S2  
52 CONTINUE  
IF(LOW)GO TO 999  
IF(P)53,53,54  
53 TPCAL=TPCAL+CT+AT*(<math>-ZL/Z</math>)**(<math>1./3.</math>)  
GO TO 999  
54 TPCAL=TPCAL+DT+BT*Z/ZL  
999 RETURN  
END
```

```
TMP00710  
TMP00720  
TMP00730  
TMP00740  
TMP00750  
TMP00760  
TMP00770  
TMP00780  
TMP00790  
TMP00800  
TMP00810  
TMP00820
```

```

SUBROUTINE TRAP(TRSK,TRWK,TRSP,H,SIGW,SIG0,TIME,SKT,WAK,SPH)
THIS SUBROUTINE PERFORMS A TRAPEZOID INTEGRATION
INPUTS
  TRSK - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
        TO THE CENTER OF THE SKIRT
  TRWK - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
        TO THE CENTER OF THE WAKE
  TRSP - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
        TO THE CENTER OF THE SPHERE
  H     - INTEGRATION STEP SIZE THROUGH THE SKIRT
  SIGW - INTEGRATION STEP SIZE THROUGH THE WAKE
  SIG0 - INTEGRATION STEP SIZE THROUGH THE SPHERE
  TIME  - TIME TRANSMITTANCE IS DESIRED
ALL OTHER NEEDED INFORMATION IS PASSED VIA COMMON BLOCKS
OUTPUTS
  SKT - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
        SIGHT FROM THE SKIRT
  WAK - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
        SIGHT FROM THE WAKE ONCE THE BUOYANT SPHERE HAS
        CONVERTED TO THE WIND MODEL
  SPH - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
        SIGHT FROM THE SPHERE ONCE IT HAS CONVERTED TO THE
        WIND MODEL
FUNCTIONS NEEDED
  CWIND - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
        (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SKIRT
  CWAKE - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
        (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE WAKE
  CSPHER - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
        (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SPHERE
*****
REAL KX,KZ
LOGICAL SWITCH,CHANGE,ONCE
DIMENSION OWF(5,2),OWFC(5),TRSK(3),TRWK(3),TRSP(3)
COMMON/LOS/TR(3),RE(3),U(3)
COMMON/ACL/CWINDS,CWINDC,CWINDW
COMMON/CARB/RCARB1,RCARB2
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON/POINTS/XNORM,DOT1,DOT2,DOT3
DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
DATA OWFC/1.,.95,.5,.2,1.E-03/
TI=TIME
SKT=0.0
WAK=0.0
SPH=0.0
SUM=0.0
SUM1=0.0
SUM2=0.0
IND=20
TRP00510
TRP00010
TRP00020
TRP00030
TRP00040
TRP00050
TRP00060
TRP00070
TRP00080
TRP00090
TRP00100
TRP00110
TRP00120
TRP00130
TRP00140
TRP00150
TRP00160
TRP00170
TRP00180
TRP00190
TRP00200
TRP00210
TRP00220
TRP00230
TRP00240
TRP00250
TRP00260
TRP00270
TRP00280
TRP00290
TRP00300
TRP00310
TRP00320
TRP00330
TRP00340
TRP00350
TRP00360
TRP00370
TRP00380
TRP00390
TRP00400
TRP00410
TRP00420
TRP00430
TRP00440
TRP00450
TRP00460
TRP00470
TRP00480
TRP00490
TRP00500
TRP00520
TRP00530
TRP00540
TRP00550
TRP00560
TRP00570
TRP00580
TRP00590
TRP00600
TRP00610
TRP00620
TRP00630
TRP00640
TRP00650
TRP00660
TRP00670
TRP00680
TRP00690
TRP00700

```

	ONCE=.FALSE.	TRP00710
	FP=0.0	TRP00720
	F=0.0	TRP00730
	J=0	TRP00740
	DO 10 I=1,IND	TRP00750
	XI=H*FLOAT(I-1)	TRP00760
	DIST=DOT1+XI	TRP00770
	IF(DIST.GT.XNORM)GO TO 10	TRP00780
	IF(DIST.LT.0.0)GO TO 10	TRP00790
	X=TRSK(1)+XI*U(1)	TRP00800
	Y=TRSK(2)+XI*U(2)	TRP00810
	Z=TRSK(3)+XI*U(3)	TRP00820
	FP=F	TRP00830
	ACL=CWIND(X,Y,Z,TI)	TRP00840
	F=ACL	TRP00850
C	CHECK TO SEE IF THE CONTRIBUTION IS NEGLIGABLE	TRP00860
C	IF(.NOT.ONCE)ACL1=ACL	TRP00870
	IF(ACL1.LT.1.E-10)GO TO 11	TRP00880
	PER=.01*ACL1	TRP00890
	ONCE=.TRUE.	TRP00900
	J=J+1	TRP00910
	IF(ACL.LT.PER)GO TO 11	TRP00920
	IF(J.LE.1)GO TO 10	TRP00930
	SUM=SUM+FP+F	TRP00940
10	CONTINUE	TRP00950
11	CONTINUE	TRP00960
	IF((TIME-TTR).LT.1.E-20)GO TO 31	TRP00970
		TRP00980
		TRP00990
C	COMPUTE THE CONTRIBUTION TO CL FROM THE SPHERE USING A TRAPEZOID	TRP01000
C	INTEGRATION ONLY AFTER THE BUOYANT FIREBALL HAS CONVERTED TO THE WIND	TRP01010
C	MODEL	TRP01020
	ONCE=.FALSE.	TRP01030
	FP=0.0	TRP01040
	F=0.0	TRP01050
	J=0	TRP01060
	DO 20 I=1,IND	TRP01070
	XI=SIG0*FLOAT(I-1)	TRP01080
	DIST=DOT3+XI	TRP01090
	IF(DIST.GT.XNORM)GO TO 20	TRP01100
	IF(DIST.LT.0.0)GO TO 20	TRP01110
	X=TRSP(1)+XI*U(1)	TRP01120
	Y=TRSP(2)+XI*U(2)	TRP01130
	Z=TRSP(3)+XI*U(3)	TRP01140
	FP=F	TRP01150
	ACL=CSPHER(X,Y,Z,TI)	TRP01160
	F=ACL	TRP01170
	IF(.NOT.ONCE)ACL1=ACL	TRP01180
	IF(ACL1.LT.1.E-05)GO TO 21	TRP01190
	PER=.01*ACL1	TRP01200
	ONCE=.TRUE.	TRP01210
	J=J+1	TRP01220
	IF(ACL.LT.PER)GO TO 21	TRP01230
	IF(J.LE.1)GO TO 20	TRP01240
	SUM1=SUM1+FP+F	TRP01250
20	CONTINUE	TRP01260
21	CONTINUE	TRP01270
		TRP01280
		TRP01290
C	COMPUTE CONTRIBUTION TO CL FROM THE WAKE AFTER THE BUOYANT FIREBALL	TRP01300
C	HAS CONVERTED TO THE WIND MODEL USING TRAPEZOID INTEGRATION WITH STEP	TRP01310
C	SIZE SIGW.	TRP01320
	ONCE=.FALSE.	TRP01330
	FP=0.0	TRP01340
	F=0.0	TRP01350
	J=0	TRP01360
	DO 30 I=1,IND	TRP01370
	XI=SIGW*FLOAT(I-1)	TRP01380
		TRP01390
		TRP01400

```

DIST=DOT2+XI
IF(DIST.GT.XNORM)GO TO 30
IF(DIST.LT.0.0)GO TO 30
X=TRUK(1)+XI*U(1)
Y=TRUK(2)+XI*U(2)
Z=TRUK(3)+XI*U(3)
FP=F
ACL=CWAKE(X,Y,Z,TI)
F=ACL
IF(.NOT.ONCE)ACL1=ACL
IF(ACL1.LT.1.E-05)GO TO 31
PER=.01*ACL1
ONCE=.TRUE.
J=J+1
IF(ACL.LT.PER)GO TO 31
IF(J.LE.1)GO TO 30
SUM2=SUM2+FP+F
30 CONTINUE
31 CONTINUE
SKT=(ABS(H)/2.)*SUM
WAK=(ABS(SIGW)/2.)*SUM2
SPH=(ABS(SIGO)/2.)*SUM1
999 RETURN
END

```

```

TRP 01 410
TRP 01 420
TRP 01 430
TRP 01 440
TRP 01 450
TRP 01 460
TRP 01 470
TRP 01 480
TRP 01 490
TRP 01 500
TRP 01 510
TRP 01 520
TRP 01 530
TRP 01 540
TRP 01 550
TRP 01 560
TRP 01 570
TRP 01 580
TRP 01 590
TRP 01 600
TRP 01 610
TRP 01 620
TRP 01 630
TRP 01 640

```

```

SUBROUTINE TRNCAL( TRN, REC, TIME, TRNLOS )
CONTROLLING ROUTINE FOR CALCULATING TRANSMITTANCES FOR CHARGE
DISTRIBUTION TYPES 1 AND 2
INPUTS
TRN - TRANSMITTER COORDINATES IN THE LOCAL COORDINATE SYSTEM
REC - RECEIVER COORDINATES IN THE LOCAL COORDINATE SYSTEM
TIME - TIME AFTER THE DETONATION AT WHICH A TRANSMITTANCE IS DESIRED
ALL OTHER NECESSARY INPUTS ARE PASSED IN COMMON BLOCKS
OUTPUT
TRNLOS - TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
SUBROUTINES NEEDED
AVRG - FINDS THE AVERAGE OF THE MOMENTS FOR THE DISCS
VSUM - ADDS TWO VECTORS
UNIT - DETERMINE A UNIT VECTOR
TRNCAL - DETERMINE THE LENGTH OF THE INTERSECTION OF THE LINE OF
SIGHT WITH THE WAKE AND SPHERE
TRAP - DOES A TRAPAZOIDAL INTEGRATION THROUGH SKIRT WAKE AND S
SPHERE FOR NON-HORIZONTAL LINES OF SIGHT
TRNCHK - CHECKS TO SEE IF THE OBSCURATION IS SUCH THAT THE
TRANSMITTANCE IS LESS THAN A SPECIFIED VALUE
FUNCTIONS NEEDED
DOTPRD - FINDS THE DOTPRODUCT OF TWO VECTORS
CWIND - FINDS THE CONCENTRATION ALONG A SPECIFIED HORIZONTAL
LINE OF SIGHT OR DETERMINES THE CONCENTRATION AT SOME
POINT ALONG THE LINE OF SIGHT FROM THE SKIRT
CWAKE - SAME AS CWIND EXCEPT FOR WAKE
CSPHER - SAME AS CWIND EXCEPT FOR BUOYANT SPHERE
*****
REAL KZ, KX
DIMENSION TRN(3), REC(3), OWF(5,2), OWFC(5), TEMP(2)
DIMENSION DIR(2), XW(3), XS(3)
DIMENSION TRSK(3), TRWK(3), TRSP(3)
LOGICAL HORIZ, SWITCH, CHANGE, TEST, SKIP
COMMON /IOUNIT/ IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUT
COMMON /CARB/ RCARB1, RCARB2
COMMON /BUOYCL/ RSPH, DELT, VZ, XCM, YCM, ZCM, XTOP, YTOP, SPHNS(3), RISTIM
COMMON /MOS/ DIFF(2,200), NCHTOT, PRSEP(200), NTOT, NARY, ITOT,
+ DMM(600), DMMY(401)
COMMON /GEOM/ COSTH2, SINTH, SINTH2, VISEXT, RTP1, SCRNC(2)
COMMON /MODE/ HORIZ
COMMON /TRAN/ VTR, KZ, KX, TTR, XTR, ZTR, QPUFF(3), SWITCH, CHANGE
COMMON /ACL/ CWINDS, CWINDC, CWINDW
COMMON /LOS/ T(3), R(3), U(3)
COMMON /SIG/ SIG02, SIGC
COMMON /CHARGE/ NCHG
COMMON /POINTS/ XNORM, DOT1, DOT2, DOT3

```

```

TRL00520
TRL00010
TRL00020
TRL00030
TRL00040
TRL00050
TRL00060
TRL00070
TRL00080
TRL00090
TRL00100
TRL00110
TRL00120
TRL00130
TRL00140
TRL00150
TRL00160
TRL00170
TRL00180
TRL00190
TRL00200
TRL00210
TRL00220
TRL00230
TRL00240
TRL00250
TRL00260
TRL00270
TRL00280
TRL00290
TRL00300
TRL00310
TRL00320
TRL00330
TRL00340
TRL00350
TRL00360
TRL00370
TRL00380
TRL00390
TRL00400
TRL00410
TRL00420
TRL00430
TRL00440
TRL00450
TRL00460
TRL00470
TRL00480
TRL00490
TRL00500
TRL00510
TRL00530
TRL00540
TRL00550
TRL00560
TRL00570
TRL00590
TRL00600
TRL00610
TRL00620
TRL00630
TRL00640
TRL00650
TRL00660
TRL00670
TRL00680
TRL00690
TRL00700

```

	COMMON/TRANNY/THRESH, TEST, NWL, NSOIL	TRL00710
	COMMON/SKIPIT/SKIP	TRL00720
	COMMON/EKTEMP/20, ZL, T0, TC1, TC2, TC3	TRL00730
	DATA ONEM/-1.0/	TRL00740
	DATA QWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/	TRL00750
	DATA QWFC/1.,.95,.5,.2,1.E-03/	TRL00760
C	PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER	TRL00770
C	NCHG=1	TRL00780
	SKIP=.FALSE.	TRL00790
	TEST=.FALSE.	TRL00800
	HSPH=0.0	TRL00810
	HWAK=0.0	TRL00820
	ACLSKT=0.0	TRL00830
	ACLWAK=0.0	TRL00840
	ACLSPH=0.0	TRL00850
	XNORM=0.0	TRL00860
	DO 10 I=1,3	TRL00870
	R(I)=REC(I)	TRL00880
	T(I)=TRN(I)	TRL00890
	U(I)=R(I)-T(I)	TRL00900
	XNORM=XNORM+U(I)**2	TRL00910
10	CONTINUE	TRL00920
	XNORM=SQRT(XNORM)	TRL00930
	U(1)=U(1)/XNORM	TRL00940
	U(2)=U(2)/XNORM	TRL00950
	U(3)=U(3)/XNORM	TRL00960
	IF((TIME-TTR).LT.1.E-20)GO TO 14	TRL00970
		TRL00980
		TRL00990
C	IF THE BUOYANT SPHERE HAS BECOME WIND BLOWN DETERMINE THE CENTER OF	TRL01000
C	MASS OF THE REFERENCE CHARGE.	TRL01010
	XCM=XTR+VTR*(TIME-TTR)	TRL01020
	ZCM=ZTR	TRL01030
		TRL01040
C	DETERMINE CENTER OF WAKE FOR REFERENCE CHARGE	TRL01050
C		TRL01060
	14 ZX=5.0	TRL01070
	CALL AVRG(ZX, TIME, QTOT, XBAVRG, SIG2X, SIG2Y)	TRL01080
	XW(1)=(XBAVRG+XCM)/2.	TRL01090
	XW(2)=YCM/2.	TRL01100
	XW(3)=(5.0+ZCM)/2.	TRL01110
		TRL01120
		TRL01130
C	IF THE DIFFERENCE BETWEEN THE TRANSMITTER AND RECEIVER IS GREATER	TRL01140
C	THAN 1 PERCENT OF THE DISTANCE BETWEEN THEM THEN THE LOS IS	TRL01150
C	CONSIDERED A SLANT PATH.	TRL01160
		TRL01170
	13 IF(ABS(U(3)).GT..01)GO TO 20	TRL01180
	IF(ABS(TRN(3)-REC(3)).LT.1.E-06)GO TO 9	TRL01190
C	COMPUTE CLOSEST POINT ALONG THE LOS TO OUR ESTIMATE OF THE CENTER OF	TRL01200
C	THE WAKE	TRL01210
	DOT=-((U(1)*(T(1)-XW(1))+U(2)*(T(2)-XW(2))+U(3)*(T(3)-XW(3)))	TRL01220
	T(3)=T(3)+U(3)*DOT	TRL01230
	R(3)=T(3)	TRL01240
C	COMPUTE CONTRIBUTIONS FOR A HORIZONTAL PATH	TRL01250
C		TRL01260
	9 HORIZ=.TRUE.	TRL01270
	CALL VSUM(REC, TRN, ONEM, DIR)	TRL01280
	CALL UNIT(DIR, DIR, RANGE)	TRL01290
	COSTH=DIR(1)	TRL01300
	SINTH=DIR(2)	TRL01310
	SINTH2=SINTH*SINTH	TRL01320
	COSTH2=COSTH**2	TRL01330
	SCRN(1)=SINTH	TRL01340
	SCRN(2)=-COSTH	TRL01350
	DO 12 J=1, ITOT	TRL01360
		TRL01370
		TRL01380
		TRL01390
		TRL01400

```

DO 11 I=1,2
TEMP(I)=DIFF(I,J)
11 CONTINUE
PRSEP(J)=DOTPRD(TEMP,SCRN)
12 CONTINUE
X=DOTPRD(SCRN,TRN)

COMPUTE THE CONTRIBUTION FROM THE SKIRT AT A HEIGHT OF T(3) WHERE T(3)
IS THE HEIGHT OF THE TRANSMITTER IF THE DIFFERENCE BETWEEN THE
TRANSMITTER AND RECEIVER HEIGHTS IS SMALL AND IS THE Z COMPONENT OF
THE POINT ON THE LINE CONNECTING THE TRANSMITTER AND RECEIVER WHICH
IS CLOSEST TO OUR ESTIMATE OF THE CENTER OF THE WAKE OTHERWISE.

ACLSKT=CWIND(X,Y,T(3),TIME)
IF( TEST)GO TO 998

TEST IS A LOGICAL VARIABLE RETURNED IN COMMON/TRANNY/ FROM SUBROUTINE
TRNCHK WHICH IS CALLED BY CWIND, CWAKE, CSPHER, TRNCLD, AND TRAP
EACH TIME A CONTRIBUTION IS MADE TO THE OPTICALLY WEIGHTED
CONCENTRATION ALONG THE OPTICAL PATH

TEST=.FALSE. TRANSMITTANCE IS GREATER THAN TRNMIN
              (A TRANSMITTANCE THRESHOLD)
              = .TRUE. TRANSMITTANCE IS LESS THAN TRNMIN

IF(TIME.GT.TTR)ACLWAK=CWAKE(X,Y,T(3),TIME)
IF( TEST)GO TO 998
IF(TIME.GT.TTR)ACLSPH=CSPHER(X,Y,T(3),TIME)
IF( TEST)GO TO 998
IF(TIME.GT.TTR)GO TO 50
CWINDS=ACLSKT
CWINDW=ACLWAK
CWINDC=ACLSPH
CALL TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)
IF( TEST)GO TO 998
GO TO 50

DO TRAPEZOIDAL INTEGRATION FOR SLANT PATH IN BOTH DIRECTIONS FROM
AN ESTIMATE OF THE LOCATION OF CENTER OF THE SKIRT USING A STEP SIZE
OF SIG, THE GEOMETRIC MEAN OF THE AVERAGE OF THE SPREADS OF THE DISCS
IN BOTH THE X AND Y DIRECTION, THEN IF THE BUOYANT SPHERE HAS
CONVERTED TO THE WIND MODEL DO THE SAME FOR THE WAKE AND SPHERE
WITH THE APPROPRIATE STEP SIZE. (CHECK TO SEE IF SPHERE HAS
CONVERTED TO THE WIND MODEL IS DONE IN TRAP.)

20 HORIC=.FALSE.
SIGX=SQRT(SIGZX)
SIGY=SQRT(SIGZY)
SIG=SQRT(SIGX*SIGY)
CALL WIND(2.0,UX,UY)
XSK(1)=TIME*UX
XSK(2)=TIME*UY
XSK(3)=2.0

FIND THE POINTS ON THE LINE CONNECTING THE TRANSMITTER AND
RECEIVER THAT ARE CLOSEST TO OUR ESTIMATE OF THE CENTER OF
THE SKIRT, WAKE, AND SPHERE.

DO 48 J=1,ITOT
XSK=XSK(1)+DIFF(1,J)
YSK=XSK(2)+DIFF(2,J)
ZSK=XSK(3)
DOT1=-(U(1)*(T(1)-XSK)+U(2)*(T(2)-YSK)+U(3)*(T(3)-ZSK))
IF(DOT1.LT.0.0)DOT1=0.0
IF(DOT1.GT.XNORM)DOT1=XNORM
XWK=XWK(1)+DIFF(1,J)
YWK=XWK(2)+DIFF(2,J)
ZWK=XWK(3)
DOT2=-(U(1)*(T(1)-XWK)+U(2)*(T(2)-YWK)+U(3)*(T(3)-ZWK))
IF(DOT2.LT.0.0)DOT2=0.0

```



```

IF(DOT2.GT.XNORM)DOT2=XNORM
XSP=XCM+DIFF(1,J)
YSP=YCM+DIFF(2,J)
ZSP=ZCM
DOT3=-(U(1)*(T(1)-XSP)+U(2)*(T(2)-YSP)+U(3)*(T(3)-ZSP))
IF(DOT3.LT.0.0)DOT3=0.0
IF(DOT3.GT.XNORM)DOT3=XNORM
NCHG=J
DO 45 II=1,3
TRSK(II)=T(II)+U(II)*DOT1
TRWK(II)=T(II)+U(II)*DOT2
TRSP(II)=T(II)+U(II)*DOT3
45 CONTINUE
DIFS=DIFFUS(Z0,ZL,2.0)
SIG2=SQRT(2.*DIFS*TIME)
IF(SIG2.LT.1.0)SIG2=1.0
H=U(1)*SIGX+U(2)*SIGY+U(3)*SIGZ
IF(TIME.LT.TTR)GO TO 46
HSPX=SQRT(SIG02+2.*KX*(TIME-TTR))
HSPY=HSPX
HSPZ=SQRT(SIG02+2.*KZ*(TIME-TTR))
HSPH=(U(1)*HSPX+U(2)*HSPY+U(3)*HSPZ)/2.
HWAK=(HSPH+H)/2.
46 CONTINUE
CALL TRAP<TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH>
ACLSKT=ACLSKT+SKT
ACLWAK=ACLWAK+WAK
ACLSPH=ACLSPH+SPH
CALL TRNCHK<ACLSKT,ACLWAK,ACLSPH>
IF<TEST>GO TO 998
H=-H
IF(TIME.LT.TTR)GO TO 47
HSPH=-HSPH
HWAK=-HWAK
47 CONTINUE
CALL TRAP<TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH>
ACLSKT=ACLSKT+SKT
ACLWAK=ACLWAK+WAK
ACLSPH=ACLSPH+SPH
CWINDS=ACLSKT
CWINDW=ACLWAK
CWINDS=ACLSPH
CALL TRNCHK<ACLSKT,ACLWAK,ACLSPH>
IF<TEST>GO TO 998
48 CONTINUE
IF(TIME.GT.TTR)GO TO 50
CALL TRNCLD<XNORM,TIME,ACLWAK,ACLSPH>
IF<TEST>GO TO 998
50 CONTINUE
ACLC=(ACLWAK+ACLSPH)*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL))
ACLS=ACLSKT*OWF(NWL,NSOIL)
COMPUTE THE TRANSMITTANCE ALONG THE LINE OF SIGHT
TRNLOS=EXP(-ACLS-ACLC)
GO TO 999
998 TRNLOS=0.0
999 RETURN
END

```

```

TRL 02110
TRL 02120
TRL 02130
TRL 02140
TRL 02150
TRL 02160
TRL 02170
TRL 02180
TRL 02190
TRL 02200
TRL 02210
TRL 02220
TRL 02230
TRL 02240
TRL 02250
TRL 02260
TRL 02270
TRL 02280
TRL 02290
TRL 02300
TRL 02310
TRL 02320
TRL 02330
TRL 02340
TRL 02350
TRL 02360
TRL 02370
TRL 02380
TRL 02390
TRL 02400
TRL 02410
TRL 02420
TRL 02430
TRL 02440
TRL 02450
TRL 02460
TRL 02470
TRL 02480
TRL 02490
TRL 02500
TRL 02510
TRL 02520
TRL 02530
TRL 02540
TRL 02550
TRL 02560
TRL 02570
TRL 02580
TRL 02590
TRL 02600
TRL 02610
TRL 02620
TRL 02630
TRL 02640
TRL 02650
TRL 02660
TRL 02670
TRL 02680
TRL 02690

```



```

SUBROUTINE TRNCLD(XNORM, TIME, ACLWAK, ACLSPH)
ROUTINE FOR DETERMINING CONTRIBUTION FROM SPHERE AND WAKE
BEFORE THE BUOYANT SPHERE WAS CONVERTED TO THE WIND MODEL.
INPUTS
  XNORM -DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER
  TIME -TIME AT WHICH TRANSMITTANCE IS DESIRED
OUTPUTS
  ACLWAK -CONTRIBUTION FROM WAKE TO OPTICALLY WEIGHTED CONCENTRATION
          ALONG GIVEN LINE OF SIGHT
  ACLSPH -CONTRIBUTION FROM BUOYANT SPHERE TO OPTICALLY WEIGHTED
          CONCENTRATION ALONG GIVEN LINE OF SIGHT
SUBROUTINES NEEDED
  AVRQ -COMPUTE THE AVERAGE OF THE MOMENTS FOR THE DISCS
  WIN -COMPUTE THE WIND SPEED AT A GIVEN HEIGHT
  AMOUNT-COMPUTE THE DISTRIBUTION OF THE LOADING BETWEEN THE BUOYANT
        SPHERE AND WAKE.
  CONLEN-COMPUTE THE LENGTH OF INTERSECTION OF NON-HORIZONTAL LINE
        OF SIGHT WITH A CONICAL SHAPED WAKE
  TRNCHK-ROUTINE TO CHECK IF THE OBSCURATION IS SUCH THAT THE
        TRANSMITTANCE IS LESS THAN A USET SPECIFIED AMOUNT
*****
LOGICAL HORIZ, TEST
DIMENSION CENTER(3)
COMMON/MODE/HORIZ
COMMON/GEOM/COSTH2, SINTH, SINTH2, VISEXT, RTP1, SCRN(2)
COMMON/BUOYCL/RSPH, DELT, VZ, XCM, YCM, ZCM, XTOP, YTOP, SPHNS(3), RISTIM
COMMON/PRTINF/R0, VGRAV(3), NPRTS
COMMON/DISCS/NDSCS, TDSC(20), XDSC(20), ZDSC(20), R2DSC(20), ODSC(20,3)
COMMON/MOS/DIFF(2,200), NCHTOT, PRSEP(200), NTOT, NARY, ITOT,
+ DMM(600), DMMY(401)
COMMON /IQUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLINT, KSTOR, NPLOTU
COMMON/LOS/TR(3), RE(3), U(3)
COMMON/TRANNY/THRESH, TEST, NWL, NSOIL
COMMON/ACL/CWINDS, CWINDC, CWINDW
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
ACLWAK=0.0
ACLSPH=0.0
DETERMINE THE RADIUS OF THE BASE OF THE CONE
ZX=5.0
CALL AVRQ(ZX, TIME, QTOT, XBAVG, SIG2X, SIG2Y)
IF(QTOT.LT.1.E-10)GO TO 33
SIGX=SQRT(SIG2X)
SIGY=SQRT(SIG2Y)
RB=SQRT(SIGX*SIGY)
GO TO 35
33 CALL WIN(5.0, UW, V)
XBAVG=UW*TIME
RB=0.0
35 VOLWAK=(PI/3.)**(ZCM-5.0)**(RSPH**2+RSPH*RB+RB**2)
VOLSPH=(4./3.)*PI*RSPH**3
CALL AMOUNT(VOLSPH, WAKAL, SPHAL)
DO 80 J=1, ITOT
IF (VOLWAK.LE.0.0) GO TO 68
XB=XBAVG+DIFF(1, J)

```

	YB=DIFF(2,J)	TRD00730
	XC=XCM+DIFF(1,J)	TRD00740
	YC=YCM+DIFF(2,J)	TRD00750
	IF(ABS(U(3)).LE..01)GO TO 110	TRD00760
	ZBOT=5.0	TRD00770
	CALL CONLENK(U,TR,ZCM,ZBOT,XC,YC,RSPH,RB,XB,YB,XNORM,PLENWK)	TRD00780
	ACLW=PLENWK*WAKAL/VOLWAK	TRD00790
	GO TO 69	TRD00800
68	ACLW=0.0	TRD00810
69	CONTINUE	TRD00820
	ACLWAK=ACLWAK+ACLW	TRD00830
	CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)	TRD00840
	IF(TEST)GO TO 999	TRD00850
C		TRD00860
C	DETERMINE CONTRIBUTION FROM SPHERE FO A SLANT PATH	TRD00870
C		TRD00880
	CENTER(1)=XCM+DIFF(1,J)-TR(1)	TRD00890
	CENTER(2)=YCM+DIFF(2,J)-TR(2)	TRD00900
	CENTER(3)=ZCM-TR(3)	TRD00910
	CLOSE=U(1)*CENTER(1)+U(2)*CENTER(2)+U(3)*CENTER(3)	TRD00920
	CON=CENTER(1)**2+CENTER(2)**2+CENTER(3)**2-RSPH**2	TRD00930
	RADIC=CLOSE**2-CON	TRD00940
	IF(RADIC.LT.0.0)GO TO 75	TRD00950
	PNEAR=CLOSE-SQRT(RADIC)	TRD00960
	PFAR=CLOSE+SQRT(RADIC)	TRD00970
	PLENSP=AMIN1(PFAR,XNORM)-AMAX1(PNEAR,0.0)	TRD00980
	IF(PLENSP.LT.0.0)PLENSP=0.0	TRD00990
	ACLS=PLENSP*SPHAL/VOLSPH	TRD01000
	GO TO 76	TRD01010
75	ACLS=0.0	TRD01020
76	ACLSPH=ACLSPH+ACLS	TRD01030
	CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)	TRD01040
	IF(TEST)GO TO 999	TRD01050
	GO TO 80	TRD01060
C		TRD01070
C	COMPUTE CONTRIBUTIONS FOR SPHERE AND WAKE FOR A HORIZONTAL PATH	TRD01080
		TRD01090
110	IF(TR(3).GT.ZCM+RSPH)GO TO 999	TRD01100
	XCEN=XCM+DIFF(1,J)	TRD01110
	YCEN=YCM+DIFF(2,J)	TRD01120
	IF(TR(3).LT.ZCM-RSPH)GO TO 130	TRD01130
	RADIUS=SQRT(RSPH**2-(TR(3)-ZCM)**2)	TRD01140
	CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENSP)	TRD01150
	IF(PLENSP.LT.0.0)PLENSP=0.0	TRD01160
	ACLSPH=ACLSPH+PLENSP*SPHAL/VOLSPH	TRD01170
	CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)	TRD01180
	IF(TEST)GO TO 999	TRD01190
130	IF(TR(3).GT.ZCM)GO TO 999	TRD01200
	IF(TR(3).LE.5.0)GO TO 999	TRD01210
	ZETA=(TR(3)-5.0)/(ZCM-5.0)	TRD01220
	XCEN=ZETA*XCEN+(1.-ZETA)*XB	TRD01230
	YCEN=ZETA*YCEN+(1.-ZETA)*YB	TRD01240
	RADIUS=ZETA*RSPH+(1.-ZETA)*RB	TRD01250
	CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENWK)	TRD01260
	IF(PLENWK.LT.0.0)PLENWK=0.0	TRD01270
	ACLWAK=ACLWAK+PLENWK*WAKAL/VOLWAK	TRD01280
	CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)	TRD01290
	IF(TEST)GO TO 999	TRD01300
80	CONTINUE	TRD01310
999	RETURN	TRD01320
	END	TRD01330

```
      SUBROUTINE UNIT(A,B,XNORM)
      DIMENSION A(2),B(2)
C ***  B IS THE NORM OF A, AND XNORM IS THE MAGNITUDE
      XNORM=SQRT(A(1)**2+A(2)**2)
      B(1)=A(1)/XNORM
      B(2)=A(2)/XNORM
      RETURN
      END
```

```
UNIT0010
UNIT0020
UNIT0030
UNIT0040
UNIT0050
UNIT0060
UNIT0070
UNIT0080
```

```

SUBROUTINE VEHCL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,PHI,NSOIL, VCL00160
1          SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR, VCL00170
2          VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH, VCL00180
3          TRNLOS,NERR) VCL00190
C THIS ROUTINE CONTROLS THE FLOW OF THE CALCULATION FOR THE VEHICLE GENERATED DUST CLOUD. VCL00020
C INPUTS VCL00030
C SEE DRTRAN FOR DETAILS VCL00040
C OUTPUTS VCL00050
C TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT VCL00060
C NERR - ERROR CODE VCL00070
C***** VCL00080
C DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2) VCL00090
C DIMENSION TRNCOR(3),RECCOR(3),TRN(3),REC(3),TRNFRM(2,2),ORIG(2), VCL00100
1          V1(2),V0(2),VDIR(2) VCL00110
C LOGICAL DHDT,ERR,NEWATM,NEWVEH VCL00120
C INTEGER VEHTYP VCL00130
C COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUV VCL00140
C COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2) VCL00150
C COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZIN VCL00200
C COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK VCL00210
C DATA RTPI /1.772453/ VCL00220
C THETA=THWND*PIRAD VCL00230
C IF(.NOT.NEWATM)GO TO 100 VCL00240
C CALL ATMCAL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,THETA,DHDT,ERR) VCL00250
C IF(.NOT.ERR)GO TO 100 VCL00260
C NERR=7 VCL00270
C GO TO 999 VCL00280
100 CONTINUE VCL00290
C IF(.NOT.NEWVEH)GO TO 5 VCL00300
C CALL VSRC(VEHSPD,VEHWID,VEHWHT,VEHTYP,NSOIL,SILT) VCL00310
C CALL PREVEH(NSOIL,NWL) VCL00320
5 CONTINUE VCL00330
C COMPUTE DIRECTION VECTOR FOR THE VEHICLE MOTION FROM USERS INPUT VCL00340
C ANGL=VEHDIR*PIRAD VCL00350
C V1(1)=VEHSPD*COS(ANGL) VCL00360
C V1(2)=VEHSPD*SIN(ANGL) VCL00370
C COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER VCL00380
C DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN VCL00390
C THE WIND DIRECTION. VCL00400
C TRNFRM(1,1)=COS(THETA) VCL00410
C TRNFRM(2,2)=TRNFRM(1,1) VCL00420
C TRNFRM(1,2)=SIN(THETA) VCL00430
C TRNFRM(2,1)=-TRNFRM(1,2) VCL00440
C ORIG(1)=V0(1) VCL00450
C ORIG(2)=V0(2) VCL00460
C COMPUTE NEW COORDINATES BY MULTIPLYING BY THE TRANSFORMATION MATRIX VCL00470
C TRN(3)=TRNCOR(3) VCL00480
C REC(3)=RECCOR(3) VCL00490
C DO 20 I=1,2 VCL00500
C TRN(I)=0.0 VCL00510
C REC(I)=0.0 VCL00520
C VDIR(I)=0.0 VCL00530
C DO 10 J=1,2 VCL00540
C TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J)) VCL00550
C REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J)) VCL00560
C VDIR(I)=VDIR(I)+TRNFRM(I,J)*V1(J) VCL00570
C VCL00580
C VCL00590
C VCL00600
C VCL00610
C VCL00620
C VCL00630
C VCL00640
C VCL00650
C VCL00660
C VCL00670
C VCL00680
C VCL00690
C VCL00700

```

C
C
C
C
10 CONTINUE
20 CONTINUE
CALL VEHTRN ROUTINE TO USE COMPUTED QUADRATIC FITS TO CALCULATE
A TRANSMITTANCE
CALL VEHTRN<TRN,REC,TIME,VDIR,TRNL0S>
999 RETURN
END

VCL00710
VCL00720
VCL00730
VCL00740
VCL00750
VCL00760
VCL00770
VCL00780
VCL00790

```

SUBROUTINE VEHTRN( TRN, REC, TIME, VDIR, TRNLOS )
THIS ROUTINE PARAMETERIZES THE LINE CONNECTING THE TRANSMITTER
AND RECEIVER IN THE LOCAL COORDINATE SYSTEM AND DOES A TRAPEZOIDAL
INTEGRATION FROM VEHICLE TIME=0.0 TO VEHICLE TIME=TIME THE
TRANSMITTANCE IS DESIRED.
INPUTS
TRN - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE
      SYSTEM
REC - THE COORDINATES OF THE RECEIVER IN THE LOCAL COORDINATE
      SYSTEM
TIME - THE PRESENT TIME AT WHICH A TRANSMITTANCE IS DESIRED
NWL - INTEGER INDEX FOR WAVELENGTH BEING USED
NSOIL - INTEGER INDEX FOR SOIL TYPE
VDIR - VECTOR INDICATING THE DIRECTION AND SPEED OF THE VEHICLE
        IN THE LOCAL COORDINATE SYSTEM
OUTPUT
TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT AT THE INDICATED
          TIME
FUNCTIONS AND SUBROUTINES NEEDED
GRAND EVALUATES THE INTEGRAND
*****
LOGICAL TEST
DIMENSION TRN(3), REC(3), TR(3), RE(3), VDIR(2), DWF(5,2), U(3)
COMMON/ MOS/ DMMY(604), DMM(600),
+          ICOUNT, TIMES(25), XC0(3,25), XC1(3,25), RT(3,25),
+          RB(3,25), Z2(3,25)
COMMON/ TRANNY/ THRESH, TEST, NWL, NSOIL
DATA DWF/ 1., .93, .52, .44, 2.E-03, 1., 1., 1., 1., 4.E-03/
TEST=.FALSE.
PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
XNORM=0.0
DO 10 I=1,3
  RE(I)=REC(I)
  TR(I)=TRN(I)
  U(I)=RE(I)-TR(I)
  XNORM=XNORM+U(I)**2
10 CONTINUE
XNORM=SQRT(XNORM)
U(1)=U(1)/XNORM
U(2)=U(2)/XNORM
U(3)=U(3)/XNORM
INCREMENT VEHICLE TRAVEL TIME AND CALL GRAND TO COMPUTE THE VALUE
OF THE INTEGRAND. IF THE VALUE OF THE VARIABLE TIME IS GREATER
THAN THE MAXIMUM TIME THAT HAS BEEN STORED (APPROX. 373 SEC) THEN
ANY DUST PRODUCED MORE THAN TIMES(20) SECONDS IS ASSUMED TO HAVE
NO EFFECT ON THE TRANSMITTANCE.
IF( TIME.LE. TIMES( ICOUNT )) GO TO 11
TINC=TIMES( ICOUNT )/400.
TSTART=TIME-TIMES( ICOUNT )
GO TO 15
11 TINC=TIME/400.
TSTART=0.0

```

```

VTN00340
VTN00010
VTN00020
VTN00030
VTN00040
VTN00050
VTN00060
VTN00070
VTN00080
VTN00090
VTN00100
VTN00110
VTN00120
VTN00130
VTN00140
VTN00150
VTN00160
VTN00170
VTN00180
VTN00190
VTN00200
VTN00210
VTN00220
VTN00230
VTN00240
VTN00250
VTN00260
VTN00270
VTN00280
VTN00290
VTN00300
VTN00310
VTN00320
VTN00330
VTN00350
VTN00360
VTN00370
VTN00380
VTN00390
VTN00400
VTN00410
VTN00420
VTN00430
VTN00440
VTN00450
VTN00460
VTN00470
VTN00480
VTN00490
VTN00500
VTN00510
VTN00520
VTN00530
VTN00540
VTN00550
VTN00560
VTN00570
VTN00580
VTN00590
VTN00600
VTN00610
VTN00620
VTN00630
VTN00640
VTN00650
VTN00660
VTN00670
VTN00680
VTN00690
VTN00700

```



```

15 SUM=0.0
   DO 50 I=1,401
   TIVEH=TSTART+FLOAT(I-1)*TINC
   CALL GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)
   IF(I.EQ.1.OR.I.EQ.401)GO TO 20
   SUM=SUM+VALUE
   GO TO 40
20 SUM=SUM+VALUE/2.
40 CONTINUE
   SUM1=SUM*TINC
   ACLWAK=0.0
   ACLSPH=0.0
   CALL TRNCHK(SUM1,ACLWAK,ACLSPH)
   IF( TEST)GO TO 998
50 CONTINUE
   SUM=SUM*TINC
   ACL=DFW(NWL,NSOIL)*SUM
   TRNLOS=EXP(-ACL)
   GO TO 999
998 TRNLOS=0.0
999 RETURN
   END

```

```

VTN00710
VTN00720
VTN00730
VTN00740
VTN00750
VTN00760
VTN00770
VTN00780
VTN00790
VTN00800
VTN00810
VTN00820
VTN00830
VTN00840
VTN00850
VTN00860
VTN00870
VTN00880
VTN00890
VTN00900
VTN00910
VTN00920

```

```

SUBROUTINE VSRCK(VSPD,VWID,VWHT,VEHTYP,NSOIL,SILT)
THIS SUBROUTINE INITIALIZES THE DUST CLOUD PRODUCED BY A VEHICLE
INPUTS
  VSPD - THE VELOCITY OF THE VEHICLE
  VWID - WIDTH OF THE VEHICLE
  VWHT - WEIGHT OF THE VEHICLE IN KGS.
  NSOIL - SOIL TYPE
  SILT - SILT CONTENT OF THE SOIL
OUTPUTS
  STOPED IN COMMON/DISCS/ AND COMMON/PRE/
  NDSCS - NUMBER OF DISCS (FOR A VEHICLE ONLY ONE)
  TDSC - TIME OF RELEASE OF THE DISCS
  XDSC - X POSITION OF DISC AT TIME OF RELEASE
  ZDSC - HEIGHT OF RELEASE OF THE DISC
  R2DSC - SQUARE OF THE RADIUS OF THE DISC
*****
INTEGER VEHTYP
COMMON/PRTINF/R0,VGRAV(3),NPRTS
COMMON /IDUNIT/IDIN, IDOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
COMMON/PRE/Z,RT2DZ
COMMON/YL/YLOAD
NDSCS=1
NPRTS=1
VGRAV(1)=0.0
QDSC(1,1)=1.0
INITIALIZE THE VEHICLE SOURCE FOR WHEELED VEHICLES
  ZDSC(1)=VWID/8.
  Z=VWID/4.
  ZZ=VWID/4.
  DZ=DIFFUS(Z0,ZL,ZZ)
  DX=DXZ0*DZ
  TDSC(1)=-9.*(VWID**2)/512./DZ
  TOF=-TDSC(1)
  CALL MOMENT(VGRAV,Z,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)
  XDSC(1)=-XBAR
  A=1.0
  B=SIGW2+SIGP2
  C=4.*(SIGW2*SIGP2)-((VWID/3. )**4))
  RAD=B**2-A*C
  R2DSC(1)=-B+SQRT(RAD)
  RT2DZ=SQRT(2.*DZ)
INITIALIZE LOADING FOR VEHICLE (VLOAD IN KG/SEC)
20 SILTPC=100.*SILT
  A=3.8E-9
  IF(VEHTYP.GT.0)A=1.52E-08
  Q=A*VSPD*VWHT*SILTPC
  ALPHA=240.

```

```

VRS00320
VRS00010
VRS00020
VRS00030
VRS00040
VRS00050
VRS00060
VRS00070
VRS00080
VRS00090
VRS00100
VRS00110
VRS00120
VRS00130
VRS00140
VRS00150
VRS00160
VRS00170
VRS00180
VRS00190
VRS00200
VRS00210
VRS00220
VRS00230
VRS00240
VRS00250
VRS00260
VRS00270
VRS00280
VRS00290
VRS00300
VRS00310
VRS00330
VRS00340
VRS00350
VRS00360
VRS00370
VRS00380
VRS00390
VRS00400
VRS00410
VRS00420
VRS00430
VRS00440
VRS00450
VRS00460
VRS00470
VRS00480
VRS00490
VRS00500
VRS00510
VRS00520
VRS00530
VRS00540
VRS00550
VRS00560
VRS00570
VRS00580
VRS00590
VRS00600
VRS00610
VRS00620
VRS00630
VRS00640
VRS00650
VRS00660
VRS00670
VRS00680
VRS00690
VRS00700

```

999 VLOAD=ALPHA*VSPD*Q
RETURN
END

VRS00710
VRS00720
VRS00730

```
SUBROUTINE VSUM(A,B,S,C)
DIMENSION A(2),B(2),C(2)
C *** C=A+S*B WHERE A,B,C ARE VECTORS AND S IS SCALAR
DO 14 J=1,2
14 C(J)=A(J)+S*B(J)
RETURN
END
```

```
VSU00010
VSU00020
VSU00030
VSU00040
VSU00050
VSU00060
VSU00070
```

```

SUBROUTINE WIN(Z,U,V)
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/EKWIND/ALP,C,PYF,PF,UHAT,VHAT
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
COMMON/BUOYCL/Y(8),SPHNS(3),RISTIM
*****
C/PURPOSE
PURPOSE
TO COMPUTE THE WIND SPEED AT A SPECIFIED HEIGHT
INPUTS
Z HEIGHT AT WHICH WIND SPEEDS ARE DESIRED
OUTPUTS
U WIND SPEED IN THE DIRECTION OF THE GROUND WIND
V WIND SPEED PERPENDICULAR TO THE GROUND WIND
CALLED BY DIFEQ
SUBROUTINES AND FUNCTIONS NEEDED
WINDCAL CALCULATES SCALED WIND SPEED
*****
C/IF(Z.GT.ZSTAR)GO TO 100
IF(Z.GT.ZSTAR)GO TO 100
U=USTAR*WINDCAL(Z0,ZL,Z)
V=0.0
GO TO 999
100 UE=C*EXP(-ALP*Z)*COS(ALP*Z)-PYF
VE=-C*EXP(-ALP*Z)*SIN(ALP*Z)+PF
U=UHAT*UE+VHAT*VE
V=-VHAT*UE+UHAT*VE
999 RETURN
END

```

```

WIN00100
WIN00200
WIN00300
WIN00400
WIN00500
WIN00600
WIN00700
WIN00800
WIN00900
WIN01000
WIN01100
WIN01200
WIN01300
WIN01400
WIN01500
WIN01600
WIN01700
WIN01800
WIN01900
WIN02000
WIN02100
WIN02200
WIN02300
WIN02400
WIN02500
WIN02600
WIN02700
WIN02800
WIN02900
WIN03000
WIN03100
WIN03200
WIN03300
WIN03400
WIN03500
WIN03600
WIN03700
WIN03800

```



```
S2=1.5
BW=1./(<math>Z0/ZL+S2</math>)+7.
DW=-1.*S2*BW
52 CONTINUE
IF(<math>LOW</math>)GO TO 999
IF(<math>P>53,53,54</math>)
53 WNCAL=WNCAL+CW+AW*(<math>-ZL/Z</math>)**(<math>1./3.</math>)
GO TO 999
54 WNCAL=WNCAL+DW+BW*Z/ZL
999 WNCAL=WNCAL/.4
998 CONTINUE
RETURN
END
```

```
WND00710
WND00720
WND00730
WND00740
WND00750
WND00760
WND00770
WND00780
WND00790
WND00800
WND00810
WND00820
WND00830
```

```

SUBROUTINE NMMW(FREQGH, ICLMAT, MMTRAN, IERR)
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
COMMON /CLYMAT/TEMP, PRESS, RH, AH, DP, VIS, CLDAMT, CLDHYT,
+ FOGPRB, WNDVEL, WNDDIR, IPASCT
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLINT, KSTOR, NPLOU
COMMON /GEOMET/PTS<15>, IGEOSW
NMM00010
NMM00020
NMM00030
NMM00040
NMM00050
NMM00060
NMM00070
NMM00080
NMM00090
NMM00100
NMM00110
NMM00120
NMM00130
NMM00140
NMM00150
NMM00160
NMM00170
NMM00180
NMM00190
NMM00200
NMM00210
NMM00220
NMM00230
NMM00240
NMM00250
NMM00260
NMM00270
NMM00280
NMM00290
NMM00300
NMM00310
NMM00320
NMM00330
NMM00340
NMM00350
NMM00360
NMM00370
NMM00380
NMM00390
NMM00400
NMM00410
NMM00420
NMM00430
NMM00440
NMM00450
NMM00460
NMM00470
NMM00480
NMM00490
NMM00500
NMM00510
NMM00520
NMM00530
NMM00540
NMM00550
NMM00560
NMM00570
NMM00580
NMM00590
NMM00600
NMM00610
NMM00620
NMM00630
NMM00640
NMM00650
NMM00660
NMM00670
NMM00680
NMM00690
NMM00700

-----
PURPOSE: TO CALCULATE THE EXTINCTION AND BACKSCATTER AT MILLIMETER
FREQUENCIES 10 TO 1000 GHZ, DUE TO
GASEOUS ABSORPTION, FOG<ICE AND WATER> AND
CLOUD BULK ATTENUATION, RAIN, SNOW EXTINCTION.
-----
*** INPUT TO THE NMMW MODULE IS PERFORMED THROUGH A CARD ORDER-
INDEPENDENT INPUT TECHNIQUE. A FOUR-LETTER IDENTIFIER IN COLS.
1-4 OF EACH RECORD SPECIFIES THE TYPE OF DATA BEING READ BY THE
MODULE. THE INPUT CARDS MAY APPEAR IN ANY ORDER WITH THE EXCEPTION
OF OF THE <GO> END OF READ SENTINEL, WHICH MUST BE THE LAST CARD
READ. ALL OF THE FOLLOWING CARDS ARE READ IN UNDER THE FORMAT
** <A4, 1X, 3(E10.4, 1X)> :
-----
CARD IDENTIFIER : PATH
VARIABLES READ : MMWPTH <PATH LENGTH <KM>>
-----
CARD IDENTIFIER : ATMO
VARIABLES READ : TEMPI <TEMPERATURE <DEG C>>
PRESS1 <PRESSURE <MB>>
ABSHUM <IF .GT. 0, ABSOLUTE
HUMIDITY <GM/M**3>>
<IF .LT. 0, RELATIVE
HUMIDITY <%>>
***NOTE: TEMPI, PRESS1, ABSHUM WILL BE PASSED FROM CLIMAT IF
ICLMAT=1. IN THAT EVENT, THE <ATMO> CARD IS NOT NEEDED.
-----
CARD IDENTIFIER : FOGD
VARIABLES READ : FOGDEN <FOG DENSITY <LIQUID WATER,
GM/M**3>>
-----
CARD IDENTIFIER : RAIN
VARIABLES READ : RAINRT <RAIN RATE <MM/HR>>
-----
CARD IDENTIFIER : SNOW
VARIABLES READ : SNOWRT <SNOW RATE <MM/HR> WATER EQUIV.>
-----
CARD IDENTIFIER : GO
VARIABLES READ : NONE <END OF READ SENTINEL>
-----
MAIN ROUTINE CALLS NMMW. NMMW THEN CALLS
(1)MMWGS<GAS ABSORPTION>, (2)MMWFG<FOG ABSORPTION, BACKSCATTER>,
(3)MMRAN<RAIN EXTINCTION, BACKSCATTER>,
(4)MMSNO<SNOW EXTINCTION, BACKSCATTER>, AND RETURNS TO MAIN NMMW
-----
LOCAL VARIABLES
REAL MMWPTH, MMTRAN, MMBSXS
DIMENSION DAT<3>, IAL<6>
DATA IAL/2HPA ,2HAT ,2HFO ,2HRA ,2HSN ,2HGO /
IERR=0
GASABS=0.
FOGEXT=0.
RAINEX=0.
FOGBS=0.
RAINBS=0.
SNOWBS=0.
5 READ<IOIN, 400>IALFA, IALFA2, <DAT<L>, L=1, 3>
400 FORMAT<2A2, 1X, 3(E10.4, 1X)>
IF<IALFA.EQ. IAL<1>> GO TO 10
IF<IALFA.EQ. IAL<2>> GO TO 20

```



```

IF<IALFA.EQ.IAL<3>> GO TO 30
IF<IALFA.EQ.IAL<4>> GO TO 40
IF<IALFA.EQ.IAL<5>> GO TO 50
IF<IALFA.EQ.IAL<6>> GO TO 60
WRITE<IOOUT,450>IALFA,IALFA2
450 FORMAT<1H0,20X,2A2,22H IS AN INCORRECT LABEL //>
GO TO 300
10 MMWPTH=DAT<1>
GO TO 5
20 TEMP1=DAT<1>
PRESS1=DAT<2>
ABSHUM=DAT<3>
GO TO 5
30 FOGDEN=DAT<1>
GO TO 5
40 RAINRT=DAT<1>
GO TO 5
50 SNOWRT=DAT<1>
GO TO 5
60 CONTINUE
IF<IGEOSW.NE.1>GO TO 99
MMWPTH=SQRT<<PTS<4>-PTS<1>>**2+<PTS<5>-PTS<2>>**2+
+<PTS<6>-PTS<3>>**2>
99 CONTINUE
IF<ICLMAT.EQ.1> TEMP1=TEMP
IF<ICLMAT.EQ.1> PRESS1=PRESS
IF<ICLMAT.EQ.1> ABSHUM=AH
WRITE<IOOUT,500> TEMP1,PRESS1,ABSHUM,FOGDEN
WRITE<IOOUT,600> RAINRT,SNOWRT,FREQGH,MMWPTH
C CHECK MODEL INPUTS FOR RANGE OF VALIDITY.
IF<PRESS1.GE.500.> GO TO 100
IERR=1
WRITE<IOOUT,800>
100 IF<FREQGH.LE.1000..AND.FREQGH.GE.10.> GO TO 150
IERR=1
WRITE<IOOUT,900>
150 IF<IERR.EQ.1> GO TO 300
C CHANGE UNITS
PRSTOR=PRESS1/TORRMB
TEMPDK=TEMP1+CDEGK
C CALL INDIVIDUAL MODULES FOR GAS, FOG/CLOUD, SNOW, RAIN EXTINCTION
CALL MMWGS<TEMPDK,PRSTOR,ABSHUM,FREQGH,GASABS>
IF<FOGDEN.GT.1.E-10>
+ CALL MMWFG<FOGDEN,TEMPDK,FREQGH,FOGEXT,FOGBS>
+ IF<RAINRT.GT.1.E-10>
+ CALL MMRAN<RAINRT,TEMPDK,FREQGH,2.,RAINEX,RAINBS>
+ IF<SNOWRT.GT.1.E-10>
+ CALL MMSNO<SNOWRT,TEMPDK,FREQGH,SNOWEX,SNOWBS>
C COMPUTE TRANSMISSION
TOTEXT=GASABS+FOGEXT+RAINEX+SNOWEX
MMBSXS=FOGBS+RAINBS+SNOWBS
MMTRAN=EXP<-MMWPTH*TOTEXT>
C CHANGE UNITS FOR ABSORPTION/EXTINCTION FROM 1/KM TO DB/KM.
DBKM=4.343
GASABS=GASABS*DBKM
FOGEXT=FOGEXT*DBKM
RAINEX=RAINEX*DBKM
SNOWEX=SNOWEX*DBKM
WRITE<IOOUT,700> GASABS,FOGEXT,RAINEX,SNOWEX,MMTRAN
WRITE<IOOUT,750> FOGBS,RAINBS,SNOWBS,MMBSXS
C COMPUTATION COMPLETED
300 RETURN
C
500 FORMAT<1H0,///,47X,12HTEMPERATURE ,14X,F8.3,
+ 10H DEGREES C,/,47X,9HPRESSURE ,17X,F8.3,
+ 3H MB,/,47X,17HABSOLUTE HUMIDITY,9X,F8.3,
+ 7H G/M**3/,47X,11HFOG DENSITY,15X,F8.3,
+ 7H G/M**3>
600 FORMAT<1H ,46X,9HRAIN RATE,17X,F8.3,6H MM/HR,/,
+ ,47X,9HSNOW RATE,17X,F8.3,6H MM/HR,/>
MMM00710
MMM00720
MMM00730
MMM00740
MMM00750
MMM00760
MMM00770
MMM00780
MMM00790
MMM00800
MMM00810
MMM00820
MMM00830
MMM00840
MMM00850
MMM00860
MMM00870
MMM00880
MMM00890
MMM00900
MMM00910
MMM00920
MMM00930
MMM00940
MMM00950
MMM00960
MMM00970
MMM00980
MMM00990
MMM01000
MMM01010
MMM01020
MMM01030
MMM01040
MMM01050
MMM01060
MMM01070
MMM01080
MMM01090
MMM01100
MMM01110
MMM01120
MMM01130
MMM01140
MMM01150
MMM01160
MMM01170
MMM01180
MMM01190
MMM01200
MMM01210
MMM01220
MMM01230
MMM01240
MMM01250
MMM01260
MMM01270
MMM01280
MMM01290
MMM01300
MMM01310
MMM01320
MMM01330
MMM01340
MMM01350
MMM01360
MMM01370
MMM01380
MMM01390
MMM01400

```

```

+          ,47X,9HFREQUENCY,17X,F8.3,4H GHZ/
+          ,47X,11HPATH LENGTH,15X,F8.3,3H KM)
700  FORMAT(1H0,46X,14HGAS ABSORPTION,10X,E10.4,6H DB/KM, /
+          ,47X,15HFOG EXTINCTION ,9X,E10.4,6H DB/KM, /
+          ,47X,16HRAIN EXTINCTION ,8X,E10.4,6H DB/KM, /
+          ,47X,16HSNOW EXTINCTION ,8X,E10.4,6H DB/KM, /
+          ,47X,13HTRANSMISSION ,11X,E10.4 /)
750  FORMAT(1H ,46X,15HFOG BACKSCATTER,9X,E10.4,10H M**2/M**3, /
+          ,47X,16HRAIN BACKSCATTER,8X,E10.4,10H M**2/M**3, /
+          ,47X,16HSNOW BACKSCATTER,8X,E10.4,10H M**2/M**3, /
+          ,47X,17HTOTAL BACKSCATTER,7X,E10.4,10H M**2/M**3)
800  FORMAT(1H0,47X,41HPRESSURE LESS THAN 500 MB, GAS ABSORPTION,
+          19H WILL BE INACCURATE)
900  FORMAT(1H0,47X,31HFREQUENCY<10. GHZ, OR >1000 GHZ,
+          42H CALULATION WILL FAIL, USE OTHER FREQUENCY)
END

```

```

NMM01410
NMM01420
NMM01430
NMM01440
NMM01450
NMM01460
NMM01470
NMM01480
NMM01490
NMM01500
NMM01510
NMM01520
NMM01530
NMM01540
NMM01550
NMM01560

```

```
FUNCTION AB(WA,A,CE,B,C)
AB=A*EXP(-ABS((ALOG10(1.E4*WA/CE))/B)**C)
RETURN
END
```

```
AB 00010
AB 00020
AB 00030
AB 00040
```

```

FUNCTION DOP(WA,A,CE1,B,C,CE2,D,E,CE3,F,G)
V=1./WA
V2=V*V
H1=CE1*CE1-V2
H2=CE2*CE2-V2
H3=CE3*CE3-V2
DOP=SQRT(A+B*H1/(H1*H1+C*V2)+D*H2/(H2*H2+E*V2)+F*H3/(H3*H3+G*V2))
RETURN
END
DOP00010
DOP00020
DOP00030
DOP00040
DOP00050
DOP00060
DOP00070
DOP00080
DOP00090

```

	SUBROUTINE INTRP(A,B,T,F,TT,FF,AA,BB,J)	INT00010
		INT00020
C	-----	INT00030
C	PURPOSE: TO DO FREQUENCY AND TEMPERATURE INTERPOLATION	INT00040
C	-----	INT00050
	DIMENSION A(9,3), B(9,3), F(9), T(3)	INT00060
		INT00070
C	IF(TT.LT.T(1)) TT=T(1)	INT00080
	DO 11 J=2,3	INT00090
	IF(TT.LT.T(J)) GO TO 14	INT00100
11	CONTINUE	INT00110
	TT=T(3)	INT00120
	J=3	INT00130
14	CONTINUE	INT00140
	DO 15 I=2,9	INT00150
	IF(FF.LT.F(I)) GO TO 16	INT00160
15	CONTINUE	INT00170
	FF=F(9)	INT00180
	I=9	INT00190
16	FF=ALOG10(FF)	INT00200
	F0=ALOG10(F(I))	INT00210
	F1=ALOG10(F(I-1))	INT00220
	FF0=(F0-FF)/(F0-F1)	INT00230
	FF1=(FF-F1)/(F0-F1)	INT00240
	TF0=(T(J)-TT)/(T(J)-T(J-1))	INT00250
	TF1=(TT-T(J-1))/(T(J)-T(J-1))	INT00260
	A11=ALOG10(A(I-1,J-1))	INT00270
	A01=ALOG10(A(I,J-1))	INT00280
	A10=ALOG10(A(I-1,J))	INT00290
	A00=ALOG10(A(I,J))	INT00300
	APJ1=A11*FF0+A01*FF1	INT00310
	APJ0=A10*FF0+A00*FF1	INT00320
	AA=APJ1*TF0+APJ0*TF1	INT00330
	B11=ALOG10(B(I-1,J-1))	INT00340
	B10=ALOG10(B(I-1,J))	INT00350
	B01=ALOG10(B(I,J-1))	INT00360
	B00=ALOG10(B(I,J))	INT00370
	BPJ1=B11*FF0+B01*FF1	INT00380
	BPJ0=B10*FF0+B00*FF1	INT00390
	BB=BPJ1*TF0+BPJ0*TF1	INT00400
	AA=10.**AA	INT00410
	BB=10.**BB	INT00420
	RETURN	INT00430
	END	INT00440

```

SUBROUTINE MMH20(V,T,PTOT,PH20,DATH20,ABH20)
C*****
C ROUTINE TO CALCULATE H2O VAPOR ABSORPTION FOR 0 TO 350 GHZ.
C INPUTS ARE: WAVENUMBER(/CM), TEMPERATURE(KELVIN), TOTAL
C PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), LINE DATA ARRAY.
C OUTPUT IS: H2O VAPOR ABSORPTION
C CALLED BY MMWGS  MAKES NO CALLS
C LOCAL VARIABLES:
C WCD  VAPOR COLUMN DENSITY(/CM/CM/KM)
C CT   LINE STRENGTH TEMPERATURE CORRECTION
C CA   LINE WIDTH SELF BROADENING AND TEMP. CORRECTION
C SA   CORRECTED LINE STRENGTH
C GA   CORRECTED LINE WIDTH
C ABS  SINGLE LINE ABSORPTION(/KM)
C*****
C DIMENSION DATH20(37,4)
C
C ABH20=0.
C WCD=7.33994E26*PH20/760./T*(PFR(T))
C CT=4.860773E-3*(T-296.)/T
C CA=(296./T)**.62*(PTOT+(4.*PH20))/760.
C
C DO 500 L=1,37
C   SA=DATH20(L,2)*WCD*EXP(DATH20(L,4)*CT)
C   GA=DATH20(L,3)*CA
C   ABS=SA*SUPK(V,DATH20(L,1),GA)
C   ABH20=ABH20+ABS
500 CONTINUE
RETURN
END

```

```

MMH00010
MMH00020
MMH00030
MMH00040
MMH00050
MMH00060
MMH00070
MMH00080
MMH00090
MMH00100
MMH00110
MMH00120
MMH00130
MMH00140
MMH00150
MMH00160
MMH00170
MMH00180
MMH00190
MMH00200
MMH00210
MMH00220
MMH00230
MMH00240
MMH00250
MMH00260
MMH00270
MMH00280
MMH00290
MMH00300
MMH00310
MMH00320
MMH00330
MMH00340
MMH00350

```

```

C      SUBROUTINE MMIDX(XL,T,ICE,H2OAB,H2OK2)
REF: RAY, APPLIED OPTICS, VOL. 11, P. 1836,(1972)
      COMPLEX CINDX,XK
      XXL=XL/10.
      TT=T-273.16
      T3=TT-25.
      IF(ICE.NE.0) GO TO 150
C  PARAMETERS FOR WATER
100  EFIN=5.27137+.0216474*TT-1.31198E-3*TT*TT
      ALFA=-16.8129/T+.0609265
      XLS=3.3836E-4*EXP(2513.98/T)
      SIGMA=1.25664E9
      ES=78.54*(1.-4.579E-3*T3+1.19E-5*T3*T3-2.8E-8*T3*T3*T3)
      GO TO 200
C
C  PARAMETERS FOR ICE
150  EFIN=3.168
      ALFA=0.288+0.0052*TT+2.3E-4*TT*TT
      XLS=9.990288E-5*EXP(1.32E4/(1.9869*T))
      SIGMA=1.26*EXP(-1.25E4/(1.9869*T))
      ES=203.168+2.5*TT+0.15*TT*TT
200  U=(ES-EFIN)*(XLS/XXL)**(1.-ALFA)
      Y=1.+2.*(XLS/XXL)**(1.-ALFA)*SIN(ALFA*1.57079633)+
      (XLS/XXL)**(2.-2*ALFA)
      EP=EFIN+((ES-EFIN)+U*SIN(ALFA*1.57079633))/Y
      EPP=(U*COS(ALFA*1.57079633))/Y+SIGMA*XXL/1.88496E11
      RE=SQRT((EP+SQRT(EP*EP+EPP*EPP))/2.)
      AI=-EPP/2./RE
      IF(ICE.NE.0) GO TO 400
C
      IF(XXL.LE..034) GO TO 307
      IF(XXL.GT..1) GO TO 311
306  R2=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,
      + 259913.,161.29,43319.7,27661.2)
      R2=R2+R2*T3*1.E-3*EXP((2.5E-5*XXL)**.25)
      RE=RE*(XXL-.034)/.066+R2*(.1-XXL)/.066
      GO TO 311
307  RE=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,
      + 259913.,161.29,43319.7,27661.2)
      RE=RE+RE*T3*1.E-3*EXP((2.5E-5*XXL)**.25)
311  CONTINUE
      IF(XXL.GT..3) GO TO 500
      AI=AI+AB(XXL,.25,300.,.47,3.)+AB(XXL,.39,17.,.45,1.3)
      + AB(XXL,.41,62.,.35,1.7)
      GO TO 500
C
400  CONTINUE
      IF(XXL.GT.0.08) GO TO 500
405  RC=DOP(XXL,1.225,1652.9,1.12082E6,46E-11,909.09,416441.,118852.,
      + 223.2,47031.8,126834.)
      RE=RE*(XXL-0.02)/0.06+R2*(0.08-XXL)/0.06
      AI=AI+AB(XXL,.242,62.,.23,1.6)+AB(XXL,.581,44.8,0.055,1.)
C
500  CINDX=CMPLX(RE,AI)
      XK=(CINDX*CINDX-1)/(CINDX*CINDX+2)
      H2OAB=AIMAG(-XK)
      H2OK2=XK*CONJG(XK)
      RETURN
      END
MMI00010
MMI00020
MMI00030
MMI00040
MMI00050
MMI00060
MMI00070
MMI00080
MMI00090
MMI00100
MMI00110
MMI00120
MMI00130
MMI00140
MMI00150
MMI00160
MMI00170
MMI00180
MMI00190
MMI00200
MMI00210
MMI00220
MMI00230
MMI00240
MMI00250
MMI00260
MMI00270
MMI00280
MMI00290
MMI00300
MMI00310
MMI00320
MMI00330
MMI00340
MMI00350
MMI00360
MMI00370
MMI00380
MMI00390
MMI00400
MMI00410
MMI00420
MMI00430
MMI00440
MMI00450
MMI00460
MMI00470
MMI00480
MMI00490
MMI00500
MMI00510
MMI00520
MMI00530
MMI00540
MMI00550
MMI00560
MMI00570
MMI00580
MMI00590

```

```

SUBROUTINE MMOXY(V,T,PTOT,PH2O,DATA02,ABS02)
*****
ROUTINE TO CALCULATE ABSORPTION DUE TO OXYGEN, METHOD IS THAT OF
LIEBE, GIMMESTAD, & HOPPONEN. IEEE TRANS. ANT. PROP. V.25, P327.
INPUTS ARE: FREQUENCY(GHZ), TEMPERATURE(KELVIN), TOTAL
PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), O2 LINE DATA
ARRAY.
OUTPUTS ARE: O2 ABSORPTION (1/KM)
CALLED FROM MMWGS CALLS NO OTHER ROUTINES.
LOCAL VARIABLES:
  T2      300./T
  PHI     TEMPERATURE CORRECTION FOR LINE STRENGTHS
  S       CORRECTED LINE STRENGTH(KHZ TORR)
  GAMMA   CORRECTED LINE WIDTH (1/GHZ)
  XIF     LINE INTERFERENCE FACTOR
  VMI     DATA02(L,1)-V
  VPL     DATA02(L,1)+V
  PROFIL  MODIFIED VANVLECK-WEISSKOPF LINE SHAPE
-----
DIMENSION DATA02(42,6)
ABS02=0.
T2=300./T
DO 500 L=1,42
  K=IFIX(DATA02(L,6))
  PHI=T2+T2+T2*EXP(-6.895E-3*K*(K+1)*(T2-1.))
  S=0.2095*PTOT*DATA02(L,2)*PHI
  GAMMA=DATA02(L,3)*( .929*PTOT+T2** .9+1.3*T2*PH2O)*1.E-3
  XIF=DATA02(L,4)*T2**DATA02(L,5)*PTOT*1.E-3
  VMI=DATA02(L,1)-V
  VPL=VMI+2*V
  PROFIL=(V/DATA02(L,1))*((GAMMA-VMI*XIF)/(VMI*VMI+GAMMA*GAMMA)+
  * (GAMMA-VPL*XIF)/(VPL*VPL+GAMMA*GAMMA))
  ABS02=ABS02+S*PROFIL*V*4.192E-5
500 CONTINUE
RETURN
END

```

```

MM000010
MM000020
MM000030
MM000040
MM000050
MM000060
MM000070
MM000080
MM000090
MM000100
MM000110
MM000120
MM000130
MM000140
MM000150
MM000160
MM000170
MM000180
MM000190
MM000200
MM000210
MM000220
MM000230
MM000240
MM000250
MM000260
MM000270
MM000280
MM000290
MM000300
MM000310
MM000320
MM000330
MM000340
MM000350
MM000360
MM000370
MM000380
MM000390
MM000400
MM000410
MM000420
MM000430
MM000440

```



```
IF<FREQ.GT.87.> IA=1
IF<FREQ.GT.82.> IB=1
DO 100 K=1,6
AA=AA+BSAT(K, IA)*<FREQ**<K-1>>
BB=BB+BSBT(K, IB)*<FREQ**<K-1>>
100 CONTINUE
AA=EXP(AA)
BSRAIN=AA*RAINRT**BB
200 RETURN
END
```

```
MMR00710
MMR00720
MMR00730
MMR00740
MMR00750
MMR00760
MMR00770
MMR00780
MMR00790
MMR00800
```

SUBROUTINE MMSNO(SNRT,TK,FQ,SNEX,SNBS)

PURPOSE: TO COMPUTE SNOW EXTINCTION AND BACKSCATTER X-SECTION.

INPUTS: SNOWFALL RATE(NM/HR), TEMPERATURE(KELVIN), FREQUENCY(GHZ)

OUTPUT: SNOW EXTINCTION (1/KM), BACKSCATTER(M**2/M**3).

CALLED FROM HMMW.

LOCAL VARIABLES:

FR,TS LOCAL FREQUENCY,TEMPERATURE
ITYPE INTEGER SNOW TYPE
XLMDA WAVELENGTH(MM)
FQ,1 FREQUENCY FITTING FACTORS
TFQ,1 TEMPERATURE FITTING FACTORS
IPJ0,1 INTERMEDIATE A VALUES
SQJ0,1 INTERMEDIATE B VALUES
AA,BB TERMS IN EXT(SNOW)=AA*SNOWRATE**BB
HRESLT TERM IN BSCAT(SNOW)=AA*SNOWRATE**1.8

DIMENSION R(9,3),B(9,3),F(9),T(3),BSAT(6,2),BSBT(6,2),SFCT(3)

DATA R 3.0E-3,2.75E-3,1.25E-2,2.50E-2,8.00E-2,1.65E-1,
2.60E-1,5.91E-1,1.68E+0,
2.07E-2,4.34E-2,1.60E-1,2.00E-1,3.10E-1,4.00E-1,
5.81E-1,6.50E-1,1.11E+0,
6.22E-2,9.67E-2,2.35E-1,3.41E-1,6.10E-1,8.52E-1,
7.83E-1,7.37E-1,5.76E-1/

DATA B 1.3,1.46,1.6,1.54,1.26,1.1, .89, .79, .6,

1.3,1.2, .95, .80, .75, .67, .65, .64, .60,

1.3,1.2, .95, .80, .75, .67, .65, .64, .60/

DATA F 10, 15, 35, 50, 95, 140, 225, 312, 1000./

DATA T 271, 273, 275, / SFCT 1, 3, 4./

DATA BSAT /-.8824881E+01, -.1029998E-01, +.2451205E-04,

+ .2462900E-07, +.6507628E-10, -.1856080E-12,

+ -.2127020E+02, +.6906017E+00, -.1924260E-01,

+ .3035233E-03, -.2545323E-05, +.8673581E-08/

DATA BSBT /+.7901887E+00, -.1900189E-02, +.6341350E-05,

+ .3186429E-08, -.5433950E-10, +.9056715E-13,

+ .1361993E+01, +.3628100E-01, -.2461284E-02,

+ .4805257E-04, -.3988064E-06, +.1193458E-08/

TS=TK

FS=FQ

ICE=1

SNEX=0.

SNBS=0.

IF(FQ.LT.10.) GO TO 200

XLMDA=299.79/FQ

CALL INTRP(A,B,T,F,TS,FS,AA,BB,J)

SNEX=AA*SNRT**BB

CALCULATIONS FOR SNOW BACKSCATTER.

AA=0.

BB=0.

IA=2

IB=2

IF(FQ.GT.87.) IA=1

IF(FQ.GT.82.) IB=1

DO 100 K=1,6

AA=AA+BSAT(K,IA)*(FQ**(K-1))

BB=BB+BSBT(K,IB)*(FQ**(K-1))

CONTINUE

AA=EXP(AA)

BB=1.2*BB

F1=(FQ-10.)/85.

FCT=.367+F1*.633

MMS00010
MMS00020
MMS00030
MMS00040
MMS00050
MMS00060
MMS00070
MMS00080
MMS00090
MMS00100
MMS00110
MMS00120
MMS00130
MMS00140
MMS00150
MMS00160
MMS00170
MMS00180
MMS00190
MMS00200
MMS00210
MMS00220
MMS00230
MMS00240
MMS00250
MMS00260
MMS00270
MMS00280
MMS00290
MMS00300
MMS00310
MMS00320
MMS00330
MMS00340
MMS00350
MMS00360
MMS00370
MMS00380
MMS00390
MMS00400
MMS00410
MMS00420
MMS00430
MMS00440
MMS00450
MMS00460
MMS00470
MMS00480
MMS00490
MMS00500
MMS00510
MMS00520
MMS00530
MMS00540
MMS00550
MMS00560
MMS00570
MMS00580
MMS00590
MMS00600
MMS00610
MMS00620
MMS00630
MMS00640
MMS00650
MMS00660
MMS00670
MMS00680
MMS00690
MMS00700

IF(FQ.GT.95.) FCT=1.
AA=FCT*AA
110 IF(TK.GE.275.) J=4
C SNBS=SFCT(J-1)*AA*SNRT**BB
C 200 RETURN
C END

MMS00710
MMS00720
MMS00730
MMS00740
MMS00750
MMS00760
MMS00770
MMS00780

```

SUBROUTINE MMWFG(FD, T, FREQ, GFOG, BSFOG)
*****
CALCULATES ABSORPTION DUE TO WATER FOGS/CLOUDS,
AND BACKSCATTER CROSS SECTION IN M**2/M**3.
INPUTS ARE: FOG DENSITY(GM/M**3), TEMPURTURE(KELVIN),
FREQUENCY(GHZ).
OUTPUTS ARE: FOG ABSORPTION(/KM), BACKSCATTER X-SECTION(M**2 M**3).
MMFOG IS CALLED FROM MMWMOD CALLS MMIDX SUBROUTINE.
LOCAL VARIABLES:
XLMDA WAVELENGTH(MM)
*****
ICE=0
IF(T.LT.243.) ICE=1
C COMPUTE FOG EXTINCTION
XLMDA=10./((FREQ/29.98)
CALL MMIDX(XLMDA, T, ICE, H2OAB, H2OK2)
GFOG=18.8498*H2OAB*FD/XLMDA
BSFOG=1.162E-06*H2OK2*FD**1.75/(XLMDA**4)
RETURN
END

```

```

MMF00010
MMF00020
MMF00030
MMF00040
MMF00050
MMF00060
MMF00070
MMF00080
MMF00090
MMF00100
MMF00110
MMF00120
MMF00130
MMF00140
MMF00150
MMF00160
MMF00170
MMF00180
MMF00190
MMF00200
MMF00210
MMF00220
MMF00230
MMF00240
MMF00250
MMF00260
MMF00270

```

```

SUBROUTINE MMWGS(T,P,AH,FRQ,GAS)
*****
SUBROUTINE COMPUTES GASEOUS ABSORPTION FROM 0 TO 1000 GHZ FOR
H2O VAPOR AND O2.
INPUTS INCLUDE: TEMPERATURE(KELVIN), PRESSURE(TORR), ABSOLUTE HUMIDITY
IN GM/M**3, FREQUENCY(GIGAHERTZ).
OUTPUTS ARE: GAS ABSORPTION (1/KM)
MMWGS IS CALLED FROM NMMW, CALLS (1)MMOXY(OXYGEN ABSORPTION),
(2) MMH2O(H2O VAPOR ABSORPTION).
LOCAL VARIABLES:
DATAO2(L,J): OXYGEN LINE DATA, L=LINE NUMBER, J=TYPE:
      J=1 : LINE FREQUENCY(GHZ)
      2 : LINE STRENGTH AT 300K
      3 : LINE WIDTH AT 300K (GHZ/TORR)
      4 : INTERFERENCE PARAMETER AT 300K
      5 : INTERFERENCE TEMPERATURE CORRECTION
      6 : LINE QUANTUM PARAMETER
DATH2O(L,J) : H2O LINE DATA, L=LINE NUMBER, J=TYPE:
      1 : WAVENUMBER(1/CM)
      2 : STRENGTH
      3 : WIDTH(1/CM/TORR)
      4 : GROUND STATE ENERGY
GH2O      : H2O VAPOR ABSORPTION (1/KM)
GO2       : O2 ABSORPTION (1/KM)
PH2O      : H2O VAPOR PRESSURE(TORR)
WVNMB     : WAVENUMBER(1/CM)
*****
DIMENSION DATAO2(42,6), DATH2O(37,4)
DIMENSION D1O2(42,3), D2O2(42,3), D1H2O(37,2), D2H2O(37,2)
EQUIVALENCE (DATAO2(1,1),D1O2(1,1)),(DATAO2(1,4),D2O2(1,1))
EQUIVALENCE (DATH2O(1,1),D1H2O(1,1)),(DATH2O(1,3),D2H2O(1,1))
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUM
*****
DATA D1O2/
1 49.451,49.961,50.473,50.987,51.503,52.021,52.542,53.066,
2 53.595,54.129,54.671,55.221,55.793,56.264,56.363,56.968,
3 57.612,58.323,58.446,59.164,59.590,60.306,60.434,61.150,
4 61.800,62.411,62.486,62.998,63.568,64.127,64.678,65.224,
5 65.764,66.302,66.836,67.369,67.900,68.430,68.960,69.488,
6 70.016,118.75,
1 7.E-5,2.2E-4,6.E-4,1.56E-3,3.86E-3,8.99E-3,1.971E-2,.04072,
2 .07919,.1448,.2489,.4012,.6056,.3487,.8539,1.1204,1.3595,
3 1.515,.9251,1.5263,1.341,1.3487,1.5626,1.5899,1.4588,1.2272,
4 .9634,.954,.6898,.4656,.2942,.1744,.0971,.0508,.025,.0116,
5 5.08E-3,2.1E-3,8.2E-4,3.E-4,1.E-4,.5973,
1 1.260,1.310,1.330,1.360,1.380,1.410,1.440,1.460,1.490,
2 1.510,1.540,1.570,1.601,2.212,1.635,1.672,1.714,1.762,
3 1.964,1.819,1.859,1.890,1.789,1.736,1.694,1.658,1.990,
4 1.627,1.598,1.568,1.540,1.510,1.490,1.460,1.440,1.410,
5 1.380,1.360,1.330,1.310,1.280,2.140,
DATA D2O2/
1 0.000,0.000,0.000,1.040,0.802,0.897,0.825,0.780,0.764,
2 0.666,0.651,0.550,0.481,0.931,0.371,0.254,0.100,-.087,
3 0.729,-.318,0.433,-.543,0.179,-.028,-.183,-.324,-.615,
4 -.419,-.537,-.591,-.693,-.703,-.796,-.808,-.849,-.916,
5 -.822,-1.05,0.000,0.000,0.000,-.054,
1 1.00,1.00,1.00,1.38,2.04,1.69,1.91,1.88,1.90,2.01,1.95,
2 2.11,2.13,0.89,2.36,2.66,4.20,-5.8,0.79,0.11,0.50,0.69,
3 -.99,7.60,3.04,2.34,0.85,2.24,2.02,2.04,1.89,1.95,1.85,
4 1.83,1.86,1.66,1.99,1.36,1.00,1.00,1.00,0.89,
1 41.,39.,37.,35.,33.,31.,29.,27.,25.,23.,21.,19.,17.,15.,13.,
2 11.,9.,7.,5.,3.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,
3 27.,29.,31.,33.,35.,37.,39.,41.,1./
*****

```

```

DATA D1H2O/
1 00.742,06.115,06.790,10.715,10.846,12.682,14.778,14.944,
2 15.707,15.834,16.294,18.270,18.577,20.704,21.960,24.860,
3 25.085,30.560,32.366,32.954,36.604,37.137,37.910,38.638,
4 38.791,39.112,40.282,40.520,42.639,43.243,43.631,44.100,
5 46.750,47.053,48.059,49.765,49.820,
1 .436E-24,.775E-22,.186E-24,.250E-23,.906E-22,.827E-21,
2 .145E-22,.863E-21,.270E-22,.108E-21,.219E-22,.985E-22,
3 .526E-19,.563E-21,.531E-22,.664E-22,.347E-19,.143E-20,
4 .160E-20,.253E-19,.164E-18,.502E-19,.333E-21,.242E-20,
5 .179E-18,.197E-21,.558E-19,.155E-21,.707E-21,.687E-21,
6 .511E-22,.568E-20,.302E-21,.142E-18,.930E-21,.399E-22,
7 .478E-22)
DATA D2H2O/
1 .081,.094,.095,.063,.087,.091,.050,.083,.061,.071,.075,
2 .111,.107,.072,.111,.103,.102,.084,.083,.101,.097,.099,
3 .094,.073,.094,.063,.093,.098,.066,.074,.060,.081,.084,
4 .091,.078,.096,.097,
1 446.512,136.164,134.800,1284.921,315.780,212.156,
2 1045.069,285.419,742.079,488.136,586.482,23.750,
3 23.794,488.110,1618.550,69.920,70.091,285.219,
4 383.843,37.137,136.761,0.000,172.880,610.345,
5 173.366,888.641,275.498,1731.890,888.607,842.361,
6 1079.088,508.814,398.392,399.459,601.553,100.391,
7 1693.650/
C *****
C
C COMPUTE WATER VAPOR PRESSURE, FREQUENCY IN WAVENUMBERS
PH20=AH*T*3.462977E-3
IF(AH.LT.0.) PH20=-PSAT(T)*AH/100.
WVNMB=FRQ/29.98
C
C COMPUTE H2O ABSORPTION
CALL MMH2O(WVNMB,T,P,PH20,DATH20,GH20)
C
C COMPUTE O2 ABSORPTION
IF(FRQ.LT.140.) CALL MMOXY(FRQ,T,P,PH20,DATA02,G02)
C
C SUM ABSORPTION
GAS=GH20+G02
C
C *****
RETURN
END
MMG00710
MMG00720
MMG00730
MMG00740
MMG00750
MMG00760
MMG00770
MMG00780
MMG00790
MMG00800
MMG00810
MMG00820
MMG00830
MMG00840
MMG00850
MMG00860
MMG00870
MMG00880
MMG00890
MMG00900
MMG00910
MMG00920
MMG00930
MMG00940
MMG00950
MMG00960
MMG00970
MMG00980
MMG00990
MMG01000
MMG01010
MMG01020
MMG01030
MMG01040
MMG01050
MMG01060
MMG01070
MMG01080
MMG01090
MMG01100
MMG01110
MMG01120
MMG01130
MMG01140

```

C	FUNCTION PFR(T)	PFR00010
C	COMPUTE H2O PARTITION FUNCTION CORRECTIONS	PFR00020
	DIMENSION VIB(3)	PFR00030
C	DATA VIB/3693.9,1614.5,3801.8/	PFR00040
	QJ=296./T	PFR00050
	QJ=QJ*SQRT(QJ)	PFR00060
	T1=-1.43879/296.	PFR00070
	T2=-1.43879/T	PFR00080
	T1S=1.	PFR00090
	T2S=1.	PFR00100
C	DO 10 J=1,3	PFR00110
	V=VIB(J)	PFR00120
	T11=1.-EXP(T1*V)	PFR00130
	T22=1.-EXP(T2*V)	PFR00140
	T1S=T1S*T11	PFR00150
10	T2S=T2S*T22	PFR00160
C	PFR=QJ*T2S/T1S	PFR00170
	RETURN	PFR00180
	END	PFR00190
		PFR00200
		PFR00210
		PFR00220
		PFR00230


```

C
FUNCTION PSAT(T)
+ DATA C1,C2,C3,C4,C5,C6,C7/-7.90298,5.02808,-1.3816E-7,11.344,
+ 8.1328E-3,-3.49149,3.005715/
DATA D1,D2,D3,D4/-9.09718,-3.56654,.876793,.785835/
DATA TS,TO/373.16,273.16/,CONV/.7500646/
IF(T.LE.TO) GO TO 100
TR=TS/T
TRI=T/TS
EW=C1*(TR-1.)+C2*ALOG10(TR)+C3*(10.**((C4*(1.-TRI))-1.))+
+ C5*(10.**((C6*(TR-1.))-1.))+C7
GO TO 200
100 TR=TO/T
TRI=T/TO
EW=D1*(TR-1.)+D2*ALOG10(TR)+D3*(1.-TRI)+D4
200 PSAT=(10.**EW)*CONV
RETURN
END

```

```

PSAT0010
PSAT0020
PSAT0030
PSAT0040
PSAT0050
PSAT0060
PSAT0070
PSAT0080
PSAT0090
PSAT0100
PSAT0110
PSAT0120
PSAT0130
PSAT0140
PSAT0150
PSAT0160
PSAT0170
PSAT0180

```

	FUNCTION SUPK(A,B,C)	SUPK0010
	COMPUTES THE SUPER KINETIC LINE PROFILE FACTOR.	SUPK0020
C	PI=3.14159265	SUPK0030
	XNORM=.998776	SUPK0040
	VM=10.*C	SUPK0050
	X=ABS((B+B-A*A)/(2.*C*A))	SUPK0060
	IF(ABS(B-A).GT.VM) GO TO 10	SUPK0070
C	SUPK=XNORM/(PI*C)/(X*X+1.)	SUPK0080
	RETURN	SUPK0090
C		SUPK0100
	CXI=10.*(B+10.*C/2.)/(B+10.*C)	SUPK0110
	CXI=(((CXI**1.88)+1.)/((CXI*CXI)+1.))	SUPK0120
10	SUPK=XNORM*CXI/(PI*C)/((X**1.88)+1.)	SUPK0130
	RETURN	SUPK0140
	END	SUPK0150
		SUPK0160
		SUPK0170

```

SUBROUTINE CLTRAN(CTrans,WAVE,IRUN,IERR)
DIMENSION TAU(20)
DIMENSION XI(20),YI(20),ZI(20),X0(20),Y0(20),Z0(20)
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
+ WNDVEL,WNDDIR,IPASCT
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTIL(20)
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
COMMON /CYL/XC,YC,RADIUS
COMMON /IQUANT/IQIN,IQOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,HPLCTU
COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)
+ ,RADICL(10)
COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)
C-----
C** DATA IS READ FROM INPUT RECORDS AND THEN TRANSFERRED TO CLTRAN
C** BY SUBROUTINE CLREAD. INPUT TO CLTRAN IS CARD ORDER-INDEPENDENT,
C** WITH A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH INPUT RECORD.
C** THE ONLY EXCEPTION TO THIS ORDER-INDEPENDENCE IS THE GO SENTINEL
C** CARD, WHICH MUST BE THE LAST RECORD READ. ALL CARDS ARE READ IN
C** UNDER THE FORMAT (A4,1X,5(E10.5,1X)), THE IDENTIFICATION AND MEANING
C** OF EACH INPUT RECORD ARE AS FOLLOWS :
C-----
C CARD IDENTIFIER : SEEK
C VARIABLES READ : XS,YS,ZS
C XS,YS,ZS = POSITION COORDINATES OF SEEKER (OR RECEIVER) (KM)
C-----
C CARD IDENTIFIER : TARG
C VARIABLES READ : XT,YT,ZT
C XT,YT,ZT = POSITION COORDINATES OF TARGET (KM)
C-----
C** THE FOLLOWING FOUR CARDS REPRESENT THE STRATIFORM CLOUDS TREATED
C** BY CLTRAN. THEY ARE TREATED AS INFINITE LAYERS IN A PLANE-PARALLEL
C** ATMOSPHERE.
C-----
C CARD IDENTIFIER : CLST (STRATUS CLOUD TYPE)
C VARIABLES READ : ZLBASE,ZLTHIC
C ZLBASE = HEIGHT OF CLOUD BASE (KM)
C ZLTHIC = VERTICAL THICKNESS OF CLOUD LAYER (KM)
C-----
C CARD IDENTIFIER : CLAS (ALTOSTRATUS CLOUD TYPE)
C VARIABLES READ : ZLBASE,ZLTHIC
C-----
C CARD IDENTIFIER : CLNS (NIMBOSTRATUS CLOUD TYPE)
C VARIABLES READ : ZLBASE,ZLTHIC
C-----
C CARD IDENTIFIER : CLSC (STRATOCUMULUS CLOUD TYPE)
C VARIABLES READ : ZLBASE,ZLTHIC
C-----
C** THE NEXT TWO CARDS REPRESENT THE CUMULUS CLOUD TYPES ADDRESSED
C** BY CLTRAN. CLOUDS OF THIS KIND ARE MODELLED AS CYLINDERS WHICH
C** HAVE VERTICAL SYMMETRY AXES. THESE TYPES ARE REPRESENTED BY
C** THE FOLLOWING RECORDS :
C-----
C CARD IDENTIFIER : CLCH (CUMULUS HUMILIS CLOUD TYPE)
C VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD
C ZCBASE = HEIGHT OF CLOUD CYLINDER'S LOWER BASE (KM)
C ZCTHIC = VERTICAL THICKNESS OF CLOUD CYLINDER (KM)
C RADICL = RADIUS OF CLOUD CYLINDER (KM)
C XCLOUD = X-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM)
C YCLOUD = Y-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM)
C-----
C CARD IDENTIFIER : CLCC (CUMULUS CONGESTUS CLOUD TYPE)
C VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD
C-----
C** THE FOLLOWING CARD MUST BE THE LAST RECORD READ :

```

```

CLT00010
CLT00020
CLT00030
CLT00040
CLT00050
CLT00060
CLT00070
CLT00080
CLT00090
CLT00100
CLT00110
CLT00120
CLT00130
CLT00140
CLT00150
CLT00160
CLT00170
CLT00180
CLT00190
CLT00200
CLT00210
CLT00220
CLT00230
CLT00240
CLT00250
CLT00260
CLT00270
CLT00280
CLT00290
CLT00300
CLT00310
CLT00320
CLT00330
CLT00340
CLT00350
CLT00360
CLT00370
CLT00380
CLT00390
CLT00400
CLT00410
CLT00420
CLT00430
CLT00440
CLT00450
CLT00460
CLT00470
CLT00480
CLT00490
CLT00500
CLT00510
CLT00520
CLT00530
CLT00540
CLT00550
CLT00560
CLT00570
CLT00580
CLT00590
CLT00600
CLT00610
CLT00620
CLT00630
CLT00640
CLT00650
CLT00660
CLT00670
CLT00680

```

```

C-----CLT00690
C-----CLT00700
C-----CLT00710
CARD IDENTIFIER : GO          CLT00720
VARIABLES READ : NONE        CLT00730
C-----CLT00740
IF<IRUN.GT.1> GO TO 5         CLT00750
NCLAY=0                      CLT00760
NCCLDS=0                     CLT00770
5 CONTINUE                   CLT00780
IUV=0
ZBMIN=999.
FOGPRB=0.
IF<WAVE.GT.0.20.AND.WAVE.LT.2.00> IUV=1 CLT00790
IF<WAVE.GT.3.00.AND.WAVE.LT.5.00> IUV=2 CLT00800
IF<WAVE.GT.8.0.AND.WAVE.LT.12.0> IUV=3 CLT00810
IF<IUV.NE.0> GO TO 50        CLT00820
IERR=1                       CLT00830
WRITE<IOOUT,22>              CLT00840
22 FORMAT<1H0,20X,94H***CLTRAN ERROR*** INPUT WAVELENGTH DOES NOT LIE CLT00850
+ WITHIN ALLOWABLE LIMITS, EXECUTION TERMINATED />
GO TO 900                    CLT00860
50 CALL CLREAD<NCLAY,NCCLDS,IERR> CLT00870
IF<IERR.EQ.1> GO TO 900      CLT00880
ISLTUP=0                     CLT00890
ISLTDN=0                     CLT00900
IHORIZ=0                     CLT00910
IVERT=0                      CLT00920
C-----CLT00930
C-----CLT00940
C**** DETERMINE SENSE OF L-O-S SLOPE FROM SEEKER'S POINT OF VIEW CLT00950
C-----CLT00960
TESVER=(XS-XT)**2+(YS-YT)**2 CLT00970
IF<TESVER.EQ.0.0> IVERT=1    CLT00980
IF<IVERT.EQ.1> GO TO 300     CLT00990
IF<ZS-ZT>200,210,220        CLT01000
200 ISLTUP=1                 CLT01010
GO TO 300                    CLT01020
210 IHORIZ=1                 CLT01030
GO TO 300                    CLT01040
220 ISLTDN=1                 CLT01050
300 CONTINUE                 CLT01060
IF<IHORIZ.EQ.1> GO TO 310    CLT01070
C-----CLT01080
C**** COMPUTE L-O-S SLOPES IN X-Z AND Y-Z VERTICAL PLANES CLT01090
C-----CLT01100
XIX=(XS-XT)/<ZS-ZT>         CLT01110
XIY=(YS-YT)/<ZS-ZT>         CLT01120
310 CONTINUE                 CLT01130
C-----CLT01140
C**** STRATIFORM CLOUD BLOCK CLT01150
C-----CLT01160
IF<NCLAY.EQ.0> GO TO 500     CLT01170
C-----CLT01180
C**** UTILIZE DEFAULT BASE OR THICKNESS VALUES IF NECESSARY CLT01190
C-----CLT01200
CALL DEFSET<1,NCLAY>        CLT01210
DO 400 N=1,NCLAY           CLT01220
NLINT(N)=0                 CLT01230
Z0(N)=ZLBASE(N)           CLT01240
Z1(N)=Z0(N)+ZLTHIC(N)     CLT01250
ZBAS=Z0(N)                 CLT01260
ZTOP=Z1(N)                 CLT01270
ICL=ILCTYP(N)             CLT01280
IF<ZBMIN.LT.ZLBASE(N)> GO TO 320
ZBMIN=ZLBASE(N)
FOGPRB=FLOAT<ICL>
320 CONTINUE
C-----CLT01290
C**** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CLOUD LAYER <IF CLT01300
C**** THERE ARE ANY>: <X1(N),Y1(N),Z1(N)>= UPPER INTERSECTION POINT, CLT01310
C**** <X0(N),Y0(N),Z0(N)>= LOWER INTERSECTION POINT CLT01320

```

```

C          CALL LAYRXY(XIX,XIY,N)
          IF( TESALL.LE.0.0)GO TO 350
          NLINT(N)=N
C**** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH
          CALL CLEXTN(TAUN,N)
          TAU(N)=TAUN
          ATTL(N)=SQRT((X1(N)-X0(N))**2+(Y1(N)-Y0(N))**2+(Z1(N)-Z0(N))**2)
          GO TO 400
C**** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH
C**** IS NOW SET TO ZERO
350 TAU(N)=0.0
400 ATTL(N)=0.0
      CONTINUE
C**** END STRATIFORM CLOUD BLOCK
C**** CUMULIFORM CLOUD BLOCK
500 IF(NCCLDS.EQ.0)GO TO 805
C**** UTILIZE DEFAULT BASE, THICKNESS, OR RADIUS IF NECESSARY
          CALL DEFSET(2,NCCLDS)
          DO 800 N=1,NCCLDS
            NN=N+NLAYER
            NCINT(N)=0
            Z0(NN)=ZCBASE(N)
            Z1(NN)=Z0(NN)+ZCTHIC(N)
            ICL=ICCTYP(N)
            IF(ZBMIN.LT.ZCBASE(N)) GO TO 520
            ZBMIN=ZCBASE(N)
            FOGPRB=FLOAT(ICL)
520 CONTINUE
            RADIUS=RADICL(N)
            XC=XCLOUD(N)
            YC=YCLOUD(N)
            ZBAS=Z0(NN)
            ZTOP=Z1(NN)
            TESARG=0.0
            TESVT=0.0
C**** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CUMULUS CLOUD CYLINDER
C**** (IF THERE ARE ANY); (X1(N),Y1(N),Z1(N))= UPPER INTERSECTION POINT,
C**** (X0(N),Y0(N),Z0(N))= LOWER INTERSECTION POINT
          CALL CYLXY(XIX,XIY,NN)
          IF( TESALL.LE.0.0.OR.TESARG.LT.0.0.OR.TESVT.LT.0.0)GO TO 650
          NCINT(N)=N
C**** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH
          CALL CLEXTN(TAUN,NN)
          TAU(NN)=TAUN
          ATTL(NN)=SQRT((X1(NN)-X0(NN))**2+(Y1(NN)-Y0(NN))**2+
          +(Z1(NN)-Z0(NN))**2)
          GO TO 800
C**** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH
C**** IS NOW SET TO ZERO
650 TAU(NN)=0.0
800 ATTL(NN)=0.0
      CONTINUE
C**** END CUMULIFORM CLOUD BLOCK

```

```

CLT01330
CLT01340
CLT01350
CLT01360
CLT01370
CLT01380
CLT01390
CLT01400
CLT01410
CLT01420
CLT01430
CLT01440
CLT01450
CLT01460
CLT01470
CLT01480
CLT01490
CLT01500
CLT01510
CLT01520
CLT01530
CLT01540
CLT01550
CLT01560
CLT01570
CLT01580
CLT01590
CLT01600
CLT01610
CLT01620
CLT01630
CLT01640
CLT01650
CLT01660
CLT01670
CLT01680
CLT01690
CLT01700
CLT01710
CLT01720
CLT01730
CLT01740
CLT01750
CLT01760
CLT01770
CLT01780
CLT01790
CLT01800
CLT01810
CLT01820
CLT01830
CLT01840
CLT01850
CLT01860
CLT01870
CLT01880
CLT01890
CLT01900
CLT01910
CLT01920
CLT01930
CLT01940
CLT01950
CLT01960
CLT01970
CLT01980

```

C
805 CONTINUE
C**** DETERMINE CUMULATIVE CLOUD OPTICAL DEPTH AND TRANSMITTANCE
TAUTOT=0.0
NN=Nlayer+NCCLDS
IF(NN.EQ.0)GO TO 820
DO 810 N=1,NN
810 TAUTOT=TAUTOT+TAU(N)
820 CTRANS=EXP(-TAUTOT)
850 CALL LISOUT(NN,CTrans,TAUTOT,TAU,Nlayer,IRUN)
900 RETURN
END

CLT01990
CLT02000
CLT02010
CLT02020
CLT02030
CLT02040
CLT02050
CLT02060
CLT02070
CLT02080
CLT02090
CLT02100

C

```

SUBROUTINE LAYRXY(XIX,XIY,IN)
LOGICAL INSEEK,INTARG
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
C**** CHECK LOS TO SEE IF IT INTERSECTS THE CLOUD LAYER
UP1=ISLTUP*(ZS-Z1(IN))
UP2=ISLTUP*(Z0(IN)-ZT)
DN1=ISLTDN*(ZT-Z1(IN))
DN2=ISLTDN*(Z0(IN)-ZS)
HOR12=IHORIZ*(ZS-Z1(IN))*(ZS-Z0(IN))*(-1,0)
UPVER1=IVERT*(ZT-ZS)*(ZS-Z1(IN))
UPVER2=IVERT*(ZT-ZS)*(Z0(IN)-ZT)
TESALL=UP1*UP2+DN1*DN2+HOR12+UPVER1*UPVER2
IF(TESALL.LT.0.0)RETURN
IF(IHORIZ.EQ.1)RETURN
C
C**** COMPUTE X,Y INTERSECTIONS OF CLOUD PLANES AND LOS
C**** ALSO, CHECK FOR THE CASE WHERE EITHER THE SEEKER
C**** OR TARGET IS INSIDE OF THE CLOUD
INSEEK=ZS.LE.Z1(IN).AND.ZS.GE.Z0(IN)
INTARG=ZT.LE.Z1(IN).AND.ZT.GE.Z0(IN)
IF(ISLTUP.EQ.1.AND.INTARG)Z1(IN)=ZT
IF(ISLTDN.EQ.1.AND.INTARG)Z0(IN)=ZT
IF(ISLTUP.EQ.1.AND.INSEEK)Z0(IN)=ZS
IF(ISLTDN.EQ.1.AND.INSEEK)Z1(IN)=ZS
X(IN)=XS+XIX*(Z1(IN)-ZS)
Y(IN)=YS+XIY*(Z1(IN)-ZS)
X0(IN)=XS+XIX*(Z0(IN)-ZS)
Y0(IN)=YS+XIY*(Z0(IN)-ZS)
RETURN
END

```

```

LAY00010
LAY00020
LAY00030
LAY00040
LAY00050
LAY00060
LAY00070
LAY00080
LAY00090
LAY00100
LAY00110
LAY00120
LAY00130
LAY00140
LAY00150
LAY00160
LAY00170
LAY00180
LAY00190
LAY00200
LAY00210
LAY00220
LAY00230
LAY00240
LAY00250
LAY00260
LAY00270
LAY00280
LAY00290
LAY00300
LAY00310
LAY00320
LAY00330

```

```

SUBROUTINE CYLXY(XIX,XIY,IN)
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
COMMON /CYL/XC,YC,RADIUS
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTLL(20)
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
C**** FIRST DETERMINE INTERSECTION PTS OF LOS WITH UPPER
C**** AND LOWER BASE PLANES OF CLOUD CYLINDER
TESALL=0.0
CALL LAYRXY(XIX,XIY,IN)
IF(TESALL.LT.0.0)RETURN
X0LAYR=X0(IN)
Y0LAYR=Y0(IN)
Z0LAYR=Z0(IN)
X1LAYR=X1(IN)
Y1LAYR=Y1(IN)
Z1LAYR=Z1(IN)
C**** CALCULATE NEXT THE INTERSECTION PTS OF THE LOS
C**** WITH THE SURFACE OF AN INFINITE VERTICAL CYLINDER WITH
C**** THE SAME RADIUS (RADIUS) AND LATERAL POSITION (XC,YC) AS THE CLOUD
IF(IVERT.NE.1)GO TO 40
VERTES=SQRT((XT-XC)**2+(YT-YC)**2)
TESVT=RADIUS-VERTES
IF(VERTES.LT.RADIUS)GO TO 400
RETURN
40 IF((XT-XS).EQ.0.0)GO TO 60
ALPHA=(YT-YS)/(XT-XS)
A2=ALPHA**2
C2=1.0+A2
C1=2.0*(ALPHA*(YT-YC-ALPHA*XT)-XC)
C0=XC**2+(YT-YC-ALPHA*XT)**2-RADIUS**2
TESARG=C1**2-4.0*C2*C0
IF(TESARG.LT.0.0)RETURN
XP=(-C1+SQRT(TESARG))/(2.0*C2)
XM=(-C1-SQRT(TESARG))/(2.0*C2)
YP=YT+ALPHA*(XP-XT)
YM=YT+ALPHA*(XM-XT)
IF(IHORIZ.EQ.1)GO TO 300
ZM=ZS+(1.0/XIX)*(XM-XS)
ZP=ZS+(1.0/XIX)*(XP-XS)
C**** CHECK FOR SKEW MISS OF CLOUD
TOPP=ZP-ZTOP
TOPM=ZM-ZTOP
BASP=ZBAS-ZP
BASM=ZBAS-ZM
IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50
TESALL=-1.0
RETURN
60 XP=XT
TESARG=RADIUS**2-(XP-XT)**2
IF(TESARG.LT.0.0)RETURN
YP=YC+SQRT(TESARG)
XM=XT
YM=YC+SQRT(TESARG)
IF(IHORIZ.EQ.1)GO TO 300
ZM=ZS+(1.0/XIY)*(YM-YS)
ZP=ZS+(1.0/XIY)*(YP-YS)
C**** CHECK FOR SKEW MISS OF CLOUD
TOPP=ZP-ZTOP
TOPM=ZM-ZTOP
BASP=ZBAS-ZP
BASM=ZBAS-ZM
IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50
TESALL=-1.0
RETURN
50 IF(ZP.LT.ZM)GO TO 100
X0CYL=XM
Y0CYL=YM
Z0CYL=ZM
X1CYL=XP

```

```

CYL00010
CYL00020
CYL00030
CYL00040
CYL00050
CYL00060
CYL00070
CYL00080
CYL00090
CYL00100
CYL00110
CYL00120
CYL00130
CYL00140
CYL00150
CYL00160
CYL00170
CYL00180
CYL00190
CYL00200
CYL00210
CYL00220
CYL00230
CYL00240
CYL00250
CYL00260
CYL00270
CYL00280
CYL00290
CYL00300
CYL00310
CYL00320
CYL00330
CYL00340
CYL00350
CYL00360
CYL00370
CYL00380
CYL00390
CYL00400
CYL00410
CYL00420
CYL00430
CYL00440
CYL00450
CYL00460
CYL00470
CYL00480
CYL00490
CYL00500
CYL00510
CYL00520
CYL00530
CYL00540
CYL00550
CYL00560
CYL00570
CYL00580
CYL00590
CYL00600
CYL00610
CYL00620
CYL00630
CYL00640
CYL00650
CYL00660
CYL00670
CYL00680
CYL00690
CYL00700

```



```

Y1CYL=YP
Z1CYL=ZP
GO TO 200
100 X0CYL=XP
Y0CYL=YP
Z0CYL=ZP
X1CYL=XM
Y1CYL=YM
Z1CYL=ZM
200 CONTINUE
GO TO 450
300 X0(IN)=XM
Y0(IN)=YM
Z0(IN)=ZT
X1(IN)=XP
Y1(IN)=YP
Z1(IN)=ZT
GO TO 500
400 X0(IN)=XT
Y0(IN)=YT
X1(IN)=XT
Y1(IN)=YT
GO TO 500
450 IF(Z0CYL.LT.Z0LAYR)GO TO 460
X0(IN)=X0CYL
Y0(IN)=Y0CYL
Z0(IN)=Z0CYL
GO TO 470
460 X0(IN)=X0LAYR
Y0(IN)=Y0LAYR
Z0(IN)=Z0LAYR
470 IF(Z1CYL.GT.Z1LAYR)GO TO 480
X1(IN)=X1CYL
Y1(IN)=Y1CYL
Z1(IN)=Z1CYL
GO TO 500
480 X1(IN)=X1LAYR
Y1(IN)=Y1LAYR
Z1(IN)=Z1LAYR
500 RETURN
END

```

```

CYL00710
CYL00720
CYL00730
CYL00740
CYL00750
CYL00760
CYL00770
CYL00780
CYL00790
CYL00800
CYL00810
CYL00820
CYL00830
CYL00840
CYL00850
CYL00860
CYL00870
CYL00880
CYL00890
CYL00900
CYL00910
CYL00920
CYL00930
CYL00940
CYL00950
CYL00960
CYL00970
CYL00980
CYL00990
CYL01000
CYL01010
CYL01020
CYL01030
CYL01040
CYL01050
CYL01060
CYL01070
CYL01080
CYL01090
CYL01100
CYL01110

```

```

SUBROUTINE CLEXTN(TAUN, IN)
DIMENSION AA(42), BB(42), CC(42), A(7,6,3)
DIMENSION X1(20), Y1(20), Z1(20), X0(20), Y0(20), Z0(20)
EQUIVALENCE (AA(1), A(1,1,1)), (BB(1), A(1,1,2)), (CC(1), A(1,1,3))
COMMON /PATHL/X0, Y0, Z0, X1, Y1, Z1, XS, YS, ZS, XT, YT, ZT, ATTL(20)
COMMON /LEVEL/ISLTUP, ISLTDN, IHORIZ, IVERT, TESALL, TESARG, TESVT
COMMON /BASTOP/ZBAS, ZTOP, ICL, IWV
DATA AA/5.3744E-2, 2.9313E-4, -1.1300E-6, 1.6228E-9, -1.0421E-12,
+2.4192E-16, 0.0000E-00, 8.4451E-2, 1.6549E-4, -4.7300E-7,
+4.4334E-10, -1.3851E-13, -2.0644E-17, 1.2569E-20, 1.0240E-1,
+4.8678E-5, -2.2195E-7, 1.4506E-10, 5.9393E-14, -8.7250E-17,
+2.1383E-20, 3.6775E-2, 1.6160E-4, -6.4666E-7, 9.5298E-10,
+-6.4023E-13, 1.6916E-16, -7.9924E-21, 1.8410E-2, 6.6870E-4,
+-2.8406E-6, 5.4478E-9, -5.4854E-12, 2.7932E-15, -5.8271E-19,
+3.3458E-2, 1.3098E-4, -4.1528E-7, 6.1166E-10, -4.6681E-13,
+1.7383E-16, -2.5043E-20/
DATA BB/7.5099E-2, 3.2061E-4, -1.7060E-6, 3.3538E-9, -3.2719E-12,
+1.5721E-15, -3.0123E-19, 1.1808E-1, 2.2387E-4, -8.7996E-7,
+1.1238E-9, -6.6823E-13, 1.7514E-16, -1.5665E-20, 1.4155E-1,
+-5.0592E-5, -1.2280E-7, 1.2715E-10, 2.7470E-14, -6.6940E-17,
+1.7734E-20, 4.9533E-2, 2.0904E-4, -1.1626E-6, 2.3531E-9,
+-2.3463E-12, 1.1472E-15, -2.2267E-19, 3.8315E-3, 1.1837E-3,
+-5.1096E-6, 9.4660E-9, -8.6426E-12, 3.6778E-15, -5.5775E-19,
+4.1534E-2, 2.0220E-4, -8.1463E-7, 1.3590E-9, -1.1142E-12,
+4.3867E-16, -6.6426E-20/
DATA CC/1.1269E-2, 2.8659E-4, -6.0210E-8, -1.5274E-9, 2.7747E-12,
+-1.8946E-15, 4.5539E-19, 1.9856E-2, 1.2292E-4, -8.4479E-8,
+8.6563E-12, 0.0000E-0, 0.0000E-0, 0.0000E-0, 3.1907E-2,
+1.9632E-4, -1.1114E-7, -3.3987E-10, 5.2528E-13, -2.7799E-16,
+5.0469E-20, 6.8522E-3, 1.5362E-4, 7.5813E-8, -1.1430E-9,
+1.8885E-12, -1.2477E-15, 2.9522E-19, 8.5792E-4, 6.4122E-5,
+7.7271E-7, -2.9750E-9, 4.4014E-12, -3.0626E-15, 8.1541E-19,
+3.7151E-3, 1.4919E-4, -9.3486E-8, -1.2183E-10, 1.8733E-13,
+-9.2167E-17, 1.5939E-20/
TAUN=0.0
ZZA=Z0(IN)-ZBAS
ZZB=Z1(IN)-ZBAS
VERDIS=ZZB-ZZA
HORDIS=SQRT((X1(IN)-X0(IN))**2+(Y1(IN)-Y0(IN))**2)
EL=SQRT(HORDIS**2+VERDIS**2)
IF(EL.EQ.0.0)RETURN
IF(IHORIZ.EQ.1)GO TO 200
XI=VERDIS/EL
ELA=ZZA*1000.0
ELB=(XI*EL+ZZA)*1000.0
POLYA=0.0
POLYB=0.0
DO 100 N=1,7
EN=FLOAT(N)
AN=A(N, ICL, IWV)
TERMB=(1.0/(XI*EN))*AN*ELB**N
TERMA=(1.0/(XI*EN))*AN*ELA**N
POLYA=POLYA+TERMA
100 POLYB=POLYB+TERMB
TAUN=POLYB-POLYA
GO TO 300
200 ZZA=ZZA*1000.0
DO 250 N=1,7
TERMK=A(N, ICL, IWV)*HORDIS*1000.0*ZZA**(N-1)
250 TAUN=TAUN+TERMK
300 RETURN
END
CLE00010
CLE00020
CLE00030
CLE00040
CLE00050
CLE00060
CLE00070
CLE00080
CLE00090
CLE00100
CLE00110
CLE00120
CLE00130
CLE00140
CLE00150
CLE00160
CLE00170
CLE00180
CLE00190
CLE00200
CLE00210
CLE00220
CLE00230
CLE00240
CLE00250
CLE00260
CLE00270
CLE00280
CLE00290
CLE00300
CLE00310
CLE00320
CLE00330
CLE00340
CLE00350
CLE00360
CLE00370
CLE00380
CLE00390
CLE00400
CLE00410
CLE00420
CLE00430
CLE00440
CLE00450
CLE00460
CLE00470
CLE00480
CLE00490
CLE00500
CLE00510
CLE00520
CLE00530
CLE00540
CLE00550
CLE00560
CLE00570
CLE00580
CLE00590
CLE00600
CLE00610
CLE00620

```

```

SUBROUTINE LISOUT(NN,CTRANS,TAUTOT,TAU,NLAYER,IRUN)
C
C*** OUTPUT CONTROL ROUTINE
DIMENSION ALPHA(6),T(20),TAU(20)
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
COMMON /PATHL/ X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
COMMON /LEVEL/ ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
COMMON /BASTH/ZBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)
+ ,RADICL(10)
COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUL
DATA ALPHA/2HST,2HAS,2HNS,2HSC,2HCH,2HCC/
IF(IWV.EQ.1)WVL=0.55
IF(IWV.EQ.2)WVL=3.80
IF(IWV.EQ.3)WVL=10.60
IF(IRUN.LT.2) WRITE(IOOUT,150)WVL
IF(IRUN.GT.1) WRITE(IOOUT,160)WVL
150 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS //)
160 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS //)
IF(IHORIZ.EQ.1)WRITE(IOOUT,200)
IF(IVERT.EQ.1)WRITE(IOOUT,210)
IF(ISLTUP.EQ.1)WRITE(IOOUT,220)
IF(ISLTDN.EQ.1)WRITE(IOOUT,230)
200 FORMAT(1H0,40X,28HLINE-OF-SIGHT IS HORIZONTAL //)
210 FORMAT(1H0,40X,26HLINE-OF-SIGHT IS VERTICAL //)
220 FORMAT(1H0,40X,28HLINE-OF-SIGHT SLANTS UPWARD //)
230 FORMAT(1H0,40X,30HLINE-OF-SIGHT SLANTS DOWNWARD //)
PTHLEN=SQRT((XS-XT)**2+(YS-YT)**2+(ZS-ZT)**2)
WRITE(IOOUT,300)PTHLEN
300 FORMAT(1H0,40X,29HTOTAL LINE-OF-SIGHT LENGTH = ,F7.3,4H KM )
ATTLEN=0.0
IF(TAUTOT.EQ.0.0)GO TO 780
DO 310 N=1,NN
310 ATTLEN=ATTLEN+ATTL(N)
WRITE(IOOUT,400)ATTLEN
400 FORMAT(1H0,40X,50HTOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD
+ = ,F7.3,4H KM )
WRITE(IOOUT,500)TAUTOT
500 FORMAT(1H0,40X,22HTOTAL OPTICAL DEPTH = ,F7.2//)
WRITE(IOOUT,600)CTRANS
600 FORMAT(1H0,40X,36HTRANSMITTANCE ALONG LINE-OF-SIGHT = ,E11.5,//)
WRITE(IOOUT,605)
605 FORMAT(1H0,40X,50HSEEKER COORDINATES (KM) TARGET COORDINATES (KML)
+ )
WRITE(IOOUT,606)
606 FORMAT(1H ,40X,23HXSEEKER YSEEKER ZSEEKER,3X,23HXTARGET YTARGET ZT
+ARGET)
WRITE(IOOUT,607)
607 FORMAT(1H ,40X,3(8H----- ),2X,3(8H----- ))
WRITE(IOOUT,609)XS,YS,ZS,XT,YT,ZT
609 FORMAT(1H0,40X,3(F6.3,2X),2X,3(F6.3,2X))
WRITE(IOOUT,613)
613 FORMAT(1H ,//)
WRITE(IOOUT,615)
615 FORMAT(1H0,40X,62H CLOUD TYPE LINE-OF-SIGHT INTERSECTION COORD
+ INATES (KM) )
WRITE(IOOUT,620)
620 FORMAT(1H ,40X,10H/ID NUMBER,4X,47HXUPPER YUPPER ZUPPER XLOWER
+ YLOWER ZLOWER)
WRITE(IOOUT,625)
625 FORMAT(1H ,40X,10H-----,4X,3(8H----- ),1X,3(8H----- ))
IF(NLAYER.EQ.0)GO TO 685
DO 670 N=1,NLAYER
IND=ILCTYP(N)
IF(NLINT(N).EQ.0)GO TO 670
WRITE(IOOUT,680)ALPHA(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N)
670 CONTINUE
680 FORMAT(1H0,42X,A2,1H/,12,6X,3(F7.3,1X),1X,3(F7.3,1X))

```

685	NLO=N_LAYER+1	LI600710
	IF(N_LAYER.EQ.NN)GO TO 699	LI600720
	LL=0	LI600730
	DO 690 N=NLO,NN	LI600740
	LL=LL+1	LI600750
	IND=ICCTYP(LL)	LI600760
	IF(NCINT(LL).EQ.0)GO TO 690	LI600770
	WRITE(IOOUT,680)ALPH<IND>,N,X1<N>,Y1<N>,Z1<N>,X0<N>,Y0<N>,Z0<N>	LI600780
690	CONTINUE	LI600790
	WRITE(IOOUT,613)	LI600800
	WRITE(IOOUT,691)	LI600810
691	FORMAT<1H0,40X,59H CLOUD TYPE CUMULUS CENTER POSITIONS <FOR CC OR	LI600820
	+ CH TYPES >	LI600830
	WRITE(IOOUT,692)	LI600840
692	FORMAT<1H ,40X,10H/ID NUMBER,6X,26HX CLOUD <KM> Y CLOUD <KM>	LI600850
	WRITE(IOOUT,693)	LI600860
693	FORMAT<1H ,40X,10H-----,6X,26H----->	LI600870
	LL=0	LI600880
	DO 695 N=NLO,NN	LI600890
	LL=LL+1	LI600900
	IND=ICCTYP(LL)	LI600910
695	WRITE(IOOUT,696)ALPH<IND>,N,X CLOUD<LL>,Y CLOUD<LL>	LI600920
696	FORMAT<1H0,42X,A2,1H/,I2,10X,F7.3,8X,F7.3>	LI600930
699	CONTINUE	LI600940
	WRITE(IOOUT,613)	LI600950
	DO 610 N=1,NN	LI600960
610	T<N>=EXP(-TAU<N>)	LI600970
	IF(NCCLDS.GT.0) WRITE(IOOUT,700)	LI600980
	IF(NCCLDS.LT.1) WRITE(IOOUT,705)	LI600990
700	FORMAT<1H1,20X,11H CLOUD TYPE ,2X,15H HEIGHT OF BASE ,2X,	LI601000
	+10H THICKNESS ,2X,16H RADIUS OF CLOUD ,2X,14H OPTICAL DEPTH ,2X,	LI601010
	+14H TRANSMITTANCE >	LI601020
705	FORMAT<1H0,20X,11H CLOUD TYPE ,2X,15H HEIGHT OF BASE ,2X,	LI601030
	+10H THICKNESS ,2X,16H RADIUS OF CLOUD ,2X,14H OPTICAL DEPTH ,2X,	LI601040
	+14H TRANSMITTANCE >	LI601050
	WRITE(IOOUT,710)	LI601060
710	FORMAT<1H ,20X,10H/ID NUMBER,9X,4H<KM>,9X,4H<KM>,12X,4H<KM>,9X,	LI601070
	+11HALONG L-O-S,5X,11HALONG L-O-S>	LI601080
	WRITE(IOOUT,720)	LI601090
720	FORMAT<1H ,20X,10H-----,3X,14H-----,3X,9H-----,	LI601100
	+3X,15H-----,3X,13H-----,3X,13H----->	LI601110
	IF(N_LAYER.EQ.0)GO TO 735	LI601120
	BLANK=0.0	LI601130
	DO 730 N=1,N_LAYER	LI601140
	IND=ILCTYP(N)	LI601150
730	WRITE(IOOUT,760)ALPH<IND>,N,ZLBASE<N>,ZLTHIC<N>,BLANK,TAU<N>,T<N>	LI601160
735	NLO=N_LAYER+1	LI601170
740	IF(NN.EQ.N_LAYER)GO TO 800	LI601180
	LL=0	LI601190
	DO 750 N=NLO,NN	LI601200
	LL=LL+1	LI601210
	IND=ICCTYP(LL)	LI601220
750	WRITE(IOOUT,760)ALPH<IND>,N,ZCBASE<LL>,ZCTHIC<LL>,RADIC<LL>,	LI601230
	+TAU<N>,T<N>	LI601240
760	FORMAT<1H0,22X,A2,1H/,I2,10X,F7.3,7X,F7.3,8X,F7.3,10X,F7.2,7X,	LI601250
	+E11.5>	LI601260
	GO TO 800	LI601270
780	WRITE(IOOUT,790)	LI601280
790	FORMAT<1H0,20X,59HNO CLOUD OBSCURATION : L-O-S DOES NOT INTERSECT	LI601290
	+ANY CLOUDS // >	LI601300
800	RETURN	LI601310
	END	LI601320

<pre> SUBROUTINE DEFSET(ISTEP,NMAX) C**** THIS ROUTINE RESETS THE CLOUD BASE HEIGHT, THICKNESS, C**** AND RADIUS VALUES TO THE NEAREST REALISTIC BOUNDARIES C**** IF THEY DO NOT LIE CLOSE TO THE RANGES SPECIFIED IN C**** R. D. H. LOW'S PAPER. C COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10) COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10) + ,RADICL(10) IF(ISTEP.NE.1)GO TO 10 DO 5 N=1,NMAX ITYPE=ILCTYP(N) IF(ITYPE.EQ.2)GO TO 3 1 IF(ZLBASE(N).LT.0.1)ZLBASE(N)=0.1 IF(ZLBASE(N).GT.1.5)ZLBASE(N)=1.5 IF(ITYPE.EQ.3)GO TO 4 2 IF(ZLTHIC(N).LT.0.2)ZLTHIC(N)=0.2 IF(ZLTHIC(N).GT.1.0)ZLTHIC(N)=1.0 GO TO 5 3 IF(ZLBASE(N).LT.2.0)ZLBASE(N)=2.0 IF(ZLBASE(N).GT.5.0)ZLBASE(N)=5.0 4 IF(ZLTHIC(N).LT.1.0)ZLTHIC(N)=1.0 IF(ZLTHIC(N).GT.4.0)ZLTHIC(N)=4.0 5 CONTINUE 10 IF(ISTEP.NE.2)GO TO 20 DO 15 N=1,NMAX IF(ZCBASE(N).LT.0.8)ZCBASE(N)=0.8 IF(ZCBASE(N).GT.1.5)ZCBASE(N)=1.5 IF(ZCTHIC(N).LT.0.2)ZCTHIC(N)=0.2 IF(ZCTHIC(N).GT.5.0)ZCTHIC(N)=5.0 IF(RADICL(N).LT.0.05)RADICL(N)=0.05 IF(RADICL(N).GT.0.6)RADICL(N)=0.6 15 CONTINUE 20 RETURN END </pre>	<pre> DEF 00010 DEF 00020 DEF 00030 DEF 00040 DEF 00050 DEF 00060 DEF 00070 DEF 00080 DEF 00090 DEF 00100 DEF 00110 DEF 00120 DEF 00130 DEF 00140 DEF 00150 DEF 00160 DEF 00170 DEF 00180 DEF 00190 DEF 00200 DEF 00210 DEF 00220 DEF 00230 DEF 00240 DEF 00250 DEF 00260 DEF 00270 DEF 00280 DEF 00290 DEF 00300 DEF 00310 DEF 00320 DEF 00330 DEF 00340 DEF 00350 DEF 00360 </pre>
---	--

```

SUBROUTINE CLREAD(NLAYER,NCCLDS,IERR)
DIMENSION IALPHA(18),DATELT(5)
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATT(20)
COMMON /GEOMET/PTS(15),IGEOSW
COMMON /BASTOP/ZBAS,ZTOP,ICL,IUV,ILCTYP(10),ICCTYP(10)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)
+ ,RADICL(10)
COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)
DATA IALPHA/2HCL,2HST,2HCL,2HAS,2HCL,2HNS,2HCL,2HSC,2HCL,2HCH,
+2HCL,2HCC,2HSE,2HEK,2HTA,2HRG,2HGO,2H /
ISFLAG=0
ITFLAG=0
IF(IGEOSW.NE.1)GO TO 60
XS=PTS(4)
YS=PTS(5)
ZS=PTS(6)
XT=PTS(1)
YT=PTS(2)
ZT=PTS(3)
60 CONTINUE
DO 900 N=1,23
READ(IOIN,100)IALPH,IALP,(DATELT(I),I=1,5)
100 FORMAT(2A2,1X,5(E10.5,1X))
INDEX=10
DO 200 K=1,17,2
IF((IALPH.EQ.IALPHA(K)).AND.(IALP.EQ.IALPHA(K+1))) GO TO 50
GO TO 200
50 REALK=FLOAT(K)
ROT=REALK/2.0
KJ=IFIX(ROT)+1
INDEX=KJ
GO TO 300
200 CONTINUE
300 IF(INDEX.LT.7)GO TO (350,350,350,350,400,400),INDEX
INM6=INDEX-6
GO TO (500,600,999,997),INM6
350 NLAYER=NLAYER+1
IF(NLAYER.GT.10) GO TO 993
ILCTYP(NLAYER)=INDEX
ZLBASE(NLAYER)=DATELT(1)
ZLTHIC(NLAYER)=DATELT(2)
GO TO 900
400 NCCLDS=NCCLDS+1
IF(NCCLDS.GT.10) GO TO 995
ICCTYP(NCCLDS)=INDEX
ZCBASE(NCCLDS)=DATELT(1)
ZCTHIC(NCCLDS)=DATELT(2)
RADICL(NCCLDS)=DATELT(3)
XCLOUD(NCCLDS)=DATELT(4)
YCLOUD(NCCLDS)=DATELT(5)
GO TO 900
500 IF(ISFLAG.EQ.1) GO TO 997
XS=DATELT(1)
YS=DATELT(2)
ZS=DATELT(3)
ISFLAG=1
GO TO 900
600 IF(ITFLAG.GT.1) GO TO 997
XT=DATELT(1)
YT=DATELT(2)
ZT=DATELT(3)
ITFLAG=1
900 CONTINUE
GO TO 999
993 IERR=1
WRITE(IOOUT,994)
994 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF STRATIFORM CLOUDS I
+INPUT EXCEEDS THE LIMIT OF 10 //)

```

```

CLR00010
CLR00020
CLR00030
CLR00040
CLR00050
CLR00060
CLR00070
CLR00080
CLR00090
CLR00100
CLR00110
CLR00120
CLR00130
CLR00140
CLR00150
CLR00160
CLR00170
CLR00180
CLR00190
CLR00200
CLR00210
CLR00220
CLR00230
CLR00240
CLR00250
CLR00260
CLR00270
CLR00280
CLR00300
CLR00310
CLR00320
CLR00330
CLR00340
CLR00350
CLR00360
CLR00370
CLR00380
CLR00390
CLR00400
CLR00410
CLR00420
CLR00430
CLR00440
CLR00450
CLR00460
CLR00470
CLR00480
CLR00490
CLR00500
CLR00510
CLR00520
CLR00530
CLR00540
CLR00550
CLR00560
CLR00570
CLR00580
CLR00590
CLR00600
CLR00610
CLR00620
CLR00630
CLR00640
CLR00650
CLR00660

```

GO TO 999	CLR00670
995 IERR=1	CLR00680
WRITE(IOOUT,996)	CLR00690
996 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF CUMULIFORM CLOUDS I	CLR00700
+NPUT EXCEEDS THE LIMIT OF 10 //)	CLR00710
GO TO 999	CLR00720
997 IERR=1	CLR00730
WRITE(IOOUT,998)	CLR00740
998 FORMAT(1H0,20X,64H***CLREAD ERROR*** IMPROPER INPUT FORMAT OR ABSE	CLR00750
+NT GO SENTINEL //)	CLR00760
999 RETURN	CLR00770
END	CLR00780

```

SUBROUTINE SCREEN(IERR,ICLMAT)
THIS MODULE 1) COMPUTES TRANSMITTANCE REQUIRED TO REDUCE THE
PROBABILITY OF STATIC TARGET DETECTION BELOW A
GIVEN LEVEL FOR CERTAIN TYPES OF IMAGERS. (ITAM)
- AND/OR -
2) COMPUTES HC AND WP 105 MM/155 MM SMOKE MUNITIONS
REQUIRED TO PRODUCE A SMOKE SCREEN OF USER-DEFINED
LENGTH AND DURATION FOR VISIBLE, NEAR, MID AND FAR
IR WAVELENGTHS.
SUBROUTINES CALLED BY SCREEN ARE - CWIC AND ITAM.
ALL OUTPUT FROM SCREEN IS TABULAR.
THE PRESENT VERSION OF CWIC USES THE XSCALE MODULE TO OPTIONALLY
CORRECT FOR EXTINCTION DUE TO FOG, RAIN AND/OR SNOW AT IR
WAVELENGTHS BASED ON VISIBILITY IN THE 5 MICROMETER REGION.
CWIC CAN ALSO COMPUTE PASQUILL CATEGORY FROM FUNDAMENTAL MET INPUTS
IF THE USER CHOOSES NOT TO PROVIDE THE CATEGORY DIRECTLY.
CLIMATOLOGICAL VALUES FROM THE CLIMAT MODULE CAN OPTIONALLY BE
USED AS "TYPICAL" MET INPUTS.
THE ITAM MODULE CAN BE USED IN A LOOPING MODE THROUGH MULTIPLE SETS
OF INPUT RECORDS TO GENERATE TABLES. THE LAST VALUE OF TRANS-
MITTANCE COMPUTED IS THAT VALUE WHICH CAN (OPTIONALLY) BE PASSED
TO CWIC AS THE THRESHOLD LEVEL FOR TOTAL PATH TRANSMITTANCE.
TWO RECORDS MUST BE PROVIDED TO SCREEN:
ONE CARD MUST BE INPUT TO SCREEN TO SELECT OPTIONS: (3(1X,I1))
COL 2 ICITAM =1 CALL ITAM, OR 0 (NO CALL).
COL 4 ICCWIC =1 CALL CWIC, OR 0 (NO CALL).
COL 6 ICCLIM =1 USE CLIMAT FOR MET INPUTS, OR 0 USE USER VALUES.
IF CHOSEN, ALL INPUT RECORDS TO ITAM ARE READ FOLLOWING THE ABOVE
RECORD. (SEE INPUT DESCRIPTION IN ITAM)
AND, IF CHOSEN, INPUT RECORDS TO CWIC ARE THEN READ. (SEE INPUT
DESCRIPTION IN CWIC)
FINALLY, ONE RECORD IS READ BY SCREEN WITH THE WORD "END" IN COLUMNS
1-3. THIS RETURNS CONTROL TO THE EOSAEL EXECUTIVE MODULE.
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOU
READ (IOIN,10) ICITAM, ICCWIC, ICCLIM
FORMAT(1X, I1, 1X, I1, 1X, I1)
IF (ICITAM.NE.0) CALL ITAM(IERR,TFNL)
IF (IERR.EQ.1) GOTO 15
IF (ICCLIM.EQ.0) GOTO 20
IF (ICLMAT.EQ.1) GOTO 20
IERR=1
WRITE (IOOUT,30)
FORMAT(1X,62H*** IN SCREEN ROUTINE, MET SOURCE SPECIFIED AS CLIMAT
*OLOGICAL,/5X,39H BUT CLIMAT ROUTINE HAD NOT BEEN CALLED.)
RETURN
15 WRITE (IOOUT,40)
TFNL=1
40 FORMAT(1X,21HIERR FLAG SET IN ITAM)
20 IF (ICWIC.NE.0) CALL CWIC (IERR,ICITAM,ICCLIM,TFNL)
RETURN
END

```

```

SCR00010
SCR00020
SCR00030
SCR00040
SCR00050
SCR00060
SCR00070
SCR00080
SCR00090
SCR00100
SCR00110
SCR00120
SCR00130
SCR00140
SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

```


SUBROUTINE CWIC (IERR, ICITAM, ICCLIM, TFNL)
 CWIC COMPUTES THE REQUIRED SMOKE MUNITIONS EXPENDITURE (NUMBER,
 RATE OF FIRE, PLACEMENT) TO PRODUCE A SCREEN OF DEFINED
 LENGTH AND DURATION USING HC OR WP 105MM OR 155MM SMOKE
 MUNITIONS. WAVELENGTHS ARE FOR RANGES OF VISIBLE, NEAR,
 MID AND FAR IR.

ON INPUT, IF ICITAM IS NON-ZERO, ALL COMPUTATIONS WILL USE
 THE INPUT TFNL AS THE TOTAL SCREEN TRANSMISSION THRESHOLD. IF
 ICITAM IS 0, THEN BUILT-IN TOTAL THRESHOLDS OF .05 ARE USED
 FOR ALL WAVELENGTH REGIONS.

INPUTS TO CWIC ARE ON STANDARDIZED RECORDS CONTAINING KEY-WORDS
 IN COLUMNS 1-4 AND REAL (IE DECIMAL) VALUES FOR ALL INPUTS,
 PLACED IN FIELDS 11-20, 21-30, ..., 71-80. KEY-WORD TYPES
 ARE SCRN FOR SCREEN, OBSERVER/TARGET LOS AND ADVERSE
 WEATHER CORRECTIONS.
 METR FOR METEOROLOGICAL CONDITIONS (MAY BE OMITTED
 IF ICCLIM IS NON-ZERO, IN WHICH CASE THE
 CLIMATOLOGICAL VALUES FROM CLIMAT ARE USED.
 PASQ FOR (OPTIONAL) MET PARAMETERS REQUIRED TO COMPUTE
 THE PASQUILL STABILITY CATEGORY. NOT REQUIRED
 IF PASQUILL CATEGORY ITSELF IS INPUT.
 DONE WHICH RETURNS EXECUTIVE CONTROL BACK TO THE SCREEN
 MODULE.

THE ORDER OF THE INPUT RECORDS IS IMMATERIAL, EXCEPT THAT THE (DONE)
 CARD MUST BE LAST.

INPUTS: (ALL VALUES REAL) FORMAT (2A2,6X,7F10.3)

KEYWORD	COLS.	VARIABLE	DESCRIPTION	
SCRN	1-4		SCREEN/LOS DEFINITION (REQUIRED)	CWC00340
	11-20	TIME	- SCREEN DURATION (MINUTES)	CWC00350
	21-30	X0	- SCREEN LENGTH (METERS)	CWC00360
	31-40	H3	- SLANT RANGE OBS-TARGET (KM)	CWC00370
	41-50	AST	- ELEVATION ANGLE OF TARGET FROM OBSERVER WRT HORIZONTAL (DEG.)	CWC00380
	51-60	DLS	- COMPASS DIRECTION (CLOCKWISE WRT NORTH) OF LINE-OF-SIGHT (DEG.)	CWC00390
	61-70	ARE	- TERRAIN ROUGHNESS ELEMENT (CM)	CWC00400
	71-80	FOG	- ADVERSE WEATHER/HAZE SELECTION 0. = NO ADVERSE WEATHER 1. = ONLY CORRECT VISIBLE WAVE- LENGTHS FOR INPUT VISIBILITY. 2. = CORRECT FOR FOG/HAZE FOR MARITIME ARCTIC AIR MASS 3. = CORRECT FOR FOG/HAZE FOR MARITIME POLAR AIR MASS 4. = CORRECT FOR FOG/HAZE FOR CONTINENTAL POLAR AIR MASS 5. = CORRECT FOR RAIN. 6. = CORRECT FOR SNOW.	CWC00410 CWC00420 CWC00430 CWC00440 CWC00450 CWC00460 CWC00470 CWC00480 CWC00490 CWC00500 CWC00510 CWC00520 CWC00530 CWC00540 CWC00550
METR	1-4		MET INPUTS (NOT REQUIRED IF ICCLIM NON-ZERO FROM SCREEN MODULE.)	CWC00560
	11-20	S3	- WINDSPEED (METERS/SEC)	CWC00570
	21-30	D0	- WIND DIRECTION (DEG) CLOCKWISE WRT NORTH, USUAL MET CONVENTION	CWC00580 CWC00590
	31-40	PCAT	- PASQUILL CATEGORY. IF INPUT AS 0, SEE PASQ RECORD BELOW. OTHERWISE, 1.=A, 2.=B, 3.=C, 4.=D, 5.=E, 6.=F	CWC00600 CWC00610 CWC00620 CWC00630
	41-50	VS	- VISIBILITY (KM) NOT REQUIRED IF FOG = 0. IS SPECIFIED.	CWC00640 CWC00650
	51-60	R0	- RELATIVE HUMIDITY (PERCENT). IF 0, THEN DEW POINT AND TEMPERATURE ARE REQUIRED BELOW TO COMPUTE R0.	CWC00660 CWC00670 CWC00680
	61-70	T0	- AIR TEMP. (DEG C) REQUIRED IF R0 NOT GIVEN.	CWC00690 CWC00700


```

900  FORMAT(2A2,6X,7F10.3)
901  FORMAT(1X,51HIN CWIC, THE FOLLOWING CARD DOES NOT CONFORM TO PRO,
*15HPER CONVENTIONS/1X,2A2,6X,7F10.3)
IF (<ICOU,LE,4) GOTO 10
WRITE (<IOOUT,902)
902  FORMAT(1X,39HINVALID INPUTS TO CWIC. IERR=1 RETURNED)
IERR=1
RETURN
20  TIME=RV(1)
X0=RV(2)
H3=RV(3)
AST=RV(4)
DLS=RV(5)
ARE=RV(6)
FOG=RV(7)
IF(<IGEOSW,NE,1) GO TO 22
DELX=PTS(1)-PTS(4)
DELY=PTS(2)-PTS(5)
DELZ=PTS(3)-PTS(6)
H3=SQRT(DELX**2+DELY**2+DELZ**2)
HDIS=SQRT(DELX**2+DELY**2)
RTDCON=57.29577951
AST=RTDCON*ACOS(HDIS/H3)
IF(HDIS.GT.1.E-20) DLS=RTDCON*ACOS(DELY/HDIS)
IF(DELX.LT.0.) DLS=360.-DLS
22  CONTINUE
GOTO 10
40  S3=RV(1)
D0=RV(2)
PCAT=RV(3)
VS=RV(4)
R0=RV(5)
T0=RV(6)
T1=RV(7)
GOTO 10
50  SLAT=RV(1)
SLONG=RV(2)
SJDATE=RV(3)
SZHOUR=RV(4)
C0=RV(5)
C1=RV(6)
GOTO 10
C*** BEGIN COMPUTATIONS. -- FIRST NET VALUES.
60  IF (<ICLIM,EQ,0) GOTO 70
C*** USE CLIMAT PASSED VALUES:
T0=TEMP
T1=DP
R0=RH
PCAT=FLOAT(IPASCT)
VS=VIS
D0=WINDDIR
S3=WINDVEL
C*** PROVIDE WINDSPEED IN KNOTS AS S0:
70  S0=S3/.515
IF (<S0,LE,1.) S3=.515
IF (<S0,LE,1.) S0=1.
C*** NOW CHECK RH AND COMPUTE IF NECESSARY.
IRNOT=0
IF (<R0,GT,0.) GOTO 80
IF(<T0,GT,0.) GO TO 76
A0=9.5
B0=265.5
IF(<T0,LE,0.) GO TO 78
76  CONTINUE
A0=7.5
B0=237.3
78  CONTINUE
IF(<T1,GE,0.) GO TO 79
A1=9.5
B1=265.5

```

```

CWC01400
CWC01410
CWC01420
CWC01430
CWC01440
CWC01450
CWC01460
CWC01470
CWC01480
CWC01490
CWC01500
CWC01510
CWC01520
CWC01530
CWC01540
CWC01550
CWC01560
CWC01570
CWC01580
CWC01590
CWC01600
CWC01610
CWC01620
CWC01630
CWC01640
CWC01650
CWC01660
CWC01670
CWC01680
CWC01690
CWC01700
CWC01710
CWC01720
CWC01730
CWC01740
CWC01750
CWC01760
CWC01770
CWC01780
CWC01790
CWC01800
CWC01810
CWC01820
CWC01830
CWC01840
CWC01850
CWC01860
CWC01870
CWC01880
CWC01890
CWC01900
CWC01910
CWC01920
CWC01930
CWC01940
CWC01950
CWC01960
CWC01970
CWC01980
CWC01990
CWC02000
CWC02010
CWC02020
CWC02030
CWC02040
CWC02050
CWC02060
CWC02070
CWC02080
CWC02090

```

79	IF(T1 .LE. 0.) GO TO 75	CWC02100
	CONTINUE	CWC02110
	A1=7.5	CWC02120
	B1=237.3	CWC02130
75	CONTINUE	CWC02140
	E0=6.11*10.**((A0*T0)/(B0+T0))	CWC02150
	E1=6.11*10.**((A1*T1)/(B1+T1))	CWC02160
	R0=(E1/E0)*100.	CWC02170
	IRNOT=1	CWC02180
C***	NOW CHECK PASQUILL CATEGORY..	CWC02190
80	IP0=IFIX(PCAT+.001)	CWC02200
	IF (IP0.GT.6) IP0=6	CWC02210
	IPNOT=0	CWC02220
	IF (IP0.GT.0) GOTO 90	CWC02230
	CALL CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE,SZHOOR,S0)	CWC02240
	IPNOT=1	CWC02250
C***	NEXT COMPUTE CL VALUE FOR THRESHOLD TRANSMITTANCE, CORRECTED FOR	CWC02260
C	FOG,RAIN,...	CWC02270
C***	VISIBILITY EXTINCTION (KM-1) .. NOTE THAT IF VS=0., THEN SET TO	CWC02280
C	CLEAR DAY AND COMPUTATIONS CONTINUE.	CWC02290
90	EX55=0.	CWC02300
	IF (VS.GT.0.) EX55=3.912/VS	CWC02310
C***	TRANSMISSION THRESHOLDS..	CWC02320
	DO 92 I=1,4	CWC02330
	T(I)=TRSH(I)	CWC02340
	IF (ICITAM.NE.0) T(I)=TFNL	CWC02350
	TR(I)=T(I)	CWC02360
92	CONTINUE	CWC02370
C***	CORRECTIONS FOR WEATHER	CWC02380
	IFOG=-1+IFIX(FOG+.0001)	CWC02390
	IF (IFOG.LE.-1) GOTO 100	CWC02400
C***	CORRECT VISIBLE FOR VISIBILITY (H3 SLANT RNG, AST ELEV. ANG.)	CWC02410
	XSTRN=0.	CWC02420
	EPTH=EX55*H3	CWC02430
	IF (EPTH.LT.12.) XSTRN=EXP(-EPTH)	CWC02440
C SET	VSET FOR NO CARD I/O IN XSCALE	CWC02450
	VSET=89.	CWC02460
	IF((AST.GT.0.1 .OR. AST.LT.-0.1) .AND. IFOG.LT.5)	CWC02470
	*CALL XSCALE(WAVE(I),VSET,EX55,XSTRN,IERR,ISLANT(I),IFOG,H3,AST)	CWC02480
	IF (XSTRN.LE.0.) XSTRN=.0001	CWC02500
	IF (XSTRN.GT.1.) XSTRN=1.	CWC02510
	T(I)=T(I)/XSTRN	CWC02520
	IF (IFOG.EQ.0) GOTO 100	CWC02530
C CORRECT	NON-VISIBLE FOR SEEABILITY.	CWC02540
94	CONTINUE	CWC02550
	DO 96 I=2,4	CWC02560
	ISLNT=ISLANT(I)	CWC02570
	IF (AST.EQ.0.) ISLNT=0	CWC02580
	IF (IFOG.GE.5) ISLNT=0	
	CALL XSCALE(WAVE(I),VSET,EX55,XSTRN,IERR,ISLNT,IFOG,H3,AST)	CWC02590
	IF (XSTRN.LE.0.) XSTRN=.0001	CWC02610
	IF (XSTRN.GT.1.) XSTRN=1.	CWC02620
	T(I)=T(I)/XSTRN	CWC02630
96	CONTINUE	CWC02640
C***	COMPUTE CL FOR THRESHOLD TRANSMISSION REQUIRED OF SMOKE.	CWC02650
100	DO 108 I=1,4	CWC02660
	IF (T(I).GT.1.) T(I)=1.	CWC02670
	IF (T(I).LE.0.) T(I)=.00001	CWC02680
	DO 105 J=1,2	CWC02690
	C(I,J)=ALOG(T(I))/(-CS(I,J))	CWC02700
105	CONTINUE	CWC02710
108	CONTINUE	CWC02720
C***	ATMOSPHERIC DIFFUSION	CWC02730
	CALL CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U,V,Y1)	CWC02740
C***	MUNITIONS EXPENDITURES	CWC02750
	CALL CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0)	CWC02760
C**	END MAIN CWIC COMPUTATIONS, FINAL OUTPUT.	CWC02770
	IF (ICITAM.GT.0) WRITE(IOOUT,10000)	CWC02780
	WRITE (IOOUT,10200)	CWC02790
	WRITE (IOOUT,10800)	CWC02800

WRITE (IOOUT,10140)	CWC02810
WRITE (IOOUT,10300)	CWC02820
WRITE (IOOUT,10400)	CWC02830
WRITE (IOOUT,10200)	CWC02840
WRITE (IOOUT,14000) H3	CWC02850
WRITE (IOOUT,14100) AST	CWC02860
WRITE (IOOUT,14200) DLS	CWC02870
WRITE (IOOUT,11900) ARE	CWC02880
WRITE (IOOUT,11400)	CWC02890
IAD=1	CWC02900
IF (IFOG.GE.0) IAD=3	CWC02910
WRITE (IOOUT,14300) JANS(IAD), JANS(IAD+1)	CWC02920
IAD=1	CWC02930
IF (IFOG.EQ.1) IAD=3	CWC02940
WRITE (IOOUT,14400) JANS(IAD), JANS(IAD+1)	CWC02950
IAD=1	CWC02960
IF (IFOG.EQ.2) IAD=3	CWC02970
WRITE (IOOUT,14500) JANS(IAD), JANS(IAD+1)	CWC02980
IAD=1	CWC02990
IF (IFOG.EQ.3) IAD=3	CWC03000
WRITE (IOOUT,14600) JANS(IAD), JANS(IAD+1)	CWC03010
IAD=1	CWC03020
IF (IFOG.EQ.4) IAD=3	CWC03030
WRITE (IOOUT,14700) JANS(IAD), JANS(IAD+1)	CWC03040
IAD=1	CWC03050
IF (IFOG.EQ.5) IAD=3	CWC03060
WRITE (IOOUT,14800) JANS(IAD), JANS(IAD+1)	CWC03070
WRITE (IOOUT,15000)	CWC03080
WRITE (IOOUT,11800) S3	CWC03090
WRITE (IOOUT,11700) D0	CWC03100
JP=JPASCT(IPO)	CWC03110
WRITE (IOOUT,12000) JP	CWC03120
WRITE (IOOUT,11300) VS	CWC03130
WRITE (IOOUT,12100) R0	CWC03140
IF (IRNOT.EQ.1) WRITE (IOOUT,11500) T0	CWC03150
IF (IRNOT.EQ.1) WRITE (IOOUT,11600) T1	CWC03160
IF (IPNOT.EQ.0) GOTO 120	CWC03170
WRITE (IOOUT,15100)	CWC03180
IF (SLAT.GE.0.) WRITE (IOOUT,10601) SLAT	CWC03190
SLATI=-SLAT	CWC03200
IF (SLAT.LT.0.) WRITE (IOOUT,10602) SLATI	CWC03210
IF (SLONG.GE.0.) WRITE (IOOUT,10701) SLONG	CWC03220
SLONGI=-SLONG	CWC03230
IF (SLONG.LT.0.) WRITE (IOOUT,10702) SLONGI	CWC03240
WRITE (IOOUT,10900) SJDATE	CWC03250
WRITE (IOOUT,11000) SZHOUR	CWC03260
WRITE (IOOUT,11100) C0	CWC03270
WRITE (IOOUT,11200) C1	CWC03280
120 CONTINUE	CWC03290
WRITE (IOOUT,15200)	CWC03300
DO 150 I=1,4	CWC03310
IWL=4*(I-1)	CWC03320
WRITE (IOOUT,15300) (LNGTH(IWL+J), J=1,4), TR(I), T(I)	CWC03330
150 CONTINUE	CWC03340
WRITE (IOOUT,10000)	CWC03350
C* PRINT MUNITION EXPENDITURES	CWC03360
WRITE (IOOUT,10130)	CWC03370
WRITE (IOOUT,16200) (LNGTH(J), J=1,8)	CWC03380
WRITE (IOOUT,10100)	CWC03390
WRITE (IOOUT,16300)	CWC03400
WRITE (IOOUT,16400) X0, TIME, X0, TIME	CWC03410
WRITE (IOOUT,10200)	CWC03420
WRITE (IOOUT,16500) ISMOKE(1), ISMOKE(1)	CWC03430
WRITE (IOOUT,10100)	CWC03440
WRITE (IOOUT,16600) IGUN(1), IGUN(2), IGUN(1), IGUN(2)	CWC03450
WRITE (IOOUT,10100)	CWC03460
WRITE (IOOUT,16700)	CWC03470
WRITE (IOOUT,16800)	CWC03480
WRITE (IOOUT,16900) Q1(1,1,1), Y(1,1,1), Q1(2,1,1), Y(2,1,1)	CWC03490
WRITE (IOOUT,17000) Q1(1,2,1), R(1,1), Y(1,2,1), P(1,1,1), Q1(2,2,1)	CWC03500

```

*R(2,1),Y(2,2,1),P(2,1,1)
WRITE<IOOUT,10200>
WRITE<IOOUT,16600> IGUN(3),IGUN(4),IGUN(3),IGUN(4)
WRITE<IOOUT,10100>
WRITE<IOOUT,16700>
WRITE<IOOUT,16800>
WRITE<IOOUT,16900> Q1(1,1,2),Y(1,1,2),Q1(2,1,2),Y(2,1,2)
WRITE<IOOUT,17000> Q1(1,2,2),R(1,1),Y(1,2,2),P(1,2,1),Q1(2,2,2),
*R(2,1),Y(2,2,2),P(2,2,1)
WRITE<IOOUT,10200>
WRITE<IOOUT,16500> ISMOKE(2),ISMOKE(2)
WRITE<IOOUT,10100>
WRITE<IOOUT,16600> IGUN(1),IGUN(2),IGUN(1),IGUN(2)
WRITE<IOOUT,10100>
WRITE<IOOUT,16700>
WRITE<IOOUT,16800>
WRITE<IOOUT,16900> Q2(1,1,1),Z(1,1,1),Q2(2,1,1),Z(2,1,1)
IF<IPO.GT.4> R(1,2)=2.
IF<IPO.GT.4> R(2,2)=2.
WRITE<IOOUT,17000> Q2(1,2,1),R(1,2),Z(1,1,1),P(1,1,2),Q2(2,2,1),
*R(2,2),Z(2,1,1),P(2,1,2)
WRITE<IOOUT,10200>
WRITE<IOOUT,16600> IGUN(3),IGUN(4),IGUN(3),IGUN(4)
WRITE<IOOUT,10100>
WRITE<IOOUT,16700>
WRITE<IOOUT,16800>
WRITE<IOOUT,16900> Q2(1,1,2),Z(1,2,2),Q2(2,1,2),Z(2,2,2)
IF<IPO.GT.4> R(1,2)=1.
IF<IPO.GT.4> R(2,2)=1.
WRITE<IOOUT,17000> Q2(1,2,2),R(1,2),Z(1,2,2),P(1,2,2),Q2(1,2,2),
*R(2,2),Z(2,2,2),P(2,2,2)
WRITE<IOOUT,10000>
WRITE<IOOUT,10130>
DO 7100 I=3,4
IWL=4*(I-1)
J1=IWL+1
J2=IWL+4
WRITE<IOOUT,12200> (LNGTH(J),J=J1,J2)
WRITE<IOOUT,10100>
WRITE<IOOUT,12300>
WRITE<IOOUT,12400> X0,TIME
WRITE<IOOUT,10200>
WRITE<IOOUT,12500> ISMOKE(2)
WRITE<IOOUT,10100>
WRITE<IOOUT,13100>
WRITE<IOOUT,13200>
WRITE<IOOUT,13300> IGUN(1),IGUN(2),Q2(1,2,1),R(1,2),P(1,1,2)
WRITE<IOOUT,13300> IGUN(3),IGUN(4),Q2(1,2,2),R(1,2),P(1,2,2)
IF<I.EQ.3> WRITE<IOOUT,10140>
IF<I.EQ.4> WRITE<IOOUT,10000>
7100 CONTINUE
RETURN
C* FORMAT STATEMENTS.
10000 FORMAT(1H1)
10100 FORMAT(1H )
10200 FORMAT(1H0)
10130 FORMAT(///)
10140 FORMAT(////)
10300 FORMAT(55X,21HMUNITION EXPENDITURES)
10400 FORMAT(56X,19HFOR HC AND WP SMOKE)
10601 FORMAT(45X,35HLATITUDE - DEG =,F7.2,6H NORTH)
10602 FORMAT(45X,35HLATITUDE - DEG =,F7.2,6H SOUTH)
10701 FORMAT(45X,35HLONGITUDE - DEG =,F7.2,6H EAST)
10702 FORMAT(45X,35HLONGITUDE - DEG =,F7.2,6H WEST)
10800 FORMAT(1H0,47X,36(1H*)/48X,1H*,34X,1H*/48X,1H*,4X,
*26HCWC MUNITION EXPENDITURES,4X,1H*/48X,1H*,34X,1H*/
*48X,36(1H*))
10900 FORMAT(45X,35HJULIAN DATE - DAY =,F7.0)
11000 FORMAT(45X,35HGMT TIME - HOUR =,F7.2)
11100 FORMAT(45X,35HCEILING - METERS =,F7.1)

```

11200	FORMAT(45X,35H CLOUD COVER	- PERCENT	=,F7.1)	CWC04210
11300	FORMAT(45X,35H VISIBILITY	- KM	=,F7.3)	CWC04220
11400	FORMAT(1H0,44X,35H ATMOSPHERIC EXTINCTION CORRECTIONS)			CWC04230
11500	FORMAT(45X,35H TEMPERATURE	- DEG C	=,F7.1)	CWC04240
11600	FORMAT(45X,35H DEW POINT	- DEG C	=,F7.1)	CWC04250
11700	FORMAT(45X,35H WIND DIRECTION	- DEG	=,F7.2)	CWC04260
11800	FORMAT(45X,35H WIND SPEED	- M/SEC	=,F7.2)	CWC04270
11900	FORMAT(45X,35H AVG ROUGHNESS ELEMENT	- CM	=,F7.1)	CWC04280
12000	FORMAT(45X,35H PASQUILL CATEGORY	-	=,5X,A2)	CWC04290
12100	FORMAT(45X,35H RELATIVE HUMIDITY	- PERCENT	=,F7.1)	CWC04300
12200	FORMAT(60X,11(1H-)/60X,1H-,1X,4A2,1H-/60X,11(1H-))			CWC04310
12300	FORMAT(62X,6H LENGTH,8X,8H DURATION/62X,6H METERS,9X,7H MINUTES)			CWC04320
12400	FORMAT(47X,6H SCREEN,8X,F7.0,9X,F7.2)			CWC04330
12500	FORMAT(59X,A2,13H SMOKE SCREEN/59X,15(1H-))			CWC04340
12600	FORMAT(59X,A2,A1,11HMM HOWITZER)			CWC04350
12700	FORMAT(47X,37H VOLLEY GUNS RATE SPACING ROUNDS)			CWC04360
12800	FORMAT(47X,30H /MIN METERS)			CWC04370
12900	FORMAT(47X,11H INITIAL: ,F5.0,6X,F8.0)			CWC04380
13000	FORMAT(47X,11H SUSTAINING: ,F5.0,F5.1,F9.0,F7.0)			CWC04390
13100	FORMAT(47X,31H ROUNDS/ RATE/ TOTAL)			CWC04400
13200	FORMAT(47X,32H 60 METERS MINUTE ROUNDS)			CWC04410
13300	FORMAT(47X,A2,A1,5HMM: ,F5.0,5X,F4.0,3X,F7.0)			CWC04420
14000	FORMAT(45X,35H SLANT RANGE OBS-TGT	- KM	=,F7.3)	CWC04430
14100	FORMAT(45X,35H ELEVATION OF TARGET	- DEG	=,F7.2)	CWC04440
14200	FORMAT(45X,35H AZIMUTH OF TARGET	- DEG	=,F7.2)	CWC04450
14300	FORMAT(45X,35H CORRECTED FOR VISIBILITY	-	,4X,2A2)	CWC04460
14400	FORMAT(45X,35H MARITIME ARCTIC AIR MASS	-	,4X,2A2)	CWC04470
14500	FORMAT(45X,35H MARITIME POLAR AIR MASS	-	,4X,2A2)	CWC04480
14600	FORMAT(45X,35H CONTINENTAL POLAR AIR MASS	-	,4X,2A2)	CWC04490
14700	FORMAT(45X,35H CORRECTED FOR RAIN	-	,4X,2A2)	CWC04500
14800	FORMAT(45X,35H CORRECTED FOR SNOW	-	,4X,2A2)	CWC04510
15000	FORMAT(1H0,44X,35H METEOROLOGICAL INPUTS			CWC04520
15100	FORMAT(1H0,44X,35H INPUTS FOR PASQUILL CATEGORY			CWC04530
15200	FORMAT(1H0,44X,42H TRANSMISSION THRESHOLDS TOTAL SMOKE)			CWC04540
15300	FORMAT(45X,4A2,18X,F5.3,6X,F5.3)			CWC04550
16200	FORMAT(33X,12(1H-),44X,12(1H-)/33X,1H-,1X,4A2,1X,1H-,44X,1H-,1X, *4A2,1X,1H-/33X,12(1H-),44X,12(1H-))			CWC04560
16300	FORMAT(1X,2(34X,6H LENGTH,8X,8H DURATION)/1X,2(34X,6H METERS,9X, *7H MINUTES))			CWC04570
16400	FORMAT(1X,2(19X,6H SCREEN,8X,F7.0,9X,F7.2))			CWC04580
16500	FORMAT(32X,A2,13H SMOKE SCREEN,41X,A2,13H SMOKE SCREEN/32X, *15(1H-),41X,15(1H-))			CWC04590
16600	FORMAT(32X,A2,A1,11HMM HOWITZER,42X,A2,A1,11HMM HOWITZER)			CWC04600
16700	FORMAT(1X,2(19X,6H VOLLEY,6X,25H GUNS RATE SPACING ROUNDS))			CWC04610
16800	FORMAT(37X,4H/MIN,3X,6H METERS,43X,4H/MIN,3X,6H METERS)			CWC04620
16900	FORMAT(20X,11H INITIAL: ,F5.0,6X,F8.0,26X,11H INITIAL: , *F5.0,6X,F8.0)			CWC04630
17000	FORMAT(1X,2(19X,11H SUSTAINING: ,F5.0,F5.1,F9.0,F7.0))			CWC04640
	END			CWC04650
				CWC04660
				CWC04670
				CWC04680
				CWC04690

```

SUBROUTINE CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE, SZHOUR,S0)
DIMENSION ITAB(7,9)
COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DATA ITAB /
1 1,1,2,3,4,6,6,
2 1,2,2,3,4,6,6,
3 1,2,2,3,4,4,6,6,
4 2,2,2,3,4,4,5,6,
5 2,2,2,3,4,4,4,5,6,
6 2,3,3,4,4,4,4,5,
7 3,3,4,4,4,4,4,
8 3,3,4,4,4,4,4,
9 3,4,4,4,4,4,4/
ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))
C*METEOROLOGICAL CALCULATIONS.
C
IF(C1 .NE. 100.) GO TO 1000
IF(C0 .GT. 2133.6042) GO TO 1000
I1=0
I2=0
GO TO 2300
1000 CONTINUE
C CALCULATE ANGULAR FRACTION OF A YEAR FOR A GIVEN JULIAN DATE.
R9=PIRAD
D9=1./R9
SLAT1=SLAT*R9
A0=((SJDATE-1.)*360.)/365.242
C CALCULATE SOLAR DECLINATION ANGLE (A4).
A1=A0*R9
A2=279.9348+A0
A2=A2+(1.914827*SIN(A1))-(0.079525*COS(A1))
A2=A2+(0.019938*SIN(2*A1))-(0.00162*COS(2*A1))
A2=A2*R9
A3=23.4438*R9
A4=SIN(A3)*SIN(A2)
A4=ASIN(A4)
C CALCULATE THE TIME OF MERIDIAN PASSAGE - TRUE SOLAR NOON (A5).
A5=12.+(0.12357*SIN(A1))-(0.004289*COS(A1))
A5=A5+(0.153809*SIN(2*A1))+(0.060783*COS(2*A1))
C CALCULATE SOLAR HOUR ANGLE (A6)*** NOTE THIS VERSION USES + SIGN
C ON SLONG DUE TO EAST-LONGITUDE POSITIVE CONVENTION.
A6=15.*(SZHOUR-A5)+SLONG
A6=A6*R9
C CALCULATE SOLAR ALTITUDE (A7).
A7=SIN(SLAT1)*SIN(A4)+COS(SLAT1)*COS(A4)*COS(A6)
A7=ASIN(A7)
1100 CONTINUE
A7=A7*D9
C CALCULATE INSOLATION CLASS NUMBER.
I2=0
IF(A7 .LE. 60.) GO TO 1200
I2=4
GO TO 1500
1200 CONTINUE
IF(A7 .LE. 35.) GO TO 1300
I2=3
GO TO 1500
1300 CONTINUE
IF(A7 .LE. 15.) GO TO 1400
I2=2
GO TO 1500
1400 CONTINUE
IF(A7 .LE. 0.) GO TO 2200
I2=1
C CALCULATE NET RADIATION INDEX FOR DAYTIME.
1500 CONTINUE
I3=0
IF(C1 .GT. 50.) GO TO 1600
I3=I2
GO TO 1900

```

```

CWX00010
CWX00020
CWX00030
CWX00040
CWX00050
CWX00060
CWX00070
CWX00080
CWX00090
CWX00100
CWX00110
CWX00120
CWX00130
CWX00140
CWX00150
CWX00160
CWX00170
CWX00180
CWX00190
CWX00200
CWX00210
CWX00220
CWX00230
CWX00240
CWX00250
CWX00260
CWX00270
CWX00280
CWX00290
CWX00300
CWX00310
CWX00320
CWX00330
CWX00340
CWX00350
CWX00360
CWX00370
CWX00380
CWX00390
CWX00400
CWX00410
CWX00420
CWX00430
CWX00440
CWX00450
CWX00460
CWX00470
CWX00480
CWX00490
CWX00500
CWX00510
CWX00520
CWX00530
CWX00540
CWX00550
CWX00560
CWX00570
CWX00580
CWX00590
CWX00600
CWX00610
CWX00620
CWX00630
CWX00640
CWX00650
CWX00660
CWX00670
CWX00680
CWX00690

```



```

1600 CONTINUE
      IF(C0 .GT. 2133.6042) GO TO 1700
      I3=I2-2
      GO TO 1900
1700 CONTINUE
      IF(C0 .GE. 4876.8096) GO TO 1800
      I3=I2-1
      GO TO 1900
1800 CONTINUE
      IF(C1 .NE. 100.) GO TO 1900
      I3=I2-1
1900 CONTINUE
      IF(I3 .NE. 0) GO TO 2000
      I3=I2
2000 CONTINUE
      IF(I3 .GT. 1) GO TO 2100
      I3=1
2100 CONTINUE
      I1=I3
      GO TO 2300
C COMPUTE NET RADIATION INDEX FOR NIGHTTIME.
2200 CONTINUE
      IF(C1 .GT. 40.) GO TO 2250
      I1=-2
      GO TO 2300
2250 CONTINUE
      I1=-1
C CALCULATE PASQUILL STABILITY CATAGORY.
2300 CONTINUE
      I4=0
      I5=0
      IF(I1 .NE. 4) GO TO 2400
      I4=1
2400 CONTINUE
      IF(I1 .NE. 3) GO TO 2420
      I4=2
2420 CONTINUE
      IF(I1 .NE. 2) GO TO 2440
      I4=3
2440 CONTINUE
      IF(I1 .NE. 1) GO TO 2460
      I4=4
2460 CONTINUE
      IF(I1 .NE. 0) GO TO 2480
      I4=5
2480 CONTINUE
      IF(I1 .NE. -1) GO TO 2500
      I4=6
2500 CONTINUE
      IF(I1 .NE. -2) GO TO 2520
      I4=7
2520 CONTINUE
      IF(S0 .GE. 2.) GO TO 2540
      I5=1
      GO TO 2700
2540 CONTINUE
      IF(S0 .GE. 4.) GO TO 2560
      I5=2
      GO TO 2700
2560 CONTINUE
      IF(S0 .GE. 6.) GO TO 2580
      I5=3
      GO TO 2700
2580 CONTINUE
      IF(S0 .GE. 7.) GO TO 2600
      I5=4
      GO TO 2700
2600 CONTINUE
      IF(S0 .GE. 8.) GO TO 2620
      I5=5

```

```

CWX00700
CWX00710
CWX00720
CWX00730
CWX00740
CWX00750
CWX00760
CWX00770
CWX00780
CWX00790
CWX00800
CWX00810
CWX00820
CWX00830
CWX00840
CWX00850
CWX00860
CWX00870
CWX00880
CWX00890
CWX00900
CWX00910
CWX00920
CWX00930
CWX00940
CWX00950
CWX00960
CWX00970
CWX00980
CWX00990
CWX01000
CWX01010
CWX01020
CWX01030
CWX01040
CWX01050
CWX01060
CWX01070
CWX01080
CWX01090
CWX01100
CWX01110
CWX01120
CWX01130
CWX01140
CWX01150
CWX01160
CWX01170
CWX01180
CWX01190
CWX01200
CWX01210
CWX01220
CWX01230
CWX01240
CWX01250
CWX01260
CWX01270
CWX01280
CWX01290
CWX01300
CWX01310
CWX01320
CWX01330
CWX01340
CWX01350
CWX01360
CWX01370
CWX01380
CWX01390

```

```

2620 GO TO 2700
      CONTINUE
      IF(S0 .GE. 10.) GO TO 2640
      I5=6
2640 GO TO 2700
      CONTINUE
      IF(S0 .GE. 11.) GO TO 2660
      I5=7
2660 GO TO 2700
      CONTINUE
      IF(S0 .GE. 12.) GO TO 2680
      I5=8
2680 GO TO 2700
      CONTINUE
      I5=9
2700 CONTINUE
      IP0=ITAB(I4,I5)
      RETURN
      END

```

```

CWX01400
CWX01410
CWX01420
CWX01430
CWX01440
CWX01450
CWX01460
CWX01470
CWX01480
CWX01490
CWX01500
CWX01510
CWX01520
CWX01530
CWX01540
CWX01550
CWX01560
CWX01570
CWX01580
CWX01590

```

C

```

SUBROUTINE CWIC3 (ARE, DLS, C2, D0, D2, H, IP0, R0, R2, U, V, Y1)
DIMENSION A(6), S(6,3), D(6,3), H(2,2), U(2,2), V(2)
DIMENSION W(6)
COMMON /CONST/ PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
DATA A/0.4, 0.32, 0.22, 0.144, 0.102, 0.076/
DATA S /
*.139085297, .122097643, .110104377, .097649832, .070772166, .055487093,
*.015017284, .010970370, .010962963, .010418519, 7.27284E-3, 6.55309E-3,
*-1.02581E-4, -6.80135E-5, -6.73401E-5, -6.83502E-5, -4.50056E-5,
*-4.01796E-5/
DATA D /
*.944814815, .894803591, .854792368, .816026936, .786026936, .726015713,
*-4.85185E-3, -4.83951E-3, -4.82716E-3, -6.07407E-3, -6.07407E-3,
*-6.06173E-3
*3.7037E-5, 3.59147E-5, 3.47924E-5, 4.7138E-5, 4.7138E-5, 4.60157E-5/
DATA W/0.016, 0.016, 0.016, 0.016, 0.016, 0.016/
C*ATMOSPHERIC DIFFUSION CALCULATIONS.
A1=-1.24+1.19*ALOG10(ARE)
Z1=10.**A1
A2=ABS(DLS-D0)*(PI/180.)
R2=SQR(13.69/(13.69*SIN(A2)*SIN(A2)+COS(A2)*COS(A2)))
Y1=1.09521547+(0.02906894*R0)-(4.9575E-04*R0*R0)+
2 (4.82E-06*R0*R0*R0)
Y2=3.364059144+(0.060502571*R0)-(1.15301E-03*R0*R0)+
2 (1.33942E-05*R0*R0*R0)
C2=S(IP0,1)+S(IP0,2)*Z1+S(IP0,3)*Z1**2
D1=D(IP0,1)+D(IP0,2)*Z1+D(IP0,3)*Z1**2
D2=1/D1
DO 5400 I=1,4
C*CALCULATE CROSSWIND INTEGRATED CONCENTRATION FOR WP SMOKE.
DO 5300 K=1,2
IF (I.LT.3.AND.IP0.GT.4) GOTO 5300
S1=U(K,1)+0.74*A(IP0)*100.**0.9
S2=U(K,2)+0.667*C2*100.**D1
V(K)=(W(IP0)*Y2*H(K,2))/(PI*S1*S2)
5300 CONTINUE
5400 CONTINUE
RETURN
END
CWY00010
CWY00020
CWY00030
CWY00040
CWY00050
CWY00060
CWY00070
CWY00080
CWY00090
CWY00100
CWY00110
CWY00120
CWY00130
CWY00140
CWY00150
CWY00160
CWY00170
CWY00180
CWY00190
CWY00200
CWY00210
CWY00220
CWY00230
CWY00240
CWY00250
CWY00260
CWY00270
CWY00280
CWY00290
CWY00300
CWY00310
CWY00320
CWY00330
CWY00340
CWY00350
CWY00360
CWY00370
CWY00380
CWY00390

```

```

SUBROUTINE CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0) CWZ00010
REAL ME(2) CWZ00020
DIMENSION C(4,2),H(2,2),P(4,2,2),Q1(4,2,2),Q2(4,2,2),R(4,2),V(2), CWZ00030
*Y(4,2,2),Z(4,2,2) CWZ00040
DATA ME/0.4,0.4/ CWZ00050
C*MUNITIONS EXPENDITURES. CWZ00060
DO 6900 I=1,4 CWZ00070
C* CALCULATE INITIAL SHELL SPACING FOR HC SMOKE CWZ00080
DO 6800 K=1,2 CWZ00090
IF(I .GT. 2) GO TO 6100 CWZ00100
Y(I,1,K)=45.*S3 CWZ00110
C* CALCULATE SUSTAINING SHELL SPACING FOR HC SMOKE CWZ00120
IF(C(I,1) .NE. 0.) GO TO 5500 CWZ00130
Y(I,2,K)=0. CWZ00140
GO TO 5600 CWZ00150
5500 CONTINUE CWZ00160
Y(I,2,K)=1/R2*((0.731*ME(K)*Y1*H(K,1))/(C2*S3*C(I,1)))*D2 CWZ00170
IF(Y(I,2,K) .GT. X0) Y(I,2,K)=X0 CWZ00180
5600 CONTINUE CWZ00190
IF(Y(I,2,K) .NE. 0.) GO TO 5700 CWZ00200
Q1(I,1,K)=1. CWZ00210
Q1(I,2,K)=1. CWZ00220
GO TO 5900 CWZ00230
5700 CONTINUE CWZ00240
C* CALCULATE INITIAL VOLLEY FOR HC SMOKE CWZ00250
IF(Y(I,1,K) .GT. Y(I,2,K)) Y(I,1,K)=Y(I,2,K) CWZ00260
Q1(I,1,K)=X0/Y(I,1,K) CWZ00270
Q5=AINT(Q1(I,1,K)) CWZ00280
Q6=Q1(I,1,K)-Q5 CWZ00290
IF(Q6 .EQ. 0.) GO TO 5800 CWZ00300
Q1(I,1,K)=Q5+1. CWZ00310
5800 CONTINUE CWZ00320
C* CALCULATE NUMBER OF GUNS FOR SUSTAINING VOLLEYS (HC) CWZ00330
Q1(I,2,K)=X0/Y(I,2,K) CWZ00340
Q5=AINT(Q1(I,2,K)) CWZ00350
Q6=Q1(I,2,K)-Q5 CWZ00360
IF(Q6 .EQ. 0.) GO TO 5900 CWZ00370
Q1(I,2,K)=Q5+1. CWZ00380
5900 CONTINUE CWZ00390
C* CALCULATE RATE OF FIRE FOR HC SMOKE CWZ00400
R(I,1)=0.5 CWZ00410
IF(C(I,1) .NE. 0.) GO TO 6000 CWZ00420
R(I,1)=0. CWZ00430
6000 CONTINUE CWZ00440
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (HC SMOKE) CWZ00450
P(I,K,1)=Q1(I,1,K)+(0.5*Q1(I,2,K))*(TIME-2.) CWZ00460
Q5=AINT(P(I,K,1)) CWZ00470
Q6=P(I,K,1)-Q5 CWZ00480
IF(Q6 .EQ. 0.) GO TO 6100 CWZ00490
P(I,K,1)=Q5+1. CWZ00500
6100 CONTINUE CWZ00510
IF(I .LT. 3 .AND. IP0 .GT. 4) GO TO 6775 CWZ00520
C* SHELL SPACING (Z( )) AND VOLLEYS (Q( )) - WP SMOKE CWZ00530
IF(C(I,2) .NE. 0.) GO TO 6200 CWZ00540
Z(I,1,K)=0. CWZ00550
Z(I,2,K)=0. CWZ00560
Q2(I,1,K)=0. CWZ00570
Q2(I,2,K)=0. CWZ00580
GO TO 6400 CWZ00590
6200 CONTINUE CWZ00600
IF(I .LT. 3) GO TO 6250 CWZ00610
IF (I .GT. 2) Q2(I,1,K)=0.6*C(I,2)/V(K) CWZ00620
GO TO 6300 CWZ00630
6250 CONTINUE CWZ00640
Z(I,1,K)=V(K)/C(I,2)*100. CWZ00650
Z(I,2,K)=Z(I,1,K) CWZ00660
Q2(I,1,K)=X0/Z(I,2,K)+1. CWZ00670
6300 CONTINUE CWZ00680
Q5=AINT(Q2(I,1,K)) CWZ00690
Q6=Q2(I,1,K)-Q5 CWZ00700

```

```

IF (Q6 .EQ. 0.) GO TO 6350
Q2(I,1,K)=Q5+1.
6350 CONTINUE
Q2(I,2,K)=Q2(I,1,K)
6400 CONTINUE
C* RATE OF FIRE FOR WP SMOKE
IF (C(I,2) .NE. 0.) GO TO 6425
R(I,2)=0.
GO TO 6600
6425 CONTINUE
IF (I .GT. 2) GOTO 6450
R(I,2)=(Z(I,2,K)+60.)/S3
GO TO 6475
6450 CONTINUE
R(I,2)=120./S3
6475 CONTINUE
R(I,2)=R(I,2)/20.
R5=AINTR(I,2)
R6=R(I,2)-R5
IF (R6 .LT. 0.5) GO TO 6500
R5=R5+1.
6500 CONTINUE
IF (R5 .NE. 0.) GO TO 6550
R5=1.
6550 CONTINUE
R(I,2)=R5*20./60.
6600 CONTINUE
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (WP)
IF (C(I,2) .NE. 0.) GO TO 6650
P(I,K,2)=0.
GO TO 6700
6650 CONTINUE
IF (I .GT. 2) GO TO 6700
P(I,K,2)=Q2(I,2,K)*(1./R(I,2))* (TIME-R(I,2))
GO TO 6750
6700 CONTINUE
P(I,K,2)=Q2(I,1,K)*(X0/60.+1.)*(1./R(I,2))* (TIME-R(I,2))
6750 CONTINUE
Q5=AINTP(I,K,2)
Q6=P(I,K,2)-Q5
IF (Q6 .EQ. 0.) GO TO 6800
P(I,K,2)=Q5+1.
GO TO 6800
6775 CONTINUE
C* CALCULATIONS FOR E AND F STABILTY CAT (STABLE FLOW)
C* INITIAL SHELL SPACING FOR WP SMOKE
IF (I .EQ. 1 .AND. K .EQ. 1) Z(I,1,K)=100.
IF (I .EQ. 2 .AND. K .EQ. 1) Z(I,1,K)=50.
IF (I .LT. 3 .AND. K .EQ. 2) Z(I,1,K)=100.
C* SUSTAINING SHELL SPACING FOR WP SMOKE
IF (I .EQ. 1 .AND. K .EQ. 1) Z(I,2,K)=100.
IF (I .EQ. 1 .AND. K .EQ. 2) Z(I,2,K)=200.
IF (I .EQ. 2 .AND. K .EQ. 1) Z(I,2,K)=50.
IF (I .EQ. 2 .AND. K .EQ. 2) Z(I,2,K)=100.
C* INITIAL VOLLEY FOR WP SMOKE
Q2(I,1,K)=X0/Z(I,1,K)+1.
C* SUSTAINING VOLLEY FOR WP SMOKE
Q2(I,2,K)=X0/Z(I,2,K)+1.
C* RATE OF FIRE FOR WP SMOKE
IF (K .EQ. 1) R(I,2)=.5
IF (K .EQ. 2) R(I,2)=1.
C* TOTAL NUMBER OF ROUNDS REQUIRED (WP)
P(I,K,2)=Q2(I,1,K)+Q2(I,2,K)*1./R(I,2)*(TIME-R(I,2))
6800 CONTINUE
R(I,2)=1./R(I,2)
6900 CONTINUE
RETURN
END

```

```

CWZ00710
CWZ00720
CWZ00730
CWZ00740
CWZ00750
CWZ00760
CWZ00770
CWZ00780
CWZ00790
CWZ00800
CWZ00810
CWZ00820
CWZ00830
CWZ00840
CWZ00850
CWZ00860
CWZ00870
CWZ00880
CWZ00890
CWZ00900
CWZ00910
CWZ00920
CWZ00930
CWZ00940
CWZ00950
CWZ00960
CWZ00970
CWZ00980
CWZ00990
CWZ01000
CWZ01010
CWZ01020
CWZ01030
CWZ01040
CWZ01050
CWZ01060
CWZ01070
CWZ01080
CWZ01090
CWZ01100
CWZ01110
CWZ01120
CWZ01130
CWZ01140
CWZ01150
CWZ01160
CWZ01170
CWZ01180
CWZ01190
CWZ01200
CWZ01210
CWZ01220
CWZ01230
CWZ01240
CWZ01250
CWZ01260
CWZ01270
CWZ01280
CWZ01290
CWZ01300
CWZ01310
CWZ01320
CWZ01330
CWZ01340
CWZ01350
CWZ01360
CWZ01370
CWZ01380

```

SUBROUTINE ITAM(IERR,TFNL)

THIS ROUTINE IS AN INVERSION OF THE NV&EOL TARGET ACQUISITION MODEL, WITH EMPHASIS ON THE DEGRADATION (TRANSMITTANCE) REQUIRED TO PREVENT DETECTION ABOVE A GIVEN PROBABILITY.

INPUTS ARE ON STANDARDIZED RECORDS:

KEY WORD - COLUMNS 1-4, FROM AMONG TARV,SENS,GO,PAGE AND DONE.
DATA FIELDS, ALL REAL - COLS 11-20, 21-30, 31-40, ..., 71-80.

INPUT CARDS ARE ORDER INDEPENDENT, WITH ARBITRARY OR SYSTEM DEFAULTS CHOSEN IF CARDS ARE LEFT OUT, OR IF VALUES ARE INPUT 0. THE EXCEPTION TO ORDER INDEPENDENCE IS THAT AFTER EACH SET OF INPUTS A (GO) CARD MUST BE PLACED TO INITIATE EXECUTION OF ONE LOOP THROUGH THE PROGRAM, FOLLOWING, OR IN PLACE OF THE LAST INPUT-SET GO CARD, A (DONE) CARD MUST BE PROVIDED TO CAUSE CONTROL TO EXIT ITAM AND RETURN TO THE SCREEN EXEC MODULE.

IN SUBSEQUENT RUNS (DELINEATED BY GO CARDS) TABLES MAY BE PRODUCED LINE-BY-LINE, AND ANY INPUT WHICH HAS NOT BEEN CHANGED (IE INPUT AS 0.) WILL USE THE VALUE GIVEN ON THE PREVIOUS RUN. EXCEPTIONS ARE THE FOV AND AMAG VALUES WHICH MUST BE SPECIFIED WHENEVER A NEW DEVICE NUMBER (LSC) IS INPUT, OR THESE VALUES ASSUME DEFAULTS FOR THE DEVICE.

KEY WORD	COLS.	VARIABLE	DESCRIPTION		
TARV	1-4		TARGET/SCENARIO DESCRIPTION	ITA00010	
	11-20	ACON	- INTRINSIC CONTRAST (DIMENSIONLESS) OR TEMPERATURE DIFFERENCE OF TARGET/BACKGROUND (DEG K) FOR THERMAL DEVICES.	ITA00020	
	21-30	SOG	- SKY/GROUND RATIO, (DIMENSIONLESS), USED FOR NON-THERMAL DEVICES. SOG SHOULD INCLUDE A FACTOR FOR CLOUD REFLECTANCE	ITA00030	
	31-40	R	- RANGE TO TARGET (KM)	ITA00040	
	41-50	TAR SZ	- TARGET MINIMUM DIMENSION (M).	ITA00050	
	51-60	ZONE	- SEARCH ZONE (DEG.**2)	ITA00060	
	61-70	ALFLG	- AMBIENT ILLUM. CATEGORY (SEE LIST BELOW) CONVERTED TO NAL, AN INTEGER, IN PROGRAM. USED FOR NON-THERMAL DEVICES.	ITA00070	
	71-80	ALIGHT	- IF ALFLG=0., THE USER MUST PROVIDE AN AMBIENT ILLUM. HERE (FT. COLS.)	ITA00080	
	SENS	1-4		SENSOR DESCRIPTION	ITA00090
		11-20	PS	- INPUT PROBABILITY OF DETECTION. (DIMENSIONLESS)	ITA00100
21-30		DVNUM	- DEVICE NUMBER. CONVERTED TO LSC, AN INTEGER, IN PROGRAM.	ITA00110	
31-40		DMODE	- OPERATIONAL MODE (1. = WIDE FOV, 2. = NARROW FOV). CONVERTED TO MODE, AN INTEGER, IN PROGRAM.	ITA00120	
41-50		FOV	- FIELD OF VIEW (DEG.)	ITA00130	
51-60		AMAG	- MAGNIFICATION (FOR VISIBLE ONLY)	ITA00140	
61-70		AJOB	- LEVEL OF ACQUISITION (MEDIAN RESOLVABLE NORMALIZED CYCLES, USUALLY 1. FOR DETECTION)	ITA00150	
PAGE	1-4		(OPTIONAL) - FORCE PAGE EJECT, WRITE NEW HEADER. (USEFUL FOR TABLE GENERATION.)	ITA00160	
GO	1-4		EXECUTE ONE LOOP WITH GIVEN INPUTS	ITA00170	
DONE	1-4		END COMPUTATIONS AND EXIT THE ITAM	ITA00180	


```

DMAG=0.
AJOB=0.
TBAR=0.
NAL=0.
TFNL=1.
ILAST=0.
IFLGFV=0
IFLCMG=0
C ***** OUTPUT LINE COUNTER
      ILINE=0
C ***** INITIALIZE VALUES (REPEATED EXECUTIONS)
C ***** INPUT CARD COUNTER
4     ICOV=0
      IF (ILAST.NE.0) GOTO 6
C ***** FLAG TO WRITE TITLE
      ITITL=0
      ITOT=1.
      RC=0.
      C=0.
      ICM=0
      INEWD=0
C ***** READ A CARD
5     READ(IQIN,900) IT(1),IT(2),(RV(I),I=1,7)
900   FORMAT(2A2,6X,7F10.2)
      ICOV=ICOV+1
      IF(IT(1).EQ.IR(1).AND.IT(2).EQ.IR(2)) GO TO 6
      IF(IT(1).EQ.IR(3).AND.IT(2).EQ.IR(4)) GO TO 3
      IF(IT(1).EQ.IR(5).AND.IT(2).EQ.IR(6)) GO TO 7
      IF(IT(1).EQ.IR(7).AND.IT(2).EQ.IR(8)) GO TO 8
      IF(IT(1).EQ.IR(9).AND.IT(2).EQ.IR(10)) GOTO 2
901   WRITE(IQOUT,901) IT(1),IT(2),(RV(I),I=1,7)
      FORMAT(1X,61HTHE FOLLOWING CARD DOES NOT CONFORM TO ITAM INPUT CON
*VENTIONS/1X,2A2,6X,7E10.3)
      IF(ICOV.LE.5) GO TO 5
      IERR=1
      GOTO 1
C ***** ALL DONE
6     IF (ICOV.GT.1) GOTO 3
1     WRITE(IQOUT,902) TFNL
902   FORMAT(1H0,5X,41H*** FINAL TOTAL TRANSMISSION FROM ITAM = ,F5.3)
      RETURN
C ***** TARGET CARD PROCESSING (TARV)
7     RCHK=ACON
      IF(RV(1).NE.0.) ACON=RV(1)
      IF(ACON.NE.RCHK) ITITL=1
      IF(RV(2).NE.0.) SOG=RV(2)
      IF(RV(3).NE.0.) RNG=RV(3)
      IF(IGEOSW.NE.1) GO TO 477
      RNG=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+(PTS(3)-PTS(6))**2)
477   CONTINUE
      RCHK=TARV
      IF(RV(4).NE.0.) TARV=RV(4)
      IF(TARV.NE.RCHK) ITITL=1
      RCHK=UZONE
      IF(RV(5).NE.0.) UZONE=RV(5)
      IF(RCHK.NE.UZONE) ITITL=1
      IF(RV(6).NE.0.) ALFLG=RV(6)
      IF(RV(7).NE.0.) ALIGHT=RV(7)
      GO TO 5
C ***** SENSOR CARD PROCESSING (SENS)
8     IF(RV(1).GT.0.) PDET=RV(1)
      RCHK=DVNUM
      IF(RV(2).NE.0.) DVNUM=RV(2)
      IF(DVNUM.NE.RCHK) ITITL=1
      IF (DVNUM.NE.RCHK) INEWD=1
      RCHK=DMODE
      IF(RV(3).NE.0.) DMODE=RV(3)
      IF(DMODE.NE.RCHK) ITITL=1
      IF (INEWD.EQ.1) DFOV=0.
      IF (INEWD.EQ.1) IFLGFV=0

```

```

ITA01410
ITA01420
ITA01430
ITA01440
ITA01450
ITA01460
ITA01470
ITA01480
ITA01490
ITA01500
ITA01510
ITA01520
ITA01530
ITA01540
ITA01550
ITA01560
ITA01570
ITA01580
ITA01590
ITA01600
ITA01610
ITA01620
ITA01630
ITA01640
ITA01650
ITA01660
ITA01670
ITA01680
ITA01690
ITA01700
ITA01710
ITA01720
ITA01730
ITA01740
ITA01750
ITA01760
ITA01770
ITA01780
ITA01790
ITA01800
ITA01810
ITA01820
ITA01830
ITA01840
ITA01850
ITA01860
ITA01870
ITA01880
ITA01890
ITA01900
ITA01910
ITA01920
ITA01930
ITA01940
ITA01950
ITA01960
ITA01970
ITA01980
ITA01990
ITA02000
ITA02010
ITA02020
ITA02030
ITA02040
ITA02050
ITA02060
ITA02070
ITA02080
ITA02090
ITA02100

```


	RCHK=DFOV	ITA02110
	IF(RV(4).NE.0.) DFOV=RV(4)	ITA02120
	IF(RV(4).NE.0.AND.IFLCFV.EQ.1) IFLCFV=0	ITA02130
	IF(DFOV.NE.RCHK) ITITL=1	ITA02140
	IF(INEWD.EQ.1) DMAG=0.	ITA02150
	IF(INEWD.EQ.1) IFLGMG=0	ITA02160
	RCHK=DMAG	ITA02170
	IF(RV(5).NE.0.) DMAG=RV(5)	ITA02180
	IF(RV(5).NE.0.AND.IFLGMG.EQ.1) IFLGMG=0	ITA02190
	IF(DMAG.NE.RCHK) ITITL=1	ITA02200
	RCHK=AJOB	ITA02210
	IF(RV(6).NE.0.) AJOB=RV(6)	ITA02220
	IF(AJOB.NE.RCHK) ITITL=1	ITA02230
	GO TO 5	ITA02240
C	***** PAGE TO NEW HEADER	ITA02250
N	ITITL=1	ITA02260
	GOTO 5	ITA02270
C	***** BEGIN PROCESSING (GO)	ITA02280
C	***** FIRST CHECK IF DEFAULTS NEEDED, AND SET INTEGER VALUES, LIGHT CA	ITA02290
SW	ILAST=1	ITA02300
	RCHK=DVNUM	ITA02310
	IF(DVNUM.EQ.0.) DVNUM=DVN	ITA02320
	IF(RCHK.NE.DVNUM) ITITL=1	ITA02330
	LSC=IFIX(DVNUM+.0001)	ITA02340
	IF(LSC.LT.1) LSC=1	ITA02350
	IF(LSC.GT.14) LSC=14	ITA02360
	RCHK=DMODE	ITA02370
	IF(DMODE.EQ.0.) DMODE=DMD	ITA02380
	IF(RCHK.NE.DMODE) ITITL=1	ITA02390
	MODE=IFIX(DMODE+.0001)	ITA02400
	IF(MODE.LT.1) MODE=1	ITA02410
	IF(MODE.GT.2) MODE=2	ITA02420
	IF(PDET.EQ.0.) PDET=PD	ITA02430
	PS=PDET	ITA02440
	IF(PS.GT.1.) PS=1.	ITA02450
	IF(PS.LT.0.) PS=0.	ITA02460
	RCHK=DFOV	ITA02470
	IF(DFOV.LE.0.0.AND.MODE.EQ.1) DFOV=FOVW(LSC)	ITA02480
	IF(DFOV.LE.0.0.AND.MODE.EQ.2) DFOV=FOVN(LSC)	ITA02490
	IF(DFOV.LE.0.0) IFLCFV=1	ITA02500
	IF(DFOV.LE.0.0.AND.MODE.EQ.1) DFOV=FV	ITA02510
	IF(DFOV.LE.0.0.AND.MODE.EQ.2) DFOV=FV/2.	ITA02520
	IF(RCHK.NE.DFOV) ITITL=1	ITA02530
	FOV=DFOV	ITA02540
	RCHK=DMAG	ITA02550
	IF(DMAG.LE.0.0.AND.MODE.EQ.1) DMAG=AMAGW(LSC)	ITA02560
	IF(DMAG.LE.0.0.AND.MODE.EQ.2) DMAG=AMAGN(LSC)	ITA02570
	IF(DMAG.LE.0.0) IFLGMG=1	ITA02580
	IF(DMAG.LE.0.0) DMAG=DMG	ITA02590
	IF(RCHK.NE.DMAG) ITITL=1	ITA02600
	AMAG=DMAG	ITA02610
	RCHK=AJOB	ITA02620
	IF(AJOB.EQ.0.0) AJOB=AJB	ITA02630
	IF(RCHK.NE.AJOB) ITITL=1	ITA02640
	RCHK=ACON	ITA02650
	IF(ACON.EQ.0.0) ACON=ACN	ITA02660
	IF(RCHK.NE.ACON) ITITL=1	ITA02670
	C=ACON	ITA02680
	IF(SOG.EQ.0.0) SOG=SGR	ITA02690
	IF(RNG.LE.0.0) RNG=RN	ITA02700
	R=RNG	ITA02710
	RCHK=TARSZ	ITA02720
	IF(TARSZ.LE.0.0) TARSZ=TGT	ITA02730
	IF(RCHK.NE.TARSZ) ITITL=1	ITA02740
	DIM=TARSZ	ITA02750
	RCHK=UZONE	ITA02760
	IF(UZONE.LE.0.0) UZONE=FOV**2	ITA02770
	IF(RCHK.NE.UZONE) ITITL=1	ITA02780
	ZONE=UZONE	ITA02790
	IF(ALFLG.LT.0.) ALFLG=0.	ITA02800

```

IF (ALFLG.GT.9.) ALFLG=9.
IAL=IFIX(ALFLG+0.0001)
IF (IAL.GT.0) AL=10.***(3-IAL)+1.E-10
IF (IAL.EQ.0.AND.ALIGHT.EQ.0.0) ALIGHT=ALV
IF (IAL.EQ.0) AL=ALIGHT
C ***** CHECK AND WRITE PAGE TITLE HEADING
10 ILINE=ILINE+1
IF (ITITL.EQ.0.AND.ILINE.LE.49) GO TO 11
IF (ITITL.EQ.0.AND.ILINE.GT.49) WRITE (IOOUT,905)
905 FORMAT(1H,35X,28H** CONTINUED ON NEXT PAGE **)
WRITE(IOOUT,910)
910 FORMAT(1H0,41X,43(1H*)/41X,1H*,41X,1H*/41X,1H*,2X,37HINVERSE S
*TATIC TARGET DETECTION MODEL,2X,1H*/41X,1H*,41X,1H*/41X,43(1H*))
WRITE(IOOUT,911)
911 FORMAT(1H0,64X,28HTARGET INTRINSIC CONTRAST OR)
WRITE(IOOUT,912) LSC,ACON
912 FORMAT(20X,13HDEVICE NUMBER,4X,12,32X,22HTEMPERATURE DIFFERENCE,
* 2X,F7.3)
IF (MODE.EQ.2) WRITE(IOOUT,913) DIM
913 FORMAT(1H0,19X,11HFOV TYPE - ,4X,6HNARROW,24X,28HMINIMUM TARGET DI
* MENSION (M),2X,F7.3)
IF (MODE.EQ.1) WRITE(IOOUT,914) DIM
914 FORMAT(1H0,19X,11HFOV TYPE - ,6X,4HWIDE,24X,28HMINIMUM TARGET DIME
* NSION (M),2X,F7.3)
IF (IFLGFV.EQ.0) WRITE(IOOUT,915) FOV,AJOB
915 FORMAT(1H0,19X,9HFOV (DEG),5X,F7.3,24X,27HACQUISITION LEVEL (50 PC
* NT),3X,F7.3)
IF (IFLGFV.EQ.1) WRITE(IOOUT,916) FOV,AJOB
916 FORMAT(1H0,19X,9HFOV (DEG),5X,F7.3,1X,19H(ARBITRARY DEFAULT),
* 4X,27HACQUISITION LEVEL (50 PCNT),3X,F7.3)
IF (IFLGMG.EQ.0) WRITE(IOOUT,917) AMAG,ZONE
917 FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,24X,24HSEARCH ZONE (DEGREE
* S**2),6X,F7.3)
IF (IFLGMG.EQ.1) WRITE(IOOUT,918) AMAG,ZONE
918 FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,1X,19H(ARBITRARY DEFAULT),
* 4X,24HSEARCH ZONE (DEGREES**2),6X,F7.3)
WRITE (IOOUT,920)
920 FORMAT(1H0,52X,35HREQUIRES (TO DEFEAT DEVICE) AT MOST/
* 4X,14HFOR NO GREATER,6X,5HUNDER,8X,3HAND,7X,2HAT,4X,
* 35(1H-))
WRITE(IOOUT,919)
919 FORMAT(8X,5HINPUT,11X,5HINPUT,7X,5HINPUT,5X,5HINPUT,3X,
* 8HCOMPUTED,2X,11HCONTRAST OR,3X,8HCOMPUTED/6X,
* 9HDETECTION,6X,11HAMB, ILLUM.,2X,10HISKY/GROUND,2X,
* 5HRANGE,2X,10HRESOLVABLE,1X,11HTEMP. DIFF.,2X,10HTOTAL PATH
* /3X,5HPROB.,2X,9HTIME(SEC),3X,9H(FT CDLS),5X,5HRATIO,5X,
* 4H(KM),3X,10HCYCLES, RC,2X,9HAT DEVICE,2X,13HTRANSMITTANCE,
* 13X,8HCOMMENTS/3X,5(1H-),2X,9(1H-),2X,11(1H-),2X,10(1H-),
* 1X,6(1H-),2X,10(1H-),2X,9(1H-),2X,13(1H-),2X,30(1H-))
ILINE=23
C ***** BEGIN COMPUTATIONS FOV,RC,TBAR
11 S=DIM/R
IF (FOV.LE.0.) FOV=.0001
TS=1.7*ZONE/FOV**2
IF (ZONE.GT.9.0.AND.FOV.GT.5.) TS=(1.7*ZONE)/(5.0*FOV)
IF (TS.LT.5) TS=.5
CALL CYCLE(PS,PB,AJOB,RC)
IF (AL.LE.0.) AL=1.E-7
ALPRNT=AL
IF (LSC.EQ.13) GOTO 13
TBAR=0.
IF (RC.LT.0.1) GOTO 14
IF ((LSC.GT.5.AND.LSC.LT.10).OR.LSC.EQ.11.OR.LSC.EQ.14) GO TO 12
RCS=RC
PINF=1.-EXP(-1.7*RCS/6.8)
IF (PINF.LE.0.) PINF=.0001
TBAR=0.5*TS*(2.-PINF)/PINF
GO TO 14
12 TBAR=0.5*TS*(2.0-PS)/PS
GO TO 14

```

```

C ***** DEVICE #13
13  NAL=1
    ICM=0
    PS=.99
    TBAR=1.8
    C=ACON
    TTOT=1.
    TFNL=TTOT
    GO TO 22
C ***** COMPUTE C BASED ON AMBIENT ILLUMINATION, INTERPOLATE IF NECESSARY
14  IF (S.LE.0.) S=.0001
    IF (RC.LT.0.) RC=0.
    RC=RC/S
    IF (AMAG.LE.0.) AMAG=.0001
    IF (LSC.EQ.1.OR.LSC.EQ.2) RC=RC/AMAG
    IF (LSC.EQ.2) AL=0.7*AL
C ***** CORRECT FOR FIELD OF VIEW
    IF (LSC.LT.6.OR.LSC.EQ.7) GOTO 140
    IF ((LSC.EQ.6).AND.(MODE.EQ.2)) RC=RC/3.
    IF ((LSC.EQ.8).AND.(MODE.EQ.1)) RC=RC*3.
    IF ((LSC.EQ.9).AND.(MODE.EQ.1)) RC=RC*4.
    IF ((LSC.EQ.11).AND.(MODE.EQ.1)) RC=RC*3.
    IF ((LSC.EQ.14).AND.(MODE.EQ.1)) RC=RC*3.
    IF ((LSC.EQ.10).AND.(MODE.EQ.1)) RC=RC*8.
    IF ((LSC.EQ.12).AND.(MODE.EQ.2)) RC=RC/4.
140 CONTINUE
    IF (LSC.EQ.2) IAL=0
    IF (IAL.GT.0) GO TO 20
    AV=100.
    IF (AL.GT.100.) AL=100.
    DO 15 I=1,9
    IF (AL.GT.(0.999*AV)) GO TO 16
    AV=AV/10.
15  CONTINUE
    I=2
    IF (LSC.GT.5) GOTO 160
    NAL=0
    ICM=0
    GO TO 22
C ***** CHECK IF INTERPOLATION NEEDED, IF NOT, GO TO 20
16  IF (LSC.GT.5) GOTO 160
    IF (AL.GT.(1.001*AV)) GO TO 17
160  IAL=1
    GO TO 20
C ***** INTERPOLATE
17  NAL1=I-1
    NAL2=I
    ICM1=0
    ICM2=0
    CALL INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA,AMRAL,
    *AMRGM)
    NAL=0
    IF (NAL1.EQ.0.AND.NAL2.EQ.0) GO TO 22
    NAL=NAL2
    IF (NAL1.GT.0) GO TO 18
    ICM=ICM2
    GO TO 21
18  IF (NAL2.GT.0) GO TO 19
    NAL=NAL1
    ICM=ICM1
    GO TO 21
19  IF (ICM1.GT.0.AND.ICM2.GT.0) ICM=1
    IF (ICM1.LT.0.AND.ICM2.LT.0) ICM=-1
    GO TO 21
C ***** NO INTERPOLATION NEEDED
20  NAL=IAL
    CALL CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)
21  IF (ICM.EQ.-1) C=.99*C
    CALL TREQ(ACON,SOC,C,LSC,TTOT)
    IF (ICM.EQ.1) TTOT=1.

```

```

ITA03510
ITA03520
ITA03530
ITA03540
ITA03550
ITA03560
ITA03570
ITA03580
ITA03590
ITA03600
ITA03610
ITA03620
ITA03630
ITA03640
ITA03650
ITA03660
ITA03670
ITA03680
ITA03690
ITA03700
ITA03710
ITA03720
ITA03730
ITA03740
ITA03750
ITA03760
ITA03770
ITA03780
ITA03790
ITA03800
ITA03810
ITA03820
ITA03830
ITA03840
ITA03850
ITA03860
ITA03870
ITA03880
ITA03890
ITA03900
ITA03910
ITA03920
ITA03930
ITA03940
ITA03950
ITA03960
ITA03970
ITA03980
ITA03990
ITA04000
ITA04010
ITA04020
ITA04030
ITA04050
ITA04040
ITA04060
ITA04070
ITA04080
ITA04090
ITA04100
ITA04110
ITA04120
ITA04130
ITA04140
ITA04150
ITA04160
ITA04170
ITA04180
ITA04190
ITA04200

```

	TFNL=TTOT	ITA04210
C *****	PRINT RESULTS	ITA04220
22	IF(NAL.GT.0) GO TO 23	ITA04230
	TFNL=1	ITA04240
	WRITE(IOOUT,930) PS,TBAR,ALPRNT,SQG,R,TFNL	ITA04250
930	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,4X,6(1H-),7X,5(1H-),5X,	ITA04260
	F8.3,6X,28HNOTE- AMBIENT ILLUM. OUTSIDE/91X,	ITA04270
	* 29HDEVICE OPERATIONAL LIMITS. NO/91X,	ITA04280
	* 21HOBSCURATION REQUIRED./1X)	ITA04290
	ILINE=ILINE+3	ITA04300
	GO TO 4	ITA04310
23	IF(TTOT.CE.1. AND, ICM.EQ.0) GOTO 24	ITA04320
	IF(ICM.EQ.0) WRITE(IOOUT,931) PS,TBAR,ALPRNT,SQG,R,RC,C,TTOT	ITA04330
	IF(ICM.EQ.1) WRITE(IOOUT,932) PS,TBAR,ALPRNT,SQG,R,RC,TTOT,C	ITA04340
	IF(ICM.EQ.1) ILINE=ILINE+6	ITA04350
	IF(ICM.EQ.-1) WRITE(IOOUT,933) PS,TBAR,ALPRNT,SQG,R,RC,C,TTOT	ITA04360
	IF(ICM.EQ.-1) ILINE=ILINE+7	ITA04370
	GO TO 4	ITA04380
24	WRITE(IOOUT,934) PS,TBAR,ALPRNT,SQG,R,RC,C,TTOT	ITA04390
934	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,	ITA04400
	* 31HNOTE- CONTRAST (OR TEMP. DIFF.)/91X,	ITA04410
	* 31HREQUIRED WOULD EXCEED INTRINSIC/91X,	ITA04420
	* 31HCONTRAST (TEMP. DIFF.). NO OBS-/91X,	ITA04430
	* 17HCURATION REQUIRED./1X)	ITA04440
	ILINE=ILINE+3	ITA04450
	GO TO 4	ITA04460
931	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3)	ITA04470
932	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,7X,5(1H-),5X,F8.3,6X,	ITA04480
	* 29HNOTE- DETECTION PROBABILITY /91X,	ITA04490
	* 30HREQUIRES CONTRAST (OR TEMP. /91X,	ITA04500
	* 29HDIFF.) AND RESOLVABLE CYCLES /91X,	ITA04510
	* 27HABOVE LIMIT FOR DEVICE. NO /91X,	ITA04520
	* 26HOBSCURANT REQUIRED. DEVICE/91X,	ITA04530
	* 18HUPPER LIMIT IS C= ,F8.3/1X)	ITA04540
933	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,	ITA04550
	* 30HNOTE- INPUT DETECTION PROBAB- /91X,	ITA04560
	* 30HILITY REQUIRES CONTRAST (OR /91X,	ITA04570
	* 30HTEMP. DIFF.) BELOW THRESHOLD. /91X,	ITA04580
	* 29HVALUES ASSUMED ARE 99 PERCENT/91X,	ITA04590
	* 29HOF THRESHOLD. ADDITIONAL OBS-/91X,	ITA04600
	* 29HCURANT WILL NOT DECREASE /91X,	ITA04610
	* 30HDETECTION PROBABILITY. /1X)	ITA04620
	END	ITA04630

```

SUBROUTINE CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)      CIN00010
DIMENSION ALFA(9),BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)        CIN00020
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCIN00030
DATA IPOB,INEG /1,-1/                                          CIN00040
THIS ROUTINE COMPUTES THE REQUIRED CONTRAST AT THE DEVICE FOR A  CIN00050
GIVEN NUMBER OF RESOLVABLE CYCLES AT THE DEVICE. EQUATIONS    CIN00060
ARE INVERSIONS OF THOSE OF THE NV&EOL TARGET ACQUISITION      CIN00070
<FOR STATIC DETECTION> MODEL.                                  CIN00080
INPUTS: LSC = DEVICE NUMBER <1-14>                             CIN00090
        RC  = RESOLVABLE CYCLES <DECIMAL>                       CIN00100
        NAL = AMBIENT ILLUMINATION CATEGORY <1-9>               CIN00110
OUTPUTS: C  = CONTRAST OR TEMPERATURE DIFFERENCE<DEG K> REQUIRED CIN00120
        ICM = OPERATIONAL LIMITS FLAG <+1 EXCEEDS LIMIT, -1    CIN00130
        NAL = SET TO 0 IF TOO MUCH OR NOT ENOUGH AMBIENT ILLUM. CIN00140
        BY DEVICE FOR GIVEN RC.                                CIN00150
        ICM = OPERATIONAL LIMITS FLAG <+1 EXCEEDS LIMIT, -1    CIN00160
        BELOW LIMIT, 0 WITHIN RANGE>                           CIN00170
        NAL = SET TO 0 IF TOO MUCH OR NOT ENOUGH AMBIENT ILLUM. CIN00180
CFUN(RX,ALF,BET,GAM)=(BET+ALF*RX)/(GAM-RX)                     CIN00190
ICM=0                                                            CIN00200
IF(NAL.EQ.0) GO TO 9020                                          CIN00210
C ***** BRANCH TO LSC                                         CIN00220
5 GO TO (10,10,30,40,50,60,60,80,90,100,111,1200,9060,1400), LSC CIN00230
10 GO TO (110,120,130,140,150,160,170,9060,9060), NAL          CIN00240
C ***** DEVICE #1 NAL #1                                       CIN00250
110 IF(2.74.LT.RC) GO TO 9030                                     CIN00260
    IF((2.133.LT.RC).AND.(RC.LE.2.74)) GO TO 112                CIN00270
    IF((.26795.LT.RC).AND.(RC.LE.2.133)) GO TO 113              CIN00280
    IF(RC.LE..26795) GO TO 114                                    CIN00290
    GO TO 9000                                                    CIN00300
112 IEQ=1                                                         CIN00310
    GO TO 9050                                                    CIN00320
113 IEQ=2                                                         CIN00330
    GO TO 9050                                                    CIN00340
114 C=0.015                                                       CIN00350
    GO TO 9015                                                    CIN00360
C ***** DEVICE #1 NAL #2                                       CIN00370
120 IF(2.74.LT.RC) GO TO 9030                                     CIN00380
    IF((2.133.LT.RC).AND.(RC.LE.2.74)) GO TO 122                CIN00390
    IF((.0001.LE.RC).AND.(RC.LE.2.133)) GO TO 123                CIN00400
    IF(RC.LT..0001) GO TO 124                                     CIN00410
    GO TO 9000                                                    CIN00420
122 IEQ=3                                                         CIN00430
    GO TO 9050                                                    CIN00440
123 IEQ=4                                                         CIN00450
    GO TO 9050                                                    CIN00460
124 C=0.025                                                       CIN00470
    GO TO 9015                                                    CIN00480
C ***** DEVICE #1 NAL #3                                       CIN00490
130 IF(2.29.LT.RC) GO TO 131                                     CIN00500
    IF((0.49585.LE.RC).AND.(RC.LE.2.29)) GO TO 132              CIN00510
    IF(RC.LT.0.49585) GO TO 133                                  CIN00520
    GO TO 9000                                                    CIN00530
131 C=0.6324                                                      CIN00540
    GO TO 9035                                                    CIN00550
132 IEQ=5                                                         CIN00560
    GO TO 9050                                                    CIN00570
133 C=0.030                                                       CIN00580
    GO TO 9015                                                    CIN00590
C ***** DEVICE #1 NAL #4                                       CIN00600
140 IF(1.5219.LT.RC) GO TO 141                                    CIN00610
    IF((0.313.LE.RC).AND.(RC.LE.1.5219)) GO TO 142              CIN00620
    IF(RC.LT.0.313) GO TO 143                                    CIN00630
    GO TO 9000                                                    CIN00640
141 C=0.70                                                         CIN00650
    GO TO 9035                                                    CIN00660
142 IEQ=6                                                         CIN00670
    GO TO 9050                                                    CIN00680
143 C=0.05                                                         CIN00690

```

```

GO TO 9015
C ***** DEVICE #1      NAL #5
150  IF<1.1959,LT,RC> GO TO 9030
      IF<<0.0,LE,RC>>,AND,<RC,LE,1.1959>> GO TO 152
      IF<RC,LT,0.0> GO TO 153
      GO TO 9000
152  IEQ=7
      GO TO 9050
153  C=0.0841
      GO TO 9015
C ***** DEVICE #1      NAL #6
160  IF<0.44767,LT,RC> GO TO 9030
      IF<<0.0309,LE,RC>>,AND,<RC,LE,0.44767>> GO TO 162
      IF<RC,LT,0.0309> GO TO 163
      GO TO 9000
162  IEQ=8
      GO TO 9050
163  C=0.18
      GO TO 9015
C ***** DEVICE #1      NAL #7
170  IF<0.14,LT,RC> GO TO 9030
      IF<<0.05,LE,RC>>,AND,<RC,LE,0.14>> GO TO 172
      IF<RC,LT,0.05> GO TO 173
      GO TO 9000
172  IEQ=9
      GO TO 9050
173  C=0.50
      GO TO 9015
C ***** DEVICE #3
30   GO TO<9060,9060,9060,340,350,360,370,380,390>, NAL
C ***** DEVICE #3      NAL #4
340  IF<0.7497,LE,RC,AND,RC,LT,2.941> GO TO 342
      IF<RC,GE,2.941> GOTO 9075
      IF<RC,LT,0.7497> GO TO 9040
      GO TO 9000
342  IEQ=1
      GO TO 9080
C ***** DEVICE #3      NAL #5
350  IF<0.3988,LE,RC,AND,RC,LT,2.4350> GO TO 352
      IF<RC,GE,2.435> GOTO 9075
      IF<RC,LT,0.3988> GO TO 9040
      GO TO 9000
352  IEQ=2
      GO TO 9080
C ***** DEVICE #3      NAL #6
360  IF<0.15965,LE,RC,AND,RC,LT,2.1060> GO TO 362
      IF<RC,GE,2.106> GOTO 9075
      IF<RC,LT,0.15965> GO TO 9040
      GO TO 9000
362  IEQ=3
      GO TO 9080
C ***** DEVICE #3      NAL #7
370  IF<0.05498,LE,RC,AND,RC,LT,2.0375> GO TO 372
      IF<RC,GE,2.0375> GOTO 9075
      IF<RC,LT,0.05498> GO TO 9040
      GO TO 9000
372  IEQ=4
      GO TO 9080
C ***** DEVICE #3      NAL #8
380  IF<0.442,LE,RC,AND,RC,LT,3.2190> GO TO 382
      IF<RC,GE,3.219> GOTO 9075
      IF<<RC,LT,0.442>>,AND,<RC,GT,0.>> GO TO 383
      IF<RC,LE,0.> GOTO 384
      GO TO 9000
382  IEQ=5
      GO TO 9080
383  C=0.57
      GO TO 9015
384  C=0.33
      GOTO 9015

```

```

CIN00710
CIN00720
CIN00730
CIN00740
CIN00750
CIN00760
CIN00770
CIN00780
CIN00790
CIN00800
CIN00810
CIN00820
CIN00830
CIN00840
CIN00850
CIN00860
CIN00870
CIN00880
CIN00890
CIN00900
CIN00910
CIN00920
CIN00930
CIN00940
CIN00950
CIN00960
CIN00970
CIN00980
CIN00990
CIN01000
CIN01010
CIN01020
CIN01030
CIN01040
CIN01050
CIN01060
CIN01070
CIN01080
CIN01090
CIN01100
CIN01110
CIN01120
CIN01130
CIN01140
CIN01150
CIN01160
CIN01170
CIN01180
CIN01190
CIN01200
CIN01210
CIN01220
CIN01230
CIN01240
CIN01250
CIN01260
CIN01270
CIN01280
CIN01290
CIN01300
CIN01310
CIN01320
CIN01330
CIN01340
CIN01350
CIN01360
CIN01370
CIN01380
CIN01390
CIN01400

```

C ***** DEVICE #3	NAL #9	CIN01410
390	IF (0.172,LE,RC) GO TO 392	CIN01420
	IF (RC,LT,0.172,AND,RC,GT,0.) GO TO 393	CIN01430
	IF (RC,LE,0.) GOTO 394	CIN01440
	GO TO 9000	CIN01450
392	IEQ=6	CIN01460
	GO TO 9080	CIN01470
393	C=0.57	CIN01480
	GO TO 9015	CIN01490
394	C=.33	CIN01500
	GOTO 9015	CIN01510
C ***** DEVICE #4		CIN01520
40	GO TO (9060,9060,9060,440,450,460,470,480,490), NAL	CIN01530
C ***** DEVICE #4	NAL #4	CIN01540
440	IF (1.199,LE,RC,AND,RC,LT,5.2180) GO TO 442	CIN01550
	IF (RC,GE,5.218) GOTO 9075	CIN01560
	IF (RC,LT,1.199) GO TO 9040	CIN01570
	GO TO 9000	CIN01580
442	IEQ=1	CIN01590
	GO TO 9080	CIN01600
C ***** DEVICE #4	NAL #5	CIN01610
450	IF (0.6402,LE,RC,AND,RC,LT,4.2770) GO TO 452	CIN01620
	IF (RC,GE,4.277) GOTO 9075	CIN01630
	IF (RC,LT,0.6402) GO TO 9040	CIN01640
	GO TO 9000	CIN01650
452	IEQ=2	CIN01660
	GO TO 9080	CIN01670
C ***** DEVICE #4	NAL #6	CIN01680
460	IF (0.2449,LE,RC,AND,RC,LT,4.1240) GO TO 462	CIN01690
	IF (RC,GE,4.124) GOTO 9075	CIN01700
	IF (RC,LT,0.2449) GO TO 9040	CIN01710
	GO TO 9000	CIN01720
462	IEQ=3	CIN01730
	GO TO 9080	CIN01740
C ***** DEVICE #4	NAL #7	CIN01750
470	IF (0.08791,LE,RC,AND,RC,LT,3.4200) GO TO 472	CIN01760
	IF (RC,GE,3.420) GOTO 9075	CIN01770
	IF (RC,LT,0.08791) GO TO 9040	CIN01780
	GO TO 9000	CIN01790
472	IEQ=4	CIN01800
	GO TO 9080	CIN01810
C ***** DEVICE #4	NAL #8	CIN01820
480	IF (0.4394,LE,RC,AND,RC,LT,8.5970) GO TO 482	CIN01830
	IF (RC,GE,8.597) GOTO 9075	CIN01840
	IF (RC,LT,0.4394),AND,(RC,GT,0.) GO TO 483	CIN01850
	IF (RC,LE,0.) GOTO 484	CIN01860
	GO TO 9000	CIN01870
482	IEQ=5	CIN01880
	GO TO 9080	CIN01890
483	C=0.33	CIN01900
	GO TO 9015	CIN01910
484	C=0.07	CIN01920
	GOTO 9015	CIN01930
C ***** DEVICE #4	NAL #9	CIN01940
490	IF (0.1605,LE,RC) GO TO 492	CIN01950
	IF (RC,LT,0.1605,AND,RC,GT,0.) GO TO 493	CIN01960
	IF (RC,LE,0.) GOTO 494	CIN01970
	GO TO 9000	CIN01980
492	IEQ=6	CIN01990
	GO TO 9080	CIN02000
493	C=0.33	CIN02010
	GO TO 9015	CIN02020
494	C=0.07	CIN02030
	GOTO 9015	CIN02040
C ***** DEVICE # 5		CIN02050
50	GO TO (9060,9060,9060,540,550,560,570,580,590), NAL	CIN02060
540	IF (0.9189,LE,RC,AND,RC,LT,2.8060) GO TO 542	CIN02070
	IF (RC,GE,2.806) GOTO 9075	CIN02080
	IF (RC,LT,0.9189) GO TO 9040	CIN02090
	GO TO 9000	CIN02100

```

542 IEQ=1
GO TO 9080
C ***** DEVICE #5 NAL #5
550 IF (<0.59035,LE,RC,AND,RC,LT,2.7010) GO TO 552
IF (<RC,GE,2.7010) GOTO 9075
IF (<RC,LT,0.59035) GO TO 9040
GO TO 9000
552 IEQ=2
GO TO 9080
C ***** DEVICE #5 NAL #6
560 IF (<0.2881,LE,RC,AND,RC,LT,2.4800) GO TO 562
IF (<RC,GE,2.480) GOTO 9075
IF (<RC,LT,0.2881) GO TO 9040
GO TO 9000
562 IEQ=3
GO TO 9080
C ***** DEVICE #5 NAL #7
570 IF (<0.12943,LE,RC,AND,RC,LT,1.6270) GO TO 572
IF (<RC,GE,1.627) GOTO 9075
IF (<RC,LT,0.12943) GO TO 9040
GO TO 9000
572 IEQ=4
GO TO 9080
C ***** DEVICE #5 NAL #8
580 IF (<0.4949,LE,RC,AND,RC,LT,2.6960) GO TO 582
IF (<RC,GE,2.696) GOTO 9075
IF (<RC,LT,0.4949,AND,RC,GT,0.) GO TO 583
IF (<RC,LE,0.) GOTO 584
GO TO 9000
582 IEQ=5
GO TO 9080
583 C=0.33
GO TO 9015
584 C=0.07
GOTO 9015
C ***** DEVICE #5 NAL #9
590 IF (<0.1894,LE,RC,AND,RC,LT,3.2898) GO TO 592
IF (<RC,GE,3.2898) GOTO 9075
IF (<RC,LT,0.1894,AND,RC,GT,0.) GO TO 593
IF (<RC,LE,0.) GOTO 594
GO TO 9000
592 IEQ=6
GO TO 9080
593 C=0.33
GO TO 9015
594 C=0.07
GOTO 9015
C ***** DEVICES #6 AND #7
60 IF (<1.7934,LT,RC) GO TO 601
IF (<<0.92376,LT,RC>>,AND,<RC,LE,1.7934>>) GO TO 602
IF (<<0.11022,LE,RC>>,AND,<RC,LE,0.92376>>) GO TO 603
IF (<RC,LT,0.11022) GO TO 9070
GO TO 9000
601 C=8.38
GO TO 9035
602 C=<1.72363*RC-0.0392775>/<1.82560-RC>
GO TO 9020
603 C=<0.09772*RC>/<1.-0.34779*RC>
GO TO 9020
C ***** DEVICE #8
80 IF (<4.702,LT,RC) GO TO 801
IF (<<0.996,LE,RC>>,AND,<RC,LE,4.702>>) GO TO 802
IF (<RC,LT,0.996) GO TO 803
GO TO 9000
801 C=2.18
GO TO 9035
802 C=<0.0298*RC>/<1.-0.199*RC>
GO TO 9020
803 C=0.037
GO TO 9015

```

```

CIN02110
CIN02120
CIN02130
CIN02140
CIN02150
CIN02160
CIN02170
CIN02180
CIN02190
CIN02200
CIN02210
CIN02220
CIN02230
CIN02240
CIN02250
CIN02260
CIN02270
CIN02280
CIN02290
CIN02300
CIN02310
CIN02320
CIN02330
CIN02340
CIN02350
CIN02360
CIN02370
CIN02380
CIN02390
CIN02400
CIN02410
CIN02420
CIN02430
CIN02440
CIN02450
CIN02460
CIN02470
CIN02480
CIN02490
CIN02500
CIN02510
CIN02520
CIN02530
CIN02540
CIN02550
CIN02560
CIN02570
CIN02580
CIN02590
CIN02600
CIN02610
CIN02620
CIN02630
CIN02640
CIN02650
CIN02660
CIN02670
CIN02680
CIN02690
CIN02700
CIN02710
CIN02720
CIN02730
CIN02740
CIN02750
CIN02760
CIN02770
CIN02780
CIN02790
CIN02800

```


C ***** DEVICE #9	
90 IF<0.37181,LE,RC,AND,RC,LT,9.14027> GO TO 901	CIN02810
IF<RC,GE,9.14027> GOTO 9075	CIN02820
IF<RC,LT,0.37181> GO TO 9070	CIN02830
GO TO 9000	CIN02840
901 C=<0.0289*RC>/<1.-0.1092*RC>	CIN02850
GO TO 9020	CIN02860
C ***** DEVICE #10	CIN02870
100 IF<NAL,GT,2> GO TO 9060	CIN02880
IF<4.227,LE,RC,AND,RC,LT,91.05956> GO TO 1010	CIN02890
IF<RC,GE,91.05956> GOTO 9075	CIN02900
C=0.01	CIN02910
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT	CIN02920
C=C/1.63	CIN02930
GO TO 9015	CIN02940
1010 C=<0.0303*RC>/<13.43-0.1473*RC>	CIN02950
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT	CIN02960
C=C/1.63	CIN02970
GO TO 9020	CIN02980
C ***** DEVICE # 11	CIN02990
1111 IF<0.5057,LE,RC,AND,RC,LT,7.73353> GO TO 1110	CIN03000
IF<RC,GE,7.73353> GOTO 9075	CIN03010
GO TO 9070	CIN03020
1110 C=<0.0207*RC>/<1.-0.1291*RC>	CIN03030
GO TO 9020	CIN03040
C ***** DEVICE # 12	CIN03050
1200 IF<NAL,GT,2> GO TO 9060	CIN03060
IF<3.48,LE,RC,AND,RC,LT,31.48113> GO TO 1210	CIN03070
IF<RC,GE,31.4883> GOTO 9075	CIN03080
C=0.01	CIN03090
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT	CIN03100
C=C/1.63	CIN03110
GO TO 9015	CIN03120
1210 C=<0.0206*RC>/<8.06-0.2559*RC>	CIN03130
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT	CIN03140
C=C/1.63	CIN03150
GO TO 9020	CIN03160
C ***** DEVICE # 14	CIN03170
1400 IF<0.9098,LE,RC,AND,RC,LT,4.57711> GO TO 1410	CIN03180
IF<RC,GE,4.57711> GOTO 9075	CIN03190
C=0.037	CIN03200
GO TO 9015	CIN03210
1410 C=<0.0297567*RC>/<0.91287-0.1991449*RC>	CIN03220
GO TO 9020	CIN03230
9000 WRITE<I00UT,9010> LSC,NAL,RC	CIN03240
9010 FORMAT<5X,46HBAD PARAMETER PASSED TO SUBRTN CINV **** LSC= ,I3,	CIN03250
* 6H NAL= ,I3,5H RC= ,F10.4>	CIN03260
9015 ICM=INEG	CIN03270
9020 RETURN	CIN03280
9030 C=0.80	CIN03290
9035 ICM=IPOS	CIN03300
GO TO 9020	CIN03310
9040 C=0.02	CIN03320
GO TO 9015	CIN03330
9050 C=CFUN<RC,ALFA<IEQ>,BETA<IEQ>,GAMA<IEQ>>	CIN03340
GO TO 9020	CIN03350
9060 NAL=0	CIN03360
GO TO 9020	CIN03370
9070 C=0.0112	CIN03380
GO TO 9015	CIN03390
9075 C=100.	CIN03400
GOTO 9035	CIN03410
9080 L=LSC-2	CIN03420
C=CFUN<RC,AMRAL<IEQ,L>,0.,AMRGM<IEQ,L>>	CIN03430
GOTO 9020	CIN03440
END	CIN03450
	CIN03460

```

SUBROUTINE INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA, INT00010
*AMRAL,AMRGM) INT00020
C INTERPOLATION ON THE SURFACE IN C, RC, AMBIENT ILLUMINATION SPACE. INT00030
C THE USER INPUTS A VALUE OF RC, AND AN AMBIENT ILLUMINATION THAT INT00040
C LIES BETWEEN ONE PAIR OF MODELED RC VS C CURVES AT TWO DISCRETELY INT00050
C MODELED AMBIENT ILLUMINATIONS. THE TECHNIQUE IS COMPLICATED BY A INT00060
C REQUIREMENT TO REPRODUCE ALMOST EXACTLY THE INTERPOLATED VALUE INT00070
C FROM RC VS C OF THE NVL TARGET ACQUISITION MODEL. THUS, WHILE INT00080
C RC IS INPUT IN THIS INVERSION MODEL, THE INTERPOLATION IS BETWEEN INT00090
C RC VALUES AT CONSTANT C OVER TWO AMB. ILLUM. REGIONS. THE INT00100
C THRESHOLDS OR OPERATIONAL LIMITS OF THE DEVICE ARE TREATED SEPAR- INT00110
C ATELY. IN ALL, FOUR TYPES OF INTERPOLATION SITUATIONS ARE INT00120
C UTILIZED. INT00130
C
C DIMENSION IRGNS(21),INDEX(21),RCL(25),RCU(25),CU(25),CL(25), INT00150
*IEQ1(36),IEQ2(36),RLW(36),RUP(36),CLW(36),CUP(36),ALFA(9), INT00160
*BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3) INT00170
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU INT00180
C** NUMBER OF C REGIONS FOR INTERPOLATION CONSIDERATION, DEV. 1-5 INT00200
DATA IRGNS /4,4,3,3,2,2,1,1,1,2,1,1,1,1,2,1,1,1,2,1,1/ INT00210
C** STARTING INDEX FOR EACH DEVICE, AMB.ILLUM. PAIR. INT00220
DATA INDEX /1,5,9,12,15,17,19,20,21,22,24,25,26,27,28,30,31, INT00230
*32,33,34,36/ INT00240
C** LOWER LIGHT LEVEL THRESHOLDS OF RC. INT00250
DATA RCL / INT00260
*.0001 ,.26795,.49585,.313 ,0. ,.0309 ,.05 ,.7497 ,.3988 , INT00270
*.15965,.055 ,.442 ,.1742 ,.1199 ,.6402 ,.2449 ,.0879 ,.4394 , INT00280
*.1605 ,.9189 ,.5903 ,.288 ,.1294 ,.4949 ,.1894 / INT00290
C** UPPER LIGHT LEVEL THRESHOLDS OF RC INT00300
DATA RCU / INT00310
*2.74 ,2.74 ,2.29 ,1.5219,1.1959, .44767,.14 ,2.941 ,2.435 , INT00320
*2.106 ,2.0375,3.219 ,3.219 ,5.218 ,4.277 ,4.124 ,3.42 ,8.579 , INT00330
*8.597 ,2.806 ,2.701 ,2.48 ,1.627 ,2.696 ,3.2898/ INT00340
C** VALUE OF C AT LOWER THRESHOLDS. INT00350
DATA CL / INT00360
*.015 ,.025 ,.03 ,.05 ,.0841 ,.18 ,.50 ,.02 ,.02 ,.02 , INT00370
*.02 ,.57 ,.57 ,.02 ,.02 ,.02 ,.02 ,.33 ,.33 ,.02 , INT00380
*.02 ,.02 ,.02 ,.33 ,.33 / INT00390
C** VALUE OF C AT UPPER THRESHOLD INT00400
DATA CU / INT00410
*.80 ,.80 ,.6324,.70 ,.80 ,.80 ,.80 ,.100 ,.100 ,.100 , INT00420
*100 ,.100 ,.16.88,100 ,.100 ,.100 ,.100 ,.100 ,.16.17,100 , INT00430
*100 ,.100 ,.100 ,.100 ,.100 / INT00440
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR UPPER AMB. ILL. INT00450
DATA IEQ1 /2,2,1,1,4,4,3,3,5,5,0,6,6,0,7,7,8,8,1,2,3,4,4,5,1,2,3, INT00460
* 4,4,5,1,2,3,4,4,5/ INT00470
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR LOWER AMB ILL. INT00480
DATA IEQ2 /-1,4,4,3,-1,5,5,0,-1,6,6,-1,7,7,-1,8,-1,9,2,3,4,-1,5, INT00490
* 6,2,3,4,-1,5,6,2,3,4,-1,5,6/ INT00500
C** LOWEST RC VALUE IN EACH REGION. INT00510
DATA RLW / INT00520
*0. ,.26794,2.118 ,2.133 ,.26794,.4958 ,1.8439,2.29 ,.312 , INT00530
*.312 ,1.5119,0. ,0. ,1.1758,0. ,.03089,.03089,.05 , INT00540
*.3987 ,.15964,.054 ,.054 ,.442 ,.174 ,.64 ,.244 ,.087 , INT00550
*.087 ,.4394 ,.1605 ,.59 ,.287 ,.129 ,.129 ,.4948 ,.1894 / INT00560
C** LARGEST RC VALUE IN EACH REGION. INT00570
DATA RUP / INT00580
*.67890,2.1335,2.1637,2.74 ,.61280,2.133 ,2.5764,2.74 ,.739 , INT00590
*2.29 ,2.29 ,.84083,1.5219,1.5219,.712 ,1.1959,.3751 ,.4477 , INT00600
*2.942 ,2.435 ,2.106 ,.9023 ,3.219 ,3.219 ,5.218 ,4.277 ,4.124 , INT00610
*1.0396,8.597 ,8.597 ,2.81 ,2.71 ,2.49 ,.9575 ,2.696 ,3.29 / INT00620
C** LOWEST VALUE OF C OVER EACH REGION. INT00630
DATA CLW / INT00640
*.015 ,.025 ,.250 ,.270 ,.025 ,.030 ,.270 ,.6324 ,.030 , INT00650
*.050 ,.6324 ,.050 ,.084107,.70 ,.0841 ,.180 ,.180 ,.500 , INT00660
*.020 ,.020 ,.020 ,.020 ,.570 ,.570 ,.020 ,.020 ,.020 , INT00670
*.020 ,.330 ,.330 ,.020 ,.020 ,.020 ,.020 ,.330 ,.330 / INT00680
C** LARGEST VALUE OF C OVER EACH REGION. INT00690
DATA CUP / INT00700

```

	*.025	..250	..270	..800	..030	..270	..6324	..800	..050	INT00710
	*.6324	..700	..084107	..70	..800	..180	..800	..500	..800	INT00720
	*100.	..100.	..100.	..570	..100.	..16.88	..100.	..100.	..100.	INT00730
	*.330	..100.	..16.17	..100.	..100.	..100.	..330	..100.	..100.	INT00740

```

C** SET LIGHT LEVELS, FRACTIONAL INTERPOLANT.
  ICM1=0
  ICM2=0
  IF (NAL1.EQ.NAL2) GOTO 10
  IF (LSC.GT.5) GOTO 10
  AL1=10.**((3-NAL1))
  AL2=10.**((3-NAL2))
  FACTR=(AL-AL2)/(AL1-AL2)
  DFAC=1.-FACTR
  GOTO (100,100,200,200,200),LSC
10  CALL CINV(LSC,RC,NAL1,C,ICM1,ALFA,BETA,GAMA,AMRAL,AMRGM)
20  NAL2=0
  RETURN
30  CALL CINV(LSC,RC,NAL2,C,ICM2,ALFA,BETA,GAMA,AMRAL,AMRGM)
40  NAL1=0
  RETURN
C** DEVICES 1,2
100 IF (NAL2.LE.7) GOTO 105
  IF (NAL1.EQ.7.AND.FACTR.GT.0.75) GOTO 10
  NAL1=0
  GOTO 20
105 NRG=IRGNS(NAL1)
  IDX=INDEX(NAL1)
  LIM1=NAL1
  LIM2=NAL2
  GOTO 500
C** DEVICES 3,4,5
200 IF (NAL1.GE.4) GOTO 205
  IF (NAL2.EQ.4.AND.FACTR.LT.0.3) GOTO 30
  NAL2=0
  GOTO 40
205 IS=6+(LSC-3)*5+(NAL1-3)
  NRG=IRGNS(IS)
  IDX=INDEX(IS)
  LIM1=7+(LSC-3)*6+(NAL1-3)
  LIM2=LIM1+1
  BVAL=0.
  IDV=LSC-2
C** FIRST, CHECK LIMIT.
500 IF (RC.GE.RCL(LIM1).AND.RC.GE.RCL(LIM2)) GOTO 510
  CALL CASE3(FACTR,CNOT,RCN,RCL(LIM1),RCL(LIM2),CL(LIM1),CL(LIM2))
  IF (RC.GT.RCN) GOTO 520
  CV=CNOT
  ICM1=-1
  ICM2=-1
  GOTO 800
510 IF (RC.LE.RCU(LIM1).AND.RC.LE.RCU(LIM2)) GOTO 520
  CALL CASE3(FACTR,CNOT,RCN,RCU(LIM1),RCU(LIM2),CU(LIM1),CU(LIM2))
  IF (RC.LT.RCN) GOTO 520
  CV=CNOT
  ICM1=1
  ICM2=1
  GOTO 800
C** RC IS IN BOUNDS OF OPERATIONAL LIMITS, BUT NOT NECESSARILLY
C BETWEEN CURVES. TEST SUB-REGIONS IN TURN.
520 DO 600 I=1,NRG
  J=IDX+I-1
  IF (RC.LT.RL(J).OR.RC.GT.RUP(J)) GOTO 600
C** RC IS IN SUB-REGION. NOW TEST C
  CV=0.
  CR1=0.
  CR2=0.
  IQ1=IEQ1(J)
  IQ2=IEQ2(J)
  IF (IQ1.GT.0) GOTO 530
  RCN=RCU(LIM1)

```

	CNOT=CU(LIM1)	INT01410
	IF (IQ1.LT.0) RCN=RCL(LIM1)	INT01420
	IF (IQ1.LT.0) CNOT=CL(LIM1)	INT01430
	IF (LSC.LE.2) CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ2)),	INT01440
	*BETA(IQ2),GAMA(IQ2))	INT01450
	IF (LSC.GT.2)CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,	INT01460
	*AMRAL(IQ2,IDV),BVAL,AMRGM(IQ2,IDV))	INT01470
	IF (IFLG.NE.0) GOTO 600	INT01480
	IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600	INT01490
	IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600	INT01500
	IF (RCN.EQ.RCU(LIM1)) ICM1=1	INT01510
	IF (RCN.EQ.RCL(LIM1)) ICM1=-1	INT01520
	GOTO 800	INT01530
530	IF (IQ2.GT.0) GOTO 540	INT01540
	RCN=RCU(LIM2)	INT01550
	CNOT=CU(LIM2)	INT01560
	IF (IQ2.LT.0) RCN=RCL(LIM2)	INT01570
	IF (IQ2.LT.0) CNOT=CL(LIM2)	INT01580
	IF (LSC.LE.2) CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ1),	INT01590
	*BETA(IQ1),GAMA(IQ1))	INT01600
	IF (LSC.GT.2)CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,	INT01610
	*AMRAL(IQ1,IDV),BVAL,AMRGM(IQ1,IDV))	INT01620
	IF (IFLG.NE.0) GOTO 600	INT01630
	IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600	INT01640
	IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600	INT01650
	IF (RCN.EQ.RCU(LIM2)) ICM2=1	INT01660
	IF (RCN.EQ.RCL(LIM2)) ICM2=-1	INT01670
	GOTO 800	INT01680
540	IF(LSC.LE.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,ALFA(IQ1),ALFA(IQ2),	INT01690
	*BETA(IQ1),BETA(IQ2),GAMA(IQ1),GAMA(IQ2))	INT01700
	IF(LSC.GT.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,AMRAL(IQ1,IDV),	INT01710
	*AMRAL(IQ2,IDV),BVAL,BVAL,AMRGM(IQ1,IDV),AMRGM(IQ2,IDV))	INT01720
	IF (IFLG.GE.2) GOTO 600	INT01730
	CV=CR1	INT01740
	IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800	INT01750
	IF (IFLG.EQ.1) GOTO 600	INT01760
	CV=CR2	INT01770
	IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800	INT01780
600	CONTINUE	INT01790
	C** NO VALUES MET INTERPOLATION CRITERIA.	INT01800
	C** FINAL TEST IS IN THE LIMITING REGIONS	INT01810
640	CONTINUE	INT01820
	WRITE (IOOUT,1000) LSC,AL1,AL2	INT01830
1000	FORMAT(1X,40H*** IN ITAM INTAL ROUTINE, DEVICE LSC = ,I2,	INT01840
	*/1X,40H*** COULD NOT INTERPOLATE BETWEEN AMBIENT ILLUM ,F11.6,	INT01850
	*5H AND ,F11.6/1X,39H*** UPPER AMBIENT ILLUM. VALUE ASSUMED.)	INT01860
	GOTO 10	INT01870
800	C=CV	INT01880
	RETURN	INT01890
	END	INT01900

	SUBROUTINE CASE1(FACTR,RC,CR1,CR2,IFLG,ALP1,ALP2,BET1,BET2, *GAM1,GAM2)	CSA00010
C	FINDS CONTRAST C INTERPOLATING ON RC VS C. RETURNS AT MOST TWO	CSA00020
C	ROOTS CR1, CR2. IFLG=1 IF ROOTS ARE IDENTICAL. IFLG=2 IF ROOTS	CSA00030
C	ARE COMPLEX.	CSA00040
C		CSA00050
	DFAC=1,-FACTR	CSA00060
	A1=RC-GAM1*FACTR-GAM2*DFAC	CSA00070
	A2=RC*(ALP1+ALP2)+(BET1-ALP2*GAM1)*FACTR+(BET2-ALP1*GAM2)*DFAC	CSA00080
	A3=ALP1*ALP2*RC+ALP2*BET1*FACTR+ALP1*BET2*DFAC	CSA00090
	IFLG=2	CSA00100
	IF (A1.LT.-1.E-10.OR.A1.GT.1.E-10) GOTO 10	CSA00110
	IF (A2.EQ.0.) RETURN	CSA00120
	IFLG=1	CSA00130
	CR1=-A3/A2	CSA00140
	CR2=CR1	CSA00150
	RETURN	CSA00160
10	DISCR=A2*A2-4.*A1*A3	CSA00170
	IF (DISCR.LT.0.) RETURN	CSA00180
	IF (DISCR.EQ.0.) IFLG=1	CSA00190
	IF (DISCR.GT.0.) IFLG=0	CSA00200
	VA=SQRT(DISCR)/(2.*A1)	CSA00210
	VB=-A2/(2.*A1)	CSA00220
	SG=-1.	CSA00230
	IF (A1.LT.0.) SG=1.	CSA00240
	CR1=VB+SG*VA	CSA00250
	CR2=VB-SG*VA	CSA00260
	RETURN	CSA00270
	END	CSA00280
		CSA00290
		CSA00300

```

SUBROUTINE CASE2(FACTR,RC,C,IFLG,CP,RNOT,CNOT,ALP,BET,GAM)
C FINDS INTERPOLATED CONTRAST C FOR CASE OF SINGLE FIXED RC AT
C SOME LIMIT.
IFLG=1
IF (FACTR.NE.1.) GOTO 10
IF (RC.NE.RNOT) RETURN
GOTO 90
10 IF (RC.EQ.RNOT) GOTO 25
DIFF=RC-RNOT*FACTR
IF (DIFF.NE.0.) GOTO 20
IF (GAM.EQ.0.) RETURN
CP=BET/GAM
GOTO 50
20 VA=BET*(1.-FACTR)+ALP*DIFF
VB=GAM*(1.-FACTR)-DIFF
IF (VB.EQ.0.) GOTO 90
CP=VA/VB
GOTO 50
25 IF (GAM.EQ.RC) RETURN
CP=(ALP*RC+BET)/(GAM-RC)
GOTO 60
50 RP=(GAM*CP-BET)/(ALP+CP)
C=CNOT+(CP-CNOT)*(RC-RNOT)/(RP-RNOT)
GOTO 100
60 C=CNOT*FACTR+CP*(1.-FACTR)
GOTO 100
90 C=CNOT
CP=CNOT
IFLG=0
RETURN
END

```

```

CSB00010
CSB00020
CSB00030
CSB00040
CSB00050
CSB00060
CSB00070
CSB00080
CSB00090
CSB00100
CSB00110
CSB00120
CSB00130
CSB00140
CSB00150
CSB00160
CSB00170
CSB00180
CSB00190
CSB00200
CSB00210
CSB00220
CSB00230
CSB00240
CSB00250
CSB00260
CSB00270
CSB00280
CSB00290
CSB00300
CSB00310
CSB00320
CSB00330

```

	SUBROUTINE CASE3(FACTR,C,RCN,RNOT1,RNOT2,CNOT1,CNOT2)	CSC00010
C	FINDS INTERPOLATED C FOR CASE OF BOTH RC VALUES FIXED AT	CSC00020
C	LIMITS.	CSC00030
C		CSC00040
	RCN=RNOT1*FACTR+RNOT2*(1.-FACTR)	CSC00050
	IF (RNOT1.EQ.RNOT2) GOTO 10	CSC00060
	C=CNOT1+(CNOT2-CNOT1)*(RCN-RNOT1)/(RNOT2-RNOT1)	CSC00070
	RETURN	CSC00080
10	C=CNOT1*FACTR+CNOT2*(1.-FACTR)	CSC00090
	RETURN	CSC00100
	END	CSC00110
		CSC00120

```

SUBROUTINE CYCLE(PS,PB,AJOB,RC)
THIS ROUTINE COMPUTES THE NORMALIZED NUMBER OF RESOLVABLE CYCLES
FOR GIVEN CUMULATIVE NORMAL DISTRIBUTION OF PROBABILITY OF
DETECTION.
INPUTS: PS = CUMULATIVE PROBABILITY OF DETECTION.
        PB = STANDARD DEVIATION FOR PS (.632)
        AJOB = MEAN RESOLVABLE CYCLES FOR 50 PERCENT PROBABILITY
              OF DETECTION.
OUTPUTS: RC = RESOLVABLE CYCLES REQUIRED FOR PS
SUBROUTINE: CYCLE CALLED BY ITAM. CALLS NONE.

COMMON /IQUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
DATA A0,A1,A2 /2.515517,0.802853,0.010328/
DATA B1,B2,B3 /1.432788,0.189269,0.001308/
ICASE=0
IF((0.0,LE,PS).AND.(PS,LE,0.0000003)) GO TO 10
IF((.9999997,LE,PS).AND.(PS,LE,1.)) GO TO 20
IF((.0000003,LT,PS).AND.(PS,LE,0.5)) GO TO 30
IF((.5,LT,PS).AND.(PS,LT,0.9999997)) GO TO 40
C ***** ERROR ROUTINE
WRITE(IOOUT,900) PS
RETURN
C ***** SPECIAL CASE I
10 X=-5.
IF (PS,EQ,0.) X=-45.
GO TO 60
C ***** SPECIAL CASE II
20 X=5.
IF (PS,EQ,1.) X=45.
GO TO 60
C ***** CASE I
30 XX=SQRT(ALOG(1/(PS*PS)))
ICASE=1
GO TO 50
C ***** CASE II
40 XX=SQRT(ALOG(1/(1,-PS)**2))
50 X=XX-(A0+A1*XX+A2*(XX*XX))/(1.+B1*XX+B2*(XX*XX)+B3*(XX*XX*XX))
IF(ICASE,EQ,1) X=X*(-1.)
60 RC=AJOB+AJOB*PB*X
RETURN
900 FORMAT(2X,39HERROR - INCORRECT PS VALUE PASSED PS = ,F10.4)
END

```



```

SUBROUTINE TREQ(ACON,SOG,C,LSC,TRQ)
C THIS ROUTINE COMPUTES THE TRANSMISSION REQUIRED TO MATCH GIVEN
C TARGET AND DETECTOR CONTRASTS, WITH MODIFICATION BY SKY/
C GROUND RATIO FOR NON-THERMAL DEVICES.
C
C INPUTS: ACON = INTRINSIC CONTRAST OF TARGET, OR TEMP. DIFF.
C          (DEG K) FOR THERMAL.
C          SOG = SKY/GROUND RATIO.
C          C   = CONTRAST OR TEMP. DIFF. SEEN AT DEVICE
C          LSC = DEVICE NUMBER.
C OUTPUTS: TRQ = REQUIRED TRANSMISSION TO REDUCE ACON TO C.
C SUBROUTINE: TREQ CALLED BY ITAM. CALLS NONE.
C
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
IF(LSC.LT.1).OR.(LSC.GT.14) GO TO 30
IF(LSC.GT.5) GO TO 20
C ***** NON-THERMAL
10 IF (SOG.EQ.0. .OR. C.EQ.0.) GOTO 45
TRQ=1./(<math>1. + \frac{ACON-C}{C+SOG}</math>))
GOTO 50
20 IF(LSC.EQ.10).OR.(LSC.EQ.12) GO TO 10
IF(LSC.EQ.13) GO TO 30
C ***** THERMAL
IF (ACON.EQ.0.) GOTO 25
TRQ=C/ACON
GOTO 50
25 TRQ=1.
GOTO 50
C ***** ERROR ROUTINE
30 WRITE(IOOUT,100) LSC
TRQ=1.
100 FORMAT(5X,19HINPUT ERROR LSC = ,I2)
RETURN
45 TRQ=0.
50 IF (TRQ.LT.0.) TRQ=0.
IF (TRQ.GT.1.) TRQ=1.
RETURN
END
TRQ00010
TRQ00020
TRQ00030
TRQ00040
TRQ00050
TRQ00060
TRQ00070
TRQ00080
TRQ00090
TRQ00100
TRQ00110
TRQ00120
TRQ00130
TRQ00140
TRQ00150
TRQ00160
TRQ00170
TRQ00180
TRQ00190
TRQ00200
TRQ00210
TRQ00220
TRQ00230
TRQ00240
TRQ00250
TRQ00260
TRQ00270
TRQ00280
TRQ00290
TRQ00300
TRQ00310
TRQ00320
TRQ00330
TRQ00340
TRQ00350
TRQ00360
TRQ00370
TRQ00380
TRQ00390

```

```

SUBROUTINE FLOUD(LAMBDA,TRANS,IERR)
REAL LAMBDA,KAPPA,LB0,LP
REAL L,M,N,LO,MO,NO,LOS,MOS,NOS,LOM,LON,NOM,MXLY,NXLZ,MZNY
DIMENSION IALPH(10)
COMMON /IUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /FGEOM/XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS
COMMON /OPT/ INDEXP,ETA,KAPPA,W0,THETA0,PHI0,LD,TAUBAR,
+RHO,LB0,TMPA,TMPC
COMMON /GEOMET/PTS(15),IGEOSW
DATA IZERO/0/
DATA IALPH/2HCP,2HRP,2HSP,2HAX,2HCL,2HAT,2HBK,
+2HSA,2HLU,2HGO /
ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
*****
SUBROUTINE FLOUD COMPUTES THE FOLLOWING QUANTITIES FOR A FINITE
ELLIPSOIDAL CLOUD:
S = LENGTH OF OPTICAL PATH IN CLOUD
TRANS = BEAM TRANSMITTANCE THROUGH CLOUD
LF = PATH RADIANCE
TC = CONTRAST TRANSMITTANCE.
INPUT OF COMPUTATIONAL PARAMETERS TAKES PLACE THROUGH AN ORDER
INDEPENDENT READ OF A GROUP OF RECORDS IDENTIFIED BY THE FOLLOWING
MNEMONICS (EACH RECORD IS FORMAT (A4,1X,5(E10.4,1X)) ):
MNEUMONIC      VARIABLES READ      DESCRIPTION
-----
CPOS           XC,YC,ZC           CLOUD CENTER POSITION
RPOS           XR,YR,ZR           RECEIVER POSITION
SPOS           XS,YS,ZS           SOURCE POSITION
AXES           AE,BE,CE           SEMI-AXES OF CLOUD ELLIPSOID
CLDS          INDEXP,ETA,KAPPA,W0,TMPC  CLOUD AEROSOL PARAMETERS
ATMO          TAUBAR,TMPA        ATMOSPHERIC PARAMETERS
BKGR          RHO,LB0           BACKGROUND PARAMETERS
SANG          THETA0,PHI0       SOLAR ANGLES
LUND          LD             LUNAR DAY
GO            TERMINATES READ
** NOTE : THE GO SENTINEL CARD MUST BE THE LAST CARD READ
THIS INPUT DATA IS STORED FOR LATER USE IN COMMON
BLOCKS /FGEOM/ AND /OPT/.
/FGEOM/ INPUT PARAMETERS:
(XC,YC,ZC) = CENTER OF ELLIPSOID
(AE,BE,CE) = SEMI-AXES OF ELLIPSOID
(XR,YR,ZR) = COORDINATES OF RECEIVER LOCATION
(XS,YS,ZS) = COORDINATES OF SOURCE LOCATION.
/OPT/ INPUT PARAMETERS:
INDEXP = PHASE FUNCTION IDENTIFIER
=0, USER SUPPLIED
=1, MARITIME ARCTIC, VIS=0.1 TO 2.0 KM
=2, MARITIME POLAR, VIS=0.2 KM
=3, MARITIME POLAR, VIS=02. KM
=4, CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM
=5, WHITE PHOSPHORUS
=6, HEXACHLOROETHANE
=7, FOG OIL
=8, DUST (MODERATE AEROSOL LOADING)
=9, DUST (HEAVY AEROSOL LOADING)
=10, MARITIME MODEL B, VIS=5KM, RH=95%
=11, MARITIME MODEL B, VIS=10KM,RH=90%

```



```

C 1 < LD < 28 THE IRRADIANCE VALUE IS FOR LUNAR DAY LD.
*****
INITIALIZATION OF INPUT DATA.
99 IF (IZERO.NE.0) WRITE (IOOUT,99)
   FORMAT(1H1)
   IF(IZERO.NE.0) GO TO 477
   XC=0.0
   YC=0.0
   ZC=0.0
   XR=0.0
   YR=0.0
   ZR=0.0
   XS=0.0
   YS=0.0
   ZS=0.0
   AE=0.0
   BE=0.0
   CE=0.0
   INDEXP=0
   ETA=0.0
   KAPPA=0.0
   WD=0.0
   IMPC=0.0
   TAUBAR=0.0
   IMPA=0.0
   RHO=0.0
   LBO=0.0
   THETA0=0.0
   PHI0=0.0
   LD=0
   ISU=0
   IZERO=1
477 CONTINUE
   IFLG=2
   IFLO=1
   DO 360 K=1,10
   READ(IOIN,334)IA,IA2,R1,R2,R3,R4,R5
334 FORMAT(2A2,1X,5(E10.4,1X))
   DO 333 I=1,11
   IF(IA.NE.IALPH(I)) GO TO 333
   IND=I
   IF(IND.EQ.10) GO TO 361
333 CONTINUE
   IF(K.EQ.10.AND.IND.NE.10) GO TO 358
   IF(IND.LT.5)IFLG=0
   IF(IND.GE.5.AND.IND.LE.9)IFLO=0
   IF(IND.EQ.11) GO TO 355
   IF(IND.LT.5) GO TO (341,342,343,344),IND
   INDM4=IND-4
   GO TO (345,346,347,348,349),INDM4
341 XC=R1
   YC=R2
   ZC=R3
   GO TO 360
342 XR=R1
   YR=R2
   ZR=R3
   GO TO 360
343 XS=R1
   YS=R2
   ZS=R3
   GO TO 360
344 AE=R1
   BE=R2
   CE=R3
   GO TO 360
345 INDEXP=IFIX(R1)

```

```

FCL01340
FCL01350
FCL01360
FCL01370
FCL01380
FCL01390
FCL01400
FCL01410
FCL01420
FCL01430
FCL01440
FCL01450
FCL01460
FCL01470
FCL01480
FCL01490
FCL01500
FCL01510
FCL01520
FCL01530
FCL01540
FCL01550
FCL01560
FCL01570
FCL01580
FCL01590
FCL01600
FCL01610
FCL01620
FCL01630
FCL01640
FCL01650
FCL01660
FCL01670
FCL01680
FCL01690
FCL01700
FCL0
FCL0172
FCL01730
FCL01740
FCL01750
FCL01760
FCL01770
FCL01780
FCL01790
FCL01800
FCL01810
FCL01820
FCL01830
FCL01840
FCL01850
FCL01860
FCL01870
FCL01880
FCL01890
FCL01900
FCL01910
FCL01920
FCL01930
FCL01940
FCL01950
FCL01960
FCL01970
FCL01980
FCL01990
FCL02000

```

```

IF (R2.GT..001) ETA=R2
KAPPA=R3
W0=R4
TMPC=R5
GO TO 360
346 TAUBAR=R1
TMPA=R2
GO TO 360
347 RH0=R1
LB0=R2
GO TO 360
348 THETA0=R1
PHI0=R2
GO TO 360
349 LD=IFIX(R1)
GO TO 360
355 WRITE(IOOUT,357)
357 FORMAT(1H0,20X,46H***F CLOUD ERROR*** INPUT RECORD DETECTED WHICH,
+ 44HDOES NOT CORRESPOND TO CORRECT INPUT FORMAT,/)
GO TO 360
358 WRITE(IOOUT,359)
359 FORMAT(1H0,35X,45H***F CLOUD ERROR*** TOO MANY INPUT CARDS OR GO,
+ 16H SENTINEL ABSENT,/)
IERR=1
GO TO 200
360 CONTINUE
361 CONTINUE
ISW=IFLG+IFLO
IF (ISW.EQ.3) ISW=2
IF (IGEOSW.NE.1) GO TO 222
XC=PTS(13)
YC=PTS(14)
ZC=PTS(15)
XR=PTS(4)
YR=PTS(5)
ZR=PTS(6)
XS=PTS(1)
YS=PTS(2)
ZS=PTS(3)
222 CONTINUE
C
C ECHO INPUT
C
WRITE(IOOUT,1000) XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS
IF (ETA.LT.1.E-20) CALL PFNN(LAMBDA,0.,INDEXP,PFN,ETA)
WRITE(IOOUT,1100) INDEXP,ETA,LAMBDA,KAPPA,W0,TAUBAR,THETA0,PHI0,
1RH0,LB0,TMPA,TMPC,LD
IF (ISW.EQ.2) GO TO 15
C
C ISW .NE. 2 INDICATES PRELIMINARY GEOMETRICAL CALCULATIONS TO
C BE PERFORMED: COMPUTE INTERSECTIONS (XM,YM,ZM) AND (XN,YN,ZN) OF
C LINE OF SIGHT WITH CLOUD, SBAR = LENGTH OF PATH FROM SOURCE TO
C RECEIVER, S = LENGTH OF PATH IN CLOUD, AND TRANS = TRANSMITTANCE
C THROUGH CLOUD
L=XS-XR
M=YS-YR
N=ZS-ZR
SBAR=SQRT(L*L+M*M+N*N)
L=L/SBAR
M=M/SBAR
N=N/SBAR
DX=XS-XC
DY=YS-YC
DZ=ZS-ZC
ASQ=AE*AE
BSQ=BE*BE
CSQ=CE*CE
ABSQ=ASQ+BSQ
ACSQ=ASQ+CSQ
FCL020200
FCL0202030
FCL020206400
FCL020206500
FCL020206500
FCL020206700
FCL020206800
FCL020206900
FCL02021000
FCL02021100
FCL02021200
FCL02021300
FCL02021400
FCL02021500
FCL02021600
FCL02021700
FCL02021800
FCL02021900
FCL02022000
FCL02022100
FCL02022200
FCL02022300
FCL02022400
FCL02022500
FCL02022600
FCL02022700
FCL02022800
FCL02022900
FCL023000
FCL023100
FCL023200
FCL023300
FCL023400
FCL023500
FCL023600
FCL023700
FCL023800
FCL023900
FCL024000
FCL024100
FCL024200
FCL024300
FCL024400
FCL024500
FCL024600
FCL024700
FCL024800
FCL024900
FCL025000
FCL025100
FCL025200
FCL025300
FCL025400
FCL025500
FCL025600
FCL025700
FCL025800
FCL025900
FCL026000
FCL026100
FCL026200
FCL026300
FCL026400
FCL026500
FCL026600
FCL026700
FCL026800
FCL026900
FCL027000

```

```

BCSQ=BSQ*CSQ
ABCSQ=ASQ*BSQ*CSQ
A=BCSQ*L*L+ACSQ*M*M+ABSQ*N*N
B=2.0*(L*DX*BCSQ+M*DY*ACSQ+N*DZ*ABSQ)
C=BCSQ*DX*DX+ACSQ*DY*DY+ABSQ*DZ*DZ-ABCSQ
DISCRM=B*B-4.0*A*C
IF(DISCRM.GE.0.0) GO TO 10
WRITE(IQOUT,1200)
S=0.0
TRANS=1.0
LP=0.0
GO TO 90
10 SRTDSC=SQRT(DISCRM)
TWOA=2.0*A
VPLUS=(-B+SRTDSC)/TWOA
VMINUS=(-B-SRTDSC)/TWOA
XM=XS+VPLUS*L
YM=YS+VPLUS*M
ZN=ZS+VPLUS*N
XN=XS+VMINUS*L
YN=YS+VMINUS*M
ZN=ZS+VMINUS*N
S=SQRT((XM-XN)*(XM-XN)+(YM-YN)*(YM-YN)+(ZN-ZN)*(ZN-ZN))
IF(S.LT.1.E-4) S=0.0
15 TRANS=EXP(-KAPPA*S)
IF(TMPA.LT.-99.0) GO TO 20
TMPA >= -99.0 SPECIFIES THERMAL CALCULATION
RBAR=(AE+BE+CE)/3.0
CALL THRMCL(RBAR,W0,TMPA,TPMC,LAMBDA,KAPPA,TRANS,LP)
GO TO 90
ISW,NE,1 SPECIFIES NEW PARAMETERS IN COMMON /OPT/;
ASSOCIATED PRELIMINARY COMPUTATIONS YIELD EXTRATERRESTRIAL
IRRADIANCE E0, COORDINATES (L0,M0,N0) OF UNIT VECTOR
POINTING TO SUN, AND TATM = TRANSMITTANCE OF ATMOSPHERE
ABOVE CLOUD
20 IF(ISW.EQ.1) GO TO 30
CALL ILLUM(LAMBDA,LD,E0)
THTO=THETA0/57.2958
PH0=PHI0/57.2958
L0=SIN(THTO)*COS(PH0)
M0=SIN(THTO)*SIN(PH0)
N0=COS(THTO)
TATM=EXP(-TAUBAR/N0)
30 WRITE(IQOUT,1600) L,M,N,L0,M0,N0
IF(INDEXP.GT.-1) GO TO 40
INDEXP < 0 SPECIFIES MULTIPLE SCATTERING COMPUTATION;
COMPUTE TAU AND TAU0 = CLOUD OPTICAL DEPTH AND THICKNESS,
AND CALL MSCLD FOR VALUE OF LP = PATH RADIANCE
XG=(XM+XN)/2.0
YG=(YM+YN)/2.0
ZG=(ZM+ZN)/2.0
H=CE*SQRT(1.0-(XG-XC)*(XG-XC)/(AE*AE)-(YG-YC)*(YG-YC)/(BE*BE))
TAU0=2.0*H*KAPPA
H=ZC-ZG+H
TAU=KAPPA*H
CALL MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP)
GO TO 90
INDEXP > -1 SPECIFIES SINGLE SCATTERING COMPUTATION;
COMPUTE SCATTERING ANGLE CHI, PHASE FUNCTION VALUE P =
PFN(LAMBDA,CHI), GEOMETRICAL PARAMETERS ALPHA, BETA, GAMMA,
DELTA, AND EPS, RGRND = GROUND RADIANCE, AND CALL SCLD
TO COMPUTE PATH RADIANCE

```

FCL02710
FCL02720
FCL02730
FCL02740
FCL02750
FCL02760
FCL02770
FCL02780
FCL02790
FCL02800
FCL02810
FCL02820
FCL02830
FCL02840
FCL02850
FCL02860
FCL02870
FCL02880
FCL02890
FCL02900
FCL02910
FCL02920
FCL02930
FCL02940
FCL02950
FCL02960
FCL02970
FCL02980
FCL02990
FCL03000
FCL03010
FCL03020
FCL03030
FCL03040
FCL03050
FCL03060
FCL03070
FCL03080
FCL03090
FCL03100
FCL03110
FCL03120
FCL03130
FCL03140
FCL03150
FCL03160
FCL03170
FCL03180
FCL03190
FCL03200
FCL03210
FCL03220
FCL03230
FCL03240
FCL03250
FCL03260
FCL03270
FCL03280
FCL03290
FCL03300
FCL03310
FCL03320
FCL03330
FCL03340
FCL03350
FCL03360
FCL03370
FCL03380
FCL03390
FCL03400

```

40 THETA= ACOS(N)
PHI=0.0
IF(M.NE.0.0.OR.L.NE.0.0) PHI=ATAN2(M,L)
COSCHI=N*NO+SIN(THETA)*SIN(THTO)*COS(PHI-PHO)
CHI= ACOS(COSCHI)*57.2958
CALL PFNN(LAMBDA,CHI,INDEXP,PFN,ETA)
P=PFN
RGRND=E0*TATM*NO
IF(LB0.LT.1.E-10) LB0=RHO*RGRND
DX=XM-XC
DY=YM-YC
DZ=ZM-ZC
LOS=L0*L0
MOS=M0*M0
NOS=N0*N0
LOM=L0*M-M0*L
LON=L0*N-N0*L
NOM=N0*M-M0*N
MXLY=M0*DX-L0*DY
NXLZ=N0*DX-L0*DZ
MZNY=M0*DZ-N0*DY
DENOM=BCSQ*LOS+ACSQ*MOS+ABSQ*NOS
ALPHA=(BCSQ*L0+ACSQ*M0+ABSQ*N0)/DENOM
BETA=- (BCSQ*L0*DX+ACSQ*M0*DY+ABSQ*N0*DZ)/DENOM
GAMMA=-ABSQ*(CSQ*LOM*L0M+BSQ*LON*L0N+ASQ*NUM*N0M)/(DENOM*DENOM)
DELTA=CSQ*MXLY*L0M+BSQ*NXLZ*L0N+ASQ*MZNY*N0M
DELTA=-2.0*ABSQ*DELTA/(DENOM*DENOM)
EPS=BCSQ*LOS+ACSQ*MOS+ABSQ*NOS-CSQ*MXLY*MXLY-BSQ*NXLZ*NXLZ
1-ASQ*MZNY*MZNY
EPS=ABSQ*EPS/(DENOM*DENOM)
IF(EPS.GE.0.0) GO TO 50
WRITE(IOOUT,1300) EPS
EPS=0.0
50 IF(GAMMA.LT.0.0) GO TO 60
WRITE(IOOUT,1400) GAMMA
GO TO 200
60 CALL SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,W0,RGRND,P,LP)
CC
CC
CC
WRITE RESULTS
90 F=LP/(LB0*TRANS)
TC=1.0/(1.0+F)
WRITE(IOOUT,1500) S,TRANS,LP,TC
200 RETURN
1000 FORMAT(1H0,43X,45H-- RADIATIVE TRANSFER THROUGH FINITE CLOUD --/
1 1H0/45X,14H(XC,YC,ZC) = (,2(F8.4,1H),F8.4,12H) KILOMETERS/
2 45X,14H(AE,BE,CE) = (,2(F8.4,1H),F8.4,1H)/
3 45X,14H(XR,YR,ZR) = (,2(F8.4,1H),F8.4,1H)/
4 45X,14H(XS,YS,ZS) = (,2(F8.4,1H),F8.4,1H)
1100 FORMAT(1H0,45X,8HINDEXP =,I9,10X,8HETA =,F11.3/
1 45X,8HLAMBDA =,F9.3,8H(MU) ,2X,9HKAPPA =,1PE10.4,7H(KM-1)/
2 45X,8HOMEGA 0=,0PF9.3,10X,8HTAUBAR =,1X,1PE10.4/
3 45X,8HETHETA0 =,0PF9.1,10X,8HPHI0 =,F11.1,10H(DEGREES)/
4 45X,8HRHO =,0PF9.3,10X,8HLB0 =,F11.3,13H(W/M2-SR-MU)/
5 45X,8HTMPA =,F9.1,8H(DEC.C),2X,8HTMPC =,F11.1/
6 45X,8HLD =,I9)
1200 FORMAT(1H0,44X,43H**LINE-OF-SIGHT MISSES CLOUD. S SET TO 0.0)
1300 FORMAT(1H0,44X,4HEPS=,E10.4,24H LT 0.0. EPS SET TO 0.0)
1400 FORMAT(1H0,44X,6HGAMMA=,E10.4,29H GE 0.0. SKIP TO NEXT CASE.)
1500 FORMAT(1H0/1H0,37X,11HPATH LENGTH,3X,15HTRANSMITTANCE ,
1 13HPATH RADIANCE,4X,8HCONTRAST/
2 38X,11H(IN CLOUD) ,19X,12H(W/M2-SR-MU),15H TRANSMITTANCE/
3 36X,4(15H+-----+))
4 1H0,40X,F6.3,7X,1PE9.3,5X,1PE9.3,5X,1PE9.3)
1600 FORMAT(1H0,45X,28HUNIT SOURCE VECTOR L, M, N =,3(1X,F7.4)/
1 1H0,45X,23HSOLAR VECTOR L0,M0,N0 =,3(1X,F7.4))
END
FCL03410
FCL03420
FCL03430
FCL03440
FCL03450
FCL03460
FCL03470
FCL03480
FCL03490
FCL03500
FCL03510
FCL03520
FCL03530
FCL03540
FCL03550
FCL03560
FCL03570
FCL03580
FCL03590
FCL03600
FCL03610
FCL03620
FCL03630
FCL03640
FCL03650
FCL03660
FCL03670
FCL03680
FCL03690
FCL03700
FCL03710
FCL03720
FCL03730
FCL03740
FCL03750
FCL03760
FCL03770
FCL03780
FCL03790
FCL03800
FCL03810
FCL03820
FCL03830
FCL03840
FCL03850
FCL03860
FCL03870
FCL03880
FCL03890
FCL03900
FCL03910
FCL03920
FCL03930
FCL03940
FCL03950
FCL03960
FCL03970
FCL03980
FCL03990
FCL04000
FCL04010
FCL04020
FCL04030
FCL04040
FCL04050
FCL04060
FCL04070

```



```

SUBROUTINE THRMCL(RBAR,W0,TMPA,TMPC,LAMBDA,KAPPA,TRANS,LP)      THR00010
REAL LAMBDA,LP,LI,KAPPA                                       THR00020
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU THR00030
BB(X,T)=1.19106E8/(X**5*(EXP(1.4388E4/(X*T))-1.0))             THR00040
C*****                                                         THR00050
C SUBROUTINE PERFORMS THERMAL RADIATION CALCULATIONS FOR FINITE  THR00060
C CLOUD, RETURNING THE VALUE OF LP = PATH RADIANCE.            THR00070
C INPUTS ARE:                                                  THR00080
C C                                                             THR00090
C RBAR = AVERAGE HALF-LENGTH OF PATH THROUGH CLOUD           THR00100
C W0 = SINGLE SCATTERING ALBEDO                                THR00110
C TMPA = TEMPERATURE OF THE ATMOSPHERE                         THR00120
C TMPC = TEMPERATURE OF THE CLOUD                             THR00130
C LAMBDA = WAVELENGTH                                          THR00140
C KAPPA = VOLUME EXTINCTION COEFFICIENT                       THR00150
C TRANS = TRANSMITTANCE THROUGH CLOUD.                         THR00160
C C                                                             THR00170
C*****                                                         THR00180
C G=1.0-EXP(-KAPPA*RBAR)                                       THR00190
B=BB(LAMBDA,273.16+TMPC)                                       THR00200
LI=BB(LAMBDA,273.16+TMPA)                                       THR00210
WRITE(IOOUT,1000) B,LI                                          THR00220
LP=(1.0-TRANS)*((1.0-W0)*(1.0+G*W0)*B+W0*(1.0-G)*LI)          THR00230
1000 FORMAT(1H0,43,40H **THERMAL CALCULATION OF PATH RADIANCE/,45X, THR00240
122H BB(LAMBDA,TMPC) = ,1PE10.4,11H W/M2-SR-MU/,45X,           THR00250
122H BB(LAMBDA,TMPA) = ,1PE10.4,11H W/M2-SR-MU)                THR00260
RETURN                                                            THR00270
END                                                                THR00280

```



```

SUBROUTINE PFNN(LAMBDA, CHI, INDEXP, PFN, ETA)
REAL LAMBDA, LAMDA1, LAM1, LAM2, KAPPA
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTTU
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
DIMENSION PHANG(65), PF1(65), PF2(65)
DATA LAMDA1, IP1, CHI1 /-1.0, -2, -1.0/
C*****
C SUBROUTINE COMPUTES THE VALUE OF THE INDEXP-TH PHASE FUNCTION
C AT SCATTERING ANGLE CHI AND WAVELENGTH LAMBDA USING BILINEAR
C INTERPOLATION.
C*****
      ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
      IERR=0
      MAXID=12
      PFNDAT ONLY CONTAINS ONE PFN FOR THE VISIBLE (.55UM) AND
      THE NEAR IR (1.06UM); THEREFORE DO NOT INTERPOLATE.
      IWAVE=0
      IF (LAMBDA.LT.2.0) IWAVE=1
      PRELIMINARIES FOR FINDING ETA
      ETA1=0.0
      ETA2=0.0
      IDCK=0
      IF (INDEXP.LT.0) IDCK=1
      IF (INDEXP.LT.0) INDEXP=-INDEXP
      IF (INDEXP.EQ.IP1.AND.CHI.EQ.CHI1.AND.LAMBDA.EQ.LAMDA1)
10      GO TO 200
      IF (INDEXP.EQ.IP1.AND.LAMBDA.EQ.LAMDA1) GO TO 70
      REWIND IPHFUN
      NRD=0
      DO 20 I=1,66,11
      READ(IPHFUN,1000) (PHANG(I+J-1),J=1,11)
      DO 10 J=1,11
      NRD=NRD+1
      IF (PHANG(NRD).GE.999.9) GO TO 30
10      CONTINUE
20      CONTINUE
30      NA=NRD-1
      DO 35 I=1,NA
      PHANG(I)=COS(PHANG(I)*PIRAD)
40      CONTINUE
      END-OF-FILE CHECK
      IF (IERR.EQ.2) GO TO 195
      READ(IPHFUN,1100) IANG1, ID, LAM1, W0, KAPPA, BETA
      IF (LAM1.GE.12.00.AND.ID.EQ.MAXID) IERR=2
      READ(IPHFUN,1200) (PF1(I),I=1,NA)
      SUM=0.
      START RENORMALIZATION OF PHASE FUNCTION - ALSO SEE BELOW
      DO 45 J=2,NA
      SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF1(J)+PF1(J-1))/4.
45      DO 46 J=1,NA
      PF1(J)=PF1(J)/SUM
      ETA1=ETAINT(PF1,PHANG,NA)
      IF (ID.NE.INDEXP) GO TO 40
      IF (IWAVE.EQ.1.AND.LAMBDA.GT.LAM1) GO TO 40
      IF (IWAVE.EQ.1) GO TO 75
      IF (LAMBDA.LT.LAM1) GO TO 190
50      CONTINUE
      IF (IERR.EQ.2) GO TO 195
      READ(IPHFUN,1100) IANG2, ID, LAM2, W0, KAPPA, BETA
      IF (LAM2.GE.12.00.AND.ID.EQ.MAXID) IERR=2
      READ(IPHFUN,1200) (PF2(I),I=1,NA)
      SUM=0.
      DO 55 J=2,NA
      SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF2(J)+PF2(J-1))/4.
55      DO 56 J=1,NA
      PF2(J)=PF2(J)/SUM
      ETA2=ETAINT(PF2,PHANG,NA)
      THE PHASE FUNCTION(S) ARE NOW NORMALIZED TO: INTEGRAL OF
C*****
PFN00010
PFN00020
PFN00030
PFN00040
PFN00050
PFN00060
PFN00070
PFN00080
PFN00090
PFN00100
PFN00110
PFN00120
PFN00130
PFN00140
PFN00150
PFN00160
PFN00170
PFN00180
PFN00190
PFN00200
PFN00210
PFN00220
PFN00230
PFN00240
PFN00250
PFN00260
PFN00270
PFN00280
PFN00290
PFN00300
PFN00310
PFN00320
PFN00330
PFN00340
PFN00350
PFN00360
PFN00370
PFN00380
PFN00390
PFN00400
PFN00410
PFN00420
PFN00430
PFN00440
PFN00450
PFN00460
PFN00470
PFN00480
PFN00490
PFN00500
PFN00510
PFN00520
PFN00530
PFN00540

```

C	PHASE FUNCTION OVER ALL SOLID ANGLE DIVIDED BY 4 PI = 1.	PFN00550
	IF<ID.NE.INDEXP> GO TO 190	PFN00560
	IF<LAMBDA.LE.LAM2> GO TO 70	PFN00570
	LAM1=LAM2	PFN00580
	IANG1=IANG2	PFN00590
	DO 60 I=1,NA	PFN00600
	PF1(I)=PF2(I)	PFN00610
60	CONTINUE	PFN00620
	GO TO 50	PFN00630
70	CONTINUE	
	IF<IWAVE.EQ.1> GO TO 75	
	DLAM=(LAMBDA-LAM1)/(LAM2-LAM1)	PFN00640
75	CONTINUE	
	IF<CHI.LT.-1.E-3.OR.CHI.GT.180.001> GO TO 190	PFN00650
	DO 80 J=2,NA	PFN00670
	IF<CHI.LE.(ACOS(PHANG(J))/PIRAD)> GO TO 90	PFN00680
80	CONTINUE	PFN00690
	J=NA	PFN00700
90	J1=J-1	PFN00710
	DCHI=(COS<CHI*PIRAD>-PHANG(J1))/(PHANG(J)-PHANG(J1))	PFN00720
	IF<IWAVE.NE.1> GO TO 95	
	PFN=PF1(J1)+DCHI*(PF1(J)-PF1(J1))	
	GO TO 96	
95	PFN=PF1(J1)+DLAM*(PF2(J1)-PF1(J1))+DCHI*(PF1(J)-PF1(J1))	PFN00730
	+DLAM*DCHI*(PF2(J)+PF1(J1)-PF2(J1)-PF1(J))	PFN00740
96	LAMDA1=LAMBDA	PFN00760
	CHI1=CHI	PFN00770
	IF<IP1.NE.-2> WRITE<IOOUT,1500> CHI,PFN	
	IP1=INDEXP	
	ETA=ETA1+DLAM*(ETA2-ETA1)	
	GO TO 200	
190	WRITE<IOOUT,1300> ID,INDEXP,LAM1,LAMBDA,CHI	PFN00790
	STOP	PFN00800
195	WRITE<IOOUT,1600> IPHFUN	PFN00810
	STOP	PFN00820
200	CONTINUE	PFN00830
	IF<IDCK.EQ.1> INDEXP=-INDEXP	
	RETURN	PFN00840
1000	FORMAT<11(F6.2,1X)>	PFN00850
1100	FORMAT<2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X)>	PFN00860
1200	FORMAT<6(E12.6,1X)>	PFN00870
1300	FORMAT<33H0ERROR IN READING PHASE FUNCTION./	PFN00880
	127H ID,INDEXP,LAM1,LAMBDA,CHI=,2I3,3E13.7>	PFN00890
1500	FORMAT<1H0,23HSCATTERING ANGLE CHI = ,F8.2,7X,24H PHASE FN P<LAMBDA	PFN00900
	+A,CHI>=,E10.4>	PFN00910
1600	FORMAT<1X,32HATTEMPT TO READ PAST EOF ON UNIT,13,18H IN SUBROUTINE	PFN00920
	1 PFN//>	PFN00930
	END	PFN00940

```

SUBROUTINE MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP) MSC00010
REAL LP,LBAR,K MSC00020
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEG MSC00030
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUMSC00040
SINH(ARG)=.5*(EXP(ARG)-EXP(-ARG))
COSH(ARG)=.5*(EXP(ARG)+EXP(-ARG))
C***** MSC00050
SUBROUTINE COMPUTES LP = PATH RADIANCE DUE TO MULTIPLE SCATTERING MSC00060
IN A FINITE CLOUD. INPUTS ARE: MSC00070
TAU = CLOUD OPTICAL DEPTH MSC00080
TAU0 = CLOUD OPTICAL THICKNESS MSC00090
TRANS = TRANSMITTANCE ALONG LINE OF SIGHT MSC00100
TATM = TRANSMITTANCE OF ATMOSPHERE ABOVE CLOUD MSC00110
E0 = EXTRATERRESTRIAL IRRADIANCE MSC00120
W0 = SINGLE SCATTERING ALBEDO MSC00130
ETA = FORWARD SCATTERING PARAMETER MSC00140
RHO = BACKGROUND SURFACE REFLECTANCE. MSC00150
C***** MSC00160
IF(W0,LT,1.0) GO TO 10 MSC00170
W0=.999 MSC00180
WRITE(IOOUT,2000) MSC00190
10 C1=1.0-ETA*W0 MSC00200
C2=(1.0-ETA)*W0 MSC00210
K=SQRT((1.0-W0)*(1.0+W0-2.0*ETA*W0)) MSC00220
TOK=K*TAU MSC00230
TOTK=K*(TAU0-TAU) MSC00240
TK2=2.0*K*TAU MSC00250
GAMMA=E0*TATM/(C1*SINH(TOK)+K*COSH(TOK)) MSC00260
DELTA=RHO*K/((C1-RHO*C2)*SINH(2.0*TOK)+K*COSH(2.0*TOK)) MSC00270
WRITE(IOOUT,1000) MSC00280
EPLUS=C2*SINH(TOK)+DELTA*(C1*SINH(TK2)+K*COSH(TK2)) MSC00290
EMINUS=C1*SINH(TOK)+K*COSH(TOK)+DELTA*C2*SINH(TK2) MSC00300
LBAR=GAMMA*(EPLUS+EMINUS)/TWOPI MSC00310
LP=W0*LBAR*(1.0-TRANS) MSC00320
WRITE(IOOUT,100)TAU,TAU0,TATM MSC00330
100 FORMAT(1H0,45X,18H OPTICAL DEPTH = ,1PE10.4/,45X, MSC00340
131H OPTICAL THICKNESS OF CLOUD = ,1PE10.4/,45X, MSC00350
244H TRANSMITTANCE OF ATMOSPHERE ABOVE CLOUD = ,1PE10.4) MSC00360
RETURN MSC00370
1000 FORMAT(36H0 **RESULTS FOR MULTIPLE SCATTERING ) MSC00380
2000 FORMAT(47H0 **OMEGA 0 WAS 1.0, NOW SET TO 0.999 IN MSCLD) MSC00390
END MSC00400
MSC00410
MSC00420
MSC00430

```

```

C      FUNCTION ETAINT(PFN,PHANG,NA)
C      THIS FUNCTION WILL DETERMINE ETA, THE FORWARD SCATTERING
C      PARAMETER: ETA=.5*INTEGRAL PFN OVER THETA, WHERE THETA GOES
C      FROM ZERO TO PI/2.
COMMON /CONST/ PI,P12,PIRAD,TWOPI,TORRMB,CDEGK
DIMENSION PHANG(65),PFN(65)
NAM1=NA-1
ETA=0.
DO 1 I=1,NAM1
IF(PHANG(I+1).GT.0.) ETA=ETA+(PHANG(I)-PHANG(I+1))*
1 (PFN(I+1)+PFN(I))/4.
CONTINUE
ETAINT=ETA
RETURN
END

```

```

SUBROUTINE OVR0ST(LAMBDA,TRANS,IERR)
REAL LC,LG,LB0,KAPPA,LAMBDA,MU,LP
DIMENSION IALPH(7)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
COMMON /GEOMET/PTS(15),IGEOSW
DATA IZERO/0/
DATA IALPH/2HUP,2HSP,2HCL,2HBK,2HGR,2HTE,2HGD /
BB(X,T)=1.191062E8/(X**5*(EXP(1.4387864E4/(X*T))-1.0))

```

```

OVR00010
OVR00020
OVR00030
OVR00040
OVR00050
OVR00060
OVR00070
OVR00080
OVR00090
OVR00100

```

```

SUBROUTINE COMPUTES BEAM TRANSMITTANCE, PATH RADIANCE, AND
CONTRAST TRANSMITTANCE ALONG AN OPTICAL PATH UNDER AN OVERCAST
SKY. ORDER-INDEPENDENT INPUT CARDS ARE AS FOLLOWS:
(INDIVIDUAL RECORD FORMAT IS (A4,1X,5(E10.4,1X))

```

```

OVR00110
OVR00120
OVR00130
OVR00140
OVR00150
OVR00160
OVR00170
OVR00180
OVR00190
OVR00200
OVR00210
OVR00220
OVR00230
OVR00240
OVR00250
OVR00260

```

MNEMONIC	VARIABLES READ	DESCRIPTION
OPOS	XO,YO,ZO	OBSERVER POSITION
SPOS	XT,YT,ZT	SOURCE POSITION
CLDS	ZC,LC,KAPPA,ETA,WO	CLOUD PARAMETERS
BKGR	LB0	BACKGROUND RADIANCE
GRND	LG	GROUND RADIANCE
TEMP	TEMP	TEMPERATURE ALONG PATH
GO		END OF READ SENTINEL

```

THE VARIABLES ZC AND LC REFER TO THE OVERCAST SKY LAYER!
THE VARIABLES KAPPA, ETA, WO, TEMP, REFER TO THE ATMOSPHERE
BETWEEN THE OVERCAST SKY AND GROUND I.E. THE INTERVENING
ATMOSPHERIC PROPERTIES (GAS OR AEROSOL)

```

** NOTE : THE GO CARD MUST BE THE LAST RECORD READ.

```

OVR00270
OVR00280
OVR00290
OVR00300
OVR00310
OVR00320
OVR00330
OVR00340
OVR00350
OVR00360
OVR00370
OVR00380
OVR00390
OVR00400
OVR00410
OVR00420
OVR00430
OVR00440
OVR00450
OVR00460
OVR00470
OVR00480
OVR00490
OVR00500
OVR00510
OVR00520
OVR00530
OVR00540
OVR00550
OVR00560
OVR00570
OVR00580
OVR00590
OVR00600
OVR00610
OVR00620
OVR00630
OVR00640
OVR00650
OVR00660

```

THE FOLLOWING ENUMERATES THE VARIABLES LISTED ON THE ABOVE CARDS :

```

(XO,YO,ZO) = OBSERVER COORDINATES
(XT,YT,ZT) = SOURCE COORDINATES
ZC = HEIGHT OF CLOUD LAYER
LC = CLOUD RADIANCE
LG = GROUND RADIANCE
LB0 = BACKGROUND RADIANCE
KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1)
ETA = FORWARD SCATTERING PARAMETER
WO = SINGLE SCATTERING ALBEDO
TEMP = TEMPERATURE ALONG PATH (DEG. C)

```

LENGTH UNITS ARE KILOMETERS; RADIANCE UNITS ARE W/M2-SR-MU.
IF TEMP >= -99., THERMAL RADIATION IS CALCULATED; IF TEMP < -99.
SINGLY SCATTERED RADIANCE IS CALCULATED.

```

SUBROUTINE RETURNS:
TRANS = BEAM TRANSMITTANCE
TO CALLING PROGRAM.

```

DATA INITIALIZATION

```

IF(IZERO.NE.0) GO TO 477
XO=0.0
YO=0.0
ZO=0.0
XT=0.0
YT=0.0
ZT=0.0
ZC=0.0
LC=0.0
KAPPA=0.0
ETA =0.0
WO=0.0
LB0=0.0

```

```

OVR00500
OVR00510
OVR00520
OVR00530
OVR00540
OVR00550
OVR00560
OVR00570
OVR00580
OVR00590
OVR00600
OVR00610
OVR00620
OVR00630
OVR00640
OVR00650
OVR00660

```

	LG=0.0	OVR00670
	TEMP=0.0	OVR00680
	IZERO=1	OVR00690
477	CONTINUE	OVR00700
	DO 360 K=1,7	OVR00710
	READ(10IN,334)IA,IA2,R1,R2,R3,R4,R5	OVR0
334	FORMAT(2A2,1X,5(E10.4,1X))	OVR0073
	DO 333 I=1,8	OVR00740
	IF(IA.NE.IALPH(I)) GO TO 333	OVR00750
	IND=I	OVR00760
	IF(IND.EQ.7) GO TO 361	OVR00770
333	CONTINUE	OVR00780
	IF(IND.EQ.8) GO TO 355	OVR00790
	IF(K.EQ.7.AND.IND.NE.7) GO TO 358	OVR00800
	GO TO (341,342,343,344,345,346),IND	OVR00810
341	XO=R1	OVR00820
	YO=R2	OVR00830
	ZO=R3	OVR00840
	GO TO 360	OVR00850
342	XT=R1	OVR00860
	YT=R2	OVR00870
	ZT=R3	OVR00880
	GO TO 360	OVR00890
343	ZC=R1	OVR00900
	LC=R2	OVR00910
	KAPPA=R3	OVR00920
	ETA=R4	OVR00930
	WO=R5	OVR00940
	GO TO 360	OVR00950
344	LB0=R1	OVR00960
	GO TO 360	OVR00970
345	LG=R1	OVR00980
	GO TO 360	OVR00990
346	TEMP=R1	OVR01000
	GO TO 360	OVR01010
355	WRITE(IOOUT,357)	OVR01020
357	FORMAT(1H0,25X,44H***OVRCSST ERROR*** INPUT CARD DETECTED WHICH,	OVR01030
	+ 36H DOES NOT MATCH CORRECT INPUT FORMAT,/))	OVR01040
	GO TO 360	OVR01050
358	WRITE(IOOUT,359)	OVR01060
359	FORMAT(1H0,34X,45H***OVRCSST ERROR*** TOO MANY INPUT CARDS OR GO,	OVR01070
	+ 16H SENTINEL ABSENT,/))	OVR01080
	IERR=1	OVR01090
	GO TO 200	OVR01100
360	CONTINUE	OVR01110
361	CONTINUE	OVR01120
	IF(IGEOSM.NE.1) GO TO 222	OVR01130
	XO=PTS(4)	OVR01140
	YO=PTS(5)	OVR01150
	ZO=PTS(6)	OVR01160
	XT=PTS(1)	OVR01170
	YT=PTS(2)	OVR01180
	ZT=PTS(3)	OVR01190
222	CONTINUE	OVR01200
	ECHO INPUT	OVR01210
	WRITE(IOOUT,1000) XO,XT,YO,YT,ZO,ZT,ZC,LC,LAMBDA,LG,	OVR01240
	1TEMP,LB0,KAPPA,WO,ETA	OVR01250
	IF(TEMP.LT.-99.0) GO TO 4	OVR01260
	BBTEMP=BB(LAMBDA,273.16+TEMP)	OVR01270
	WRITE(IOOUT,1800) BBTEMP	OVR01280
	BTE=(WO-1.0)*BBTEMP	OVR01290
	GO TO 8	OVR01300
4	WRITE(IOOUT,1700)	OVR01310
8	F=2.0*(1.0-ETA)	OVR01320
	ZLEN=ZT-ZO	OVR01330
	S=SQRT((XT-XO)**2+(YT-YO)**2+ZLEN**2)	OVR01340
	MU=ABS(ZLEN)/S	OVR01350
	TO=KAPPA*ZO	OVR01360


```

TT=KAPPA*ZT
SO=KAPPA*(ZC-ZO)
ST=KAPPA*(ZL-ZT)
TRANS=EXP(-KAPPA*S)
CONST=W0*ETA /2.0
IF(ZLEN) 10,40,70
OVR01370
OVR01380
OVR01390
OVR01400
OVR01410
OVR01420
OVR01430
OVR01440
OVR01450
OVR01460
OVR01470
OVR01480
OVR01490
OVR01500
OVR01510
OVR01520
OVR01530
OVR01540
OVR01550
OVR01560
OVR01570
OVR01580
OVR01590
OVR01600
OVR01610
OVR01620
OVR01630
OVR01640
OVR01650
OVR01660
OVR01670
OVR01680
OVR01690
OVR01700
OVR01710
OVR01720
OVR01730
OVR01740
OVR01750
OVR01760
OVR01770
OVR01780
OVR01790
OVR01800
OVR01810
OVR01820
OVR01830
OVR01840
OVR01850
OVR01860
OVR01870
OVR01880
OVR01890
OVR01900
OVR01910
OVR01920
OVR01930
OVR01940
OVR01950
OVR01960
OVR01970
OVR01980
OVR01990
OVR02000
OVR02010
OVR02020
OVR02030
OVR02040
OVR02050
OVR02060

CCC
HEIGHT OF OBSERVER > HEIGHT OF SOURCE
10 G21=G2(TO,TT,MU)
G22=G2(SO,ST,-MU)
T1=EXP(-TO/MU)
T2=EXP(-SO/MU)
F1=W0*F*KAPPA*S*T1/2.0
XINT1=CONST*T1*G21
XINT2=CONST*T2*G22
IF(TEMP,GE,-99.0) GO TO 20
LP=LG*(F1+XINT1)+LC*XINT2
GO TO 100
20 LP=(LG+BTE)*(F1+XINT1)+(LC+BTE)*XINT2+W0*BTE*(TRANS-1.0)
GO TO 100
CCC
HEIGHT OF OBSERVER = HEIGHT OF SOURCE
40 E21=1.0
E22=1.0
IF(TO,NE,0.0) E21=EXP(-TO)-TO*E1(TO)
IF(SO,NE,0.0) E22=EXP(-SO)-SO*E1(SO)
IF(TEMP,GE,-99.0) GO TO 50
LP=CONST*(1.0-TRANS)*(LG*E21+LC*E22)
GO TO 100
50 LP=CONST*(E21*(LG+BTE)+E22*(LC+BTE))
LP=(LP-W0*BTE)*(1.0-TRANS)
GO TO 100
CCC
HEIGHT OF OBSERVER < HEIGHT OF SOURCE
70 G21=G2(TO,TT,-MU)
G22=G2(SO,ST,MU)
T1=EXP(-SO/MU)
T2=EXP(-TO/MU)
F1=W0*F*KAPPA*S*T1/2.0
XINT1=CONST*T2*G21
XINT2=CONST*T1*G22
IF(TEMP,GE,-99.0) GO TO 80
LP=LC*(F1+XINT2)+LG*XINT1
GO TO 100
80 LP=(LC+BTE)*(F1+XINT2)+(LG+BTE)*XINT1+W0*BTE*(TRANS-1.0)
GO TO 100
CCC
WRITE RESULTS
100 TC=1.0/(1.0+LP/(LB0*TRANS))
WRITE(1000,2000) S,TRANS,LP,TC
200 RETURN
1000 FORMAT(1H0/1H0,55X,34H-- RADIATION UNDER OVERCAST SKY --/
1 1H0,43X,8HX0 =,5X,F6.3,5H (KM),8X,5HXT =,5X,F6.3,5H (KM)/
2 44X,8HY0 =,5X,F6.3,13X,5HYT =,5X,F6.3/
3 44X,8HZ0 =,5X,F6.3,13X,5HZT =,5X,F6.3/
4 44X,8HLC =,5X,F6.3,13X,5HLC =,1X,1PE10.4,13H (W/M2-SR-MU)/
5 44X,8HLAMBDA =,5X,OPF6.3,5H (MU),8X,5HLG =,1X,1PE10.4/
6 44X,8HTEMP =,3X,OPF6.1,10H (DEG.C),5X,5HLB0 =,1X,1PE10.4/
7 44X,8HKAPPA =,1X,E10.4,7H (KM-1),6X,5HW0 =,3X,OPF6.3/
8 44X,8HETA =,6X,OPF5.3)
2000 FORMAT(1H0,39X,4HPATH,7X,13HTRANSMITTANCE,2X,13HPATH RADIANCE,
1 4X,8HCONTRAST/
2 38X,11HLENGTH (KM),19X,12H(W/M2-SR-MU),15H TRANSMITTANCE/
3 36X,4(15H+-----+))
4 1H0,39X,F6.3,9X,F7.5,7X,1PE9.3,7X,OPF7.5)
1700 FORMAT(1H0,50X,31H**RESULTS FOR SINGLE SCATTERING)
1800 FORMAT(1H0,46X,38H**THERMAL CALCULATION OF PATH RADIANCE/

```

1 1H0,50X,8HBBTEMP= ,1PE10.4,11H W/M2-SR-MU>
END

OVR02070
OVR02080

```

FUNCTION G2(TAU1,TAU2,MU)
*****
SUBROUTINE COMPUTES IN CLOSED FORM INTEGRALS OF THE FUNCTION
EXP(TAU/MU)*E2(TAU)
WHERE E2 IS THE SECOND EXPONENTIAL INTEGRAL, FOR DETAILS SEE
KOURGANOFF, 'BASIC METHODS IN TRANSFER PROBLEMS', APPENDIX I
(PAGES 256-257 OF FIRST EDITION, 1952, OXFORD UNIVERSITY PRESS.)
*****
REAL MU
DATA GAMMA/,5772156649/
IF(MU.LT..9999) GO TO 50
MU EQ 1.0
IF(TAU1.NE.0.0) GO TO 10
G2=(TAU2-1.0)*EXP(TAU2)*E1(TAU2)-GAMMA-ALOG(TAU2)
GO TO 100
10 IF(TAU2.NE.0.0) GO TO 20
G2=GAMMA+ALOG(TAU1)+EXP(TAU1)*(1.0-TAU1)*E1(TAU1)
GO TO 100
20 G2=EXP(TAU1)*(1.0-TAU1)*E1(TAU1)-EXP(TAU2)*(1.0-TAU2)*E1(TAU2)
G2=G2+(ALOG(TAU1)-ALOG(TAU2))
GO TO 100
MU NE 1.0
50 RM=1.0-1.0/MU
IF(TAU1.NE.0.0) GO TO 60
G2=EXP(TAU2/MU)*(TAU2-MU)*E1(TAU2)
G2=G2+MU*E1(TAU2*RM)
G2=G2+1.0+MU*ALOG(ABS(RM))-EXP(-RM*TAU2)
GO TO 100
60 IF(TAU2.NE.0.0) GO TO 70
G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1)
G2=G2-MU*E1(TAU1*RM)
G2=G2+EXP(-RM*TAU1)-MU*ALOG(ABS(RM))-1.0
GO TO 100
70 G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1)
G2=G2-EXP(TAU2/MU)*(MU-TAU2)*E1(TAU2)
G2=G2+(EXP(-TAU1*RM)-EXP(-TAU2*RM))
G2=G2+MU*(E1(TAU2*RM)-E1(TAU1*RM))
100 RETURN
END

```

```

FUG00010
FUG00020
FUG00030
FUG00040
FUG00050
FUG00060
FUG00070
FUG00080
FUG00090
FUG00100
FUG00110
FUG00120
FUG00130
FUG00140
FUG00150
FUG00160
FUG00170
FUG00180
FUG00190
FUG00200
FUG00210
FUG00220
FUG00230
FUG00240
FUG00250
FUG00260
FUG00270
FUG00280
FUG00290
FUG00300
FUG00310
FUG00320
FUG00330
FUG00340
FUG00350
FUG00360
FUG00370
FUG00380
FUG00390
FUG00400
FUG00410
FUG00420
FUG00430
FUG00440
FUG00450
FUG00460
FUG00470

```

```

FUNCTION E1(X) FUE00010
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NOIRTU,NCLIMT,KSTOR,NPLOTUFUE00020
***** FUE00030
FUNCTION COMPUTES THE VALUE OF THE FIRST EXPONENTIAL INTEGRAL FUE00040
-E1(-X) WHERE -174 < X < 170. FOR X OUTSIDE THESE BOUNDS, AN FUE00050
OVERFLOW OR UNDERFLOW MIGHT OCCUR, SO PROGRAM EXECUTION IS FUE00060
HALTED. POLYNOMIAL AND RATIONAL FUNCTION APPROXIMATIONS ARE FUE00070
ADAPTED FROM THE IBM SCIENTIFIC SUBROUTINE PACKAGE, SUBROUTINE FUE00080
EXPI. FUE00090
***** FUE00100
DATA GAMMA/.5772156649/ FUE00110
DATA C1,C2,C3,C4,C5,C6,C7,C8/674.567029,57.411833,6.05529232, FUE00120
11699.06552,841.654932,49.3133893,8.01957683,.99979204/ FUE00130
DATA D1,D2,D3,D4,D5,D6,D7,D8,D9/248.6697,224.4234,32.43665, FUE00140
13.061037,.05176245,180.7837,22.63818,38.93944,3.995161/ FUE00150
DATA F1,F2,F3,F4,F5,F6,F7,F8,F9/9.999999E-1,2.500001E-1, FUE00160
15.555682E-2,1.041576E-2,1.664156E-3,2.335379E-4,2.928433E-5, FUE00170
21.766345E-6,7.122452E-7/ FUE00180
DATA G1,G2,G3,G4,G5,G6,G7,G8/.2677737343,8.6347608925, FUE00190
118.059016973,8.5733287401,3.9584969228,21.0996530827, FUE00200
225.6329561486,9.5733223454/ FUE00210
IF(X.GT.-174.0.AND.X.LE.170.0) GO TO 10 FUE00220
WRITE(IOOUT,1000) FUE00230
STOP FUE00240
10 IF(X.GT.-9.0) GO TO 20 FUE00250
E1=1.0-(C1+C2*X-C3*X*X-X*X*X)/(C4+C5*X+C6*X*X-C7*X*X*X-C8*X*X*X*X) FUE00260
E1=E1*EXP(-X)/X FUE00270
GO TO 100 FUE00280
20 IF(X.GT.-3.0) GO TO 30 FUE00290
E1=D1+D2*X+D3*X*X+D4*X*X*X+D5*X*X*X*X FUE00300
E1=(1.0-E1/(D6+D7*X+D8*X*X+D9*X*X*X+X*X*X*X))*EXP(-X)/X FUE00310
GO TO 100 FUE00320
30 IF(X.GT.1.0) GO TO 40 FUE00330
E1=F1-X*(F2-X*(F3-X*(F4-X*(F5-X*(F6-X*(F7-X*(F8-X*F9)))))) FUE00340
E1=X*E1-GAMMA-ALOG(ABS(X)) FUE00350
GO TO 100 FUE00360
40 E1=(G1+X*(G2+X*(G3+X*(G4+X))))/(G5+X*(G6+X*(G7+X*(G8+X))) FUE00370
E1=E1*EXP(-X)/X FUE00380
100 RETURN FUE00390
1000 FORMAT(4H0X =,3X,E10.4,39H OUT-OF-RANGE FOR E1. EXECUTION HALTED.) FUE00400
END FUE00410
FUE00420
FUE00430

```

```

SUBROUTINE GRNADE(WAVE1,ICLMAT,TRANS,IERR)
C/*****/GRN00010
C/* SUBROUTINE GRNADE *GRN00020
C/* MAIN GRNADE MODULE *GRN00030
C/* EOSAEL80 *GRN00040
C/*****/GRN00050
DIMENSION XA(3),XTRAN(7) GRN00060
COMMON /MOS/ XDTA(1000),CDTA(1000),CL(1000) GRN00070
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK GRN00080
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU GRN00090
COMMON/CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDMAT,CLDHYT,FOGPRB, GRN00100
* WNDVEL,WNDDIR,IPASCT GRN00110
COMMON/MECH0/XM,YM,ZM,XO,YO,ZO,XT,YT,ZT,ISTO,IETO,IDTO,XN,FW, GRN00120
*TBURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG, GRN00130
*DLEN,WPOWR,EXTC(8),XMIS(8),XNORTH GRN00140
COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP GRN00150
COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH GRN00160
COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM GRN00170
C*****/GRN00180
C** THE FOLLOWING VARIABLES SUPPLIED BY THE USER. GRN00190
C** FIELD DATA GRN00200
C** XNORTH FIELD COORDINATES FROM NORTH (DEGREES) GRN00210
C** HEAD GRENADE TANK HEADING CLOCKWISE FROM NORTH (DEGREES) GRN00220
C** RNG DISTANCE OF GRENADES FROM TANK (METERS) GRN00230
C** DLEN GRENADE SPACING (PERPENDICULAR TO HEADING) GRN00240
C** XO,YO,ZO COORDINATES OF OBSERVER (METERS) GRN00250
C** XM,YM,ZM COORDINATES OF TANK (METERS) GRN00260
C** XT,YT,ZT TARGET COORDINATES (METERS) GRN00270
C** METEOROLOGICAL DATA GRN00280
C** WINDP WIND PROFILE EXPONENT GRN00290
C** HM HEIGHT OF INVERSION LAYER GRN00300
C** WD WIND DIRECTION FROM NORTH (DEGREES) GRN00310
C** WS WIND SPEED (METERS PER SECOND) GRN00320
C** RH RELATIVE HUMIDITY GRN00330
C** ICAT PASQUILL CATEGORY GRN00340
C** YF SMOKE YIELD FACTOR GRN00350
C** MUNITION DATA GRN00360
C** EFF CLOUD-MAKING EFFICIENCY OF MUNITION GRN00370
C** QMUN TOTAL MASS OF SMOKE AGENT (GRAMS) GRN00380
C** DETECTOR DATA GRN00390
C** WAVE1 WAVELENGTH OF INTEREST (MICRONS) GRN00400
C** DIFFUSION PARAM METERS GRN00410
C** SIGZ REFERENCE SIGMA (METERS) GRN00420
C** XREF REFERENCE DISTANCE (METERS) GRN00430
C** ZDIFF VERTICAL DIFFUSION CONSTANT GRN00440
C** YDIFF CROSSWIND DIFFUSION CONSTANT GRN00450
C** HK TERRAIN SCAVENGING COEFFICIENT GRN00460
C** VS PARTICLE SETTLING VELOCITY (CM/SEC) GRN00470
C** RC TERRAIN REFLECTION COEFFICIENT GRN00480
C*****/GRN00490
C** DEFINITIONS OF OTHER VARIABLES** GRN00500
C** IXMAX NUMBER OF POINTS ALONG LINE-OF-SIGHT FOR CL COMPUTATIO GRN00510
C** NBPT NUMBER OF GRENADE LINES (NBPT=1) GRN00520
C** NTARG NUMBER OF TARGETS (NTARG=1) GRN00530
C** TBURST MUNITION DETONATION TIME GRN00540
C*****/GRN00550
C** ERROR CODES AND OPTION CODES: GRN00560
C** IWRIT=1 DEPRESSES RAW DATA PRINTOUT GRN00570
C** IFLAG=4 INVALID DATA CARD (BUT IGNORED) GRN00580
C** IFLAG=3 OVER 11 DATA CARDS ENTERED BEFORE GO GRN00590
C** REMAINDER IGNORED GRN00600
C** IFLAG=2 NORMAL READ TERMINATION GRN00610
C** IFLAG=1 WAVELENGTH OF INTEREST NOT IN DEFINED BANDS GRN00620
C** TRANS SET TO 1.0 GRN00630
C*****SET DEFAULTS***** GRN00640
TBURST=0.0 GRN00650
CALL GOGET(WAVE1,KWAVE) GRN00660
C*****READ DATA AND WRITE HEADING***** GRN00670
WRITE(IOOUT,8000) GRN00680
8000 FORMAT(1H0,20X,40(2H**),/,21X,1H*,34X,14HPROGRAM GRNADE,30X,1H*,/, GRN00690
GRN00700

```

```

*2IX,1H*,37X,8HEOSAEL80,33X,1H*,/,21X,40(2H**))
IWRIT=1
IFLAG=0
6 CALL DATRD(IWRIT,IFLAG)
IERR=IFLAG
IF(IFLAG.EQ.4)GO TO 9999
C*****CALCULATE INTEGRATION INCREMENT
XX=(XT-XO)**2+(YT-YO)**2+(ZT-ZO)**2
DLOS=SQRT(XX)
IXMAX=IFIX(DLOS)
IF(IXMAX.GT.1000)IXMAX=1000
C*****DEFAULT TO CLIMATE DATA OPTION
IF(ICLMAT.NE.1)GO TO 12
RHA=RH
UW=WINDVEL
WD=WINDDIR
ICAT=IFASCT
12 CONTINUE
IF(DLEN.EQ.0.0)DLEN=10.0
IF(BRATE.EQ.0.0)BRATE=(1.0/14.3)
QLENTH=XN*DLEN
BREXP=BRATE
QMUN=XN*FW
EMUN=QMUN*YF*EFF/100.0
WS=UW
IF(XMIS(1).LE.0.0)GO TO 4
XREFZ=100.0
SIGZR=XMIS(1)
ZDIFF=XMIS(2)
YDIFF=XMIS(3)
HM=XMIS(4)
HK=XMIS(5)
RC=XMIS(6)
VS=XMIS(7)
GO TO 5
4 CALL PARMS(ICAT)
5 CONTINUE
C*****REDEFINE WIND PROFILE EXPONENT IF READ IN POSITIVE
IF(WPOWR.GE.0.0)WINDP=WPOWR
IF(WS.LE.0.0) WS=0.1
U2=WS
C-----
C LOCATE GENERATING LINE
C-----
THETA=(HEAD-WD)*PIRAD
C*** LOCATE CENTER OF GENERATING LINE.
YHEAD=(XNORTH+90.0-HEAD)*PIRAD
XBURST=RNC*SIN(YHEAD)+XM
YBURST=RNC*COS(YHEAD)+YM
ZBURST=ZM
C-----
C TRANSFORM TO OBSERVER COORDINATES
C-----
C*** TRANSFORM TO ORIGIN UNDER OBSERVER AND X-AXIS UNDER TARGET.
ANGLR=ATAN2(YT-YO,XT-XO)
XI(1)=(XBURST-XO)*COS(ANGLR)+(YBURST-YO)*SIN(ANGLR)
YI(1)=(YBURST-YO)*COS(ANGLR)-(XBURST-XO)*SIN(ANGLR)
ZI(1) = ZBURST
TTI(1) = TBURST
C*** GET THE WIND DIRECTION ANGLE WITH THE NEW X-AXIS.
ANGLD = ANGLR*180./PI
WX=-(WD+XNORTH+ANGLD)
WX=AMOD(WX,360.0)
WDA = WX*PI/180.
C-----
C WRITE INPUT DATA AND HEADINGS
C-----
WRITE(IOOUT,8001)
WRITE(IOOUT,1000)
WRITE(IOOUT,1001)WS,WD,ICAT,RHA,XNORTH

```

```

GRN00710
GRN00720
GRN00730
GRN00740
GRN00750
GRN00760
GRN00770
GRN00780
GRN00790
GRN00800
GRN00810
GRN00820
GRN00830
GRN00840
GRN00850
GRN00860
GRN00870
GRN00880
GRN00890
GRN00900
GRN00910
GRN00920
GRN00930
GRN00940
GRN00950
GRN00960
GRN00970
GRN00980
GRN00990
GRN01000
GRN01010
GRN01020
GRN01030
GRN01040
GRN01050
GRN01060
GRN01070
GRN01080
GRN01090
GRN01100
GRN01110
GRN01120
GRN01130
GRN01140
GRN01150
GRN01160
GRN01170
GRN01180
GRN01190
GRN01200
GRN01210
GRN01220
GRN01230
GRN01240
GRN01250
GRN01260
GRN01270
GRN01280
GRN01290
GRN01300
GRN01310
GRN01320
GRN01330
GRN01340
GRN01350
GRN01360
GRN01370
GRN01380
GRN01390
GRN01400

```

```

WRITE(IOOUT,998)
WRITE(IOOUT,995)
WRITE(IOOUT,997)X0,EXTC(1),Y0,EXTC(2),Z0,EXTC(3),
*XT,EXTC(4),YT,EXTC(5),ZT,EXTC(6),EXTC(7)
WRITE(IOOUT,1002)
WRITE(IOOUT,1003)XM,SIGZR,YM,XREFZ,ZM,MM
WRITE(IOOUT,1004)HEAD,HK,RNG,RC,XN,VS
WRITE(IOOUT,1005)QMUH,WINDP,QLENTH,ZDIFF,BREXP,YDIFF
WRITE(IOOUT,1006)EFF,YF
8001 FORMAT(//,21X,36H*****INPUT*****ALL LENGTHS IN METERS,/,
*21X,38(2H--))
1000 FORMAT(21X,15HMETEOROLOGICAL;)
1001 FORMAT(24X,10HWIND SPEED,10X,F6.1,1X,3HM/S,/,
* 24X,14HWIND DIRECTION,6X,F6.1,1X,3HDEG,/,
* 24X,17HPASQUILL CATEGORY,3X,I3,/,
* 24X,17HRELATIVE HUMIDITY,3X,F6.1,1X,1HX,/,
* 24X,21HNOTE: X AXIS HEADING: 1X,F6.1,1X,3HDEG,1X,
* 28HCLOCKWISE FROM NORTH (DCWFN))
1002 FORMAT(20X,19HTANK/MUNITION DATA: 16X,21HDIFFUSION PARAMETERS;)
1003 FORMAT(24X,7HX(TANK),12X,F6.1,10X,10HSIGZ(XREF),21X,F6.1,/,
* 24X,7HY(TANK),12X,F6.1,10X,4HXREF,27X,F6.1,/,
* 24X,7HZ(TANK),12X,F6.1,10X,17HMXING HEIGHT(HM),14X,F6.1)
1004 FORMAT(24X,14HHEADING(DCWFN),5X,F6.1,10X,20HSCAVENGING COEFF(HK),
*11X,F6.3,/,24X,5HRANGE,14X,F6.1,10X,20HREFLECTION COEFF(RC),11X,
*F6.3,/,24X,10HNO GRNADES,9X,F6.1,10X,21HSETTLING VELOCITY(VS),
*5X,F6.3,1X,4HCM/S)
1005 FORMAT(24X,14HSMOKE MASS(GM),5X,F6.1,10X,29HVERTICAL WIND EXPONENT
*(WPOWR),2X,F6.3,/,24X,11HLINE LENGTH,8X,F6.1,10X,
*29HVERTICAL DIFF CONSTANT(ZDIFF),2X,F6.3,/,
*24X,13HBURN CONSTANT,2X,F6.3,1X,3H1/S,10X,
*30HCROSSWIND DIFF CONSTANT(YDIFF),1X,F6.3)
1006 FORMAT(24X,10HEFFICIENCY,9X,F6.1,10X,12HYIELD FACTOR,17X,F6.1)
998 FORMAT(21X,16HOBSERVER/TARGET: 8X,24HEXTINCTION COEFFICIENTS;)
995 FORMAT(49X,7HMICRONS,3X,7HM**2/GM)
997 FORMAT(24X,6HX(OBS),3X,F6.1,10X,7H0.4-0.7,3X,F6.3,/,
* 24X,6HY(OBS),3X,F6.1,10X,7H0.7-1.2,3X,F6.3,/,
* 24X,6HZ(OBS),3X,F6.1,10X,7H1.06,3X,F6.3,/,
* 24X,6HX(TAR),3X,F6.1,10X,7H3.0-5.0,3X,F6.3,/,
* 24X,6HY(TAR),3X,F6.1,10X,7H8.0-12,3X,F6.3,/,
* 24X,6HZ(TAR),3X,F6.1,10X,7H10.6,3X,F6.3,/,
* 49X,4H94.0,1X,3HGHZ,2X,F6.3)
WRITE(IOOUT,996)
996 FORMAT(1H1,21X,16H*****OUTPUT*****/,21X,38(2H--))
WRITE(IOOUT,3000)
3000 FORMAT(24X,4HTIME,6X,2HCL,23X,12HTRANSMISSION,/,24X,5H(SEC),2X,
*9H(GM/M**2),2X,7H0.4-0.7,1X,7H0.7-1.2,3X,4H1.06,2X,7H3.0-5.0,1X,
*7H8.0-12.,2X,4H10.6,4X,5H94GHZ)
C-----
C BEGIN CL CALCULATIONS
C-----
DO 400 IT=ISTO,IETO,IDTO
ITT=IT
C * SET UP LOOP ON SPACIAL DISTRIBUTION
XC = 0.0
YC = 0.0
ZC = Z0
DELX=SQRT((XT-X0)**2+(YT-Y0)**2)/IXMAX
DELZ = (ZT-Z0)/IXMAX
C** FOR EACH TIME GET THE CONCENTRATION AT IXMAX POINTS ALONG LINE-OF-SIGHT
DO 300 IX=1,IXMAX
XC = XC + DELX
ZC = ZC + DELZ
XA(1) = XC
XA(2) = YC
XA(3) = ZC
T = FLOAT(IT)
UBXB=U2
CALL CONCN(XA,T,C)
XDTA(IX)=XC
CDTA(IX)=C

```

```

GRN01410
GRN01420
GRN01430
GRN01440
GRN01450
GRN01460
GRN01470
GRN01480
GRN01490
GRN01500
GRN01510
GRN01520
GRN01530
GRN01540
GRN01550
GRN01560
GRN01570
GRN01580
GRN01590
GRN01600
GRN01610
GRN01620
GRN01630
GRN01640
GRN01650
GRN01660
GRN01670
GRN01680
GRN01690
GRN01700
GRN01710
GRN01720
GRN01730
GRN01740
GRN01750
GRN01760
GRN01770
GRN01780
GRN01790
GRN01800
GRN01810
GRN01820
GRN01830
GRN01840
GRN01850
GRN01860
GRN01870
GRN01880
GRN01890
GRN01900
GRN01910
GRN01920
GRN01930
GRN01940
GRN01950
GRN01960
GRN01970
GRN01980
GRN01990
GRN02000
GRN02010
GRN02020
GRN02030
GRN02040
GRN02050
GRN02060
GRN02070
GRN02080
GRN02090
GRN02100

```

300 CONTINUE	GRN02110
C*** INTEGRATE ALONG LINE-OF-SIGHT TO GET CL.	GRN02120
CALL SUMA (XDTA,CDTA, CL ,IXMAX)	GRN02130
C-----	GRN02140
C CALCULATE TRANSMITTANCE	GRN02150
C-----	GRN02160
DO 500 J=1,7	GRN02170
XTRAN(J)=EXP(-1.0*EXTC(J)*CL(IXMAX))	GRN02180
500 CONTINUE	GRN02190
WRITE(I0OUT,2000)T,CL(IXMAX),(XTRAN(J),J=1,7)	GRN02200
2000 FORMAT(23X,F5.1,2X,F8.3,2X,7(1X,F7.3))	GRN02210
400 CONTINUE	GRN02220
C*****	GRN02230
C*****SET TRANSMISSION FOR RETURN TO EOSAEL	GRN02240
C*****	GRN02250
IF(KWAVE.GT.0)GO TO 3	GRN02260
TRANS=1.0	GRN02270
IFLAG=1	GRN02280
3 TRANS=XTRAN(KWAVE)	GRN02290
GO TO 6	GRN02300
9999 WRITE(I0OUT,9000)	GRN02310
9000 FORMAT(21X,28H*****PROGRAM GRNAE END*****)	GRN02320
RETURN	GRN02330
END	GRN02340


```

SUBROUTINE CONCNC(XA,T,C2)
C/*****
C/          SUBROUTINE CONCNC
C/          GRNAD MODULE
C/          EOSAEL80
C/*****
C/ PURPOSE:
C/          CALCULATES CONCENTRATION AT A SPECIFIED POSITON AND TIME.
C/ USAGE:
C/          THE CONCENTRATION IS USED WITH THE EXTINCTION COEFFICIENT
C/          TO COMPUTE TRANSMITTANCE.
C/ DESCRIPTION OF PARAMETERS:
C/          XA      - POSITION IN METERS, INPUT.
C/          T        - TIME IN SECONDS, INPUT.
C/          C2      - CONCENTRATION, OUTPUT.
C/ SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
C/          LOCAT   UMEAN
C/ COMMON BLOCK STATEMENTS REQUIRED:
C/          MECH1   MECH2   MECH3
C/ REMARKS:
C/          CONCNC COMPUTES FIVE TERMS AND MULTIPLIES THEM TO
C/          GET CONCENTRATION.
C/*****
      DIMENSION XA(3)
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
C-----
C          CONCENTRATION=TERM1*TERM2*TERM3*TERM4*TERMS
C-----
      C2 = 0.0
      X = XA(1)
      Y = XA(2)
      Z = XA(3)
C*** GET CROSSWIND AND DOWNWIND COMPONENTS OF GENERATING LINE.
      QC=ABS(QLENTH*COS(THETA))
      QD=ABS(QLENTH*SIN(THETA))
      NBPT=1
      DO 390 J = 1,NBPT
      TJ = T- TTI(J)
C-----
C          TERM1 IS CLOUD MASS AS A FUNCTION OF TIME.
C-----
      IF(BREXP*TJ.GE.200.0)STOP01
      TRM1=EMUN*(1.0-EXP(-BREXP*TJ))
C-----
C          TERM2 IS TERRAIN SCAVENGING TERM.
C-----
C** CHANGE TO SMOKE COORDINATES.
      CALL LOCAT(J,X,Y,Z,XB,YB,ZB)
C*** UPWIND END OF GENERATING LINE WILL BE ORIGIN OF SMOKE SYSTEM.
      XB=XB+0.5*QD
      CALL UMEAN(J,TJ)
      IF(HK*XB/UBXB.GE.200.0)STOP01
      TRM2 = EXP(-HK*XB/UBXB)
C-----
C          TERM3 IS DOWNWIND PROBABILITY DENSITY.
C-----
      UT=UBXB*TJ
      IF(XB.LE.0.) GO TO 999
      IF(XB.GT.0. AND.XB.LE.UT) TERM3=1./<UT+0.5*QD>
      IF(XB.GT.UT AND.XB.LE.UT+QD) TERM3=<UT+QD-XB>/<UT*QD+0.5*QD*QD>
      IF(XB.GT.UT+QD) TERM3=0.
      TRM3=TERM3
C-----
C          TERM4 IS CROSSWIND PROBABILITY DENSITY.
C-----
      YWIDTH=YDIFF*XB+QC
      TRM4= 1./YWIDTH

```

```

COC00010
COC00020
COC00030
COC00040
COC00050
COC00060
COC00070
COC00080
COC00090
COC00100
COC00110
COC00120
COC00130
COC00140
COC00150
COC00160
COC00170
COC00180
COC00190
COC00200
COC00210
COC00220
COC00230
COC00240
COC00250
COC00260
COC00270
COC00280
COC00290
COC00300
COC00310
COC00320
COC00330
COC00340
COC00350
COC00360
COC00370
COC00380
COC00390
COC00400
COC00410
COC00420
COC00430
COC00440
COC00450
COC00460
COC00470
COC00480
COC00490
COC00500
COC00510
COC00520
COC00530
COC00540
COC00550
COC00560
COC00570
COC00580
COC00590
COC00600
COC00610
COC00620
COC00630
COC00640
COC00650
COC00660
COC00670
COC00680
COC00690
COC00700

```

```

IF(YB .GT. 0.5*YWIDTH ) TRM4=0.
IF( YB .LT. -0.5*YWIDTH) TRM4=0.
-----
TERMS IS VERTICAL PROBABILITY DENSITY.
-----
SIGZT=SIGZR*(XB/XREFZ)**ZDIFF
SZT2=SIGZT*SIGZT
HMVT = ZI(J) - (VS/100.0)*(XB/UBXB)
TV1 = EXP(-(HMVT-ZB)**2/(2.0*SZT2))
TV2 = RC*EXP(-(HMVT+ZB)**2/(2.0*SZT2))
TERR = 1.0E-06
JC = 0
T1 = 0.0
T2 = 0.0
T3 = 0.0
T4 = 0.0
TS = 0.0
370 JC = JC+1
R1 = RC**(JC-1)
R2 = RC**JC
R3 = RC**(JC+1)
IF( (2.*JC*HM+HMVT+ZB)**2 .GT. 600.*2.*SZT2) GO TO 371
T1 = R1*EXP(-(2.0*JC*HM-HMVT-ZB)**2/(2.0*SZT2))
T2 = R2*EXP(-(2.0*JC*HM-HMVT+ZB)**2/(2.0*SZT2))
T3 = R3*EXP(-(2.0*JC*HM+HMVT-ZB)**2/(2.0*SZT2))
T4 = R3*EXP(-(2.0*JC*HM+HMVT+ZB)**2/(2.0*SZT2))
T1234 = T1 + T2 + T3 + T4
TS = TS + T1234
IF(T1234 - TERR) 371,371,370
371 CONTINUE
TV3 = TS
TRM5 = TV1+TV2+TV3
TRM5=TRM5/( SIGZT*SQRT(2.*PI) )
C1 = TRM1*TRM2*TRM3*TRM4*TRM5
C2 = C2 + C1
380 CONTINUE
999 RETURN
END

```

```

COC00710
COC00720
COC00730
COC00740
COC00750
COC00760
COC00770
COC00780
COC00790
COC00800
COC00810
COC00820
COC00830
COC00840
COC00850
COC00860
COC00870
COC00880
COC00890
COC00900
COC00910
COC00920
COC00930
COC00940
COC00950
COC00960
COC00970
COC00980
COC00990
COC01000
COC01010
COC01020
COC01030
COC01040
COC01050
COC01060
COC01070
COC01080

```

```

SUBROUTINE GOGET(WAVE1,KWAVE)
C/*****
C/* SUBROUTINE GOGET
C/* GRNAD MODULE
C/* EOSAEL80
C/*****
C****SUBROUTINE FINDS SPECTRAL BAND FOR GIVEN SINGLE WAVELENGTH
KWAVE=0
IF(WAVE1.GE.0.40.AND.WAVE1.LT.0.70)KWAVE=1
IF(WAVE1.GE.0.70.AND.WAVE1.LT.1.20)KWAVE=2
IF(WAVE1.GE.1.20.AND.WAVE1.LT.3.00)KWAVE=4
IF(WAVE1.GE.3.00.AND.WAVE1.LT.5.00)KWAVE=4
IF(WAVE1.GE.5.00.AND.WAVE1.LT.12.0)KWAVE=5
IF(WAVE1.GE.1.06) KWAVE=3
IF(WAVE1.GE.10.6) KWAVE=6
IF(WAVE1.GE.94.0) KWAVE=7
IF(WAVE1.GT.3188.0.AND.WAVE1.LT.3195.0)KWAVE=7
RETURN
END
GOG00010
GOG00020
*/GOG00030
*/GOG00040
*/GOG00050
*/GOG00060
GOG00070
GOG00080
GOG00090
GOG00100
GOG00110
GOG00120
GOG00130
GOG00140
GOG00150
GOG00160
GOG00170
GOG00180

```

```

SUBROUTINE PARM5(ICAT)
C/*****
C/* SUBROUTINE PARM5
C/* GRNAD MODULE
C/* EDSAE80
C/*****
C*****SETS DIFFUSION PARAMETER DEFAULTS AS FUNCTION OF PASQUILL CATEGORY
COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
YDIFF=0.355
HK=0.002
VS=0.021
RC=0.70
XREFZ=100.0
IF(ICAT.GT.3)GO TO 1
WINDP=0.10
ZDIFF=2.08
HM=1000.0
SIGZR=14.0
GO TO 3
1 IF(ICAT.GT.4)GO TO 2
WINDP=0.20
ZDIFF=1.40
HM=300.0
SIGZR=7.2
GO TO 3
2 WINDP=0.40
ZDIFF=1.04
HM=50.0
SIGZR=5.0
3 RETURN
END
PAR00010
PAR00020
*/PAR00030
*/PAR00040
*/PAR00050
*/PAR00060
PAR00070
PAR00080
PAR00090
PAR00100
PAR00110
PAR00120
PAR00130
PAR00140
PAR00150
PAR00160
PAR00170
PAR00180
PAR00190
PAR00200
PAR00210
PAR00220
PAR00230
PAR00240
PAR00250
PAR00260
PAR00270
PAR00280
PAR00290
PAR00300
PAR00310

```

```

SUBROUTINE EXTIN(EX)
C/*****
C/*          SUBROUTINE EXTIN
C/*          GRNAD MODULE
C/*          EOSAEL80
C/*****
C*****PROGRAM TO SET EXTINCTION COEFFICIENTS FOR WP/RP SMOKE*****
  DIMENSION EX(7),CX(7)
  DATA CX/4.304,2.166,1.541,0.350,0.338,0.364,0.001/
  DO 1 I=1,7
    EX(I)=CX(I)
1 CONTINUE
  RETURN
  END

```

```

EXTN0010
EXTN0020
*/EXTN0030
*/EXTN0040
*/EXTN0050
*/EXTN0060
EXTN0070
EXTN0080
EXTN0090
EXTN0100
EXTN0110
EXTN0120
EXTN0130
EXTN0140

```

```

SUBROUTINE UMEAN(J,TJ)
C/*****
C/* SUBROUTINE UMEAN
C/* GRNAD MODULE
C/* EGSSEL80
C/*****
C*****CALCULATES MEAN WIND SPEED OVER EXTENT OF CLOUD
COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
IC = 0
QD = ABS(QLENTH*SIN(THETA))
P = WINDP + 1.0
C1=P*2.0**WINDP
Z2=ZI(J)
IF(Z2-2.0)10,10,20
10 CONTINUE
Z2=2.0
Z1=0.01
GO TO 200
20 Z1=ZI(J)-1.5*SIGZR
IF(Z1)30,30,40
30 Z1=0.01
40 CONTINUE
UH = (U2/((Z2-Z1)*C1))*(Z2**P-Z1**P)
US=UH
50 CONTINUE
XC = UH*TJ
Z2 = ZI(J) - (VS/100.0)*(XC/UH)
IF(Z2-2.0)151,151,155
151 CONTINUE
Z2=2.0
Z1=0.01
GO TO 160
155 CONTINUE
SIGZT = SIGZR*((XC+QD)/XREFZ)**ZDIFF
Z1 = ZI(J) - 1.5*SIGZT
IF(Z1)159,159,160
159 CONTINUE
Z1=0.01
160 UBXJ = (U2/((Z2-Z1)*C1))*(Z2**P-Z1**P)
GO TO 1000
200 CONTINUE
US=U2
UH=US
UBXJ=US
1000 CONTINUE
UBXJ = SQRT((UBXJ**2 + UH**2)/2.0)
IC = IC + 1
IF(IC,EQ,1) UH = UBXJ
IF(IC,EQ,1) GO TO 50
UBXB=UBXJ
RETURN
END

```

```

/UME00010
/UME00020
*/UME00030
*/UME00040
*/UME00050
/UME00060
/UME00070
/UME00080
/UME00090
/UME00100
/UME00110
/UME00120
/UME00130
/UME00140
/UME00150
/UME00160
/UME00170
/UME00180
/UME00190
/UME00200
/UME00210
/UME00220
/UME00230
/UME00240
/UME00250
/UME00260
/UME00270
/UME00280
/UME00290
/UME00300
/UME00310
/UME00320
/UME00330
/UME00340
/UME00350
/UME00360
/UME00370
/UME00380
/UME00390
/UME00400
/UME00410
/UME00420
/UME00430
/UME00440
/UME00450
/UME00460
/UME00470
/UME00480
/UME00490
/UME00500
/UME00510
/UME00520
/UME00530
/UME00540

```

```

SUBROUTINE SUMA(X,Y,Z,NDIM)
C/*****
C/* SUBROUTINE SUMA
C/* GRNAD MODULE
C/* EOSAEL80
C/*****
C/* PURPOSE:
C/* GENERAL PURPOSE INTEGRATION SUBROUTINE
C/* USAGE:
C/* CALLED FROM GRNAD TO GET INTEGRAL OF CONCENTRATION ALONG
C/* LINE-OF-SIGHT.
C/* DESCRIPTION OF PARAMETERS:
C/* X INDEPENDENT VARIABLE
C/* Y DEPENDENT VARIABLE
C/* Z INTEGRAL OF Y OVER X
C/* INDIM NUMBER OF POINTS
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
C/* NONE
C/* COMMON BLOCK STATEMENTS REQUIRED:
C/* NONE
C/* REMARKS:
C/* AS USED BY GRENADE, X IS POSITION ONLINE-OF-SIGHT, Y IS
C/* LOCAL CONCENTRATION, Z IS TOTAL CONCENTRATION.
C/* METHOD:
C/* SUMA INCREMENTS INTEGRAL BY AVERAGE OF LAST TWO Y VALUES
C/* TIMES DELTA X.
C/*****
DIMENSION X(1000),Y(1000),Z(1000)
SUM2 = 0.0
IF (NDIM - 1) 4,3,1
  *** INTEGRATION LOOP ***
1 DO 2 I = 2,NDIM
  SUM1 = SUM2
  SUM2 = SUM2 + 0.500*(X(I)-X(I-1))*(Y(I)+Y(I-1))
2 Z(I-1) = SUM1
3 Z(NDIM) = SUM2
4 RETURN
END
SUMA0010
SUMA0020
SUMA0030
SUMA0040
SUMA0050
SUMA0060
SUMA0070
SUMA0080
SUMA0090
SUMA0100
SUMA0110
SUMA0120
SUMA0130
SUMA0140
SUMA0150
SUMA0160
SUMA0170
SUMA0180
SUMA0190
SUMA0200
SUMA0210
SUMA0220
SUMA0230
SUMA0240
SUMA0250
SUMA0260
SUMA0270
SUMA0280
SUMA0290
SUMA0300
SUMA0310
SUMA0320
SUMA0330
SUMA0340
SUMA0350
SUMA0360
SUMA0370
SUMA0380
SUMA0390

```

```

SUBROUTINE LOCAT(I,X,Y,Z,XB,YB,ZB)
C/*****
C/* SUBROUTINE LOCAT
C/* GRNAD MODULE
C/* EOSAEL80
C/*****
C/* PURPOSE:
C/* LOCAT TRANSLATES POSITION INTO BURST AND WIND COORDINATES.
C/* USAGE:
C/* CALLED BY CONCH
C/* DESCRIPTION OF PARAMETERS:
C/* T TIME (SECONDS), INPUT
C/* X,Y,Z ORIGINAL POSITION (METERS), INPUT
C/* XI,YI,ZI ORIGINAL BURST POSITION (METERS), INPUT
C/* WDA ANGLE BETWEEN WIND VECTOR AND LINE OF SIGHT
C/* (RADIAN), INPUT
C/* XB,YB,ZB TRANSLATED POSITION (METERS), OUTPUT
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
C/* NONE
C/* COMMON BLOCK STATEMENTS REQUIRED:
C/* MECH1 MECH2
C/* REMARKS:
C/* ORIGINAL COORDINATES HAVE ORIGIN AT OBSERVER AND X-AXIS
C/* THROUGH TARGET. NEW COORDINATES HAVE ORIGIN AT BURST
C/* AND X-AXIS IN DIRECTION OF WIND VECTOR.
C/* METHOD:
C/* STANDARD ROTATION AND TRANSLATION OF AXES
C/*****
COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
XB = (X - XI(1))*COS(WDA) + (Y - YI(1))*SIN(WDA)
YB = -(X - XI(1))*SIN(WDA) + (Y - YI(1))*COS(WDA)
ZB = Z
IF(ZB.LT.0.0.AND.ABS(ZB).GT.ZI(1)) ZB=0.0
RETURN
END

```

```

LOC00010
LOC00020
LOC00030
LOC00040
LOC00050
LOC00060
LOC00070
LOC00080
LOC00090
LOC00100
LOC00110
LOC00120
LOC00130
LOC00140
LOC00150
LOC00160
LOC00170
LOC00180
LOC00190
LOC00200
LOC00210
LOC00220
LOC00230
LOC00240
LOC00250
LOC00260
LOC00270
LOC00280
LOC00290
LOC00300
LOC00310
LOC00320
LOC00330
LOC00340
LOC00350
LOC00360

```



```

SUBROUTINE DATRD(IWRIT,IFLAG)
C/*****
C/*          SUBROUTINE DATRD
C/*          GRNAD MODULE
C/*          EOSAEL80
C/*****
C*****THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS
C          THE SMOKE(EOSAEL) PROGRAM
C          INPUTS
C          EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,
C          FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.
C          PER FIELD BEGINNING IN COL 11. THE CARDS ARE NOT ORDER
C          DEPENDENT.
C          NAME          IGNORED
C          BURN          IGNORED
C          MUNC          XM,YM,ZM      COORDINATES OF GRENADE FIRING TANK
C          HEAD          HEAD         HEADING OF GRENADE FIRING TANK CLOCKWISE
C          RING          RING         FROM NORTH
C          DLEN          DLEN        GRENADE FIRING RANGE OF TANK
C          OBSC          XO,YO,ZO     SPACING OF GRENADES ALONG LINE PERPENDICULAR
C          TARC          XT,YT,ZT     TO HEADING
C          BART          STO          COORDINATES OF THE OBSERVER (M,M,M)
C          MUNT          ETO          COORDINATES OF THE TARGET (M,M,M)
C          FW           STO          STARTING TIME (ELAPSED TIME SINCE BLAST)
C          ITYPE        ETO          ENDING TIME FOR CALCULATION
C          EFF          DTD          TIME INCREMENT FOR CALCULATION
C          YF           XNORTH       X AXIS HEADING CLOCKWISE FROM NORTH
C          BRATE        XN           NUMBER OF MUNITIONS FIRED AT THE SAME
C          MISC         FW           LOCATION AND AT THE SAME TIME
C          MISC         TBURN        FILL WEIGHT (LBS)
C          MISC         ITYPE       BURN TIME OF SMOKE TYPE (IGNORED)
C          MISC         EFF         TYPE OF SMOKE (DEFAULTS TO 1)
C          MISC         YF         1.=WP, 2.=PWP, 3.=HC, 4.=FOG OIL
C          MISC         BRATE      EFFICIENCY OF BURN (PERCENT). IF 0.0,
C          MISC         MISC       DEFAULTS TO 62.0%.
C          MISC         MISC       YIELD FACTOR IF 0.0,DEFAULTS TO ANALYTICAL
C          MISC         MISC       MODEL
C          MISC         MISC       EXPONENTIAL BURN RATE PARAMETER
C          MISC         RHA        RELATIVE HUMIDITY (PERCENT)
C          MISC         UW        WIND VELOCITY (M/S)
C          MISC         WD        WIND DIRECTION (DEGREES)
C          MISC         ICAT      PASQUILL CATEGORY
C          MISC         AIRT      1-A, 2-B, 3-C, 4-D, 5-E, 6-F
C          MISC         TGRAD     SURFACE AIR TEMPERATURE (IGNORED)
C          MISC         WPOWR     VERT TEMP GRADIENT (IGNORED)
C          MISC         MISC       WIND PROFILE EXPONENT (DIMENSIONLESS)
C          MISC         MISC       DESIRED CHANGES IN EXTINCTION COEFF.
C          MISC         MISC       (OPTIONAL). IF NOT USED OR READ AS 0,
C          MISC         MISC       DEFAULTS TO ALPHA ARRAY VALUE IN STRNS.
C          MISC         MISC       BANDS ARE:
C          MISC         MISC       0.4-0.7 MICRONS
C          MISC         MISC       0.7-1.2 MICRONS
C          MISC         MISC       1.06 MICRONS
C          MISC         MISC       3.0-5.0 MICRONS
C          MISC         MISC       8.0-12. MICRONS
C          MISC         MISC       10.6 MICRONS
C          MISC         MISC       94.0 GHZ
C          MISC         SIGZR     DIFFUSION PARAMETER OPTION CARD FOR GRNAD
C          MISC         ZDIFF     DOWNWIND REFERENCE AT 100 M REFERENCE DIST.
C          MISC         YDIFF     VERTICAL DIFFUSION COEFFICIENT
C          MISC         HM       CROSSWIND DIFFUSION COEFFICIENT
C          MISC         HK       HEIGHT OF MIXING LAYER (METERS)
C          MISC         RC       TERRAIN SCAVENGING COEFFICIENT
C          MISC         VS       TERRAIN REFLECTION COEFFICIENT
C          MISC         MISC     SETTLEING VELOCITY (CM SEC)
C          GO           SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT
C          DONE        END OF JOB.

```

```

C*****
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /GEOMET/PTS<15>,IGEOSW
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLOHYT,FOGPRB,
* WNDVEL,WNDDIR,IPASCT
COMMON /MECH0/XM,YM,ZM,XO,YO,ZO,XT,YT,ZT,ISTO,IETO,IDTO,XN,FW,
*TBURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG,
*DLN,WPOWR,EXTC<8>,XMIS<8>,XNORTH
DIMENSION DUMY<8>,IR<26>,IR1<10>,R1<10>,EX<7>,INAME<35>
DATA IR/2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA,
*2HRC,2HEX,2HTC,2HBU,2HRN,2HMI,2HSC,2HGO,2H ,2HDO,2HNE,2HNA,2HME,
*2HOU,2HTP/
IF (IFLAG.GT.0) GO TO 8
DO 2 J=1,8
XMIS<J>=0.0
2 EXTC<J>=0.
C****
C**** BEGINNING OF READ LOOP
C****
8 CONTINUE
IF<IWRIT.EQ.0>GO TO 6
WRITE<IOOUT,200>
200 FORMAT<1H0,21X,20H*****CARD INPUT*****,/,21X,40<2H-->>
201 FORMAT<2A2,6X,35A2>
202 FORMAT<1H0,21X,2A2,6X,35A2>
6 DO 0 I=1,13
IF<I.EQ.13> GO TO 310
IF<IFLAG.GT.0> GO TO 4
IFLAG=1
READ<IOIN,201>IR1<1>,IR1<2>,<INAME<J>,J=1,35>
IF<IWRIT.EQ.0> GO TO 4
WRITE<IOOUT,202>IR1<1>,IR1<2>,<INAME<J>,J=1,35>
4 READ<IOIN,20>IR1<1>,IR1<2>,<R1<J>,J=2,8>
IF<IWRIT.EQ.0> GO TO 5
WRITE<IOOUT,30>IR1<1>,IR1<2>,<R1<J>,J=2,8>
5 IF<IR1<1>.EQ.IR<21>.AND.IR1<2>.EQ.IR<22>> GO TO 998
20 FORMAT<2A2,6X,7F10.3>
30 FORMAT<1H0,21X,2A2,6X,7F10.3>
C*****
C**** RELATING INPUT DATA TO VARIABLE NAMES.
C*****
IF<IR1<1>.EQ.IR<1>.AND.IR1<2>.EQ.IR<2>> GO TO 90
IF<IR1<1>.EQ.IR<3>.AND.IR1<2>.EQ.IR<4>> GO TO 100
IF<IR1<1>.EQ.IR<5>.AND.IR1<2>.EQ.IR<6>> GO TO 110
IF<IR1<1>.EQ.IR<7>.AND.IR1<2>.EQ.IR<8>> GO TO 120
IF<IR1<1>.EQ.IR<9>.AND.IR1<2>.EQ.IR<10>> GO TO 130
IF<IR1<1>.EQ.IR<11>.AND.IR1<2>.EQ.IR<12>> GO TO 140
IF<IR1<1>.EQ.IR<13>.AND.IR1<2>.EQ.IR<14>> GO TO 150
IF<IR1<1>.EQ.IR<15>.AND.IR1<2>.EQ.IR<16>> GO TO 155
IF<IR1<1>.EQ.IR<17>.AND.IR1<2>.EQ.IR<18>> GO TO 165
IF<IR1<1>.EQ.IR<19>.AND.IR1<2>.EQ.IR<20>> GO TO 175
IF<IR1<1>.EQ.IR<21>.AND.IR1<2>.EQ.IR<22>> GO TO 998
IF<IR1<1>.EQ.IR<23>.AND.IR1<2>.EQ.IR<24>> GO TO 70
IF<IR1<1>.EQ.IR<25>.AND.IR1<2>.EQ.IR<26>> GO TO 70
C*****
C**** ERROR CAUTION FOR INVALID DATA CARD
C*****
IFLAG=2
WRITE<IOOUT,80>
80 FORMAT<21X,35H*****CAUTION***** INVALID DATA CARD>
GO TO 70
90 RHA = R1<2>
UW = R1<3>
WD = R1<4>
ICAT = IFIX<R1<5>>
AIRT = R1<6>
TGRAD = R1<7>
WPOWR=R1<8>
GO TO 70
100 XN = R1<2>

```

```

      FW      =453.6*R1(3)
      TBURN   = R1(4)
      ITYPE   = IFIX(R1(5))
      EFF     = R1(6)
      YF=R1(7)
      BRATE=R1(8)
110  GO TO 70
      ISTO    = IFIX(R1(2))
      IETO    = IFIX(R1(3))
      IDTO    = IFIX(R1(4))
      XNORTH  = R1(5)
      IF(ISTO.LE.0)ISTO=1
120  GO TO 70
      XM      = R1(2)
      YM      = R1(3)
      ZM      = R1(4)
      HEAD    = R1(5)
      RNG     = R1(6)
      DLEN    = R1(7)
130  GO TO 70
      XU      = R1(2)
      YU      = R1(3)
      ZU      = R1(4)
140  GO TO 70
      XT      = R1(2)
      YT      = R1(3)
      ZT      = R1(4)
C*****
C*****BURN CARD DATA DUMMYED BY PROGRAM GRNAD*****
C*****
155  DO 156 J=1,7
156  DUMY(J)=R1(J+1)
      GO TO 70
150  DO 152 J=1,7
152  EXTC(J)=R1(J+1)
      GO TO 70
165  DO 166 J=1,7
166  XMIS(J)=R1(J+1)
      GO TO 70
175  GO TO 311
C*****
C*****CAUTION FOR TOO MANY CARDS
C*****
310  WRITE(IOOUT,320)
      IFLAG=3
C*****
C*****DEFAULT NON USER DEFINED INPUT*****
C*****
311  IF(ITYPE.EQ.1)GO TO 3
      ITYPE=1
      WRITE(IOOUT,171)
171  FORMAT(1H ,21X,17H*****CAUTION*****/,1H ,21X,54HWRONG SMOKE TYPE
*FOR PROGRAM GRNAD--DEFAULTED TO WP/RP)
3  IF(EFF.EQ.0.0)EFF=62.0
   IF(YF.EQ.0.0)YF=3.14+0.032*RHA
   IF(EXTC(1).GT.0.0)GO TO 1
   CALL EXTIN(EX)
   DO 7 I=1,7
7  EXTC(I)=EX(I)
1  CONTINUE
320  FORMAT(21X,17H*****CAUTION*****/,
*21X,56HMORE THAN 10 DATA CARDS ENTERED--REMAINING CARDS IGNORED)
      GO TO 9999
998  IFLAG=4
9999 IF(IGEOSW.NE.1) GO TO 555
      DISKTM=1000.
C***  CONVERT KM TO M.
      XT=PTS(1)*DISKTM
      YT=PTS(2)*DISKTM
      ZT=PTS(3)*DISKTM

```

```

DAT01410
DAT01420
DAT01430
DAT01440
DAT01450
DAT01460
DAT01470
DAT01480
DAT01490
DAT01500
DAT01510
DAT01520
DAT01530
DAT01540
DAT01550
DAT01560
DAT01570
DAT01580
DAT01590
DAT01600
DAT01610
DAT01620
DAT01630
DAT01640
DAT01650
DAT01660
DAT01670
DAT01680
DAT01690
DAT01700
DAT01710
DAT01720
DAT01730
DAT01740
DAT01750
DAT01760
DAT01770
DAT01780
DAT01790
DAT01800
DAT01810
DAT01820
DAT01830
DAT01840
DAT01850
DAT01860
DAT01870
DAT01880
DAT01890
DAT01900
DAT01910
DAT01920
DAT01930
DAT01940
DAT01950
DAT01960
DAT01970
DAT01980
DAT01990
DAT02000
DAT02010
DAT02020
DAT02030
DAT02040
DAT02050
DAT02060
DAT02070
DAT02080

```

555 X0=PTSC(4)*DISKTM
Y0=PTSC(5)*DISKTM
Z0=PTSC(6)*DISKTM
RETURN
END

DAT02090
DAT02100
DAT02110
DAT02120
DAT02130

```

SUBROUTINE LT4MCH1, H2, ANGLE, ITYPE, IXY, TRAN, RADA, RADG, IEMISS, LEN, L4M00010
+MODEL, VIS, V11, V22, T1, ICLMAT, IERR, NR, IHAZE, MULDV) L4M00020
LOGICAL ISPOT, LOREAD, N16 L4M00030
COMMON /CONST/PI, PI2, CA, TWOP1, TORRMB, CDEGK L4M00040
COMMON /M01/EH(16, 34), P(34), T(34), WH(34), Z(34), WA(34), RE, M, NL L4M00050
COMMON /M02/WO(34), RO, TBOUND, JP, IM, ML, IP, JSTOR L4M00060
COMMON /M05/ C1(501), C2(258), C3(86), C4(33), C5(6), C5DUM(9), C8(82), L4M00070
1 C11(4), C12(15), C14(21), C15(6) L4M00080
COMMON /M07/TR(67), FW(67), FO(67) L4M00090
COMMON /M08/SUM4, SUM5, SUM8, SUM11, SUM6 L4M00100
COMMON /M09/RADMAX, RADMIN, VRMAX, VRMIN L4M00110
COMMON /M03/ FS(9), S1(9), S2(9), FNH3(9), FH1(9), FH2(9), FNO2(9), L4M00120
1 O1(9), O2(9), PPMO2, PPMNH3, PPMNO2 L4M00130
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUL L4M00140
COMMON /LOWEX/WPATH(68, 16), WLAY(34, 16), TBBY(68), TX(16), BETAEX, L4M00150
1 CLDHGT, NCLD L4M00160
COMMON /EM1/HMIN, KMAX, IJ, J1, J2, JMIN, JEXTRA, NP1 L4M00170
COMMON /EM2/W(16), E(16), IL, IKMAX, LENTOR, NLL L4M00180
COMMON /BASPOT/ ANG(65), SUM(65), WVL(16), NWVL, ALBB(16), BS(16), L4M00190
1 BE(16), SINGWV, PF(65), LMAX L4M00200
COMMON /SPOTLO/ISPOT, LOREAD, N16 L4M00210
DIMENSION TRAN(16) L4M00220
DIMENSION RADA(16), RADG(16) L4M00230
PLANCK RADIANCE FUNCTION L4M00240
FF(T,V)=1.190956E-16*(V**5)/(EXP(1.43879*V/T)-1.) L4M00250
WATT CM-2 ST-1 MICRON-1 L4M00260
***** L4M00280
PROGRAM MODIFIED LOWTRAN CALCULATES THE TRANSMITTANCE L4M00290
OF THE ATMOSPHERE FROM 830 TO 1250, 2010 TO 3330, AND L4M00300
5010 TO 39990 CM-1 (0.25 TO 2.0, 3.0 TO 5.0, AND 8.0 TO 12.0 L4M00310
MICRONS) AT 20 CM-1 SPECTRAL INTERVALS ON A LINEAR WAVENUMBER L4M00320
SCALE. L4M00330
REFRACTION AND EARTH CURVATURE EFFECTS ARE EXCLUDED. ATMOSPHERE L4M00340
IS LAYERED IN ONE KM INTERVALS BETWEEN 0 AND 25 KM, 5 KM INTER- L4M00360
VALS FROM 25 TO 50 KM, A 20 KM LAYER FROM 50 TO 70 KM, A 30 KM L4M00370
LAYER FROM 70 TO 100 KM, AND ONE FROM 100 KM TO INFINITY. L4M00380
***** L4M00390
PROGRAM ACTIVATED BY SUBMISSION OF CARD SEQUENCE AS FOLLOWS L4M00400
CARD 1 MODEL, IHAZE, ITYPE, LEN, JP, NPLT, IM, ML, IEMISS, L4M00410
RO, TBOUND, BETAEX, FORMAT ( 9I3, 3F10.3) L4M00420
MODEL =0, METEOROLOGICAL DATA SPECIFIED L4M00430
=1, TROPICAL MODEL ATMOSPHERE L4M00440
=2, MIDLATITUDE SUMMER L4M00450
=3, MIDLATITUDE WINTER L4M00460
=4, SUBARCTIC SUMMER L4M00470
=5, SUBARCTIC WINTER L4M00480
=6, 1962 US STANDARD L4M00490
=7, NEW MODEL ATMOSPHERE L4M00500
=8, ISRAELI STANDARD ATMOSPHERE (YEAR, DAYTIME) L4M00510
=9, ISRAELI STANDARD ATMOSPHERE (YEAR, NIGHTTIME) L4M00520
** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK ** L4M00530
FOR SLANT PATHS IHAZE = 1, 2, OR 3 ARE THE ONLY ALLOWED VALUES. L4M00540
IHAZE =0, NO AEROSOL ATTENUATION L4M00550
=1, MARITIME POLAR L4M00560
=2, MARITIME ARCTIC L4M00570
=3, CONTINENTAL POLAR L4M00580
=4, RAIN L4M00590
=5, SNOW L4M00600
=7, USER SUPPLIED EXTINCTION COEFFICIENT L4M00610
(READ ON ATM CARD - SEE CARD 3 BELOW) L4M00620
ITYPE =1, HORIZONTAL (CONSTANT PRESSURE) PATH L4M00630
=2, VERTICAL OR SLANT PATH BETWEEN 2 ALTITUDES L4M00640
=3, VERTICAL OR SLANT PATH TO SPACE L4M00650
LEN =0, NORMAL OPERATION L4M00660
=1, DOWNWARD LONG PATH L4M00670
JP =0, NORMAL OPERATION L4M00680
=1, SUPPRESS PRINT OF HORIZ AND VERTICAL PROFILES L4M00690
NPLT =0, NORMAL OPERATION L4M00700
L4M00710

```

```

=1, IN TRANSMISSION MODE WRITE, WAVELENGTH (UM), L4M00720
H2O, CO2+, OZONE, N2 C, H2O C, MOL SCAT, L4M00730
NITRIC, SO2, HNO3, NO2, IN EMISSION MODE L4M00740
WRITE WAVELENGTH (UM) AND RADIANCE PER MICRON, L4M00750
RESULTS WILL BE WRITTEN ON NPLTU (SEE COMMON BLOCK IOUNIT) L4M00760
IM =1, RADIOSONDE DATA TO BE READ INITIALLY L4M00770
=0, NORMAL OPERATION OR WHEN SUBSEQUENT CALCULATIONS L4M00780
ARE TO BE RUN WITH MODEL = 7 L4M00790
ML =NUMBER OF LEVELS TO BE READ IN FOR MODEL = 7 L4M00800
***IM AND ML ONLY USED WHEN MODEL = 7 AND THEN ONLY ON L4M00810
FIRST CALCULATIONS WHEN DATA READ IN L4M00820
IEMISS DETERMINES MODE OF EXECUTION OF PROGRAM L4M00830
=0, TRANSMITTANCE MODE L4M00840
=1, RADIANCE MODE L4M00850
RO RADIUS OF THE EARTH (KM) AT LOCATION OF CALCULATION L4M00860
***DEFAULT WILL BE MIDLATITUDE VALUE OF 6371.23 KM WHEN L4M00870
MODEL = 0 OR = 7 OTHERWISE DEFAULT IS EARTH RADIUS L4M00880
FOR STANDARD MODEL ATMOSPHERE SPECIFIED BY MODEL L4M00890
TBOUND TEMPERATURE OF EARTH (DEGREES K) AT LOCATION OF CALCUL L4M00900
***USED ONLY IN RADIANCE MODE FOR SLANT PATHS WHICH INTERSECT EARTH L4M00910
***DEFAULT IS TEMPERATURE OF FIRST LAYER BOUNDARY TEMPERATURE L4M00920
BETAEX USER SUPPLIED EXTINCTION COEFFICIENT, INPUT ONLY L4M00930
WHEN IHAZE=7 L4M00940
CARD 2 H1,H2,ANGLE,RANGE,BETA,VIS,CLDHGT FORMAT (7F10.3) L4M00950
H1 INITIAL ALTITUDE (KM) L4M00960
H2 FINAL ALTITUDE (KM) L4M00970
ANGLE INITIAL ZENITH ANGLE (DEG) L4M00980
RANGE PATH LENGTH (KM) L4M00990
BETA EARTH CENTER ANGLE SUBTENDED BY H1 AND H2 (DEG) L4M01000
VIS SEA LEVEL VISUAL RANGE (KM) L4M01010
CLDHGT HEIGHT OF BOTTOM OF CLOUD LAYER (KM), WHEN IHAZE NE 0 L4M01020
***VIS NOT REQUIRED ON THIS CARD IF ICLMAT (EOMAIN) =1 OR L4M01030
***THIS IS FIRST LOOP THROUGH LT4 L4M01040
***SEE MANUAL FOR MORE DETAIL L4M01050
CARD 2A V1,V2,MULDV FORMAT (2F10.3,I2) L4M01060
V1 INITIAL FREQUENCY (CM**-1) L4M01070
V2 FINAL FREQUENCY (CM**-1) L4M01080
MULDV MULTIPLIER FOR FREQUENCY INCREMENT, WHERE THE L4M01090
INCREMENT IS A MULTIPLE OF 20 (CM**-1). L4M01100
OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY NR=1 IN EOMAIN) L4M01110
CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2). L4M01120
CARDS 2 - NUMBER OF VALUES: FORMAT (2(E10.4,1X)) L4M01130
ONE VALUE OF WAVELENGTH (UM) AND RESPONSE FUNCTON PER CARD L4M01140
CARD 3 IXY FORMAT (I3) L4M01150
IXY =0, EXIT LOWTRAN MODULE L4M01160
=1, SELECT NEW WAVE FREQUENCY RANGE (CARD 2A) L4M01170
=2, SELECT NEW DATA SEQUENCE (CARDS 1,2,2A,3) L4M01180
=3, SELECT NEW CARD 2 AND CARD 3 L4M01190
=4, SELECT NEW CARD 1 AND CARD 3 L4M01200
L4M01210
***FOR NON-STANDARD CONDITIONS SEE MANUAL L4M01220
C***** L4M01230
V1=V11 L4M01240
V2=V22 L4M01250
KMAX=16 L4M01260
ISPOT1=0 L4M01270
RESPFN=0. L4M01280
SUMRPF=0. L4M01290
SUMINT=0. L4M01300
IF (ISPOT) NPLT=0 L4M01310
200 CONTINUE L4M01320
IF (.NOT.LOREAD) GO TO 400 L4M01330
LOREAD=.FALSE. L4M01340
READ (LOUNIT,3300) IATM,NL L4M01350
NL4=4*NL L4M01360
DO 299 I=1,NL4 L4M01370
299 READ (LOUNIT,3500) DUMMY L4M01380
READ (LOUNIT,3510) PPMO2,PPMNH3,PPMNO2 L4M01390
READ (LOUNIT,3700)(TR(I),FW(I),FO(I),I=1,67) L4M01400
READ (LOUNIT,3800)(C(I),I=1,501) L4M01410

```

```

READ (LOUNIT,3800)(C2(I),I=1,258)
READ (LOUNIT,3800)(C3(I),I=1,86)
READ (LOUNIT,3900)(C4(I),I=1,33)
READ (LOUNIT,3810)(C5(I),I=1,6)
READ (LOUNIT,3900)(C8(I),I=1,82)
READ (LOUNIT,4000)(C11(I),I=1,4)
READ (LOUNIT,4010)(C12(I),I=1,15)
READ (LOUNIT,4020)(C14(I),I=1,21)
READ (LOUNIT,4020)(C15(I),I=1,6)
READ (LOUNIT,4010)(FS(I),S1(I),S2(I),I=1,9)
READ (LOUNIT,4010)(FNH3(I),FH1(I),FH2(I),I=1,9)
READ (LOUNIT,4010)(FNO2(I),O1(I),O2(I),I=1,9)
REWIND LOUNIT
IF (ISPOT) GO TO 710
400 CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(I))
IF (IERR.EQ.1) RETURN
JP=0
JSTOR=0
IP=0
IF (.NOT.ISPOT) GO TO 700
BETA=0.
RANGE=0.
RO=0.
IF (ITYPE.EQ.1) RANGE=ANGLE
IF (IXY.EQ.0) GO TO 700
401 GO TO (500,700,600,680),IXY
500 AVW=1.E+04/V1
ALAM=1.E+04/V2
SUMA=0.
GO TO 1100
600 IF (MODEL.EQ.0) GO TO 800
ISPOT1=1
GO TO 1000
700 CONTINUE
RE=6371.23
680 IF (.NOT.ISPOT) READ (IOIN,3300) MODEL,HAZE,ITYPE,
1 LEN,JP,NPLT,IM,ML,IEMISS,RO,TBOUND
C*****IEMISS=0=TRANSMISSION MODE / IEMISS=1=EMISSION MODE
IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,4100)
IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4200)
IF (ISPOT) GO TO 800
IF (MODEL.EQ.0.OR.MODEL.EQ.7) GO TO 210
710 READ(LOUNIT,3300)IATH,NL
MSKIP=(MODEL-1)/2
IF(MSKIP.EQ.0) GO TO 220
IF (MSKIP.EQ.4) MSKIP=3
DO 230 J=1,MSKIP
DO 230 I=1,NL
230 READ(LOUNIT,3500) DUMMY
220 CONTINUE
C ISRAELI STD ATM READS
IF (MODEL.EQ.8) GO TO 270
IF (MODEL.EQ.9) GO TO 250
IF (2*(MODEL/2).EQ.MODEL) GO TO 250
270 DO 240 I=1,NL
240 READ(LOUNIT,3500)Z(I),P(I),T(I),WA(I),WH(I),WO(I)
GO TO 210
250 DO 260 I=1,NL
260 READ(LOUNIT,3550)Z(I),P(I),T(I),WA(I),WH(I),WO(I)
210 REWIND LOUNIT
IF (ISPOT) RETURN
800 M=MODEL
900 IF (RO.GT.0) RE=RO
LENTOR=LEN
1000 CALL ABSORB(IXY,IERR,W,V1,V2,DV,SUMA,MULDV,ANGLE,LEN,ITYPE,H1,H2,
1 MODEL,ISPOT1,RANGE,BETA,VIS,ICLMAT,IV1,IV2,IDV)
IF (IERR.EQ.1) TRAN(I)=1.
IF (IERR.EQ.1) RETURN
1100 CONTINUE
IF (.NOT.ISPOT) WRITE (IOOUT,4300)

```

```

L4M01420
L4M01430
L4M01440
L4M01450
L4M01460
L4M01470
L4M01480
L4M01490
L4M01500
L4M01510
L4M01520
L4M01530
L4M01540
L4M01550
L4M01560
L4M01570
L4M01580
L4M01590
L4M01600
L4M01610
L4M01620
L4M01630
L4M01640
L4M01650
L4M01660
L4M01670
L4M01680
L4M01690
L4M01700
L4M01710
L4M01720
L4M01730
L4M01740
L4M01750
L4M01760
L4M01770
L4M01780
L4M01790
L4M01800
L4M01810
L4M01820
L4M01830
L4M01840
L4M01850
L4M01860
L4M01870
L4M01880
L4M01890
L4M01900
L4M01910
L4M01920
L4M01930
L4M01940
L4M01950
L4M01960
L4M01970
L4M01980
L4M01990
L4M02000
L4M02010
L4M02020
L4M02030
L4M02040
L4M02050
L4M02060
L4M02070
L4M02080
L4M02090
L4M02100
L4M02110

```

```

IF (.NOT.ISPOT) WRITE(IOOUT,4400) (W(I),I=1,6),W(8),W(10)
IF (.NOT.ISPOT) WRITE(IOOUT,4401)
IF (.NOT.ISPOT) WRITE (IOOUT,4500) W(11),W(12),W(13),W(14),
* W(15),W(16)
1200 CONTINUE
C NCLD WILL BE THE INDEX OF THE LAYER ABOVE CLDHGT
DO 5 ICLD=2,6
NCLD=ICLD
IF (CLDHGT.LT.Z(ICLD)) GO TO 6
5 CONTINUE
6 CONTINUE
7 IF (CLDHGT.GT.Z(6)) WRITE (IOOUT,7)
FORMAT (1H,25H**** WARNING FROM LOWTRAN,/,1X,14H CLOUD BASE IS ,
1 23HLIMITED TO 4 KM MAXIMUM/)
I=1
L=1
IV=IV1
ICOUNT=0
IF (N16) KWAVE=0
IF (IEMISS.EQ.0) GO TO 1300
RADSUM=0.0
FACTOR=0.5
CALL LTPATH(WLAY,WPATH,TBBY,ANGLE,LEN,ITYPE,H1,H2,MODEL)
IF (.NOT.ISPOT) WRITE (IOOUT,4600)
IF (.NOT.ISPOT) WRITE (IOOUT,4700)
C**** BEGINNING OF TRANSMITTANCE CALCULATIONS
1300 CONTINUE
IF (N16) KWAVE=KWAVE+1
SUMV=0.
TLOLD=1.
TSOLD=1.
TX7=1.
TX10=1.
IKLO=1
TOLD=1
IF (IEMISS.EQ.0) IKMAX=IKLO
C ONLY ONE LOOP FOR TRANSMISSION: LOOP OVER LAYERS FOR EMISSION
DO 2300 IK=IKLO,IKMAX
IF (IEMISS.EQ.0) GO TO 1500
C TRANSFER CUMULATIVE ABSORBER AMOUNTS FOR TH IK TH LEVEL AND
C THE K TH ABSORBER - EMISSION ONLY.
DO 1400 K=1,KMAX
W(K)=WPATH(IK,K)
1400 CONTINUE
1500 IJ=IK
IF (ICOUNT.EQ.0) GO TO 1600
IF (ICOUNT.EQ.50) GO TO 1600
GO TO 1700
1600 ICOUNT=0
IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4800)
1700 DO 1800 K=1,KMAX
TX(K)=1.0
1800 CONTINUE
ICOUNT=ICOUNT+1
V=FLOAT(IV)
ALAM=1.E+04/V
I=(IV-830)/20+1
SUM4=0.
SUM5=0.
SUM6=0.
SUM8=0.
SUM11=0.
CALL FREQSL(I,IV,W,TX)
TX(9)=SUM4+SUM5+SUM8+SUM11+SUM6
IF (TX(9).EQ.0.0) GO TO 2000
IF (TX(9).LE.0.1) GO TO 1900
IF (TX(9).GT.20.) GO TO 2100
TX(9)=EXP(-TX(9))
GO TO 2200
1900 TX(9)=1.0-TX(9)+0.5*TX(9)*TX(9)

```

```

L4M02120
L4M02130
L4M02140
L4M02150
L4M02160
L4M02170
L4M02180
L4M02190
L4M02200
L4M02210
L4M02220
L4M02230
L4M02240
L4M02250
L4M02260
L4M02270
L4M02280
L4M02290
L4M02300
L4M02310
L4M02320
L4M02330
L4M02340
L4M02350
L4M02360
L4M02370
L4M02380
L4M02390
L4M02400
L4M02410
L4M02420
L4M02430
L4M02440
L4M02450
L4M02460
L4M02470
L4M02480
L4M02490
L4M02500
L4M02510
L4M02520
L4M02530
L4M02540
L4M02550
L4M02560
L4M02570
L4M02580
L4M02590
L4M02600
L4M02610
L4M02620
L4M02630
L4M02640
L4M02650
L4M02660
L4M02670
L4M02680
L4M02690
L4M02700
L4M02710
L4M02720
L4M02730
L4M02740
L4M02750
L4M02760
L4M02770
L4M02780
L4M02790
L4M02800
L4M02810

```



```

2000 GO TO 2200
TX(9)=1.0
GO TO 2200
2100 TX(9)=0.
2200 TX(9)=TX(1)*TX(2)*TX(3)*TX(9)*TX(12)*TX(13)*TX(14)
C AEROSOL COMPUTATIONS UNTIL LABEL 1
IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 1
C
C IF SPOT IS CALLING LT4M WITH ITYPE = 3, DO NOT INCLUDE AEROSOLS
C
IF (ISPOT.AND.ITYPE.EQ.3) GO TO 1
EXT55=3.914/VIS
UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE
PASS HORIZONTAL DIST IF ITYPE=1, SLANT DISTANCE IF ITYPE GT 1.
IF (ITYPE.EQ.1) RNG=RANGE
IF (ITYPE.EQ.2.AND.(H2.GT.H1).AND.(RANGE.GT..5/COS(ANGLE*CA)))
1 RNG=.5/COS(ANGLE*CA)
IF (ITYPE.EQ.2.AND.(H2.LT.H1).AND.
1 (RANGE.GT..5/COS((180.-ANGLE)*CA)))
2 RNG=.5/COS((180.-ANGLE)*CA)
IF (ITYPE.EQ.3.AND.(RANGE.GT..5/COS(ANGLE*CA)))
1 RNG=.5/COS(ANGLE*CA)
IF (ITYPE.EQ.3.AND.RANGE.LT..0001)
1 RNG=.5/COS(ANGLE*CA)
ISLANT=ITYPE-1
CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL
CALL XSCALE(ALAM,88.,EXT55,TX7,IERR,ISLANT,IHAZE,RNG,ANGLE)
IF (IERR.EQ.1) RETURN
C USER OPTIONS
IF (IHAZE.EQ.7) TX7=EXP(-BETAEX*RANGE)
IF (ISPOT.AND.IHAZE.EQ.8) TX7=EXP(-BE(KWAVE)*RANGE)
CONTINUE
TX(9)=TX(9)*TX7
IF (IV.GE.13000) TX(3)=TX(8)
TNEW=TX(9)
IF (IEMISS.EQ.0) GO TO 2500
C COMPUTER LIMITS
BBIK=0.0
IF (ABS(1.43879*V/TBKY(1K)).LT.85.) BBIK=FF(TBKY(1K),V)
C AEROSOL COMPUTATIONS UNTIL LABEL 2
IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 2
C FIND AEROSOL ABSORPTION IN DIFFERENT WAVELENGTH BANDS FROM EXTN
IF (ALAM.LT.2.) TX10=1.
IF (ALAM.GE.3. .AND. ALAM.LE.5.) TX10=TX7**.2
IF (ALAM.GE.8. .AND. ALAM.LE.12.05) TX10=TX7**.45
2 CONTINUE
TLNEW=TX(9)*TX10/(TX(6)*TX7)
TSNEW=TX7*TX(6)/TX10
DTAU=ABS(TLOLD-TLNEW)
IF (DTAU.LT.1.0E-5.AND.TLNEW.LT.1.0E-5) GO TO 2400
SUMV=SUMV+(TOLD-TNEW)*BBIK
TLOLD=TLNEW
TSOLD=TSNEW
TOLD=TNEW
2300 CONTINUE
2400 CONTINUE
TAUG=0.
IF (HMIN.LE.0.0.AND.IL.EQ.1) TAUG=TX(9)
T1=T(1)
IF (TBOUND.GT.0.0) T1=TBOUND
C COMPUTER LIMITS
BBG=0.0
IF (ABS(1.43879*V/T1).LT.85.) BBG=FF(T1,V)*TAUG
IF (N16) RADG(KWAVE)=BBG*1.E+04
IF (N16) RADA(KWAVE)=SUMV*1.E+04
IF (HMIN.LE.0) SUMV=SUMV+BBG
SUMVV=SUMV
IF (IV.GT.IV1) FACTOR=1.0
IF (IV.GE.IV2) FACTOR=0.5
SUMV=(1.0E+04/V**2)*SUMV

```

```

L4M02820
L4M02830
L4M02840
L4M02850
L4M02860
L4M02870
L4M02880
L4M02890
L4M02900
L4M02910
L4M02920
L4M02930
L4M02940
L4M02950
L4M02960
L4M02970
L4M02980
L4M02990
L4M03000
L4M03010
L4M03020
L4M03030
L4M03040
L4M03050
L4M03060
L4M03070
L4M03080
L4M03090
L4M03100
L4M03110
L4M03120
L4M03130
L4M03140
L4M03150
L4M03160
L4M03170
L4M03180
L4M03190
L4M03200
L4M03210
L4M03220
L4M03230
L4M03240
L4M03250
L4M03260
L4M03270
L4M03280
L4M03290
L4M03300
L4M03310
L4M03320
L4M03330
L4M03340
L4M03350
L4M03360
L4M03370
L4M03380
L4M03390
L4M03400
L4M03410
L4M03420
L4M03430
L4M03440
L4M03450
L4M03460
L4M03470
L4M03480
L4M03490
L4M03500
L4M03510

```

```

RADSUM=RADSUM+DV*FACTOR*SUMV
IF (.NOT. ISPOT) WRITE (IOOUT,4900) V,
1 ALAM,SUMV,SUMVV,RADSUM,TX(9),TX7,TX10
98 IF (NPLT.EQ.1) WRITE (NPLOTU,98) ALAM,SUMVV
FORMAT (F7.4,1X,E13.5)
IF (SUMV.GE.RADMAX) VRMAX=V
IF (SUMV.GE.RADMAX) RADMAX=SUMV
IF (SUMV.LE.RADMIN) VRMIN=V
IF (SUMV.LE.RADMIN) RADMIN=SUMV
2500 AB=1.-TX(9)
IF (IV.EQ.IV1.OR.IV.GE.IV2) AB=0.5*AB
SUMA=SUMA+AB*DV
IF (IEMISS.EQ.1) GO TO 2600
IF (.NOT. ISPOT) WRITE (IOOUT,5000) IV,
1 ALAM,(TX(K),K=1,6),TX(11),TX(12),TX(13),TX(14),
2 SUMA,TX(9),TX7
99 IF (NPLT.EQ.1) WRITE (NPLOTU,99) ALAM,(TX(K),K=1,6),(TX(J),J=11,14)
FORMAT (F7.4,10(1X),F6.4)
2600 CONTINUE
RESPFN=RESFN(NR,ALAM)
SUMRPF=SUMRPF+RESPFN
IF (IV.GT.IV1) SUMINT=SUMINT+.5*(OLDTX9*OLDRFN+TX(9))*RESPFN*
+ABS(1./FLOAT(IV)-1./FLOAT(IV-IDV))*1.E+04
IF (N16) TRAN(KWAVE)=TX(9)
OLDTX9=TX(9)
OLDRFN=RESPFN
IV=IV+IDV
IF (IV.GE.IV2) GO TO 2700
GO TO 1300
2700 CONTINUE
IF (NR.NE.1) SUMRPF=1.
SUMINT=SUMINT/(SUMRPF*1.E+04*ABS(1./FLOAT(IV1)-1./FLOAT(IV2)))
IF (.NOT. ISPOT) TRAN(1)=SUMINT
IF (.NOT. ISPOT.AND.NR.EQ.1) WRITE (IOOUT,3250) SUMINT
IF (.NOT. ISPOT.AND.NR.NE.1) WRITE (IOOUT,3275) SUMINT
RESPFN=0.
SUMRPF=0.
SUMINT=0.
IF ((IEMISS.EQ.1).AND.(.NOT. ISPOT)) WRITE (IOOUT,5100)
1 VRMIN,RADMIN,VRMAX,RADMAX
JSTOR=0
AB=1.0-SUMA/FLOAT(IV-IV1)
IF (ISPOT) RETURN
WRITE (IOOUT,5200) IV1,IV,SUMA,AB
IF ((IEMISS.EQ.1).AND.(.NOT. ISPOT)) WRITE (IOOUT,5300) RADSUM
IF (.NOT. ISPOT) READ (IOIN,3300) IXY
IF (IXY.EQ.0) GO TO 3100
GO TO (2800,700,2900,680,3100),IXY
2800 CONTINUE
READ (IOIN,5400) V1,V2,MULDV
CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(1))
IF (IERR.EQ.1) RETURN
AVW=10000./V1
ALAM=10000./V2
WRITE (IOOUT,5500) V1,V2,DV,ALAM,AVW
SUMA=0.0
GO TO 1100
2900 IF (MODEL.EQ.0) GO TO 800
GO TO 401
3000 CONTINUE
READ (IOIN,3300) MODEL,IHAZE,ITYPE,LEN,JP,NPLT,IM,
1 ML,IEMISS,RO,TBOUND,BETAEX
IF (IEMISS.EQ.1) WRITE (IOOUT,4100)
IF (IEMISS.EQ.0) WRITE (IOOUT,4200)
LENTOR=LEN
GO TO 800
3100 RETURN
3250 FORMAT (/,1X,48HWAVELENGTH AND SENSOR INTEGRATED TRANSMISSION =
+,E10.4)

```

```

L4M03520
L4M03530
L4M03540
L4M03550
L4M03560
L4M03570
L4M03580
L4M03590
L4M03600
L4M03610
L4M03620
L4M03630
L4M03640
L4M03650
L4M03660
L4M03670
L4M03680
L4M03690
L4M03700
L4M03710
L4M03720
L4M03730
L4M03740
L4M03750
L4M03760
L4M03770
L4M03780
L4M03790
L4M03800
L4M03810
L4M03820
L4M03830
L4M03840
L4M03850
L4M03860
L4M03870
L4M03880
L4M03890
L4M03900
L4M03910
L4M03920
L4M03930
L4M03940
L4M03950
L4M03960
L4M03970
L4M03980
L4M03990
L4M04000
L4M04010
L4M04020
L4M04030
L4M04040
L4M04050
L4M04060
L4M04070
L4M04080
L4M04090
L4M04100
L4M04110
L4M04120
L4M04130
L4M04140
L4M04150
L4M04160
L4M04170
L4M04180
L4M04190
L4M04200
L4M04210

```

```

3275 FORMAT (/,1X,37HWAVELENGTH INTEGRATED TRANSMISSION = ,E10.4) L4M04220
3300 FORMAT (/,9I3,3F10.3) L4M04230
3500 FORMAT (F6.1,2(E9.3,F5.1,E9.3,2E7.1)) L4M04240
3510 FORMAT(3F8.3) L4M04250
3550 FORMAT(F6.1,37X,E9.3,F5.1,E9.3,2E7.1) L4M04260
3700 FORMAT (4(F6.3,2F7.4)) L4M04270
3800 FORMAT (15F5.2) L4M04280
3810 FORMAT(15F5.3) L4M04290
3900 FORMAT (8E9.2) L4M04300
4000 FORMAT (12F6.3) L4M04310
4010 FORMAT(10F8.4) L4M04320
4020 FORMAT(10F8.3) L4M04330
4100 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,/,1X, L4M04340
1 45HPROGRAM WILL BE EXECUTED IN THE EMISSION MODE) L4M04350
4200 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,/,1X, L4M04360
1 49HPROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE) L4M04370
4300 FORMAT (/,10X,38H EQUIVALENT SEA LEVEL ABSORBER AMOUNTS L4M04380
1/,21X,56HWATER VAPOUR CO2 ETC. OZONE NITROGEN (CONT), L4M04390
2 42H H2O (CONT) MOL SCAT OZONE(U-V)/24X, L4M04400
3 7HGM CM-2,10X,2HKM,10X,6HATM CM,10X,2HKM,9X,7HGM CM-2, L4M04410
4 10X,2HKM,10X,6HATM CM) L4M04420
4400 FORMAT(/10X,10H W(1-6,8)=,7(E14.3)/,1X,10X,7H W(10)=,58X,E14.3/) L4M04430
4401 FORMAT(/,23X,11HNITRIC ACID,8X,3HSO2,11X,3HNH3,11X,3HNO2/) L4M04440
4500 FORMAT(/10X,10H W(11-16)=,6(E14.3)/) L4M04450
4600 FORMAT (1H1,30X,28HRADIANCE(WATTS/CM2-STER-XXX)) L4M04460
4700 FORMAT (1H ,10X,37HFR(CM-1) WVL(MICRON) PER CM-1 L4M04470
1 10HPER MICRON,26H INTEGRAL TRANS,1X,4(1H-), L4M04480
2 11H AERO TRAN ,4(1H-),/,1X,84X,17H EXTN ABS) L4M04490
4800 FORMAT (1H1,/,1X,2X,15HFREQ WAVELENGTH,2X,3HH2O,3X,4HCO2+,4X, L4M04500
1 30HOZONE N2 C H2O C MOL S ,1X, L4M04510
2 22HNITRIC SO2 HNO3,4X,16HNO2 INTEGRATED, L4M04520
3 2X,13HTOTAL AEROSOL/1X,1X,13H CM-1 MICRONS,10(3X,5HTRANS), L4M04530
4 2X,24HABSORPTION TRANS TRANS) L4M04540
4900 FORMAT (1H ,10X,F6.1,F13.6,3E13.5,F13.6,1X,F7.5,3X,F7.5) L4M04550
5000 FORMAT (1H ,16,11F8.4,F11.4,F8.4,1X,F7.5) L4M04560
5100 FORMAT (1H0,8H RADMIN ,F12.3,E12.5,/,8H RADMAX ,F12.3, L4M04570
1 E12.5) L4M04580
5200 FORMAT (1H0,26H INTEGRATED ASORPTION FROM,15,4H TO ,15, L4M04590
1 7H CM-1 =,F10.2,25H, AVERAGE TRANSMITTANCE =,F6.4) L4M04600
5300 FORMAT (1H ,22H INTEGRATED RADIANCE =,E12.5,13H WATT CM -2 ,2HSR) L4M04610
5400 FORMAT (2F10.3,12) L4M04620
5500 FORMAT (/,10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO , L4M04630
1 4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1 ( L4M04640
2 ,F6.2,3H - ,F5.2,10H MICRONS )) L4M04650
END L4M04660

```

```

SUBROUTINE ABSORB(IXY, IERR, W, V1, V2, DV, SUMA, MULDV, ANGLE, LEN, ITYPE, ABS00010
1 COMMON H1, H2, MODEL, ISPOT1, RANGE, BETA, VIS, ICLMAT, IV1, IV2, IDV) ABS00020
COMMON /GEOMET/PTS(15), IGEOSW ABS00030
COMMON /CLYMAT/TEMP, PRESS, RH1, AH1, DP1, VIS1, CLDAMT, ABS00040
1 CLDHYT, FQGPRB, WNDVEL, WNDDIR, IPASCT ABS00050
COMMON /CONST/PI, PI2, CA, TWOPI, TORRMB, CDEGK ABS00060
COMMON /IUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU ABS00070
COMMON /MO1/EH(16, 34), P(34), T(34), WH(34), Z(34), WAC(34), RE, M, NL ABS00080
COMMON /MO2/ WO(34), RO, TBOUND, JP, IM, ML, IP, JSTOR ABS00090
COMMON /MO9/RADMAX, RADMIN, VRMAX, VRMIN ABS00100
COMMON /EM1/HMIN, KMAX, IJ, J1, J2, JMIN, JEXTRA, NP1 ABS00110
COMMON /LQWEX/WPATH(68, 16), WLAY(34, 16), TBBY(68), TX(16), BETAEX, ABS00120
1 CLDHGT, NCLD ABS00130
COMMON /SPOTLO/ISPOT, LOREAD, N16 ABS00140
COMMON /MO3/ FS(9), S1(9), S2(9), FNH3(9), FH1(9), FH2(9), FNO2(9), ABS00150
1 O1(9), O2(9), PPMO2, PPMNH3, PPMNO2 ABS00160
LOGICAL ISPOT, N16, LOREAD ABS00170
DIMENSION VH(16), W(16), E(16) ABS00180
EH(7, 1) REPLACES HSTOR ABS00190
EH(9, 1) REPLACES HMIX ABS00200
DATA (EH(9, I), I=1, 34) /9*0., 0.1, 0.33, 0.8, 1.2, 1.4, 1.6, 1.8, 1.9, ABS00210
1 2.0, 2.1, 2.3, 3.0, 3.7, 4.2, 5.2, 6.0, 3.8, 2.6, 0.22, 6*0.0, ABS00220
F(A)=EXP(-18.9766-14.9595*A-2.43882*A*A)*A ABS00230
TMPVIS=VIS ABS00240
IF (ISPOT1.EQ.1) GO TO 200 ABS00250
IF (MODEL.EQ.0) GO TO 400 ABS00260
IF (IXY.EQ.3) GO TO 100 ABS00270
IF (M.EQ.7.AND.IM.NE.0) GO TO 400 ABS00280
IF (IXY.GT.3) GO TO 1500 ABS00290
WHEN IXY=0 VIS IS READ IN MAIN ABS00300
100 IF (.NOT.ISPOT) READ (IOIN, 6200) H1, H2, ANGLE, RANGE, ABS00310
1 BETA, VIS, CLDHGT ABS00320
IF (IGEOSW.NE.1) GO TO 111 ABS00330
H1=PTS(3) ABS00340
H2=PTS(6) ABS00350
RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+ ABS00360
+(PTS(3)-PTS(6))**2) ABS00370
111 CONTINUE ABS00380
IF (IXY.EQ.0) VIS=TMPVIS ABS00390
200 X1=RE+H1 ABS00400
X2=RE+H2 ABS00410
IF (ITYPE.EQ.3) GO TO 1000 ABS00420
IF (ITYPE.EQ.1) GO TO 1500 ABS00430
IF (RANGE.EQ.0.) GO TO 1200 ABS00440
IF (.NOT.ISPOT) WRITE (IOOUT, 6300) H1, H2, ANGLE, RANGE, ABS00450
1 BETA, VIS ABS00460
IF (H2.EQ.0.AND.ANGLE.NE.0) GO TO 300 ABS00470
ANGLE=ACOS(0.5*((H2-H1)*(1.+X2/X1)/RANGE-RANGE/X1))/CA ABS00480
GO TO 1400 ABS00490
300 X2=SQRT((X1/RANGE+RANGE/X1+2.0*COS(ANGLE*CA))*X1* ABS00500
1 RANGE) ABS00510
H2=X2-RE ABS00520
GO TO 1400 ABS00530
400 CONTINUE ABS00540
IF (ML.LE.0) ML=1 ABS00550
DO 900 K=1, ML ABS00560
C CLIMATE OPTION - SEE COMMON /CLYMAT/ ABS00570
IF (M.EQ.0.AND..NOT.ISPOT) READ (IOIN, 6400) H1, P(1), ABS00580
1 TMP, DP, RH, WH(K), WO(K), VIS, RANGE ABS00590
IF (IGEOSW.NE.1) GO TO 444 ABS00600
H1=PTS(3) ABS00610
H2=PTS(6) ABS00620
RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+ ABS00630
+(PTS(3)-PTS(6))**2) ABS00640
444 CONTINUE ABS00650
IF (ICLMAT.NE.1) GO TO 500 ABS00660
TMP=TEMP ABS00670
P(1)=PRESS ABS00680
DP=DP1 ABS00690
RH=RH1 ABS00700

```

```

WH(K)=AH1
VIS=VIS1
500 IF (IXY.EQ.0.AND.ICLMAT.NE.1) VIS=TMPVIS
IF ((M.GT.0).AND.(.NOT.ISPOT)) READ (IOIN,6400) Z(K),
1 P(K),TMP,DP,RH,WH(K),WO(K)
IF ((M.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,6500) H1,
1 P(1),TMP,DP,RH,WH(K),WO(K),VIS,RANGE
IF (M.EQ.0) Z(K)=H1
J=IFIX(Z(K)+1.0E-6)+1
IF (Z(K).GE.25.) J=(Z(K)-25.)/5.0+26.
IF (Z(K).GE.50.0) J=(Z(K)-50.)/20.+31.
IF (Z(K).GE.70.) J=(Z(K)-70.)/30.+32.
IF (J.GT.33) J=33
FAC=Z(K)-FLOAT(J-1)
IF (J.LT.26) GO TO 600
FAC=(Z(K)-5.0*FLOAT(J-26)-25.)/5.
IF (J.GE.31) FAC=(Z(K)-50.)/20.
IF (J.GE.32) FAC=(Z(K)-70.)/30.
IF (FAC.GT.1.0) FAC=1.0
600 L=J+1
T(K)=TMP+CDEGK
TT=CDEGK/T(K)
IF (RH.LE.0.0) TT=CDEGK/(CDEGK+DP)
IF (WH(K).LE.0.0) WH(K)=F(TT)
IF (RH.GT.0.0) WH(K)=0.01*RH*WH(K)
EH(7,K)=0.0
IF (EH(9,J).LE.0.) GO TO 700
EH(7,K)=EH(9,J)*(EH(9,L)/EH(9,J))**FAC
700 CONTINUE
IF (MODEL.EQ.0) GO TO 1500
IF ((K.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,6600)
IF (.NOT.ISPOT) WRITE (IOOUT,6400) Z(K),P(K),TMP,DP,
1 RH,WH(K),WO(K)
900 CONTINUE
IM=0
NL=ML
C NOTE THAT Z(I) MAY NOT CORRESPOND TO THE VALUES GIVEN FOR STANDARD
C MODEL ATMOSPHERES
IF (IXY.GE.3) GO TO 1500
GO TO 100
1000 IF (RANGE.GT.0.0) GO TO 1100
GO TO 1500
1100 ITYPE=2
BETA=ACOS(0.5*(RANGE*RANGE/(X1*X2)-X2/X1-X1/X2))/CA
1200 IF (BETA.EQ.0.) GO TO 1300
BET=CA*BETA
X2=RE+H2
ANGLE=ATAN(X2*SIN(BET)/(X2*COS(BET)-X1))/CA
IF (ANGLE.LT.0.) ANGLE=ANGLE+PI
RANGE=X2*SIN(BET)/SIN(ANGLE*CA)
BET=BETA
GO TO 1500
1300 RANGE=(X2/X1)**2-(SIN(ANGLE*CA))**2
IF (RANGE.GE.0.0) RANGE=X1*(SQRT(RANGE)-ABS(COS(ANGLE*CA)))
1400 IF (ANGLE.NE.0.0,OR,ANGLE.NE.180.) BET=ASIN(RANGE*SIN(ANGLE*CA)/X2)
IF (ANGLE.LT.0.) ANGLE=ANGLE+PI
IF (RANGE.LT.0.0) RANGE=-RANGE
BET=BET/CA
IF (.NOT.ISPOT) WRITE (IOOUT,6300) H1,H2,ANGLE,RANGE.
1 BET,VIS
1500 CONTINUE
DO 1600 I=1,NL
DO 1600 J=1,KMAX
1600 WLAY(I,J)=0.
SUMA=0.
C WHEN IXY=0 V1,V2,MULDY ARE READ IN EOSAEL.MAIN
IF ((IXY.EQ.1,OR,IXY.EQ.2).AND.(.NOT.ISPOT))
1 READ (IOIN,6250) V1,V2,MULDY
IF ((IXY.EQ.1,OR,IXY.EQ.2).AND.(.NOT.ISPOT))
+ CALL CKER (V1,V2,DV,IV1,IV2,IOV,IERR,MULDY,ISFJT,TMPVIS)
ABS00710
ABS00720
ABS00730
ABS00740
ABS00750
ABS00760
ABS00770
ABS00780
ABS00790
ABS00800
ABS00810
ABS00820
ABS00830
ABS00840
ABS00850
ABS00860
ABS00870
ABS00880
ABS00890
ABS00900
ABS00910
ABS00920
ABS00930
ABS00940
ABS00950
ABS00960
ABS00970
ABS00980
ABS00990
ABS01000
ABS01010
ABS01020
ABS01030
ABS01040
ABS01050
ABS01060
ABS01070
ABS01080
ABS01090
ABS01100
ABS01110
ABS01120
ABS01130
ABS01140
ABS01150
ABS01160
ABS01170
ABS01180
ABS01190
ABS01200
ABS01210
ABS01220
ABS01230
ABS01240
ABS01250
ABS01260
ABS01270
ABS01280
ABS01290
ABS01300
ABS01310
ABS01320
ABS01330
ABS01340
ABS01350
ABS01360
ABS01370
ABS01380
ABS01390
ABS01400

```

```

IF (IERR.EQ.1) RETURN
IF ((ITYPE.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,6700)
1 H1,RANGE
IF ((ITYPE.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,6800)
1 H1,H2,ANGLE
IF ((ITYPE.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,6900)
1 H1,ANGLE
IF (MODEL.EQ.0) M=7
IF ((M.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,7200) M
IF ((M.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,7300) M
IF ((M.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,7400) M
IF ((M.EQ.4).AND.(.NOT.ISPOT)) WRITE (IOOUT,7500) M
IF ((M.EQ.5).AND.(.NOT.ISPOT)) WRITE (IOOUT,7600) M
IF ((M.EQ.6).AND.(.NOT.ISPOT)) WRITE (IOOUT,7700) M
IF ((M.EQ.8).AND.(.NOT.ISPOT)) WRITE (IOOUT,7800) M
IF ((M.EQ.9).AND.(.NOT.ISPOT)) WRITE (IOOUT,7900) M
AVW=10000./V1
ALAM=10000./V2
RADMIN=1.0E+38
RADMAX=0.
VRMIN=0.
VRMAX=0.
IF (.NOT.ISPOT) WRITE (IOOUT,8000) V1,V2,DV,ALAM,AVW
AVW=0.5E-4*(V1+V2)
AVW=AVW*AVW
IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8100)
IF (ITYPE.EQ.1) GO TO 2100
DO 1800 K=1,KMAX
VH(K)=0.0
1800 CONTINUE
BETA=0.0
SR=0.0
IP=0
C**** NOW DEFINE CONSTANT PRESSURE PATH QUANTITIES EH(1-8)
Y=CA*ANGLE
SPHI=SIN(Y)
R1=(RE+H1)*SPHI
IF (H1.GT.Z(NL)) GO TO 1900
GO TO 2100
1900 X=(RE+Z(NL))/(RE+H1)
IF (SPHI.GT.X) GO TO 2000
H1=Z(NL)
J1=NL
SPHI=SPHI/X
ANGLE=180.0-ASIN(SPHI)/CA
R1=(RE+H1)*SPHI
GO TO 2100
2000 HMIN=R1-RE
IF (.NOT.ISPOT) WRITE (IOOUT,8200) HMIN
GO TO 6000
2100 DO 2400 I=1,NL
PS=P(I)/1013.0
TS=CDEGK/T(I)
X=PS*TS
C--- COMPUTE MASS DENSITY (G M-3) FROM IDEAL GAS LAW ---
C--- 1292.02 = DENSITY OF STANDARD COMPOSITION AIR AT STP ---
WA(I) = 1292.02*X
PT=PS*SQR(TS)
D=0.1*W(I)
EH(1,I)=D*PT**0.9
EH(2,I)=X*PT**0.75
EH(4,I)=0.8*PT*X
PPW=4.56E-5*D*CDEGK/TS
TS1=(296.0/CDEGK)*TS
EH(5,I)=D*PPW*EXP(6.08*(TS1-1.0))+0.002*D*(PS-PPW)
EH(10,I)=D*(PPW+0.12*(PS-PPW))*EXP(4.56*(TS1-1.0))
EH(6,I)=X
EH(8,I)=46.6667*W(I)
EH(3,I)=EH(8,I)*PT**0.4
C EH(11,I)=HN03 ABSORBER AMOUNT (ATM-CM)/KM

```

```

ABS01410
ABS01420
ABS01430
ABS01440
ABS01450
ABS01460
ABS01470
ABS01480
ABS01490
ABS01500
ABS01510
ABS01520
ABS01530
ABS01540
ABS01550
ABS01560
ABS01570
ABS01580
ABS01590
ABS01600
ABS01610
ABS01620
ABS01630
ABS01640
ABS01650
ABS01660
ABS01670
ABS01680
ABS01690
ABS01700
ABS01710
ABS01720
ABS01730
ABS01740
ABS01750
ABS01760
ABS01770
ABS01780
ABS01790
ABS01800
ABS01810
ABS01820
ABS01830
ABS01840
ABS01850
ABS01860
ABS01870
ABS01880
ABS01890
ABS01900
ABS01910
ABS01920
ABS01930
ABS01940
ABS01950
ABS01960
ABS01970
ABS01980
ABS01990
ABS02000
ABS02010
ABS02020
ABS02030
ABS02040
ABS02050
ABS02060
ABS02070
ABS02080
ABS02090
ABS02100

```

```

EH(11,I)=PS*TS*EH(9,I)*1.0E-04
IF (MODEL.EQ.0.OR.MODEL.EQ.7) EH(11,I)=PS*TS*EH(7,I)*1.0E-04
C*****EH(12,I) = SO2 ABSORBER AMOUNT (ATM-CM)/KM
EH(12,I)=0.772E-04*PPMSO2*WA(I)*PS**0.07122*TS**0.06159
EH(13,I)=0.772E-04*PPMNH3*WA(I)*PS**0.52125*TS**(-0.60438)
EH(14,I)=0.772E-04*PPMNO2*WA(I)*PS**0.18066*TS**0.20911
C***K=15 FOR ASL 3.3 - 4.3 MICRON H2O CONTINUUM
EH(15,I)=PPW*0
C***K=16 FOR 4.6 - 4.8 MICRON H2O CONTINUUM
EH(16,I)=PPW*(PPW+3.0E-03*(PS-PPW))
IF (I.EQ.NL) GO TO 2300
IF (MODEL.EQ.0.AND.I.GE.1) GO TO 3600
T2=T(I+1)
W2=W(I+1)
PPW=4.56E-6*W2*T2
2300 IF (HI.GE.Z(I)) J1=I
IF ((JP.EQ.0).AND.(.NOT.ISPOT))
1WRITE (IOOUT,8300)I,Z(I),(EH(K,I),K=1,6),EH(8,I),(EH(K,I),K=10,14)
2400 CONTINUE
X1=H1
CALL POINT (H1,N,NP1,TX)
J1=N
DO 2500 K=1,KMAX
2500 E(K)=TX(K)
JEXTRA=0
JMIN=0
C**ITYPE=1 MEANS HORIZONTAL PATH *****
IF (ITYPE.EQ.1) GO TO 3600
IF (ITYPE.EQ.3) H2=Z(NL)
C** ANGLE GREATER THAN 90 DEGREES MEANS DOWNWARD TRAJECTORY *****
IF (ANGLE.GT.90.0) GO TO 3800
C** IF THE PATH IS NOT HORIZONTAL OR DOWNWARD THEN IT IS UPWARD TRAJECTORY
2600 IF (ANGLE.GT.90.0.AND.NP1.GT.0) J1=J1+1
J2=NL
IF (ITYPE.EQ.3) GO TO 2700
CALL POINT (H2,N,NP,TX)
J2=N
IF (NP.GT.0) J2=J2-1
2700 DO 2800 K=1,KMAX
IF (K.EQ.9.OR.K.EQ.7) GO TO 2800
EH(K,J1)=E(K)
IF (ITYPE.EQ.3) GO TO 2800
EH(K,J2+1)=TX(K)
2800 CONTINUE
C***** NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)
IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8400)
DO 2900 K=1,KMAX
2900 W(K)=0.
DO 3500 I=J1,J2
X1=Z(I)
IF (I.LT.NL) X2=Z(I+1)
IF (I.EQ.NL) X2=Z(I)
IF (I.EQ.J1) X1=H1
IF (I.EQ.J2) X2=H2
DZ=X2-X1
IF (I.EQ.NL) DZ=Z(I)-Z(I-1)
DS=DZ
C***** UPWARD TRAJECTORY
RX=(RE+X1)/(RE+X2)
THETA=ASIN(SPHI)/CA
PHI=ASIN(SPHI*RX)/CA
BET=THETA-PHI
SALP=RX*SPHI
IF (SPHI.GT.1.E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI
BETA=BETA+BET
PSI=BETA+PHI-ANGLE
PHI=180.-PHI
SR=SR+DS
JEXTRA=0
DO 3400 K=1,KMAX

```

```

ABS02110
ABS02120
ABS02130
ABS02140
ABS02150
ABS02160
ABS02170
ABS02180
ABS02190
ABS02200
ABS02210
ABS02220
ABS02230
ABS02240
ABS02250
ABS02260
ABS02270
ABS02280
ABS02290
ABS02300
ABS02310
ABS02320
ABS02330
ABS02340
ABS02350
ABS02360
ABS02370
ABS02380
ABS02390
ABS02400
ABS02410
ABS02420
ABS02430
ABS02440
ABS02450
ABS02460
ABS02470
ABS02480
ABS02490
ABS02500
ABS02510
ABS02520
ABS02530
ABS02540
ABS02550
ABS02560
ABS02570
ABS02580
ABS02590
ABS02600
ABS02610
ABS02620
ABS02630
ABS02640
ABS02650
ABS02660
ABS02670
ABS02680
ABS02690
ABS02700
ABS02710
ABS02720
ABS02730
ABS02740
ABS02750
ABS02760
ABS02770
ABS02780
ABS02790
ABS02800

```

	IF (K.EQ.7.OR.K.EQ.9) GO TO 3100	ABS02810
	EV=DS*EH(K,I)	ABS02820
	IF (I.EQ.NL) GO TO 3000	ABS02830
	IF (EH(K,I).EQ.0.0.OR.EH(K,I+1).EQ.0.0) GO TO 3100	ABS02840
	IF (EH(K,I).EQ.EH(K,I+1)) GO TO 3200	ABS02850
	EV=DS*(EH(K,I)-EH(K,I+1))/ALOG(EH(K,I)/EH(K,I+1))	ABS02860
	GO TO 3200	ABS02870
3000	IF (EH(K,I).EQ.0.0) GO TO 3100	ABS02880
	IF (EH(K,I-1).EQ.0.0) GO TO 3100	ABS02890
	IF (EH(K,I).EQ.EH(K,I-1)) GO TO 3200	ABS02900
	EV=EV/ALOG(EH(K,I-1)/EH(K,I))	ABS02910
	GO TO 3200	ABS02920
3100	EV=0.	ABS02930
3200	VH(K)=VH(K)+EV	ABS02940
	IF (I.EQ.JSTOR) GO TO 3300	ABS02950
	W(LAY(I,K))=EV+W(K)	ABS02960
	W(K)=0.	ABS02970
	GO TO 3400	ABS02980
3300	W(K)=EV	ABS02990
	IF (J1.NE.J2) GO TO 3400	ABS03000
	W(LAY(J2+1,K))=W(K)	ABS03010
	W(K)=0.	ABS03020
	JEXTRA=1	ABS03030
3400	CONTINUE	ABS03040
	IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8500) I,	ABS03050
	1 X1,(VH(L),L=1,6),VH(8),PSI,PHI,BETA,THETA,SR	ABS03060
	IF (I.GE.NL) GO TO 3500	ABS03070
	SPHI=SPHI+RX	ABS03080
	IF (SALP.GE.1.) SPHI=SALP	ABS03090
3500	CONTINUE	ABS03100
	GO TO 5800	ABS03110
C****	HORIZONTAL PATH	ABS03120
3600	DO 3700 K=1,KMAX	ABS03130
	IF (K.EQ.7.OR.K.EQ.9) GO TO 3700	ABS03140
	W(K)=RANGE*EH(K,1)	ABS03150
	IF (MODEL.GT.0) W(K)=RANGE*TX(K)	ABS03160
	VH(K)=W(K)	ABS03170
3700	CONTINUE	ABS03180
	GO TO 6100	ABS03190
3800	CONTINUE	ABS03200
C****	DOWNWARD TRAJECTORY	ABS03210
	K2=0	ABS03220
	IF (NP1.EQ.1) J1=J1-1	ABS03230
	IF (J1.LE.0) J1=1	ABS03240
	J2=J1+1	ABS03250
	J=J1+1	ABS03260
	IF (H2.GT.Z(J1+1).OR.H1.EQ.H2) GO TO 4000	ABS03270
	IF (NP1.EQ.1.AND.H2.GE.Z(J1+1)) GO TO 4000	ABS03280
	CALL POINT (H2,N,NP2,TX)	ABS03290
	DO 3900 K=1,KMAX	ABS03300
3900	W(K)=TX(K)	ABS03310
	IF (H2.LT.H1) H=H2	ABS03320
	J2=N	ABS03330
4000	A0=(RE+H1)*SPHI	ABS03340
	DO 4100 I=1,J1	ABS03350
	HMIN=A0-RE	ABS03360
	JMIN=1	ABS03370
	IF (HMIN.LE.Z(I+1)) GO TO 4200	ABS03380
4100	CONTINUE	ABS03390
4200	X=HMIN	ABS03400
	IF (HMIN.LE.0) GO TO 4400	ABS03410
	CALL POINT (X,N,NP,TX)	ABS03420
	JMIN=N	ABS03430
	HMIN=A0-RE	ABS03440
	IF (ABS(X-HMIN).GT.0.0001) GO TO 4200	ABS03450
	IF (H2.GE.H1) J2=N	ABS03460
	IF (H2.GE.H1.OR.H2.LT.HMIN) H=HMIN	ABS03470
	IF (.NOT.ISPOT) WRITE (IOOUT,8600) HMIN	ABS03480
	IF (H2.LT.HMIN) J2=N	ABS03490
	IF ((H2.LT.HMIN).AND.(.NOT.ISPOT)) WRITE (IOOUT,8700) HMIN	ABS03500


```

GO TO 4500
4400 IF (.NOT.ISPOT) WRITE (IOOUT,8600) HMIN
      IF (H2.LT.H1) GO TO 4500
      IF ((ITYPE.EQ.3.OR.H2.GE.H1).AND.(.NOT.ISPOT))
1      WRITE (IOOUT,8800)
      ITYPE=2
      JMIN=0
      J2=1
      H2=0.0
      H=0.0
C**** NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)
4500 IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8400)
      JSTOR=J-1
      DO 5100 I=1,NL
      J=J-1
      IF (I.NE.1) X1=Z(J+1)
      X2=Z(J)
      IF (J.EQ.J2.AND.K2.EQ.0) X2=H
      IF (J.EQ.JMIN.AND.K2.EQ.1) X2=HMIN
      HM=(RE+X1)*SPHI-RE
      IF (HM.GT.Z(J).AND.HM.GT.X2) X2=HM
      RX=(RE+X1)/(RE+X2)
      DS=X1-X2
      ALP=90.0
      THET=ASIN(SPHI)/CA
      SALP=RX*SPHI
      IF (ABS(2-HM).GT.1.0E-5) ALP=ASIN(SALP)/CA
      BET=ALP-THET
      IF (SPHI.GT.1.0E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI
      THETA=180.0-THET
      BETA=BETA+BET
      PSI=BETA-ALP-ANGLE+180.0
      SR=SR+DS
      DO 5000 K=1,KMAX
      IF (K.EQ.7.OR.K.EQ.9) GO TO 5000
      AJ=EH(K,J)
      BJ=EH(K,J+1)
      IF (J.EQ.J1) BJ=E(K)
      IF (J.EQ.J2.AND.H2.LT.H1.AND.H2.GT.0.0) AJ=W(K)
      IF (J.EQ.JMIN.AND.H2.GE.H1) AJ=TX(K)
      IF (J.EQ.JMIN.AND.ABS(H2-HM).LT.1.0E-5) AJ=TX(K)
      IF (K2.EQ.0) GO TO 4600
      IF (J.EQ.J2) BJ=W(K)
      IF (J.EQ.JMIN) AJ=TX(K)
4600 IF (AJ.EQ.0.0.OR.BJ.EQ.0.0) GO TO 4800
      IF (AJ.EQ.BJ) GO TO 4700
      EV=DS*(AJ-BJ)/ALOG(AJ/BJ)
      GO TO 4900
4700 EV=DS*AJ
      GO TO 4900
4800 EV=0.0
4900 VH(K)=VH(K)+EV
5000 WLAY(J,K)=EV
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8500) J,
1      X1,(VH(L),L=1,6),VH(8),PSI,ALP,BETA,THETA,SR
      IF (J.EQ.J2.AND.H2.GE.H1) GO TO 5600
      IF (J.EQ.JMIN.AND.K2.EQ.1) GO TO 5400
      SPHI=SALP
      IF (J.EQ.J2.AND.K2.EQ.0) GO TO 5200
5100 CONTINUE
5200 IF (HMIN.LE.0) GO TO 5800
      IF ((LEN.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8900)
      IF ((LEN.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,9000)
      IF (LEN.EQ.0) GO TO 5800
      K2=1
      X1=X2
      IF (ABS(X1-HMIN).LE.0.001) GO TO 5800
      H=HMIN
      J=J2+1
      IF (NP2.EQ.1) J=J-1

```

```

ABS03510
ABS03520
ABS03530
ABS03540
ABS03550
ABS03560
ABS03570
ABS03580
ABS03590
ABS03600
ABS03610
ABS03620
ABS03630
ABS03640
ABS03650
ABS03660
ABS03670
ABS03680
ABS03690
ABS03700
ABS03710
ABS03720
ABS03730
ABS03740
ABS03750
ABS03760
ABS03770
ABS03780
ABS03790
ABS03800
ABS03810
ABS03820
ABS03830
ABS03840
ABS03850
ABS03860
ABS03870
ABS03880
ABS03890
ABS03900
ABS03910
ABS03920
ABS03930
ABS03940
ABS03950
ABS03960
ABS03970
ABS03980
ABS03990
ABS04000
ABS04010
ABS04020
ABS04030
ABS04040
ABS04050
ABS04060
ABS04070
ABS04080
ABS04090
ABS04100
ABS04110
ABS04120
ABS04130
ABS04140
ABS04150
ABS04160
ABS04170
ABS04180
ABS04190
ABS04200

```

```

B=BETA
PH=180.0-ASIN(SPHI)/CA
TS=180.0-ASIN(SPHI)/CA
DO 5300 K=1,KMAX
E(K)=VH(K)
GO TO 4500
5300 BETA=2.*BETA-B
PSI=2.*PSI-PS
SR=2.*SR-TS
C LONG PATH TAKEN
PHI=PH
DO 5500 K=1,KMAX
VH(K)=2.*VH(K)-E(K)
GO TO 5800
5600 DO 5700 K=1,KMAX
VH(K)=2.0*VH(K)
BETA=2.0*BETA
SR=2.0*SR
IF (H2.EQ.H1) GO TO 5800
SPHI=SIN(ANGLE*CA)
GO TO 2600
5800 CONTINUE
DO 5900 K=1,KMAX
W(K)=VH(K)
5900 CONTINUE
6000 CONTINUE
6100 RETURN
C
6200 FORMAT (7F10.3)
6250 FORMAT (2F10.3,12)
6300 FORMAT (1H0.9X,4H H1=,F7.3,6HKM,H2=,F7.3,9HKM,ANGLE=,
1 F8.4,13HGEOM. RANGE =,F7.2,8HKM,BETA=,F8.5,
2 5H,VIS=,F6.1)
6400 FORMAT (3F10.3,2F5.1,2E10.3,2F10.3)
6500 FORMAT (10X,26HINPUT METEOROLOGICAL DATA:/10X,2HZ=,
1 F7.2,7H KM, P=,F7.2,6H MB,T=,F5.1,8H C, DEW ,
2 7HPT,TEMP,F5.1,17H C, REL HUMIDITY=,F5.1,
3 16H %, H2O DENSITY=,1PE9.2,7H GM M-3/10X,
4 15H OZONE DENSITY=,E9.2,20H GM-3, VISUAL RANGE=
5 2OPF6.1,10H KM,RANGE=,F10.3,4H KM )
6600 FORMAT (24H MODEL ATMOSPHERE NO. 7,/4X,6HZ (KM),3X,
1 6HP (MB),4X,30HT (C) DEW PT %RH H2O(GM.M-3) ,
2 19H03(GM.M-3) NO. DEN.)
6700 FORMAT (/10X,28H HORIZONTAL PATH, ALTITUDE =,F7.3,
1 11H KM,RANGE =,F7.3,3H KM)
6800 FORMAT (/10X,37H SLANT PATH BETWEEN ALTITUDES H1 AND ,
1 13HH2 WHERE H1 =,F7.3,8H KM H2 =,F7.3,
2 18H KM,ZENITH ANGLE =,F7.3,8H DEGREES)
6900 FORMAT (/10X,39H SLANT PATH TO SPACE FROM ALTITUDE H1 =
1 ,F7.3,19H KM, ZENITH ANGLE =,F7.3,
2 8H DEGREES)
7200 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,11H = TROPICAL)
7300 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,
1 21H = MIDLATITUDE SUMMER)
7400 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,
1 21H = MIDLATITUDE WINTER)
7500 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,14H = SUB-ARCTIC ,
1 7HSUMMER )
7600 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,14H = SUB-ARCTIC ,
1 7HWINTER )
7700 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,11H = 1962 US ,
1 10HSTANDARD )
7800 FORMAT(/20X,18H MODEL ATMOSPHERE ,11,20H = ISRAELI STANDARD ,
1 16H(YEAR, DAYTIME) )
7900 FORMAT(/20X,18H MODEL ATMOSPHERE ,11,20H = ISRAELI STANDARD ,
1 18H(YEAR, NIGHTTIME) )
8000 FORMAT (/10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO ,
1 4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1 (
2 ,F6.2,3H - ,F5.2,10H MICRONS ))
ABS04210
ABS04220
ABS04230
ABS04240
ABS04250
ABS04260
ABS04270
ABS04280
ABS04290
ABS04300
ABS04310
ABS04320
ABS04330
ABS04340
ABS04350
ABS04360
ABS04370
ABS04380
ABS04390
ABS04400
ABS04410
ABS04420
ABS04430
ABS04440
ABS04450
ABS04460
ABS04470
ABS04480
ABS04490
ABS04500
ABS04510
ABS04520
ABS04530
ABS04540
ABS04550
ABS04560
ABS04570
ABS04580
ABS04590
ABS04600
ABS04610
ABS04620
ABS04630
ABS04640
ABS04650
ABS04660
ABS04670
ABS04680
ABS04690
ABS04700
ABS04710
ABS04720
ABS04730
ABS04740
ABS04750
ABS04760
ABS04770
ABS04780
ABS04790
ABS04800
ABS04810
ABS04820
ABS04830
ABS04840
ABS04850
ABS04860
ABS04870
ABS04880
ABS04890
ABS04900

```

8100	FORMAT	(1H1,///10X,20H HORIZONTAL PROFILES)	ABS04910
8200	FORMAT	(38H TRAJECTORY MISSES EARTHS ATMOSPHERE,	ABS04920
	1	31HCLOSEST DISTANCE OF APPROACH IS,F10.2,1X,/ 1X,18HEND OF CALCULATION)	ABS04930
	2		ABS04940
8300	FORMAT	(1X,14,F6.1,12(E9.3))	ABS04950
8400	FORMAT	(1H1,///10X,21H VERTICAL PROFILES ,53X,3HPSI, 6X,3HPHI,6X,4HBETA,3X,13HTHETA RANGE)	ABS04960
	1		ABS04970
8500	FORMAT	(13,F6.1,7E10.3,4F9.4,F6.1)	ABS04980
8600	FORMAT	(8H HMIN = ,F10.3)	ABS04990
8700	FORMAT	(40H H2 WAS SET LESS THAN HMIN AND HAS BEEN , 34HRESET EQUAL TO HMIN I.E. H2 = ,F10.3)	ABS05000
	1		ABS05010
8800	FORMAT	(41H PATH INTERSECTS EARTH - PATH CHANGED TO , 23HTYPE 2 WITH H2 = 0.0 KM)	ABS05020
	1		ABS05030
8900	FORMAT	(36H CHOICE OF TWO PATHS FOR THIS CASE - , 42HSHORTEST PATH TAKEN. FOR LONGER PATH SET , 6HLEN=1.)	ABS05040
	1		ABS05050
	2		ABS05060
9000	FORMAT	(44H CHOICE OF TWO PATHS FOR THIS CASE -LONGEST , 41HPATH TAKEN. FOR SHORT PATH SET LEN = 0)	ABS05070
	1		ABS05080
		END	ABS05090

```

SUBROUTINE CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN)      CKR00010
LOGICAL ISPOT                                                    CKR00020
COMMON /IGUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU CKR00030
IERR=0                                                            CKR00040
IF (MULDV.LE.0) MULDV=1                                         CKR00050
C CHECK FOR PROPER WINDOW REGION                                CKR00060
IF (V2.LE.830.0) GO TO 1090                                     CKR00070
IF (V1.GE.1250.0.AND.V2.LE.2010.0) GO TO 1090                CKR00080
IF (V1.GE.3330.0.AND.V2.LT.5010.0) GO TO 1090                CKR00090
IF (V1.GE.39990.0) GO TO 1090                                  CKR00100
GO TO 1100                                                       CKR00110
1090 WRITE(IOOUT,3190)                                           CKR00120
TRAN=1.                                                           CKR00130
IERR=1.                                                           CKR00140
RETURN                                                            CKR00150
C CHECK FOR PROPER INTEGER VALUES                               CKR00160
1100 IV1=20*FIX((V1-10.0)/20.0)+10                               CKR00170
IV2=20*FIX((V2-10.0)/20.0+0.99)+10                               CKR00180
IF (IV1.LT.830) IV1=830                                         CKR00190
IF (IV1.LT.1250.AND.IV2.GT.1250) IV2=1250                     CKR00200
IF (IV1.LT.2010.AND.IV2.GT.2010) IV1=2010                     CKR00210
IF (IV1.LT.3330.AND.IV2.GT.3330) IV2=3330                     CKR00220
IF (IV1.LT.5010.AND.IV2.GT.5010) IV1=5010                     CKR00230
IF (IV1.LT.39990.AND.IV2.GT.39990) IV2=39990                  CKR00240
V1=FLOAT(IV1)                                                    CKR00250
V2=FLOAT(IV2)                                                    CKR00260
DV=20.*FLOAT(MULDV)                                              CKR00270
IWVCK=0                                                           CKR00280
IDV=FIX(DV)                                                       CKR00290
C WHEN CALLED FROM SPOT CHECK FOR MORE THAN 15 DIVISIONS      CKR00300
C (16 WAVENUMBER VALUES) WHICH IS ARRAY SIZE.                CKR00310
C CHECK HERE FOR ROUNDING PROBLEMS ON V1, V2 CAUSING          CKR00320
C TOO SMALL AN INCREMENT                                       CKR00330
IF (ISPOT.AND.(FLOAT(IV2-IV1)/15..GT.(FLOAT(IDV)+.001))) IWVCK=1 CKR00340
IF (IWVCK.NE.1) GO TO 91                                         CKR00350
CKDV=FLOAT(IV2-IV1)/(15.*20.)                                    CKR00360
MULDV=FIX(CKDV)                                                  CKR00370
IF (FLOAT(MULDV)/CKDV.LT..99) MULDV=MULDV+1                    CKR00380
DVHOLD=20.*FLOAT(MULDV)                                          CKR00390
WRITE (IOOUT,93) DV,DVHOLD                                       CKR00400
DV=DVHOLD                                                         CKR00410
IF (DV.LT.20.) DV=20.                                           CKR00420
IDV=FIX(DV)                                                       CKR00430
93  FORMAT (1X,'DIVISION LIMITS CHANGED FROM ',F10.3,          CKR00440
+ ' TO ',F10.3)                                                  CKR00450
91  CONTINUE                                                       CKR00460
RETURN                                                            CKR00470
3190 FORMAT (6X,'*****FREQUENCY IS OUTSIDE OF THE WINDOW*****' / CKR00480
1      6X,'*****TOTAL TRANSMITTANCE IS 1.0000*****')          CKR00490
END                                                                CKR00500

```

	SUBROUTINE FREQSL(I, IV, W, TX)		FRE00010
	COMMON /MO5/ C1(501), C2(258), C3(86), C4(33), C5(6), C5DUM(9), C8(82),		FRE00020
	1 C11(4), C12(15), C14(21), C15(6)		FRE00030
	COMMON /MO7/ TR(67), FW(67), FO(67)		FRE00040
	COMMON /MO8/ SUM4, SUM5, SUM8, SUM11, SUM6		FRE00050
	DIMENSION W(16), TX(16)		FRE00060
	IF (I.EQ.1) GO TO 10		FRE00070
	IF (I.GE.2.AND.I.LE.3) GO TO 11		FRE00080
	IF (I.GE.4.AND.I.LE.5) GO TO 12		FRE00090
	IF (I.GE.6.AND.I.LE.12) GO TO 13		FRE00100
	IF (I.GE.13.AND.I.LE.21) GO TO 15		FRE00110
	IF (I.EQ.22) GO TO 16		FRE00120
	IF ((I.GE.23.AND.I.LE.59).OR.(I.GE.127.AND.I.LE.209)) RETURN		FRE00130
	IF ((I.GE.60.AND.I.LE.63)) GO TO 14		FRE00140
	IF (I.GE.64.AND.I.LE.76) GO TO 18		FRE00150
	IF ((I.GE.77.AND.I.LE.81).OR.(I.GE.87.AND.I.LE.96)) GO TO 18		FRE00160
	IF (I.GE.82.AND.I.LE.86) GO TO 30		FRE00170
	IF ((I.GE.97.AND.I.LE.101).OR.(I.GE.105.AND.I.LE.109)) GO TO 14		FRE00180
	IF (I.GE.102.AND.I.LE.104) GO TO 9		FRE00190
	IF (I.GE.110.AND.I.LE.112) GO TO 21		FRE00200
	IF (I.GE.113.AND.I.LE.123) GO TO 22		FRE00210
	IF ((I.GE.124.AND.I.LE.126).OR.(I.GE.210.AND.I.LE.363)) GO TO 23		FRE00220
	IF ((I.GE.364.AND.I.LE.419).OR.(I.GE.454.AND.I.LE.599)) GO TO 24		FRE00230
	IF ((I.GE.420.AND.I.LE.453).OR.(I.GE.600.AND.I.LE.606).OR.(I.GE		FRE00240
	1,1160.AND.I.LE.1334)) GO TO 35		FRE00250
	IF (I.GE.607.AND.I.LE.609) GO TO 25		FRE00260
	IF (I.GE.610.AND.I.LE.621) GO TO 26		FRE00270
	IF ((I.GE.622.AND.I.LE.629).OR.(I.GE.686.AND.I.LE.1159).OR.(I.GE		FRE00280
	1,1335)) GO TO 27		FRE00290
	IF (I.GE.630.AND.I.LE.685) GO TO 28		FRE00300
4	CALL H2OVAP(I, W, C1, TX)		FRE00310
	GO TO 40		FRE00320
6	CALL OZONE(I, W, C3, TX)		FRE00330
7	CALL UNIMIX(I, W, C2, TX)		FRE00340
	GO TO 4		FRE00350
8	CALL NO2(I, W, C15, TX)		FRE00360
	GO TO 6		FRE00370
9	CALL H2O410(I, IV, W, C5, TX, SUM5)		FRE00380
	GO TO 8		FRE00390
10	CALL NH3(I, W, C14, TX)		FRE00400
	GO TO 9		FRE00410
11	CALL NITRIC(I, W, C11, SUM11, TX)		FRE00420
	GO TO 10		FRE00430
12	CALL NITRIC(I, W, C11, SUM11, TX)		FRE00440
13	CALL NH3(I, W, C14, TX)		FRE00450
14	CALL H2O410(I, IV, W, C5, TX, SUM5)		FRE00460
	GO TO 6		FRE00470
15	CALL SO2(I, W, C12, TX)		FRE00480
	GO TO 13		FRE00490
16	CALL SO2(I, W, C12, TX)		FRE00500
	GO TO 14		FRE00510
18	CALL H2O410(I, IV, W, C5, TX, SUM5)		FRE00520
19	CALL NITRO(I, W, C4, TX, SUM4)		FRE00530
	GO TO 6		FRE00540
20	CALL SO2(I, W, C12, TX)		FRE00550
	GO TO 10		FRE00560
21	CALL MOLSCT(IV, W, TX, SUM6)		FRE00570
	GO TO 14		FRE00580
22	CALL MOLSCT(IV, W, TX, SUM6)		FRE00590
	GO TO 6		FRE00600
23	CALL MOLSCT(IV, W, TX, SUM6)		FRE00610
	GO TO 7		FRE00620
24	CALL MOLSCT(IV, W, TX, SUM6)		FRE00630
	GO TO 4		FRE00640
25	CALL UNIMIX(I, W, C2, TX)		FRE00650
35	CALL MOLSCT(IV, W, TX, SUM6)		FRE00660
	GO TO 40		FRE00670
26	CALL UVOZNE(I, W, C8, TX, SUM8)		FRE00680
	GO TO 25		FRE00690
27	CALL UVOZNE(I, W, C8, TX, SUM8)		FRE00700

CALL MOLSCT<IV,W, TX, SUM6>
GO TO 40
28 CALL H20VAP<I,W,C1, TX>
GO TO 27
30 CALL H20410<I, IV, W, C5, TX, SUM5>
GO TO 20
40 RETURN
END

FRE00710
FRE00720
FRE00730
FRE00740
FRE00750
FRE00760
FRE00770
FRE00780

```

SUBROUTINE H2OVAP(I,W,C1,TX)
DIMENSION C1(501),TX(1),W(1)
C*****TRANSMITTANCE FOR WATER VAPOR*****
IF (W(1).LT.1.0E-20) GO TO 500
IF (I.LE.22) I1=I
IF (I.GE.60.AND.I.LE.126) I1=I-37
IF (I.GE.210.AND.I.LE.419) I1=I-120
IF (I.GE.454.AND.I.LE.599) I1=I-154
IF (I.GE.630) I1=I-184
WS1=ALOG10(W(1))+C1(I1)
TX(1)=EXP(-10**(-1.14619+0.55013*WS1))
500 RETURN
END

```

```

H2V00010
H2V00020
H2V00030
H2V00040
H2V00050
H2V00060
H2V00070
H2V00080
H2V00090
H2V00100
H2V00110
H2V00120
H2V00130

```

SUBROUTINE H20410(I,IV,W,C5,TX,SUM5)	H2F00010
C***WATER VAPOR CONTINUUM, 3-5 AND 8-12 MICRON REGIONS	H2F00020
DIMENSION C5(6),TX(5),W(16)	H2F00030
V=FLOAT(IV)	H2F00040
IF(I.GT.109) GO TO 100	H2F00050
IF(I.GT.22) GO TO 200	H2F00060
C***1.LE.I.LE.22:CALCULATE OPTICAL DEPTH DUE TO 8-12 MICRON CONTINUUM	H2F00070
TX(5)=(4.18+5578.0*EXP(-7.87E-03*V))*W(5)	H2F00080
GO TO 300	H2F00090
200 CONTINUE	H2F00100
IF(I.LT.63) GO TO 100	H2F00110
IF(I.GT.68) GO TO 400	H2F00120
C***63.LE.I.LE.68:CALCULATE OPTICAL DEPTH DUE TO 4.6-4.8 MICRON	H2F00130
C***CONTINUUM MODEL FROM BEN-SHALOM ET AL.1980,SPIE,VOL.253,261.	H2F00140
II=I-62	H2F00150
TX(5)=C5(II)*577.6*W(16)	H2F00160
GO TO 300	H2F00170
400 CONTINUE	H2F00180
IF(I.LT.77) GO TO 100	H2F00190
C***77.LE.I.LE.109:CALCULATE OPTICAL DEPTH DUE TO 3.3-4.3 MICRON	H2F00200
C***CONTINUUM MODEL FROM WATKINS ET AL.1979,APPL.OPT.,VOL.18,1149.	H2F00210
V=V*1.0E-03	H2F00220
V2=V*V	H2F00230
V3=V2*V	H2F00240
CBURCH=46.4745-48.0898*V+16.3988*V2-1.83217*V3	H2F00250
CASL=-370.082+508.137*V-225.822*V2+32.7744*V3	H2F00260
TX(5)=CBURCH*W(10)+CASL*W(15)	H2F00270
GO TO 300	H2F00280
100 TX(5)=0.0	H2F00290
300 SUM5=TX(5)	H2F00300
IF(TX(5).LT.1.0E-05) GO TO 500	H2F00310
IF(TX(5).GT.20.0) GO TO 600	H2F00320
TX(5)=EXP(-TX(5))	H2F00330
RETURN	H2F00340
500 TX(5)=1.0	H2F00350
RETURN	H2F00360
600 TX(5)=0.0	H2F00370
RETURN	H2F00380
END	H2F00390


```

SUBROUTINE LTPATH(WLAY, WPATH, TBBY, ANGLE, LEN, ITYPE, H1, H2, MODEL) LTP00010
COMMON /M01/EH(16,34), P(34), T(34), WH(34), Z(34), WA(34), RE, M, NL LTP00020
DIMENSION WLAY(34,16), TBBY(68), WPATH(68,16) LTP00030
COMMON /EM1/HMIN, KMAX, IJ, J1, J2, JMIN, JEXTRA, NP1 LTP00040
COMMON /EM2/W(16), E(16), IL, IKMAX, LENTOR, NLL LTP00050
COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUL LTP00060
COMMON /SPOTLO/ISPOT, LOREAD, N16 LTP00070
LOGICAL ISPOT, N16, LOREAD LTP00080
555 IL=0 LTP00090
IF (ITYPE.EQ.1) GO TO 1000 LTP00100
IF (ITYPE.EQ.2.AND.H1.EQ.H2) J2=J1 LTP00110
IF (H2.GT.H1.AND.ANGLE.GT.90..AND.NP1.EQ.1) J1=J1-1 LTP00120
IF (JEXTRA.EQ.1) J2=J2+1 LTP00130
IF ((ITYPE.EQ.2).AND.(H1.GT.H2).AND.(LENTOR.EQ.1)) J2=
J2-1 LTP00140
1 IF (ITYPE.EQ.3) J2=NLL LTP00150
IF (.NOT.ISPOT) WRITE (IOOUT,1200) LTP00170
DO 100 IK=1,68 LTP00180
TBBY(IK)=0. LTP00190
DO 100 K=1,KMAX LTP00200
WPATH(IK,K)=0. LTP00210
100 CONTINUE LTP00220
LEN=0 LTP00230
NLL=NLL-1 LTP00240
IL=J1+1 LTP00250
IJ=IL+NLL LTP00260
DO 200 K=1,KMAX LTP00270
E(K)=0. LTP00280
200 CONTINUE LTP00290
IF (ANGLE.GT.90.0) GO TO 300 LTP00300
LEN=1. LTP00310
IL=J1-1 LTP00320
HMIN=1.0E-6 LTP00330
IJ=NLL LTP00340
300 CONTINUE LTP00350
DO 800 IK=1,68 LTP00360
IF (LEN.EQ.0) IL=IL-1 LTP00370
IF (LEN.EQ.1) IL=IL+1 LTP00380
IJ=IJ-1 LTP00390
IF (IL.EQ.0) GO TO 800 LTP00400
DO 400 K=1,KMAX LTP00410
W(K)=E(K)+WLAY(IL,K) LTP00420
WPATH(IK,K)=W(K) LTP00430
400 CONTINUE LTP00440
IF (IL.LE.0.OR.IL.GE.NLL) GO TO 500 LTP00450
TBAR=(T(IL)+T(IL+1))*0.5 LTP00460
C JEXTRA IS 1 ONLY WHEN PROGRAM NEVER LEAVES ONE LAYER LTP00470
IF (JEXTRA.EQ.1) TBAR=(T(J1)+T(J1+1))*0.5 LTP00480
500 CONTINUE LTP00490
TBBY(IK)=TBAR LTP00500
DO 600 K=1,KMAX LTP00510
E(K)=W(K) LTP00520
600 CONTINUE LTP00530
IF (ANGLE.LE.90.0.AND.IL.EQ.NLL) GO TO 900 LTP00540
IF (ITYPE.EQ.3.AND.ANGLE.LE.90.0) GO TO 700 LTP00550
IF (ITYPE.EQ.3.AND.LEN.EQ.1.AND.IL.EQ.J2) GO TO 900 LTP00560
IF (ITYPE.EQ.2.AND.LENTOR.EQ.0.AND.IL.EQ.J2) GO TO 900 LTP00570
IF (IL.EQ.JMIN.AND.HMIN.GT.0) LEN=1 LTP00580
IF (IL.EQ.1.AND.HMIN.LE.0.0) GO TO 900 LTP00590
IF (LEN.EQ.0) GO TO 700 LTP00600
IF (IL.EQ.JMIN.AND.IJ.EQ.IL+NLL) IL=IL-1 LTP00610
IF (ITYPE.EQ.2.AND.IL.EQ.J2) GO TO 900 LTP00620
700 CONTINUE LTP00630
IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=
1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK) LTP00640
1 800 CONTINUE LTP00650
IKMAX=68 LTP00660
LEN=LENTOR LTP00670
RETURN LTP00680
900 CONTINUE LTP00690
LTP00700

```

```

IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=
1 1,6),WPATH(IK,8),WPATH(IK,K),K=10,14),TBBY(IK) LTP00710
IKMAX=IK LTP00720
LEN=LENTOR LTP00730
RETURN LTP00740
1000 DO 1100 K=1,KMAX LTP00750
WPATH(1,K)=W(K) LTP00760
1100 CONTINUE LTP00770
IF (MODEL.EQ.0) J1=1 LTP00780
J2=J1 LTP00790
TBBY(1)=T(J1) LTP00800
IKMAX=1 LTP00810
IK=1 LTP00820
IF (.NOT.ISPOT) WRITE (IOOUT,1200) LTP00830
IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K= LTP00840
1 1,6),WPATH(IK,8),WPATH(IK,K),K=10,14),TBBY(IK) LTP00850
HMIN=1.0E-6 LTP00860
RETURN LTP00870
C LTP00880
1200 FORMAT (//,20X,37H CUMULATIVE ABSORBER AMOUNTS FOR THE , LTP00890
1 16HATMOSPHERIC PATH,// 8X,3HH2O,5X,4HC02+,6X, LTP00910
2 2H03,7X,2HN2,6X,5HH2O C,4X,5HMOL S,4X,5H03 UV, LTP00920
3 4X,5HH2O C,5X,4HHNO3,6X,3HS02,6X,3HHH3,6X,3HN02, LTP00930
4 5X,4HTAVE) LTP00940
1300 FORMAT (I5,12E9.3,F10.3) LTP00950
END LTP00960

```

```

SUBROUTINE MOLSCT(IV,W,TX,SUM6)
C*****TRANSMITTANCE FOR MOLECULAR SCATTERING *****
C C6 EXPRESSION MODIFIED AS PER SHETTLER ET AL 1980,APPL.OPT.,VOL.19,
C 2873
DIMENSION TX(6),W(6)
V=FLOAT(IV)
C6=(V**4)/(9.67578E+18-1.11836E+09*V**2)
TX(6)=C6*W(6)
SUM6=TX(6)
IF (TX(6).EQ.0.0) GO TO 200
IF (TX(6).LE.0.1) GO TO 100
IF (TX(6).GT.20.) GO TO 300
TX(6)=EXP(-TX(6))
GO TO 400
100 TX(6)=1.0-TX(6)+0.5*TX(6)*TX(6)
GO TO 400
200 TX(6)=1.0
GO TO 400
300 TX(6)=0.0
400 RETURN
END
MOL00010
MOL00020
MOL00030
MOL00040
MOL00050
MOL00060
MOL00070
MOL00080
MOL00090
MOL00100
MOL00110
MOL00120
MOL00130
MOL00140
MOL00150
MOL00160
MOL00170
MOL00180
MOL00190
MOL00200
MOL00210

```

	SUBROUTINE NH3(I,W,C13,TX)	NHA00010
	COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),	NHA00020
	1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2	NHA00030
	DIMENSION C13(21),TX(13),W(13)	NHA00040
C	*****	NHA00050
C	THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NH3 (PPM READ IN	NHA00060
C	THE MAIN PROGRAM).	NHA00070
C	*****	NHA00080
	I1=1	NHA00090
	IF (W(13).LT.1.0E-20) GO TO 3	NHA00100
	WS13=ALOG10(W(13))+C13(I1)	NHA00110
	DO 1 J=1,9	NHA00120
	IF (WS13-FNH3(J)) 2,2,1	NHA00130
1	CONTINUE	NHA00140
2	TX(13)=EXP(-10**((FH1(J)+FH2(J))*WS13))	NHA00150
	RETURN	NHA00160
	END	NHA00170
		NHA00180
		NHA00190

SUBROUTINE NITRIC(I,W,C11,SUM11,TX)	NITC0010
DIMENSION C11(4),TX(11),W(11)	NITC0020
C***** TRANSMITTANCE FOR NITRIC ACID*****	NITC0030
HABS=0.	NITC0040
IF (I.LT.2.OR.I.GT.46) GO TO 100	NITC0050
IF (I.GT.5.AND.I.LT.23) GO TO 100	NITC0060
I1=I-1	NITC0070
HABS=C11(I1)	NITC0080
100 CONTINUE	NITC0090
TX(11)=HABS*W(11)	NITC0100
SUM11=TX(11)	NITC0110
IF (TX(11).EQ.0.0) GO TO 300	NITC0120
IF (TX(11).LE.0.1) GO TO 200	NITC0130
IF (TX(11).GT.20.) GO TO 400	NITC0140
TX(11)=EXP(-TX(11))	NITC0150
GO TO 500	NITC0160
200 TX(11)=1.0-TX(11)+0.5*TX(11)*TX(11)	NITC0170
GO TO 500	NITC0180
300 TX(11)=1.0	NITC0190
GO TO 500	NITC0200
400 TX(11)=0.0	NITC0210
500 RETURN	NITC0220
END	NITC0230

SUBROUTINE NITRO(I,W,C4,TX,SUM4)	NITR0010
DIMENSION C4(33),TX(4),W(4)	NITR0020
C*****TRANSMITTANCE FOR NITROGEN CONTINUUM*****	NITR0030
IF (I.LT.64) GO TO 200	NITR0040
I1=i-63	NITR0050
C TEMP FIX FOLLOWS	NITR0060
C IF (I1.GT.10) GO TO 300	NITR0070
TX(4)=C4(I1)*W(4)	NITR0080
SUM4=TX(4)	NITR0090
IF (TX(4).EQ.0.0) GO TO 200	NITR0100
IF (TX(4).LE.0.1) GO TO 100	NITR0110
IF (TX(4).GT.20.) GO TO 300	NITR0120
TX(4)=EXP(-TX(4))	NITR0130
GO TO 400	NITR0140
100 TX(4)=1.0-TX(4)+0.5*TX(4)*TX(4)	NITR0150
GO TO 400	NITR0160
200 TX(4)=1.0	NITR0170
GO TO 400	NITR0180
300 TX(4)=0.0	NITR0190
400 RETURN	NITR0200
END	NITR0210

```

SUBROUTINE NO2(I,W,C14,IX)
COMMON /NO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FN02(9),
1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2
DIMENSION C14(6),TX(14),W(14)
*****
C *****
C THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NO2 ( PPM READ IN
C THE MAIN PROGRAM).
C *****
IF (I.GE.102.AND.I.LE.104) I1=I-98
IF (I.LE.3) I1=1
IF (W(14).LT.1.0E-20) GO TO 3
WS14=ALOG10(W(14))+C14(I1)
DO 1 J=1,9
IF (WS14-FN02(J)) 2,2,1
CONTINUE
TX(14)=EXP(-10** (O1(J)+O2(J)*WS14))
RETURN
END

```

NO2X0010
NO2X0020
NO2X0030
NO2X0040
NO2X0050
NO2X0060
NO2X0070
NO2X0080
NO2X0090
NO2X0100
NO2X0110
NO2X0120
NO2X0130
NO2X0140
NO2X0150
NO2X0160
NO2X0170
NO2X0180
NO2X0190
NO2X0200

SUBROUTINE OZONE(I,W,C3,TX)	OZN00010
DIMENSION C3(86),TX(3),W(3)	OZN00020
C*****TRANSMITTANCE FOR OZONE*****	OZN00030
IF (W(3).LT.1.0E-20) GO TO 500	OZN00040
IF (I.LE.22) I1=I	OZN00050
IF (I.GE.60) I1=I-37	OZN00060
WS3=ALOG10(W(3))+C3(I1)	OZN00070
TX(3)=1/(1+EXP(-3.08019+2.11127*WS3))	OZN00080
500 RETURN	OZN00090
END	OZN00100


```

C      FUNCTION RESFN (NR,WAVE)
C      THIS FUNCTION WILL READ IN UP TO 20 VALUES OF A RESPONSE FUNCTION
C      IF THE RESF CARD IS READ IN EOMAIN. ONLY ONE RESPONSE FUNCTION
C      PER RUN IS ALLOWED. THIS FUNCTION WILL ALSO DO A LINEAR INTERPOLARE
C      OVER WAVELENGTH. IF RESF CARD IS NOT READ A VALUE OF 1 IS RETURNED
C      TO THE CALLING PROGRAM WHEN THIS FUNCTION IS REFERENCED.
C      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUR
C      DIMENSION WAVELN(20),RESPFN(20)
C      DATA WAVELN,RESPFN,ICOUNT,NBR /20*0.,20*0.,0,1/
C      IF (NR.NE.1) GO TO 6
C      ICOUNT=ICOUNT+1
C      IF (ICOUNT.GT.1) GO TO 2
C      READ (IOIN,100) NBR
C      IF (NBR.GT.20) WRITE (IOOUT,102)
C      IF (NBR.GT.20) STOP
C      WRITE (IOOUT,103)
C      DO 3 I=1,NBR
C      READ (IOIN,101) WAVELN(I),RESPFN(I)
C      WRITE (IOOUT,104) WAVELN(I),RESPFN(I)
C      IF (WAVE.LT.(WAVELN(1)-.0001).OR.WAVE.GT.(WAVELN(NBR)+.0001))
C      + GO TO 6
C      DO 4 I=1,NBR
C      K=I
C      IF (WAVE.GE.WAVELN(I)) GO TO 5
C      IF (WAVE/WAVELN(K).GE..99.AND.WAVE/WAVELN(K).LE.1.01) GO TO 7
C      IF (K.EQ.NBR) GO TO 8
C      RESFN=(WAVE-WAVELN(K))*((RESPFN(K+1)-RESPFN(K)))/
C      + (RESPFN(K+1)-RESPFN(K))+RESPFN(K)
C      RETURN
C      RESFN=RESPFN(K)
C      RETURN
C      RESFN=RESPFN(NBR)
C      RETURN
C      RESFN=1.
C      RETURN
C      100  FORMAT (I2)
C      101  FORMAT (2(E10.4,1X))
C      102  FORMAT (1H,5I10,THE NUMBER OF VALUES FOR THE RESPONSE FUNCTION IS G
C      +,5B,REATER THAN THE DIMENSIONS LIMITS OF WAVELN( ) AND RESPFN( ),
C      +/,1X,19HPROGRAM TERMINATED.)
C      103  FORMAT (1H,20X,23HINPUT RESPONSE FUNCTION,/,1X,15X,10HWAVELENGTH,
C      +5X,10HR FUNCTION)
C      104  FORMAT (1H,15X,2(E10.4,1X))
C      END
RES00010
RES00020
RES00030
RES00040
RES00050
RES00060
RES00070
RES00080
RES00090
RES00100
RES00110
RES00120
RES00130
RES00140
RES00150
RES00160
RES00170
RES00180
RES00190
RES00200
RES00210
RES00220
RES00230
RES00240
RES00250
RES00260
RES00270
RES00280
RES00290
RES00300
RES00310
RES00320
RES00330
RES00340
RES00350
RES00360
RES00370
RES00380
RES00390
RES00400
RES00410
RES00420
RES00430
RES00440

```


SUBROUTINE UNIMIX(I,W,C2,TX)	UNI00010
DIMENSION C2(258),TX(2),W(2)	UNI00020
C*****TRANSMITTANCE FOR UNIFORMLY MIXED GASES*****	UNI00030
IF (W(2).LT.1.0E-20) GO TO 500	UNI00040
IF (I.LE.22) I1=I	UNI00050
IF (I.GE.60.AND.I.LE.126) I1=I-37	UNI00060
IF (I.GE.210.AND.I.LE.363) I1=I-120	UNI00070
IF (I.GE.607) I1=I-363	UNI00080
WS2=ALOG10(W(2))+C2(I1)	UNI00090
TX(2)=EXP(-10**(-1.14619+0.55013*WS2))	UNI00100
500 RETURN	UNI00110
END	UNI00120

	SUBROUTINE UVOZNE(I,W,C8,TX,SUM8)	UVZ00010
	DIMENSION C8(82),TX(8),W(8)	UVZ00020
C*****	TRANSMITTANCE FOR UV OZONE *****	UVZ00030
	AI=I	UVZ00040
	IF(I.LE.1159) GO TO 90	UVZ00050
	IF(I.GE.1335) GO TO 100	UVZ00060
90	XX=10.0	UVZ00070
	XI=(AI-610.0)/XX+1.0	UVZ00080
	L1=1	UVZ00090
	L2=53	UVZ00100
	GO TO 200	UVZ00110
100	XX=25.0	UVZ00120
	XI=(AI-1335.0)/XX+57.0	UVZ00130
	L1=57	UVZ00140
	L2=102	UVZ00150
200	DO 300 N=L1,L2	UVZ00160
	XD=XI-FLOAT(N)	UVZ00170
	IF(XD) 500,400,300	UVZ00180
300	CONTINUE	UVZ00190
400	TX(8)=W(8)*C8(N)	UVZ00200
	GO TO 600	UVZ00210
500	TX(8)=C8(N)+XD*(C8(N)-C8(N-1))	UVZ00220
	TX(8)=W(8)*TX(8)	UVZ00230
600	SUM8=TX(8)	UVZ00240
	IF(TX(8).EQ.0.0) GO TO 800	UVZ00250
	IF(TX(8).LE.0.1) GO TO 700	UVZ00260
	IF(TX(8).GT.20.0) GO TO 900	UVZ00270
	TX(8)=EXP(-TX(8))	UVZ00280
	GO TO 1000	UVZ00290
700	TX(8)=1.0-TX(8)+0.5*TX(8)*TX(8)	UVZ00300
	GO TO 1000	UVZ00310
800	TX(8)=1.0	UVZ00320
	GO TO 1000	UVZ00330
900	TX(8)=0.0	UVZ00340
1000	RETURN	UVZ00350
	END	UVZ00360

```

SUBROUTINE SPOT(WAVN1,WAVN2,VIS,NR,IERR,MULDV)
*****
INPUT: EXCLUDING THE OPTIONAL RESPONSE FUNCTION CARDS,
THERE IS A MAXIMUM OF 7 CARDS TO EXECUTE THIS MODULE.
THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OF
THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.
THE CARDS ARE INPUT WITH FORMAT (A4,6X,7E10.4)
EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4
FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL PER
FIELD BEGINNING IN COL 11.
THE CARDS ARE NOT ORDER DEPENDENT.
*****
CARD 1
ENVR  ISORC, ITARG, IHAZE, MODEL, NLAM
      ISORC = 0 SUNLIGHT ONLY
            1 MOONLIGHT ONLY
            2 EMISSION ONLY
            3 SUNLIGHT AND EMISSION
            4 MOONLIGHT AND EMISSION
      ITARG = 0 BACKGROUND ONLY
            1 GROUND REFLECTANCE/EMISSION
            2 TARGET REFLECTANCE/EMISSION
** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK **
FOR SLANT PATHS IHAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES.
      IHAZE = 0, NO AEROSOL ATTENUATION
            = 1, MARITIME POLAR
            = 2, MARITIME ARCTIC
            = 3, CONTINENTAL POLAR
            = 4, RAIN
            = 5, SNOW
            = 7, USER SUPPLIED EXTINCTION COEFFICIENT
              (READ ON ATM CARD - SEE CARD 3 BELOW)
            = 8, EXTINCTION COEFFICIENT WILL BE READ FROM
              PHASE FUNCTION DATA FILE
      MODEL = 1 TROPICAL MODEL ATMOSPHERE
            2 MIDLATITUDE SUMMER
            3 MIDLATITUDE WINTER
            4 SUBARCTIC SUMMER
            5 SUBARCTIC WINTER
            6 1962 US STANDARD
            8 ISRAELI STANDARD (YEAR, DAYTIME)
            9 ISRAELI STANDARD (YEAR, NIGHTTIME)
      NLAM  = 0 NO AEROSOL ATTENUATION
            = 1 READ PHASE FUNCTION DATA SET - ALSO SEE
              ID BELOW AND EXPLN OF PFN DATA SET BELOW
      ID    = 0, USER SUPPLIED
            = 1, MARITIME ARCTIC, VIS=0.1 TO 2.0 KM
            = 2, MARITIME POLAR, VIS=0.2 KM
            = 3, MARITIME POLAR, VIS=0.2, KM
            = 4, CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM
            = 5, WHITE PHOSPHORUS
            = 6, HEXACHLOROETHANE
            = 7, FOG OIL
            = 8, DUST (MODERATE AEROSOL LOADING)
            = 9, DUST (HEAVY AEROSOL LOADING)
            =10, MARITIME MODEL B, VIS=5KM, RH=95%
            =11, MARITIME MODEL B, VIS=10KM,RH=90%
            =12, MARITIME MODEL B, VIS=50KM,RH=50%
CARD 2  **** IF ISORC LT 2 OR ITARG LT 1 THIS CARD IS NOT NEEDED
EMIS  EM(1), TM(1), EM(2), TM(2)
      EM(1)  EMISSIVITY OF GROUND
      TM(1)  TEMPERATURE OF GROUND (KELVIN)
      EM(2)  EMISSIVITY OF TARGET
      TM(2)  TEMPERATURE OF TARGET (KELVIN)
CARD 3
ATM   ZENTH, CLDHGT, PHASE, BETAEX
      ZENTH  INCIDENT ANGLE OF SUNLIGHT OR MOONLIGHT (DEGREES)
      CLDHGT HEIGHT OF BOTTOM OF CLOUD LAYER (KM)
              ONLY NEEDED WHEN IHAZE NE 0 (DEFAULT IS 0.)

```

```

SPOT0010
SPOT0020
SPOT0030
SPOT0040
SPOT0050
SPOT0060
SPOT0070
SPOT0080
SPOT0090
SPOT0100
SPOT0110
SPOT0120
SPOT0130
SPOT0140
SPOT0150
SPOT0160
SPOT0170
SPOT0180
SPOT0190
SPOT0200
SPOT0210
SPOT0220
SPOT0230
SPOT0240
SPOT0250
SPOT0260
SPOT0270
SPOT0280
SPOT0290
SPOT0300
SPOT0310
SPOT0320
SPOT0330
SPOT0340
SPOT0350
SPOT0360
SPOT0370
SPOT0380
SPOT0390
SPOT0400
SPOT0410
SPOT0420
SPOT0430
SPOT0440
SPOT0450
SPOT0460
SPOT0470
SPOT0480
SPOT0490
SPOT0500
SPOT0510
SPOT0520
SPOT0530
SPOT0540
SPOT0550
SPOT0560
SPOT0570
SPOT0580
SPOT0590
SPOT0600
SPOT0610
SPOT0620
SPOT0630
SPOT0640
SPOT0650
SPOT0660
SPOT0670
SPOT0680
SPOT0690
SPOT0700

```

```

C          PHASE          PHASE ANGLE FOR MOONLIGHT (DEGREES)          SPOT0710
          BETAEX          USER SUPPLIED EXTINCTION COEFFICIENT          SPOT0720
          VALID ONLY WHEN IHAZE=7          SPOT0730
CARD 4
TARG      RTARG,COSX,COSY,COSZ          SPOT0740
          RTARG          SLANT RANGE FROM RECEIVER TO TARGET (KM)          SPOT0750
          COSX          X-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)          SPOT0760
          COSY          Y-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)          SPOT0770
          COSZ          Z-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)          SPOT0780
CARD 5      ***** IF ITARG LT 1 THIS CARD IS NOT NEEDED          SPOT0790
REFL      A0(1),A1(1),IALB(1),A0(2),A1(2),IALB(2)          SPOT0800
          A0(1)          ALBEDO COEFFICIENT FOR GROUND          SPOT0810
          A1(1)          ALBEDO COEFFICIENT FOR GROUND          SPOT0820
          IALB(1)        TYPE OF REFLECTION DISTRIBUTION FOR GROUND          SPOT0830
          A0(2)          ALBEDO COEFFICIENT FOR TARGET          SPOT0840
          A1(2)          ALBEDO COEFFICIENT FOR TARGET          SPOT0850
          IALB(2)        TYPE OF REFLECTION DISTRIBUTION FOR TARGET          SPOT0860
          IALB = 0 LAMBERTIAN REFLECTION SURFACE          SPOT0870
          1 ISOTROPIC          SPOT0880
CARD 6
SENS      ALT, THETA, PHI, SANG2          SPOT0890
          ALT          ALTITUDE OF RECEIVER (KM)          SPOT0900
          THETA        POLAR DIRECTION OF LOOK ANGLE (DEGREES)          SPOT0910
          PHI          AZIMUTH DIRECTION OF LOOK ANGLE (DEGREES)          SPOT0920
          SPOT0930
          SPOT0940
          SPOT0950
          SPOT0960
          SPOT0970
          SPOT0980
          SANG2        HALF ANGLE DEFINING RECEIVER FIELD-OF-VIEW          SPOT0990
          (DEGREES)          SPOT1000
CARD 7
GO          SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FOR          SPOT1010
          THIS CALL. NOTE THAT IF A DATA CARD IS NOT READ          SPOT1020
          THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLS          SPOT1030
          TO THE MODULE ARE STILL IN EFFECT.          SPOT1040
          SPOT1050
          SPOT1060
OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY NR=1 IN EOMAIN)          SPOT1070
THESE CARDS MUST FOLLOW THE GO CARD AND CAN ONLY BE INSERTED ONCE          SPOT1080
CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2).          SPOT1090
CARDS 2 - NUMBER OF VALUES: FORMAT (2(E10.4,1X))          SPOT1100
          ONE VALUE OF WAVE (UM) AND RESPONSE FUNCON PER CARD          SPOT1110
          N.B. ONLY ONE RESPONSE FUNCTION PER EOSAEL RUN.          SPOT1120
          SPOT1130
AUXILLARY READ FROM UNIT IPHFUN (ASL DATA SET PROVIDED WITH EOSAEL)          SPOT1140
ANG        ANGLES AT WHICH PHASE MATRIX IS DEFINED,          SPOT1150
          NANG VALUES (DEFAULT IS 65), FORMAT(11(F6.2,1X))          SPOT1160
NANG, ID, WAVE, OMEGA0, BETAEX, BETABS          SPOT1170
          NUMBER OF ANGLES AT WHICH THE PHASE FUNCTION HAS          SPOT1180
          VALUES, PFN IDENTIFIER, WAVELENGTH(UM), SINGLE          SPOT1190
          SCATTERING ALBEDO, EXTINCTION COEFFICIENTS (TOTAL          SPOT1200
          AND SCATTERING).          SPOT1210
          FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X)).          SPOT1220
PF         PHASE FUNCTION AT SPECIFIED ANGLES,          SPOT1230
          FORMAT (6(E12.6,1X))          SPOT1240
C*****          SPOT1250
LOGICAL L1,L2,L3,L4,L5,L6,L7,ISPOT,N16,LOREAD,HORIZ          SPOT1260
DIMENSION DUMMY(16)          SPOT1270
EQUIVALENC (ITARG,IT)          SPOT1280
COMMON /ANSW2/TTR(16),TBR(16),CNTRST(16)          SPOT1290
COMMON /ALBED/A0(2),A1(2),IALB(2)          SPOT1300
COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB          SPOT1310
COMMON /CGEOM/COSGM,COSBT,COSIN          SPOT1320
COMMON /COM11/ISORC,ITARG,IWN,JHL          SPOT1330
COMMON /CONST/P1,P12,PIRAD,TWOPI,TORRMB,CDEGK          SPOT1340
COMMON /CTARG/RTARG,COSX,COSY,COSZ          SPOT1350
COMMON /EMISS/EM(2),TM(2)          SPOT1360
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUS          SPOT1370
COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,          SPOT1380
+          RRS(16,34),SCOE(16,34),          SPOT1390
+          TRANS(16,3),RADA(16,2),WAVE(16),SS(16),          SPOT1400

```

```

1      DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),          SPOT1410
2      PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)      SPOT1420
COMMON /MO2/ W0(34),R0,TBOUND,JP,IM,ML,IP,JSTOR          SPOT1430
COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL              SPOT1440
COMMON /SPOTLO/ ISPOT,L0READ,N16                          SPOT1450
COMMON /LOWEX/ WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX, SPOT1460
1      CLDHGT,NCLD                                         SPOT1470
COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16), SPOT1480
1      BE(16),SINGWV,PF(65),LMAX                           SPOT1490
COMMON /LOGIC/L1,L2,L3,L4,L5,L6,L7                       SPOT1500
DATA L1,L2,L3,L4,L5,L6,L7/7*.FALSE./                    SPOT1510
DATA ITR1,ITR2,ITR3,ITR4/2,3,5,1/                       SPOT1520
N16=.TRUE.                                               SPOT1530
INBR=16                                                  SPOT1540
DUM=1.                                                  SPOT1550
ICLMAT=0                                                SPOT1560
L0READ=.TRUE.                                          SPOT1570
ISPOT=.TRUE.                                           SPOT1580
C      INITIALIZE AND READ INPUT PARAMETERS                SPOT1590
CALL ZERO                                              SPOT1600
CALL INDAT(IEMISS,IHAZE,IM,LEN,ML,MODEL,                SPOT1610
1      TBOUND,0,CLDHGT,BETAEX)                          SPOT1620
C      CHECK FOR ERROR IN INPUT DATA                     SPOT1630
IF (IHAZE.EQ.9) IERR=1                                  SPOT1640
IF (IERR.EQ.1) RETURN                                   SPOT1650
IF (ISORC.NE.2) ISWTC=1                                 SPOT1660
C      FIRST CALL IS TO READ LOWTRAN DATA FILE ONLY     SPOT1670
CALL LT4M(ALT,DUM,ZENTH,3,0,TRANS(1,1),DUMMY,DUMMY,    SPOT1680
1      IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,                SPOT1690
2      ICLMAT,IERR,NR,IHAZE,MULDV)                       SPOT1700
V1=WAVN1                                                SPOT1710
V2=WAVN2                                                SPOT1720
CALL CKER(V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,DUM)   SPOT1730
WAVE(1)=10000./V1                                       SPOT1740
DO 300 IW=2,INBR                                        SPOT1750
V2=V1+20.*FLOAT(MULDV)*FLOAT(IW-1)                    SPOT1760
IF (V2.GE.WAVN2) GO TO 400                              SPOT1770
WAVE(IW)=10000./V2                                       SPOT1780
300 CONTINUE                                           SPOT1790
L1=.TRUE.                                              SPOT1800
C      MAXIMUM NO. OF WAVELENGTHS                        SPOT1810
400 IWN=IW-1                                            SPOT1820
IF (L1) IWN=INBR                                        SPOT1830
NWVL=IWN                                               SPOT1840
C      ARRAY WVL IS USED ONLY IN SUBROUTINE PFUNC. THE WAVELENGTHS SPOT1850
C      IN THIS ARRAY INCREASE WITH INCREASING ARRAY INDEX. THE SPOT1860
C      VALID RANGES FOR VALUES IN THE WVL ARRAY ARE : 0.2-2.0, 3.0-5.0, SPOT1870
C      AND 8.0-12.0 MICROMETERS.                         SPOT1880
C      SPOT1890
DO 355 JX=1,NWVL                                       SPOT1900
INDM=NWVL-JX+1                                         SPOT1910
355 WVL(JX)=WAVE(INDM)                                  SPOT1920
DO 500 I=1,NL                                          SPOT1930
IF (ALT.LE.Z(I)) GO TO 600                              SPOT1940
500 CONTINUE                                           SPOT1950
WRITE (IOOUT,3700) ALT,I,Z(I)                          SPOT1960
IERR=1                                                  SPOT1970
RETURN                                                 SPOT1980
600 CONTINUE                                           SPOT1990
IF (I.EQ.1) WRITE (IOOUT,3800) ALT                      SPOT2000
IF (I.EQ.1) IERR=1                                     SPOT2010
IF (IERR.EQ.1) RETURN                                  SPOT2020
JHL=I-1                                                SPOT2030
NLL=NL-1                                               SPOT2040
SANG=TWOP1*(1.0-COS(SANG2*PIRAD))                     SPOT2050
CZNTH=COS(ZENTH*PIRAD)                                 SPOT2060
SZNTH=SIN(ZENTH*PIRAD)                                 SPOT2070
CTHTA=COS(THETA*PIRAD)                                 SPOT2080
STHTA=SIN(THETA*PIRAD)                                 SPOT2090
SPOT2100

```



```

CPHI=COS(PHI*PIRAD)
SPHI=SIN(PHI*PIRAD)
IF (ISORC.EQ.2) GO TO 900
C CALCULATE SOURCE TERM ... FOR SUNLIGHT (ISORC=0,3)
C OR MOONLIGHT (ISORC=1,4)
DO 700 IW=1,IWN
SS(IW)=0.0
IF (ISORC.EQ.0.OR.ISORC.EQ.3) SS(IW)=SOLARS(WAVE(IW))
IF (ISORC.EQ.1.OR.ISORC.EQ.4) SS(IW)=SMOON(WAVE(IW),PHASE)
700 CONTINUE
C CALCULATE DIRECT INTENSITY ... FOR SUNLIGHT (ISORC=0,3)
C OR MOONLIGHT (ISORC=1,4)
COSIN=STHTA*CPHI*SZNTH+CTHTA*CZNTH
ANGIN=ACOS(COSIN)/PIRAD
READ PHASE FUNCTION FILE
CALL INDAT(IEMISS,IHAZE,IM,LEN,ML,MODEL,
1 TBOUND,1,CLDHGT,BETAEX)
IF (ANGIN.GT.SANG2) L2=.TRUE.
IF (ZENITH.GT.80.0) L3=.TRUE.
CALL LT4M(ALT,DUM,ZENITH,3,0,TRANS(1,1),DUMMY,DUMMY,
1 IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,
2 ICLMAT,IERR,NR,IHAZE,MULDV)
IF (L2.OR.L3) GO TO 900
DO 800 IW=1,IWN
DIR(IW)=SS(IW)*TRANS(IW,1)
800 CONTINUE
900 IF (ITARG.EQ.0.AND.THETA.GT.90.0) L7=.TRUE.
IF (L7) GO TO 3500
IF (ITARG.EQ.0) GO TO 1200
IF (ITARG.EQ.1) GO TO 1000
C TARGET...
ZTARG=RTARG*CTHTA+ALT
COSTX=STHTA*CPHI
COSTY=STHTA*SPHI
COSTZ=CTHTA
COSBT=COSX*SZNTH+COSY*CZNTH
COSGM=-(COSTX*COSX+COSTY*COSY+COSTZ*COSZ)
IF (THETA.LE.90.0) GO TO 1100
COSBTG=CZNTH
COSGMG=COS((180.0-THETA)*PIRAD)
GO TO 1100
C GROUND...
1000 ZTARG=0.0
COSBT=CZNTH
COSGMG=COS((180.0-THETA)*PIRAD)
COSBTG=COSBT
COSGMG=COSGM
C CALCULATE ATMOSPHERIC TRANSMISSION/RADIANCE FOR VARIOUS PATHS
1100 IF (COSGM.LE.0.0) L4=.TRUE.
IF (COSBT.LE.0.0) L5=.TRUE.
IF (THETA.LE.90.0.AND.ITARG.EQ.1) L6=.TRUE.
IF (L4.OR.L6) GO TO 3500
1200 CONTINUE
IF (THETA.EQ.90.0)
1 CALL LT4M(ALT,DUM,1000.0,1,2,TRANS(1,4),RADA(1,1),
2 DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2,
3 TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
IF (THETA.EQ.90.0.AND.ITARG.EQ.2)
1 CALL LT4M(ALT,DUM,RTARG,1,2,TRANS(1,2),RADA(1,2),
2 DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2,
3 TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
IF (THETA.EQ.90.0) GO TO 1300
IF (THETA.LT.90.0)
1 CALL LT4M(ALT,DUM,THETA,3,2,TRANS(1,4),RADA(1,2),DUMMY,
2 IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,
3 ICLMAT,IERR,NR,IHAZE,MULDV)
IF (ITARG.EQ.0) GO TO 2000
IF (THETA.LT.90.0)
1 CALL LT4M(ALT,ZTARG,THETA,2,2,TRANS(1,2),RADA(1,2)
2 ,DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2,

```

```

SPOT2110
SPOT2130
SPOT2140
SPOT2150
SPOT2160
SPOT2170
SPOT2180
SPOT2190
SPOT2200
SPOT2210
SPOT2220
SPOT2230
SPOT2240
SPOT2250
SPOT2260
SPOT2270
SPOT2280
SPOT2290
SPOT2300
SPOT2310
SPOT2320
SPOT2330
SPOT2340
SPOT2350
SPOT2360
SPOT2370
SPOT2380
SPOT2390
SPOT2400
SPOT2410
SPOT2420
SPOT2430
SPOT2440
SPOT2450
SPOT2460
SPOT2470
SPOT2480
SPOT2490
SPOT2500
SPOT2510
SPOT2520
SPOT2530
SPOT2540
SPOT2550
SPOT2560
SPOT2570
SPOT2580
SPOT2590
SPOT2600
SPOT2610
SPOT2620
SPOT2630
SPOT2640
SPOT2650
SPOT2660
SPOT2670
SPOT2680
SPOT2690
SPOT2700
SPOT2710
SPOT2720
SPOT2730
SPOT2740
SPOT2750
SPOT2760
SPOT2770
SPOT2780
SPOT2790
SPOT2800

```

```

3   TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
IF (THETA.LT.90.0) GO TO 1300
CALL LT4M<ALT,ZTARG,THETA,2,2,TRANS<1,2>,RADA<1,2>,
1   RADG<1>, IEMISS,LEN,MODEL,VIS, V1,V2,
2   TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
IF (ITARG.NE.2) ITR4=2
IF (ITARG.NE.2) GO TO 1300
IF (ZTARG.LE.0.0) ITR4=2
IF (ZTARG.LE.0.0) GO TO 1300
ITR1=4
CALL LT4M<ALT,0.0,THETA,2,2,TRANS<1,4>,RADA<1,1>,
1   RADG<1>, IEMISS,LEN,MODEL,VIS, V1,V2,
2   TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
1300 IF (ISORC.LT.2) GO TO 1600
C   CALCULATE UNCOLLIDED EMISSION ... FROM GROUND (ITARG=1)
C   OR TARGET (ITARG=2)
DO 1500 IW=1,IWN
WAVEM=WAVE<IW>/1.0E+4
IF (ITARG.EQ.2) GO TO 1400
BKG<IW>=BLACK<WAVEM,TGRD>*EM<1>
RADG<IW>=RADG<IW>*EM<1>*COSGMG
BK<IW>=BKG<IW>
UTEM<IW>=RADG<IW>
GO TO 1500
1400 BK<IW>=BLACK<WAVEM,TM<2>>*EM<2>
UTEM<IW>=BK<IW>*COSGM*TRANS<IW,2>
IF (THETA.LE.90.0) GO TO 1500
BKG<IW>=BLACK<WAVEM,TGRD>*EM<1>
RADG<IW>=RADG<IW>*EM<1>*COSGMG
1500 CONTINUE
C   1600 IF (ISORC.EQ.2) GO TO 2900
C   CALCULATE UNCOLLIDED REFLECTANCE ... FROM GROUND (ITARG=1)
C   OR TARGET (ITARG=2)
IEMISS=0
IF (L5) GO TO 2000
HORIZ=ABS<ZTARG-ALT>.LT.0.001
IF (HORIZ) ITR2=1
IF (HORIZ) GO TO 1700
CALL LT4M<ZTARG,DUM,ZENTH,3,2,TRANS<1,3>,DUMMY,DUMMY,
1   IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,
2   ICLMAT,IERR,NR,IHAZE,MULDV)
1700 IF (ZTARG.LE.0.0) ITR3=ITR2
IF (ZTARG.LE.0.0) GO TO 1800
IF (THETA.LE.90.0) GO TO 1800
CALL LT4M<0.0,DUM,ZENTH,3,2,TRANS<1,5>,DUMMY,DUMMY,
1   IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,
2   ICLMAT,IERR,NR,IHAZE,MULDV)
1800 DO 1900 IW=1,IWN
ALB=ALBEDO<IT>
UTRF<IW>=SS<IW>*COSBT*ALB*TRANS<IW,2>*TRANS<IW,ITR2>
IF (THETA.LE.90.0) GO TO 1900
UERF<IW>=SS<IW>*COSBTG*ALBEDO<1>*TRANS<IW,ITR1>*
1   TRANS<IW,ITR3>
C   1900 CONTINUE
C   CALCULATE SINGLE-SCATTERED PATH RADIANCE ...
C   FROM SUNLIGHT (ISORC=0,3)
C   OR MOONLIGHT (ISORC=1,4)
2000 CALL COEFS<P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX>
IF (IERR.EQ.1) RETURN
IF (ITARG.EQ.0) GO TO 2700
CALL PATHR<CTHTA,ALT,RTARG,1,IHAZE,NR,
1   IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
DO 2100 IW=1,IWN
2100 PATHR2<IW>=SS<IW>*PATHR<IW>
IF (ABS<CTHTA>.LE.1.0E-3) GO TO 2600
IF (CTHTA.LT.0.0) GO TO 2300
2200 Z=Z<NLL>
IO=2
RT=(Z<NLL>-ZTARG)/CTHTA
GO TO 2800
SPOT2810
SPOT2820
SPOT2830
SPOT2840
SPOT2850
SPOT2860
SPOT2870
SPOT2880
SPOT2890
SPOT2900
SPOT2910
SPOT2920
SPOT2930
SPOT2940
SPOT2950
SPOT2960
SPOT2970
SPOT2980
SPOT2990
SPOT3000
SPOT3010
SPOT3020
SPOT3030
SPOT3040
SPOT3050
SPOT3060
SPOT3070
SPOT3080
SPOT3090
SPOT3100
SPOT3110
SPOT3120
SPOT3130
SPOT3140
SPOT3150
SPOT3160
SPOT3170
SPOT3180
SPOT3190
SPOT3200
SPOT3210
SPOT3220
SPOT3230
SPOT3240
SPOT3250
SPOT3260
SPOT3270
SPOT3280
SPOT3290
SPOT3300
SPOT3310
SPOT3320
SPOT3330
SPOT3340
SPOT3350
SPOT3360
SPOT3370
SPOT3380
SPOT3390
SPOT3400
SPOT3410
SPOT3420
SPOT3430
SPOT3440
SPOT3450
SPOT3460
SPOT3470
SPOT3480
SPOT3490
SPOT3500

```

```

2300 IF (ZTARG.GT.0.0) GO TO 2500
      DO 2400 IW=1,IWN
2400 PATHR(IW)=0.0
      GO TO 2900
2500 Z2=0.0
      IO=1
      RT=-ZTARG/CTHTA
      GO TO 2800
2600 Z2=ZTARG
      RT=1000.0
      IO=3
      GO TO 2800
2700 IEMISS=0
      ZTARG=ALT
      IF (ABS(CTHTA).LE.1.0E-03) GO TO 2600
      GO TO 2200
2800 CALL PATHRD(CTHTA,ZTARG,RT,IO,IHAZE,NR,
1      IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
C      CALCULATE BACKGROUND AND TOTAL INTENSITY, PLUS
C      CONTRAST RATIO
2900 DO 3100 IW=1,IWN
      PATHR(IW)=PATHR2(IW)+SS(IW)*PATHR(IW)
      TTR(IW)=PATHR2(IW)+UTRF(IW)+RADA(IW,2)+UTEM(IW)
      TBR(IW)=PATHR(IW)+UERF(IW)+RADA(IW,ITR4)+RADG(IW)
      DIF=TTR(IW)-TBR(IW)
      IF (TBR(IW).GT.0.0) GO TO 3000
      IF (TTR(IW).EQ.TBR(IW)) CNTRST(IW)=0.0
      IF (TTR(IW).NE.TBR(IW)) CNTRST(IW)=1.0E30
      GO TO 3100
3000 CNTRST(IW)=DIF/TBR(IW)
C 3100 CONTINUE
C      CALCULATE TOTAL RADIANCES INTEGRATED OVER DETECTOR RESPONSE
      DV2=DV*0.5
      SUMRPF=0.
      DO 3200 IW=1,IWN
      NW=10000./WAVE(IW)
      W2=10000./(<FLOAT(NW)-DV2)
      W1=10000./(<FLOAT(NW)+DV2)
      IF (IW.EQ.1) W2=WAVE(1)
      IF (IW.EQ.IWN) W1=WAVE(IWN)
      RESPFN=RESFNF(NR,WAVE(IW))
      SUMRPF=SUMRPF+RESPFN
      DW=(W2-W1)*RESPFN
      TOT(1)=TOT(1)+DW*UTEM(IW)
      TOT(2)=TOT(2)+DW*UTRF(IW)
      TOT(3)=TOT(3)+DW*RADA(IW,2)
      TOT(4)=TOT(4)+DW*PATHR2(IW)
      TOT(5)=TOT(5)+DW*TTR(IW)
      TOT(6)=TOT(6)+DW*RADG(IW)
      TOT(7)=TOT(7)+DW*UERF(IW)
      TOT(8)=TOT(8)+DW*RADA(IW,ITR4)
      TOT(9)=TOT(9)+DW*PATHR(IW)
      TOT(10)=TOT(10)+DW*TBR(IW)
3200 TOT(11)=TOT(11)+DW*DIR(IW)
      IF (NR.NE.1) SUMRPF=1.
      DO 3250 I=1,11
3250 TOT(I)=TOT(I)/SUMRPF
      IF (TOT(10).GT.0.0) GO TO 3300
      IF (TOT(5).EQ.TOT(10)) TOT(12)=0.0
      IF (TOT(5).NE.TOT(10)) TOT(12)=1.0E30
      GO TO 3400
3300 TOT(12)=(TOT(5)-TOT(10))/TOT(10)
3400 CONTINUE
3500 CALL OUTPUT(MODEL,IHAZE,CLDHGT)
      RETURN
C
3700 FORMAT (1H ,11H ALTITUDE (<,F10.3,17H) GREATER THAN Z<,
1      12,2H)=,F10.3,27H CONTROL RETURNED TO MAIN
2      10HFROM SPOT.)
3800 FORMAT (1H ,11H ALTITUDE (<,F10.3,16H) LESS THAN ZERO,
SPOT3510
SPOT3520
SPOT3530
SPOT3540
SPOT3550
SPOT3560
SPOT3570
SPOT3580
SPOT3590
SPOT3600
SPOT3610
SPOT3620
SPOT3630
SPOT3640
SPOT3650
SPOT3660
SPOT3670
SPOT3680
SPOT3690
SPOT3700
SPOT3710
SPOT3720
SPOT3730
SPOT3740
SPOT3750
SPOT3760
SPOT3770
SPOT3780
SPOT3790
SPOT3800
SPOT3810
SPOT3820
SPOT3830
SPOT3840
SPOT3850
SPOT3860
SPOT3870
SPOT3880
SPOT3890
SPOT3900
SPOT3910
SPOT3920
SPOT3930
SPOT3940
SPOT3950
SPOT3960
SPOT3970
SPOT3980
SPOT3990
SPOT4000
SPOT4010
SPOT4020
SPOT4030
SPOT4040
SPOT4050
SPOT4060
SPOT4070
SPOT4080
SPOT4090
SPOT4100
SPOT4110
SPOT4120
SPOT4130
SPOT4140
SPOT4150
SPOT4160
SPOT4170
SPOT4180
SPOT4190
SPOT4200

```

1 37H CONTROL RETURNED TO MAIN FROM SPOT. >
END

SPOT4210
SPOT4220

C
C
C

```
FUNCTION ALBEDO(I)
COMMON /ALBED/A0(2),A1(2),IALB(2)
COMMON /CGEOM/COSGM,COSBT,COSIN
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
CALCULATE ALBEDO FOR GROUND (ITARG=1) OR TARGET (ITARG=2)
IALB(I) = 0  LAMBERTIAN REFLECTION SURFACE
           1  ISOTROPIC
A=A0(I)+A1(I)*COSBT
IF (IALB(I).EQ.0) ALBEDO=A*COSGM/PI
IF (IALB(I).EQ.1) ALBEDO=A/TWOPI
RETURN
END
```

ALB00010
ALB00020
ALB00030
ALB00040
ALB00050
ALB00060
ALB00070
ALB00080
ALB00090
ALB00100
ALB00110
ALB00120

C00000

FUNCTION BLACK(W,T)
BLACK(W,T) = PLANCK FUNCTION (UNITS: WATT PER SQUARE METER PER
MICROMETER PER STERADIAN), GIVEN WAVELENGTH W IN CM AND TEMP-
ERATURE T IN K

EXP OVERFLOW PROTECTION

ARG=1.43879/(W*T)
IF(ARG.LT.88.) GO TO 1
BLACK=0.0
RETURN
1 BLACK=1.19106E-12/(W**5*(EXP(ARG)-1.0))
RETURN
END

BLA00010
BLA00020
BLA00030
BLA00040
BLA00050
BLA00060
BLA00070
BLA00080
BLA00090
BLA00100
BLA00110
BLA00120
BLA00130
BLA00140

```

SUBROUTINE COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)      COE00010
COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB        COE00020
COMMON /CTARG/RTARG,COSX,COSY,COSZ                      COE00030
COMMON /M01/DUMMIE(715),MHOLD,NL,                     COE00040
+      RRS(16,34),SCOE(16,34),                        COE00050
1      TRANS(16,5),RADA(16,2),WAVE(16),SS(16),        COE00060
1      DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),     COE00070
2      PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)  COE00080
COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16), COE00090
1 BE(16),SINGWV,PF(65)                                  COE00100
COMMON /COMI1/ISORC,ITARG,IWN,JHL                      COE00110
COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK        COE00120
DIMENSION P(34),T(34)                                  COE00130
C      CALCULATE THE WAVELENGTH-DEPENDENT CONSTANT PRESSURE COEFFICIENTS COE00140
C      FOR MOLECULAR SCATTERING.                        COE00150
C      LOOP OVER LAYERS                                 COE00160
DO 600 I=1,NL                                          COE00170
PS=P(I)/1013.0                                         COE00180
TS=CDEGK/T(I)                                          COE00190
RSCAT=PS*TS                                           COE00200
C      LOOP OVER WAVELENGTHS                            COE00210
DO 600 IW=1,IWN                                       COE00220
RAYS=0.0                                              COE00230
NW=10000./WAVE(IW)                                    COE00240
C      RAYLEIGH SCATTERING = 0.FOR WAVELENGTH GT 3.33 UM COE00250
IF (NW.LT.3000) GO TO 200                             COE00260
WN=FLOAT(NW)                                          COE00270
RAYS=RSCAT*(WN**4)/(9.67578E+18-1.11836E+09*WN**2)  COE00280
200 CONTINUE                                           COE00290
AEXT=1.                                               COE00300
AABS=1.                                               COE00310
IF (IHAZE.EQ.0.OR.I.NE.(NCLD-1)) GO TO 1             COE00320
EXT55=3.912/VIS                                       COE00330
C      UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE COE00340
ZTALT=ZTARG/ALT                                       COE00350
IF (ABS(ZTALT-1.)>.01) RNG=RTARG                     COE00360
IF ((ZTARG.GT.ALT).AND.(RTARG.GT..5/COS(THETA*PIRAD))) COE00370
1  RNG=.5/COS(THETA*PIRAD)                             COE00380
IF (ZTARG.LT.RTARG.AND.(RTARG.GT..5/COS((180.-THETA)*PIRAD))) COE00390
1  RNG=.5/COS((180.-THETA)*PIRAD)                     COE00400
IF (ITARG.EQ.0.AND.(RTARG.GT..5/COS(THETA*PIRAD)))   COE00410
1  RNG=.5/COS(THETA*PIRAD)                             COE00420
ISLANT=1                                              COE00430
IF (ABS(ZTALT-1.)>.01) ISLANT=0                       COE00440
C      CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL COE00450
CALL XSCALE(WAVE(IW),88.,EXT55,XSTRN,IERR,ISLANT,IHAZE,RNG,THETA) COE00460
IF (IERR.EQ.1) RETURN                                  COE00470
AEXT=-ALOG(XSTRN)/RNG                                  COE00480
C      USER SUPPLIED COEFF(IHAZE=7), OR READ FROM PFN DATA FILE(IHAZE=8) COE00490
IF (IHAZE.EQ.7) AEXT=BETAEX                            COE00500
IF (IHAZE.EQ.8) AEXT=BE(IW)                           COE00510
IF (WAVE(IW).LT.2.) AABS=1.                            COE00520
IF (WAVE(IW).GE.3..AND.WAVE(IW).LE.5.) AABS=AEXT*.2  COE00530
IF (WAVE(IW).GE.8..AND.WAVE(IW).LE.12.) AABS=AEXT*.45 COE00540
1  CONTINUE                                           COE00550
SCOE(IW,I)=AEXT-AABS+RAYS                              COE00560
C      CHECK FOR NO AEROSOL PRESENT                    COE00570
IF (SCOE(IW,I).LT.1.E-20) RRS(IW,I)=1.0              COE00580
C      AEROSOL AND RAYLEIGH PRESENT                    COE00590
IF (SCOE(IW,I).GE.1.E-20) RRS(IW,I)=RAYS/SCOE(IW,I) COE00600
C      CHECK FOR NO RAYLEIGH SCATTERING                COE00610
IF (RAYS.LT.1.E-20) RRS(IW,I)=0.0                    COE00620
600 CONTINUE                                           COE00630
RETURN                                                 COE00640
END                                                    COE00650

```

C	SUBROUTINE DIAG	DIAG0010
	THIS SUBROUTINE PRODUCES DIAGNOSTIC COMMENTARY FOR THE	DIAG0020
	SPOT PROGRAM.	DIAG0030
C	CALLING SEQUENCE: CALL DIAG	DIAG0040
	EXTERNAL VARIABLES REQUIRED:	DIAG0050
	THETA <COMMON BLOCK BKDAT>	DIAG0060
	HIGHLIGHT, HTYPE <COMMON BLOCK HOLRTH>	DIAG0070
	L1, L2, L3, L4, L5, L6 <COMMON BLOCK LOGIC>	DIAG0080
		DIAG0090
		DIAG0100
		DIAG0110
	LOGICAL L1, L2, L3, L4, L5, L6, L7	DIAG0120
	COMMON /BKDAT, ALT, THETA, PHI, SANG2, ZENTH, PHASE, ALB	DIAG0130
	COMMON /HOLRTH, HITARG(8,3), HISORC(6,5),	DIAG0140
1	HMODEL(5,6), HLIGHT(3), HMNLT(3), HSNLT(3),	DIAG0150
2	HTRGT(2), HTYPE(2), HGRND(2)	DIAG0160
	COMMON /LOGIC, L1, L2, L3, L4, L5, L6, L7	DIAG0170
	COMMON /IOUNIT, IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLINT, KSTOR, NPLOTU	DIAG0180
	DATA I, I/	DIAG0190
C	WRITE HEADING,	DIAG0200
	WRITE (IOOUT, 900)	DIAG0210
C	IF ERRORS, GO TO 5; OTHERWISE PRINT CLEAN RUN MESSAGE.	DIAG0220
	IF (L1.OR.L2.OR.L3.OR.L4.OR.L5.OR.L6.OR.L7) GO TO 100	DIAG0230
	WRITE (IOOUT, 1000)	DIAG0240
	GO TO 700	DIAG0250
C	HERE IF THERE WERE ERRORS	DIAG0260
100	WRITE (IOOUT, 1100)	DIAG0270
	IF (.NOT.L1) GO TO 200	DIAG0280
	WRITE (IOOUT, 1200) I	DIAG0290
	I=I+1	DIAG0300
200	IF (.NOT.L2) GO TO 300	DIAG0310
	WRITE (IOOUT, 1300) I, HLIGHT	DIAG0320
	I=I+1	DIAG0330
300	IF (.NOT.L3) GO TO 400	DIAG0340
	WRITE (IOOUT, 1400) I, HLIGHT	DIAG0350
	I=I+1	DIAG0360
400	IF (.NOT.L4) GO TO 500	DIAG0370
	WRITE (IOOUT, 1500) I, HTYPE	DIAG0380
	I=I+1	DIAG0390
500	IF (.NOT.L5) GO TO 600	DIAG0400
	WRITE (IOOUT, 1600) I, HLIGHT	DIAG0410
	I=I+1	DIAG0420
600	IF (.NOT.L6) GO TO 700	DIAG0430
	WRITE (IOOUT, 1700) I, THETA	DIAG0440
	I=I+1	DIAG0450
700	IF (.NOT.L7) GO TO 800	DIAG0460
	WRITE (IOOUT, 1800) I, THETA	DIAG0470
C	WRITE FOOTING.	DIAG0480
800	WRITE (IOOUT, 1900)	DIAG0490
C	900 FORMAT (1H0, 21X, 90(1H*), 3(/, 21X, 1H*, 88X, 1H*))	DIAG0500
	1000 FORMAT (21X, 1H*, 28X, 29HNO SPOT DIAGNOSTICS FOR THIS	DIAG0510
1	3HRUN, 28X, 1H*, /, 21X, 1H*, 28X, 8H-- ----, 11(1H-)	DIAG0520
2	13H ----, 28X, 1H*)	DIAG0530
1100	FORMAT (21X, 1H*, 28X, 25HSPOT DIAGNOSTIC MESSAGES	DIAG0540
1	7HFOLLOW:, 28X, 1H*, /, 21X, 1H*, 28X, 5H----, 10(1H-	DIAG0550
2), 1X, 8(1H-), 1X, 6(1H-), 29X, 1H*, 2(/, 21X, 1H*, 88X,	DIAG0560
3	1H*)	DIAG0570
1200	FORMAT (21X, 1H*, 9X, 11, 30H. NUMBER OF WAVELENGTHS (IWN)	DIAG0580
1	9H EXCEEDS, 31HALLOWABLE DIMENSIONS; IWN RESET	DIAG0590
2	, 8X, 1H*, /, 21X, 1H*, 13X, 6HTO 16., 69X, 1H*, /, 21X,	DIAG0600
3	1H*, 88X, 1H*)	DIAG0610
1300	FORMAT (21X, 1H*, 9X, 11, 12H. NO DIRECT, 3A4, 9HINCIDENT	DIAG0620
1	7HWITHIN, 25HRECEIVER'S FIELD OF VIEW., 13X, 1H*	DIAG0630
2	, /, 21X, 1H*, 88X, 1H*)	DIAG0640
1400	FORMAT (21X, 1H*, 9X, 11, 25H. ANGLE OF INCIDENCE FOR,	DIAG0650
1	3A4, 8HGREATER, 21HTHAN 90.0 DEGREES; NO, 12X,	DIAG0660
2	1H*, /, 21X, 1H*, 13X, 26HCALCULATIONS WILL BE MADE	DIAG0670
3	14HFOR ITARG = 0., 35X, 1H*, /, 21X, 1H*, 88X, 1H*)	DIAG0680
1500	FORMAT (21X, 1H*, 9X, 11, 1H., 1X, 2A4, 14HDOES NOT FACE	DIAG0690
		DIAG0700

1	9HRECEIVER . 45X, 1H*, /, 21X, 1H*, 88X, 1H*)	DIAG0710
1600	FORMAT (21X, 1H*, 9X, 11, 1H, 3A4, 22HILLUMINATES BACK SIDE	DIAG0720
1	3HOF, 7HTARGET . 33X, 1H*, /, 21X, 1H*, 88X, 1H*)	DIAG0730
1700	FORMAT (21X, 1H*, 9X, 11, 30H. THETA LESS THAN 90 DEGREES	DIAG0740
1	10HAND ITARG 13H = 1; THETA = ,F6.4, 19X, 1H*, /	DIAG0750
2	21X, 1H*, 88X, 1H*)	DIAG0760
1800	FORMAT (21X, 1H*, 9X, 11, 25H. THETA GREATER THAN 90	DIAG0770
1	13HDEGREES AND 19HITARG = 0; THETA = ,F7.4,	DIAG0780
2	13X, 1H*, /, 21X, 1H*, 88X, 1H*)	DIAG0790
1900	FORMAT (2<21X, 1H*, 88X, 1H*, />, 21X, 90<1H*))	DIAG0800
	RETURN	DIAG0810
	END	DIAG0820

```

SUBROUTINE INDAT(IEMISS, IHAZE, IM, LEN, ML,          IND00010
MODEL, TBOUND, ISW, CLDHGT, BETAEX)                IND00020
1 COMMON /ALBED/A0(2), A1(2), IALB(2)              IND00030
COMMON /MO1/DUMMIE(715), MHOLD, NLHOLD, DUMMYS(1088), IND00040
+   TRANS(16,5), RADA(16,2), WAVE(16), SS(16),    IND00050
1   DIR(16), RADG(16), UTEM(16), UTRF(16), BK(16),  IND00060
2   PATHR(16), UERF(16), PATHR2(16), TOT(12), BKG(16) IND00070
COMMON /BKDAT/ALT, THETA, PHI, SANG2, ZENTH, PHASE, ALB IND00080
COMMON /COMI1/ISORC, ITARG, IWN, JHL              IND00090
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK IND00100
COMMON /CTARG/RTARG, COSX, COSY, COSZ            IND00110
COMMON /EMISS/EM(2), TM(2)                       IND00120
COMMON /IQUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU IND00130
COMMON /GEOMET/PTS(15), IGEOSW                   IND00140
COMMON /BASPOT/ANG(65), SUM(65), WVL(16), NMVL, ALBB(16), BS(16), IND00150
1 BE(16), SINGWV, PF(65), LMAX                     IND00160
DIMENSION DAT(7), TYPE(7)                         IND00170
DATA TYPE /4HENVR, 4HEMIS, 4HATM, 4HTARG, 4HREFL, 4HSENS, 4HGO / IND00180
DATA IZERO /0/                                    IND00190
IND00200
SUBROUTINE INDAT IS CALLED UPON TO A) READ INPUT CONTROL IND00210
PARAMETERS, WITH CARD ORDER INDEPENDENT INPUT (SEE SPOT IND00220
FOR MORE DETAIL) AND, B) TO READ VALUES OF THE PHASE IND00230
FUNCTION AT SPECIFIED ID AND WAVELENGTH.          IND00240
NOTE. IF ISORC LT 2 OR ITARG LT 1 EM(1), TM(1), EM(2), TM(2) IND00250
ARE NOT NEEDED.                                  IND00260
IF ITARG LT 1 A0(2), A1(2), IALB(2) ARE NOT NEEDED. IND00270
IND00280
ISW=1 ON SECOND CALL TO INDAT                     IND00290
IF (ISW.EQ.1) GO TO 400                           IND00300
IF (IZERO.GT.0) GO TO 9                           IND00310
ISORC=0                                           IND00320
ITARG=0                                           IND00330
MODEL=0                                           IND00340
IHAZE=0                                           IND00350
NLAM=0                                           IND00360
EM(1)=0.                                          IND00370
TM(1)=0.                                          IND00380
EM(2)=0.                                          IND00390
TM(2)=0.                                          IND00400
ZENTH=0.                                          IND00410
CLDHGT=0.                                         IND00420
PHASE=0.                                          IND00430
RTARG=0.                                          IND00440
COSX=0.                                           IND00450
COSY=0.                                           IND00460
COSZ=0.                                           IND00470
A0(1)=0.                                          IND00480
A1(1)=0.                                          IND00490
IALB(1)=0.                                        IND00500
A0(2)=0.                                          IND00510
A1(2)=0.                                          IND00520
IALB(2)=0.                                        IND00530
ALT=0.                                             IND00540
THETA=0.                                          IND00550
PHI=0.                                             IND00560
SANG2=0.                                          IND00570
IZERO=1                                           IND00580
9 CONTINUE                                        IND00590
WRITE(IOOUT,600)                                  IND00600
600 FORMAT(1H0, 'SPOT CONTROL CARDS READ FOR THIS RUN: '//) IND00610
DO 10 I=1,7                                       IND00620
READ (IOIN,11) (DAT(J), J=1,7)                   IND00630
11 FORMAT (A4,6X,7E10.4)                          IND00640
WRITE(IOOUT,610)(DAT(J), J=1,7)                  IND00650
610 FORMAT(1H, A4,6X,7E10.4)                     IND00660
IF (DAT(1).EQ.TYPE(1)) GO TO 1                    IND00670
IF (DAT(1).EQ.TYPE(2)) GO TO 2                    IND00680
IF (DAT(1).EQ.TYPE(3)) GO TO 3                    IND00690
IF (DAT(1).EQ.TYPE(4)) GO TO 4                    IND00700

```

	IF (DAT(1).EQ.TYPE(5)) GO TO 5	IND00710
	IF (DAT(1).EQ.TYPE(6)) GO TO 6	IND00720
	IF (DAT(1).EQ.TYPE(7)) GO TO 7	IND00730
C	ERROR RETURN	IND00740
	WRITE (IOOUT,101)	IND00750
101	FORMAT(1H,48HINCORRECT INPUT CARD FOR SPOT, CONTROL RETURNED ,	IND00760
	1,18HTO MAIN FROM INDAT)	IND00770
	IHAZE=9	IND00780
	RETURN	IND00790
C	OPERATING ENVIRONMENT	IND00800
1	ISORC=IFIX(DAT(2))	IND00810
	ITARG=IFIX(DAT(3))	IND00820
	IHAZE=IFIX(DAT(4))	IND00830
	MODEL=IFIX(DAT(5))	IND00840
	NLAM=IFIX(DAT(6))	IND00850
	ID=IFIX(DAT(7))	IND00860
	GO TO 8	IND00870
C	EMISSIVITY AND TEMPERATURE OF GROUND AND TARGET, RESPECTIVELY	IND00880
2	EM(1)=DAT(2)	IND00890
	TM(1)=DAT(3)	IND00900
	EM(2)=DAT(4)	IND00910
	TM(2)=DAT(5)	IND00920
	TBOUND=TM(1)	IND00930
	GO TO 8	IND00940
C	INCIDENT ZENITH ANGLE OF RADIATION, CLOUD BOTTOM HEIGHT,	IND00950
3	PHASE ANGLE OF MOON, OPTIONAL EXTN COEF (VALID WHEN IHAZE=8)	IND00960
	ZENITH=DAT(2)	IND00970
	CLDHGT=DAT(3)	IND00980
	PHASE=DAT(4)	IND00990
	BETAEX=DAT(5)	IND01000
	GO TO 8	IND01010
C	TARGET PROPERTIES	IND01020
4	RTARG=DAT(2)	IND01030
	COSX=DAT(3)	IND01040
	COSY=DAT(4)	IND01050
	COSZ=DAT(5)	IND01060
	GO TO 8	IND01070
C	ALBEDO COEFFICIENTS AND TYPE OF REFLECTION SURFACE FOR	IND01080
5	GROUND AND TARGET, RESPECTIVELY	IND01090
	A0(1)=DAT(2)	IND01100
	A1(1)=DAT(3)	IND01110
	IALB(1)=IFIX(DAT(4))	IND01120
	A0(2)=DAT(5)	IND01130
	A1(2)=DAT(6)	IND01140
	IALB(2)=IFIX(DAT(7))	IND01150
	GO TO 8	IND01160
C	SENSOR CHARACTERISTICS	IND01170
6	ALT=DAT(2)	IND01180
	THETA=DAT(3)	IND01190
	PHI=DAT(4)	IND01200
		IND01210
C	EXPECTING INPUT AZIMUTH IN METEOROLOGICAL CONVENTION	IND01220
C	(I.E., N = 0 DEG, E = 90 DEG, S = 180 DEG, W = 270 DEG),	IND01230
C	SO CONVERT TO MATHEMATICAL CONVENTION FOR PURPOSES OF	IND01240
C	SPOT (ASSUMING Y-AXIS IS POSITIVE NORTHWARD, X-AXIS POSI-	IND01250
C	TIVE EASTWARD).	IND01260
		IND01270
	PHI=90.-PHI	IND01280
	SANGZ=DAT(5)	IND01290
8	CONTINUE	IND01300
10	CONTINUE	IND01310
7	CONTINUE	IND01320
	IEMISS=0	IND01330
	IM=0	IND01340
	LEN=0	IND01350
	ML=0	IND01360
	IF (ISORC.GT.1) IEMISS=1	IND01370
	COSX=COS(COSX*PIRAD)	IND01380
	COSY=COS(COSY*PIRAD)	IND01390
	COSZ=COS(COSZ*PIRAD)	IND01400

C	GEOMETRICAL OPTION	IND01410
	IF<IGEOSW.NE.1>GO TO 311	IND01420
	RTARG=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+	IND01430
	+(PTS(6)-PTS(3))**2)	IND01440
	THETA=ACOS((PTS(3)-PTS(6))/RTARG)	IND01450
	RTDCON=1.0/PIRAD	IND01460
	THETA=THETA*RTDCON	IND01470
	ALT=PTS(6)	IND01480
	DELY=PTS(1)-PTS(4)	IND01490
	DELY=PTS(2)-PTS(5)	IND01500
	HDIS=SQRT(DELY**2+DELY**2)	IND01510
	PHI=ACOS(DELY/HDIS)	IND01520
	PHI=PHI*RTDCON	IND01530
	IF<DELY.LT.0.0>PHI=360.0-PHI	IND01540
311	CONTINUE	IND01550
	IF<ITARG.EQ.1> RTARG=ALT/ABS(COS(THETA*PIRAD))	IND01560
	IF<ITARG.EQ.0> RTARG=1000.0	IND01570
	RETURN	IND01580
400	CONTINUE	IND01590
	REWIND IPHFUN	IND01600
	DO 500 I=1,IWN	IND01610
500	PF(I)=0.	IND01620
	IF<NLAM.NE.0> CALL PFUNC(ID)	IND01630
	RETURN	IND01640
	END	IND01650

```

SUBROUTINE OUTPUT(MODEL, IHAZE, CLDHGT)
LOGICAL LMNLT
COMMON /MO1/ DUMMIE(715), MHOLD, NLHOLD, DUMMYS(1088),
+      TRANS(16,5), RADA(16,2), WAVE(16), SS(16),
1      DIR(16), RADG(16), UTEM(16), UTRF(16), BK(16),
2      PATHR(16), UERF(16), PATHR2(16), TOT(12), BKG(16)
COMMON /BKDAT/ ALT, THETA, PHI, SANG2, ZENTH, PHASE, ALB
COMMON /COMI1/ ISORC, ITARG, IWN, JHL
COMMON /ANSW2/ TTR(16), TBR(16), CNTRST(16)
COMMON /HOLRTH/ HITARG(8,3), HISORC(6,5),
1      HMODEL(5,6), HLIGHT(3), HMNLT(3), HSNLT(3),
2      HTRGT(2), HTYPE(2), HGRND(2)
COMMON /IOUNIT/ IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU
DATA HGRND/4H GRO,4HUND /
DATA HISORC/4HSUNL,4HIGHT,4H ONL,4HY,4H,4H
1      4HMOON,4HLIGH,4HT ON,4HLY,4H,4H,4HEMIS,
2      4HSION,4H ONL,4HY,4H,4H,4HSUNL,4HIGHT,
3      4H AND,4H EMI,4HSSIO,4HN,4HMOON,4HLIGH,4HT AN,
4      4HD EM,4HISSI,4HON /
DATA HITARG/4HND R,4HEFLE,4HCTAN,4HCE,4H,4H
1      4H,4H,4HGROU,4HND R,4HEFLE,4HCTAN,4HCE /,
2      4H EMI,4HSSIO,4HN,4HTARG,4HET R,4HEFLE,4HCTAN,
3      4HCE /,4H EMI,4HSSIO,4HN /
DATA HMODEL/4HTROP,4HICAL,4H,4H,4H,4HMIDA,
1      4HLTIT,4HUDE,4HSUMM,4HER,4HMIDA,4HLTIT,4HUDE /,
2      4HWINT,4HER,4HSUBA,4HRCTI,4HC SU,4HMMER,4H,
3      4HSUBA,4HRCTI,4HC WI,4HNTER,4H,4H1962,4H U.S.,
4      4H, ST,4HANDA,4HRD /
DATA HMNLT/4H MO,4HONLI,4HGHT /
DATA HSNLT/4H SU,4HNLI,4HHT /
DATA HTRGT/4H TAR,4HGET /
DATA HTYPE/4H,4H /
DATA LMNLT/ FALSE /
IF (ISORC.NE.0.AND.ISORC.NE.3) GO TO 200
DO 100 I=1,3
100 HLIGHT(I)=HSNLT(I)
GO TO 400
DO 200 I=1,3
200 HLIGHT(I)=HMNLT(I)
LMNLT=.TRUE.
400 IF (ITARG.EQ.0) GO TO 600
IF (ITARG.EQ.2) GO TO 500
HTYPE(1)=HGRND(1)
HTYPE(2)=HGRND(2)
GO TO 600
500 HTYPE(1)=HTRGT(1)
HTYPE(2)=HTRGT(2)
600 CALL DIAG
IF (MODEL.GT.7) GO TO 700
WRITE (IOOUT,1400) ISORC,(HISORC(I,ISORC+1),I=1,6),
1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,
2      (HMODEL(I,MODEL),I=1,5),IHAZE
GO TO 750
700 IF (MODEL.EQ.8) WRITE (IOOUT,1450) ISORC,(HISORC(I,ISORC+1),I=1,6),
1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE
IF (MODEL.EQ.9) WRITE (IOOUT,1500) ISORC,(HISORC(I,ISORC+1),I=1,6),
1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE
750 IF (IHAZE.GT.0) WRITE (IOOUT,1600) CLDHGT
800 WRITE (IOOUT,1700)
IF (LMNLT) WRITE (IOOUT,1800) PHASE
WRITE (IOOUT,1900) HLIGHT
DO 900 I=1,IWN
NW=10000./WAVE(I)
900 WRITE (IOOUT,2000) WAVE(I),NW,SS(I),BK(I),BKG(I)
WRITE (IOOUT,2100)
DO 1000 I=1,IWN
NW=10000./WAVE(I)
1000 WRITE (IOOUT,2200) WAVE(I),NW,UTEM(I),UTRF(I),
1      RADA(I,2),PATHR2(I),TTR(I)
WRITE (IOOUT,2300)

```

```

OUTP0010
OUTP0020
OUTP0030
OUTP0040
OUTP0050
OUTP0060
OUTP0070
OUTP0080
OUTP0090
OUTP0100
OUTP0110
OUTP0120
OUTP0130
OUTP0140
OUTP0150
OUTP0160
OUTP0170
OUTP0180
OUTP0190
OUTP0200
OUTP0210
OUTP0220
OUTP0230
OUTP0240
OUTP0250
OUTP0260
OUTP0270
OUTP0280
OUTP0290
OUTP0300
OUTP0310
OUTP0320
OUTP0330
OUTP0340
OUTP0350
OUTP0360
OUTP0370
OUTP0380
OUTP0390
OUTP0400
OUTP0410
OUTP0420
OUTP0430
OUTP0440
OUTP0450
OUTP0460
OUTP0470
OUTP0480
OUTP0490
OUTP0500
OUTP0510
OUTP0520
OUTP0530
OUTP0540
OUTP0550
OUTP0560
OUTP0570
OUTP0580
OUTP0590
OUTP0600
OUTP0610
OUTP0620
OUTP0630
OUTP0640
OUTP0650
OUTP0660
OUTP0670
OUTP0680
OUTP0690
OUTP0700

```

```

DO 1100 I=1, IWN
NW=10000./WAVE(I)
1100 WRITE (IOOUT,2200) WAVE(I),NW,RADG(I),UERF(I),
1 RADA(I,1),PATHR(I),TBR(I)
WRITE (IOOUT,2400) HLIGHT
IF (LMNLT) WRITE (IOOUT,1800) PHASE
WRITE (IOOUT,2500) HLIGHT,HLIGHT
DO 1200 I=1, IWN
NW=10000./WAVE(I)
1200 WRITE (IOOUT,2600) WAVE(I),NW,SS(I),DIR(I)
WRITE (IOOUT,2700)
DO 1300 I=1, IWN
NW=10000./WAVE(I)
1300 WRITE (IOOUT,2800) WAVE(I),NW,TTR(I),TBR(I),CNTRST(I)
WRITE (IOOUT,2900) (TOT(I),I=1,10),TOT(12),HLIGHT,
1 TOT(11)
RETURN
C
1400 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1 8HFOLLOWS:,,43X,10(1H-),6H -- ,7(1H-),2X,
2 10(1H-),2X,7(1H-),,,,43X,9HPARAMETER,3X,
3 5HVALUE,3X,11HDESCRIPTION,,43X,9(1H-),3X,5(
4 1H-),3X,11(1H-),,,45X,5HISORC,7X,11,5X,6A4,/,
5 45X,5HITARG,7X,11,5X,8A4,/,45X,5HMODEL,7X,
6 11,5X,5A4,/,45X,5HIHAZE,7X,11,/)
1450 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1 8HFOLLOWS:,,43X,10(1H-),6H -- ,7(1H-),2X,
2 10(1H-),2X,7(1H-),,,,43X,9HPARAMETER,3X,
3 5HVALUE,3X,11HDESCRIPTION,,43X,9(1H-),3X,5(
4 1H-),3X,11(1H-),,,45X,5HISORC,7X,11,5X,6A4,/,
5 45X,5HITARG,7X,11,5X,8A4,/,45X,5HMODEL,7X,
6 11,5X,32HISRAELI STANDARD (YEAR, DAYTIME)//45X,5HIHAZE,7X,11/)
1500 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1 8HFOLLOWS:,,43X,10(1H-),6H -- ,7(1H-),2X,
2 10(1H-),2X,7(1H-),,,,43X,9HPARAMETER,3X,
3 5HVALUE,3X,11HDESCRIPTION,,43X,9(1H-),3X,5(
4 1H-),3X,11(1H-),,,45X,5HISORC,7X,11,5X,6A4,/,
5 45X,5HITARG,7X,11,5X,8A4,/,45X,5HMODEL,7X,
6 11,5X,34HISRAELI STANDARD (YEAR, NIGHTTIME)//45X,5HIHAZE,7X,11/)
1600 FORMAT (1H,45X,22H CLOUD BOTTOM HEIGHT = ,F5.3,3H KM)
1700 FORMAT (1H1,56X,19HSOURCE INTENSITIES,/,56X,7H-----
1 12H -----)
1800 FORMAT (44X,27H PHASE ANGLE FOR MOONLIGHT: ,F6.2,
1 10H (DEGREES),/)
1900 FORMAT (15X,10HWAVELENGTH,3X,10HWAVENUMBER,3X,3A4,
1 6HSOURCE,11X,13HTARGET SOURCE,15X,7HGROUND
2 6HSOURCE,/,15X,9H(MICRONS),6X,6H(CM-1),11X,
3 8HSTRENGTH,17X,8HSTRENGTH,20X,8HSTRENGTH,/,
4 41X,20H(WATTS M-2 MICRON-1),2(3X,9H(WATTS M-
5 16H2 MICRON-1 SR-1)),/,15X,10(1H-),3X,10(1H-),
6 3X,20(1H-),3X,25(1H-),3X,25(1H-),/)
2000 FORMAT (15X,1PE10.4,3X,17,11X,1PE10.4,15X,1PE10.4,18X,
1 1PE10.4)
2100 FORMAT (1H1,46X,33HCOMPONENTS FOR RADIANCE FROM
1 6HTARGET,/,46X,29H-----
2 10H-----,/,53X,23H(WATTS M-2 MICRON-1 SR-
3 2H1),,,,22X,10HWAVELENGTH,3X,10HWAVENUMBER,
4 5X,6HTARGET,7X,6HTARGET,7X,7HPARTIAL,6X,
5 7HPARTIAL,6X,5HTOTAL,/,22X,9H(MICRONS),6X,
6 6H(CM-1),6X,8HEMISSION,4X,11HREFLECTANCE,2X,
7 11HATMOSPHERIC,5X,4HPATH,8X,6HTARGET,/,75X,
8 8HEMISSION,5X,8HRADIANCE,5X,8HRADIANCE,/,
9 22X,10(1H-),3X,10(1H-),4X,8(1H-),4X,11(1H-),
X 2X,11(1H-),3X,8(1H-),5X,8(1H-),/)
2200 FORMAT (22X,1PE10.4,3X,17,3X,1P5E13.4)
2300 FORMAT (1H1,47X,29HCOMPONENTS FOR BACKGROUND
1 8HRADIANCE,/,47X,27H-----
2 10H-----,/,53X,23H(WATTS M-2 MICRON-1 SR-
3 2H1),,,,22X,10HWAVELENGTH,3X,10HWAVENUMBER,
4 5X,6HGROUND,7X,6HGROUND,8X,5HTOTAL,8X,5HTOTAL,

```

```

5      7X, 5HTOTAL, /, 22X, 9H(MICRONS), 6X, 6H(CM-1), 6X,      OUTP1410
6      8HEMISSION, 4X, 11HREFLECTANCE, 2X, 11HATMOSPHERIC      OUTP1420
7      5X, 4HPATH, 6X, 10HBACKGROUND, /, 75X, 8HEMISSION,      OUTP1430
8      2(5X, 8HRADIANCE), /, 22X, 10(1H-), 3X, 10(1H-), 4X,      OUTP1440
9      8(1H-), 4X, 11(1H-), 2X, 11(1H-), 3X, 8(1H-), 4X, 10(      OUTP1450
X      1H-), /, /)      OUTP1460
2400  FORMAT (1H1, 58X, 6HDIRECT, 3A4, /, 57X, 6(1H-), 1X, 12(1H-), /      OUTP1470
1      /, 56X, 20H(WATTS M-2 MICRON-1), /, /)      OUTP1480
2500  FORMAT (41X, 10HWAVELENGTH, 3X, 10HWAVENUMBER, 2X, 3A4, 1X,      OUTP1490
1      3A4, /, 41X, 9H(MICRONS), 6X, 6H(CM-1), 7X, 6HSOURCE,      OUTP1500
2      8X, 4HFLUX, /, 68X, 8HSTRENGTH, /, 41X, 10(1H-), 3X,      OUTP1510
3      10(1H-), 2X, 12(1H-), 1X, 12(1H-), /, /)      OUTP1520
2600  FORMAT (41X, 1PE10.4, 3X, 17, 6X, 1PE10.4, 3X, 1PE10.4)      OUTP1530
2700  FORMAT (1H1, 58X, 15HTOTAL RADIANCE, /, 58X, 11H-----      OUTP1540
1      4H-----, /, 53X, 25H(WATTS M-2 MICRON-1 SR-1), /, /, /      OUTP1550
2      , 35X, 10HWAVELENGTH, 3X, 10HWAVENUMBER, 5X,      OUTP1560
3      6HTARGET, 5X, 10HBACKGROUND, 4X, 8HCONTRAST, /      OUTP1570
4      , 35X, 9H(MICRONS), 6X, 6H(CM-1), 33X, 5HRATIO, /      OUTP1580
5      , 35X, 10(1H-), 3X, 10(1H-), 5X, 6(1H-), 5X, 10(1H-),      OUTP1590
6      4X, 8(1H-), /, /)      OUTP1600
2800  FORMAT (35X, 1PE10.4, 3X, 17, 3X, 1P3E13.4)      OUTP1610
2900  FORMAT (1H1, 46X, 30HDETECTOR-RESPONSE WAVELENGTH-      OUTP1620
1      10HINTEGRATED, /, 46X, 17(1H-), 2X, 21(1H-), /, /      OUTP1630
2      , 58X, 16H(WATTS M-2 SR-1), /, /, 46X, 7HTARGET      OUTP1640
3      8HEMISSION, 16X, 1PE10.4, /, 46X, 7HTARGET      OUTP1650
4      11HREFLECTANCE, 13X, 1PE10.4, /, 46X, 8HPARTIAL      OUTP1660
5      20HATMOSPHERIC EMISSION, 3X, 1PE10.4, /, 46X,      OUTP1670
6      21HPARTIAL PATH RADIANCE, 10X, 1PE10.4, /, 46X,      OUTP1680
7      21HTOTAL TARGET RADIANCE, 10X, 1PE10.4, /, 46X,      OUTP1690
8      15HGROUND EMISSION, 16X, 1PE10.4, /, 46X, 7HGROUND      OUTP1700
9      11HREFLECTANCE, 13X, 1PE10.4, /, 46X, 6HTOTAL      OUTP1710
X      20HATMOSPHERIC EMISSION, 5X, 1PE10.4, /, 46X,      OUTP1720
1      19HTOTAL PATH RADIANCE, 12X, 1PE10.4, /, 46X,      OUTP1730
2      25HTOTAL BACKGROUND RADIANCE, 6X, 1PE10.4, /, /      OUTP1740
3      , 46X, 41(1H*), /, /, 46X, 8HCONTRAST, 22X, 1PE11.4, /, /      OUTP1750
4      , 46X, 41(1H*), /, /, 46X, 6HDIRECT, 3A4, 13X, 1PE10.4, /      OUTP1760
5      , 46X, 11H(WATTS M-2))      OUTP1770
      END      OUTP1780

```

```

SUBROUTINE PATHRD(CTHED,HP,RT,IO,IHAZE,NR,
1 IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
LOGICAL HORIZ
DIMENSION TR1(16),TR2(16),DUMMY(16),ANS(16)
COMMON /BASPO/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),
1 BE(16),SINGWV,SAER(65),LOUM
COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
COMMON /CGEOM/COSGM,COSAT,CSA
COMMON /COMI1/ISORC,ITARG,IWN,JHL
COMMON /MQ1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,
+ RRS(16,34),SCOE(16,34),
+ TRANS(16,5),RADA(16,2),WAVE(16),SS(16),
1 DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),
2 PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)
C **** CONST1 = (1.-.0295)*.5*3.14159/(1.+5*.0295)
C **** CONST2 = .0295/(1.+5*.0295)*3.14159
DATA CONST1,CONST2/.0570805145,.0034701189/
NLL=NL-1
SRAYL=CONST1*(1.+CSA*CSA)+CONST2
C **** INITIALIZE VARIABLES ****
DO 800 IW=1,IWN
800 PATHR(IW)=0.0
DO 900 J=2,NLL
900 IF (HP.LT.Z(J)) GO TO 1000
CONTINUE
J=NLL
JU=J
JL=JU-1
HORIZ=.FALSE.
DS1=0.0
H2=HP
RAT=(H2-Z(JL))/(Z(JU)-Z(JL))
GO TO (1100,1200,1300),IO
1100 IX=RT*1.99999+1
DX=RT/FLOAT(IX)
DH=DX*CTHED
HORIZ=ABS(CTHED).LT.1.E-3
DIST=0.5*DX
GO TO 1400
1200 IX=NLL-JL
DH=Z(JU)-HP
DX=DH/CTHED
GO TO 1400
C HORIZONTAL PATH FOLLOWS
1300 IX=100
DIST=DIST-0.5*DX+0.25
DX=0.5
HORIZ=.TRUE.
GO TO 1500
1400 H2=HP+DH*0.5
1500 DO 2600 K=1,IX
IF (HORIZ) GO TO 2000
CALL LT4M(HP,H2,THETA,2,2,TR1,DUMMY,DUMMY,
1 IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,ICLMAT,IERR,
2 NR,IHAZE,MULDV)
CALL LT4M(H2,0UM,ZENTH,3,2,TR2,DUMMY,DUMMY,
1 IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,ICLMAT,IERR,
2 NR,IHAZE,MULDV)
IF (IO.EQ.2) GO TO 1800
DO 1600 J=2,NLL
IF (H2.LT.Z(J)) GO TO 1700
1600 CONTINUE
J=NLL
JU=J
JL=JU-1
RAT=(H2-Z(JL))/(Z(JU)-Z(JL))
1800 DO 1900 IW=1,IWN
SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL))
RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))
SSCAT=RS*SRAYL+(1.0-RS)*SAER(IW)

```


1900	PATHR(IW)=PATHR(IW)+TR1(IW)*TR2(IW)*SC*SSCAT*DX	PAD00710
	GO TO (2400,2500,2300),IO	PAD00720
2000	CONTINUE	PAD00730
	CALL LT4M(HP,DUM,DIST,1,2,TR1,DUMMY,DUMMY,	PAD00740
1	IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,ICLMAT,IERR,	PAD00750
2	NR,IHAZE,MULDV)	PAD00760
	IF (K.EQ.1) CALL LT4M(HP,DUM,ZENTH,3,2,TR2,DUMMY,DUMMY,	PAD00770
1	IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,	PAD00780
2	ICLMAT,IERR,	PAD00790
NR	NR,IHAZE,MULDV)	PAD00800
	DS=0.0	PAD00810
	DO 2200 IW=1,IWN	PAD00820
	IF (K.GT.1) GO TO 2100	PAD00830
	SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL))	PAD00840
	RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))	PAD00850
	ANS(IW)=TR2(IW)*SC*(RS*SRAYL+(1.0-RS)*SAER(IW))	PAD00860
2100	DPATH=ANS(IW)*TR1(IW)*DX	PAD00870
	DS=DS+DPATH	PAD00880
2200	PATHR(IW)=PATHR(IW)+DPATH	PAD00890
	DS1=DS1+DS	PAD00900
	DS=DS*0.5/DX	PAD00910
	IF (IO.EQ.3.AND.K.GT.1.AND.DS/DS1.LT.0.001) RETURN	PAD00920
	IK=(K/20)*20	PAD00930
	IF (IO.EQ.3.AND.IK.EQ.K) DX=DX*2.0	PAD00940
2300	DIST=DIST+DX	PAD00950
	GO TO 2600	PAD00960
2400	H2=H2+DH	PAD00970
	GO TO 2600	PAD00980
2500	JU=JU+1	PAD00990
	JL=JL+1	PAD01000
	IF (K.EQ.1) RAT=0.5	PAD01010
	H2=(Z(JU)+Z(JL))*0.5	PAD01020
	DX=(Z(JU)-Z(JL))/CTHED	PAD01030
2600	CONTINUE	PAD01040
	RETURN	PAD01050
	END	PAD01060

```
SUBROUTINE ZERO  
COMMON /MO1/ DUMMIE(715),M,NL,DUMMYS(1088),RDATA(300)  
DO 100 I=1,300  
RDATA(I)=0.0  
100 CONTINUE  
RETURN  
END
```

```
ZER00010  
ZER00020  
ZER00030  
ZER00040  
ZER00050  
ZER00060  
ZER00070
```

SUBROUTINE CLIMAT(LOCAT, MONTH, NHOUR, IWIND, NPRT, TEMP, PRESS, RH, AH, DP, CL100010
 1, VIS, WNDVEL, WINDIR, IPASCT) CL100020

***** CL100030

CLIMATOLOGY MODULE - CLIMAT CL100040

PURPOSE - TO PROVIDE THE CENTRAL EUROPEAN AND MID-EASTERN CLIMATOLOGY DATA REQUIRED BY OTHER MODULES OF EOSAEL. CL100050
 CL100060
 CL100070
 CL100080
 CL100090

PARAMETER DESCRIPTION CL100100

LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER CL100110
 (1-4) FOR CENTRAL EUROPE AND CL100120
 (5-10) FOR MID-EAST. CL100130
 CL100140
 CL100150

- REGION 1 - EUROPEAN LOWLANDS, CL100160
- REGION 2 - EUROPEAN RHINE VALLEY, CL100170
- REGION 3 - EUROPEAN HIGHLANDS, CL100180
- REGION 4 - EUROPEAN ALPINE, CL100190
- REGION 5 - MIDEAST DESERTS, CL100200
- REGION 6 - MIDEAST COASTAL, CL100210
- REGION 7 - MIDEAST PERSIAN GULF, CL100220
- REGION 8 - MIDEAST RED SEA, CL100230
- REGION 9 - MIDEAST EASTERN MOUNTAINS, AND CL100240
- REGION 10 - MIDEAST INDUS VALLEY. CL100250

MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR. CL100260
 MONTH IS USED TO SELECT THE SEASON WHICH IS CL100270
 APPLICABLE TO THE REGION LOCAT. CL100280

NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL CL100290
 STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OF CL100300
 FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14, CL100310
 AND 15-19. CL100320

IWIND - *** NOT USED *** CL100330

NPRT - A PRINT SELECTOR. CL100340
 NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA. CL100350
 NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD CL100360
 DEVIATIONS, AND PERCENT OCCURRENCES. CL100370
 CL100380

TEMP - MEAN TEMPERATURE (C). CL100390
 PRESS - MEAN SEA LEVEL PRESSURE (MB). CL100400
 RH - MEAN RELATIVE HUMIDITY (PERCENT). CL100410
 AH - MEAN ABSOLUTE HUMIDITY (GM/CM³). CL100420
 DP - MEAN DEW-POINT TEMPERATURE (C). CL100430
 VIS - MEAN HORIZONTAL VISIBILITY (KM). CL100440
 WNDVEL - MEAN WIND SPEED (MPS). CL100450
 WINDIR - MOST PROBABLE WIND DIRECTION (DEGREES). WINDIR IS CL100460
 GIVEN IN 30 DEGREE INCREMENTS (015,045,075,... CL100470
 345). CL100480

IPASCT - INDICATOR (1-6) FOR THE MOST PROBABLE PASQUILL CL100490
 STABILITY CATEGORY (A-F). CL100500

CLDHT - MEAN CLOUD HEIGHT (KM). CL100510
 CLDCVR - MEAN TOTAL CLOUD COVER (PERCENT). CL100520
 WNDDIR - WIND DIRECTION (DEGREES). CL100530
 CL100540

SUBROUTINES AND FUNCTIONS - NONE CL100550

CARD INPUT - NONE CL100560

TAPE INPUT - YES. BE SURE TO ASSIGN THE CLIMATOLOGY DATA TAPE TO CL100570
 UNIT NCLINT. CL100580
 CL100590
 CL100600

***** CL100610

COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLINT, KSTOR, NPLOU CL100620
 DIMENSION REGION(70), SEASON(8), HOUR(8), DATA(18), DIR(13) CL100630
 CL100640

DATA REGION/4HEURO, 4HPEAN, 4H LOW, 4HLAND, 4HS CL100650
 1 2*4H , 4HEURO, 4HPEAN, 4H RHI, 4HNE V, 4HALLE, 4HY , 4H , 4HEURO, CL100660
 2 4HPEAN, 4H HIG, 4HHLAN, 4HDS , 2*4H , 4HEURO, 4HPEAN, 4H ALP, 4HINE , CL100670
 3 3*4H , 4HMIDE, 4HAST , 4HDESE, 4HRYS , 3*4H , 4HMIDE, 4HAST , 4HCOACLI00700

```

4S,4HTAL,3*4H,4HMIDE,4HAST,4HPERS,4HIAN,4HGULF,2*4H,4HMI00710
5DE,4HAST,4HRED,4HSEA,3*4H,4HMIDE,4HAST,4HEAST,4HERN,4HMOU00720
6N,4HTAIN,4HS,4HMIDE,4HAST,4HINDU,4HS VA,4HLEY,2*4H,00730
DATA SEASON/4HWINT,4HER,4HSPRI,4HNG,4HSUMM,4HER,4HAUTU,4HMN00740
1/00750
DATA HOUR/4H20-0,4H2,4H03-0,4H9,4H10-1,4H4,4H15-1,4H9,00760
DATA DIR/4H 015,4H 045,4H 075,4H 105,4H 135,4H 165,4H 195,4H 225,00770
14H 255,4H 285,4H 315,4H 345,4H VBL/00780
POSITION THE TAPE NCLIMT FOR READING00790
REWIND NCLIMT00800
SKIP OVER ALL DATA FOR REGIONS 1,2,...,LOCAT-100810
IF(LOCAT.LT.1.OR.LOCAT.GT.10) LOCAT=100820
LSKIP=1056*(LOCAT-1)00830
SKIP OVER DATA FOR SEASONS 1,2,...,SEASON-1 FOR REGION LOCAT00840
NSEASN=100850
IF(MONTH.GE.3.AND.MONTH.LE.5) NSEASN=200860
IF(MONTH.GE.6.AND.MONTH.LE.8) NSEASN=300870
IF(MONTH.GE.9.AND.MONTH.LE.10) NSEASN=400880
IF(LOCAT.GE.5.AND.MONTH.EQ.11) NSEASN=400890
NSKIP=LSKIP+176*(NSEASN-1)00900
SKIP OVER DATA FOR TIME PERIODS 0,1,...,PERIOD-1 FOR REGION00910
LOCAT DURING SEASON.00920
NTIME=100930
IF(NHOUR.GE.3.AND.NHOUR.LE.9) NTIME=200940
IF(NHOUR.GE.10.AND.NHOUR.LE.14) NTIME=300950
IF(NHOUR.GE.15.AND.NHOUR.LE.19) NTIME=400960
NSKIP=NSKIP+44*(NTIME-1)00970
IF(NSKIP.LE.0) GO TO 200980
DO 1 J=1,NSKIP00990
READ(NCLIMT,9) A01000
1 CONTINUE01010
IF NPRT GT 0, PRINT A HEADING FOR THE THERMODYNAMIC DATA01020
2 CONTINUE01030
IF(NPRT.LE.0) GO TO 301040
WRITE(IOOUT,15)01050
ILOC=7*LOCAT-601060
ILO6=ILOC+601070
WRITE(IOOUT,10) (REGION(J),J=ILOC,ILOC6),SEASON(2*NSEASN-1),01080
1SEASON(2*NSEASN),HOUR(2*NTIME-1),HOUR(2*NTIME)01090
READ THE THERMODYNAMIC DATA FOR REGION LOCAT AT NTIME01100
DURING NSEASN.01110
3 DO 4 J=1,2201120
READ(NCLIMT,11) NCLASS,(DATA(K),K=1,18)01130
CONVERT FROM METERS TO KILOMETERS01140
DATA(6)=0.001*DATA(6)01150
DATA(10)=0.001*DATA(10)01160
IF NPRT GT 0, PRINT THE THERMODYNAMIC DATA01170
IF(NPRT.LE.0) GO TO 401180
WRITE(IOOUT,12) NCLASS,(DATA(K),K=1,18)01190
4 CONTINUE01200
EXTRACT THE VALUES OF TEMP, PRESS, RH, AH, DP, VIS, AND01210
WINDVEL.01220
01230
01240
01250
01260
01270
01280
01290
01300
01310
01320
01330
01340
01350
01360
01370
01380
01390
01400

```

```

TEMP=DATA(2)
DP=DATA(3)
AH=DATA(4)
RH=DATA(5)
VIS=DATA(6)
PRESS=DATA(7)
WINDVEL=DATA(8)
CCC
    DETERMINE THE VALUE OF IPASCT
IPASCT=1
FREQ=DATA(13)
DO 5 J=2,6
IF<DATA<J+12>,LE.FREQ> GO TO 5
FREQ=DATA<J+12>
IPASCT=J
5 CONTINUE
CCC
    GET TO THE WIND DATA ON NCLIMT
NSKIP=(NSKIP-LSKIP)/44
NSKIP=44*(15-NSKIP)+22*NSKIP
DO 16 J=1,NSKIP
READ<NCLIMT,9> A
16 CONTINUE
CCC
    IF NPRT GT 0, PRINT A HEADING FOR THE WIND DATA
IF<NPRT,LE.0> GO TO 6
WRITE<IOOUT,13> <DIR<J>,J=1,13>
CCC
    READ THE WIND DATA FOR REGION LOCAT AT NTIME DURING NSEASN
6 DO 7 J=1,22
READ<NCLIMT,14> NCLASS,<DATA<K>,K=1,14>
    IF NPRT GT 0, PRINT THE WIND DATA
IF<NPRT,LE.0> GO TO 7
WRITE<IOOUT,17> NCLASS,<DATA<K>,K=1,14>
7 CONTINUE
CCC
    DETERMINE THE VALUE OF WINDIR
NDIR=1
FREQ=DATA(2)
DO 8 J=2,12
IF<DATA<J+1>,LE.FREQ> GO TO 8
FREQ=DATA<J+1>
NDIR=J
8 CONTINUE
WINDIR=30*NDIR-15
CCC
    RETURN FROM CLIMAT
RETURN
CCC
    FORMAT STATEMENTS
9 FORMAT(A1)
10 FORMAT(25H1 EOSAEL CLIMATOLOGY FOR ,7A4,8H DURING ,2A4,4H AT ,2A4,
17H<LST>, //126H CLASS FREQCY MEAN MEAN MEAN MEAN CLIO2020
2 MEAN MEAN/STDEV MEAN MEAN/STDEV FREQCY FREQCY FREQCY FREQCY FREQCY CLIO2030
3CY FREQCY/126H NO. CLASS TEMP DP AH C RH D VIS PCLIO2040
4RESS WINDVEL CLDHT CLDCVR A B C D E CLIO2050
5 F /126H (<C> <C> <C> <GM/CM> (<C> <C> <C> <C> CLIO2060
6MB> <MPS> <KM> (<C> (<C> (<C> (<C> (<C> (<C> (<C> (<C> CLIO2070
7 (<C> /) CLIO2080
11 FORMAT(6X,13,5F5.1,F7.0,F6.1,2F5.1,F6.0/8F5.1) CLIO2090
12 FORMAT(15,F9.1,4F7.1,F7.3,F7.1,F4.1,1H/,F4.1,F7.3,F6.1,1H/,F5.1,6FCLIO2100

```

```

17.1)
13 FORMAT(1H0,7H CLASS ,14(7H FREQCY)/14H NO. CLASS,13(7H WND DIR) CLI02110
1 /10X,3H(%) ,1X,13(2%,A4,1X)/14X,13(3X,3H(%) ,1X)/ CLI02120
14 FORMAT(6X,13,14F5.1) CLI02130
15 FORMAT(18H1CLIMATOLOGY MODEL//39H DEFINITIONS OF METEOROLOGICAL CLI02140
1 CLASSES//48H CLASS 1 = FOG, HAZE AND MIST WITH VIS LT 1 KM./53H CLASS 2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM./53H CLASS 3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM./48H CLASS 4 = FOG, HAZE AND MIST WITH VIS GE 7 KM./34H CLASS 5 = DUST WITH VIS LT 3 KM./34H CLASS 6 = DUST WITH VIS GE 3 KM./53H CLASS 7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM./58H CLASS 8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM./58H CLASS 9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM./53H CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM./34H CLASS 11 = SNOW WITH VIS LT 1 KM./39H CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM./39H CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM./34H CLASS 14 = SNOW WITH VIS GE 7 KM./59H CLASS 15 = NO WEATHER CER AND ABSOLUTE HUMIDITY LT 10 GM/CM M./59H CLASS 16 = NO WEATHER GAND ABSOLUTE HUMIDITY GE 10 GM/CM M./52H CLASS 17 = VIS LT 1 KM AND ED CEILING HEIGHT LT 300 M./53H CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M./36H CLASS 19 = CEILING HEIGHT LT 300 M./37H CLASS 20 = CEILING HEIGHT LT 1000 M./23H CLASS 21 = NO CEILING./36H CLASS 22 = ALL CONDITIONS COMBINED.)
17 FORMAT(15,F9.1,13F7.1)
END

```

PROGRAM AGAUS

PROGRAM AGAUS

REVISION DATE 22 JANUARY, 1982

PURPOSE:

TO CALCULATE EXTINCTION COEFFICIENTS, ETC., AND PRODUCE LEGENDRE EXPANSION COEFFICIENTS, PHASE FUNCTIONS AND (OPTIONALLY) SCATTERING FRACTIONS UNDER A VARIETY OF CONDITIONS AND AEROSOL DISTRIBUTIONS AT ONE OR MORE WAVELENGTHS. THE PHASE FUNCTION IS NORMALIZED TO 4 PI OMEGA ZERO AND MAY BE RENORMALIZED BY DIVISION BY THE APPROPRIATE CONSTANT(S).

***** INPUT *****

CARD 1 - IDENTIFIER - 80 ALPHA CHARACTERS

CARD 2 - INTEGER CONTROL PARAMETERS: NWAWE, NINDX, IW, IDSTP, NRADI, IT, MQRTE, IANG, IEO, NEOU
FORMAT (10I5)

NWAWE: IS THE NO. OF WAVELENGTHS, OR REL. HUMIDITY VALUES TO BE TREATED IN THIS RUN. SEE COMMENTS CIRCA READ OF WAVE, DWAVE, ETC. N.B. NWAWE MUST BE LE. 10 - TO CHANGE THE NUMBER OF WAVELENGTHS CHANGE THE FIRST INDEX OF ARRAY OUT(I,J) TO AGREE WITH NWAWE.

NINDX: IS THE NBR OF AEROSOL COMPONENTS WHICH WILL HAVE DIFFERENT OPTICAL CONSTANTS, MASS DENSITIES OR MASS CONCENTRATIONS.

IW: =0 WILL SET THE REFRACTIVE INDEX OF THE AEROSOL EQUAL TO THAT OF WATER AT THE INPUT WAVELENGTH AND TEMP. IF IW .NE. 0 AND HANEL'S GROWTH FACTOR IS ZERO (EMUA=0. - CARD 5), THEN THE INPUT REFRACTIVE INDEX (EMA,CAYA) WILL BE USED FOR THE AEROSOL. OTHERWISE THE REFRACTIVE INDEX IS ADJUSTED PER HANEL (SEE BELOW).

IDSTP: IDENTIFIES TYPE OF AEROSOL SIZE DISTRIBUTION TO BE USED.

NRADI: NO. OF PARTICLE RADII TO BE EXPECTED FOR IDSTP=0 OR 3; THE INPUT VALUE OF NRADI IS IGNORED FOR IDSTP NOT ZERO OR 3.

NRADI MUST BE LE. 1+2**JDIMCK(2) - C.F. BLOCK DATA
IT: IS THE NUMBER OF GAUSS-LEGENDRE ANGLES (ORDER OF EXPANSION) IF ONLY EXTINCTION COEFFICIENTS, ETC. ARE DESIRED, I.E. NOT PHASE FUNCTIONS, THEN SET -IT- EQUAL TO ONE.

MQRTE: =12345 WILL CAUSE PRINTS OF MIE EFFICIENCY FACTORS AT EVERY VALUE OF PARTICLE RADIUS USED IN THE MIE CALCULATIONS; SET MQRTE = 0 IF SUCH PRINTS ARE NOT DESIRED.

IANG: =0 FOR COMPUTATIONS OF PHASE FN. AT -IT- GAUSS LEGENDRE QUADRATURE ANGLES; IANG=1 FOR COMPUTATIONS OF PHASE FN AT -IT- EQUALLY SPACED ANGLES BETWEEN 0 AND 180 DEGREES.

IANG=2 WILL ALLOW -IT- USER SUPPLIED ANGLES TO BE READ - FORMAT (16F5.1). THIS REQUIRES AT LEAST ONE CARD OF TYPE 2A. IF IANG.GT.0 NO LEGENDRE COEFFICIENTS WILL BE GENERATED.

IEO=1,2,3,4 WILL CONSTRUCT A PHASE FUNCTION FILE (ON NEOU).

IEO=1 65 PREDETERMINED ANGLES INDIVIDUAL WAVELENGTHS ONLY

IEO=2 65 PREDETERMINED ANGLES COMPOSITE WAVELENGTH ONLY

IEO=3 USER INPUT ANGLES INDIVIDUAL WAVELENGTHS ONLY

IEO=4 USER INPUT ANGLES COMPOSITE WAVELENGTH ONLY

IEO=5 65 PREDETERMINED ANGLES INDIVIDUAL & COMPOSITE WAVELENGTHS

THE COMPOSITE WILL BE THE LAST DATA SET WRITTEN ON UNIT -NEOU-

FOR USER INPUT ANGLES SEE -IT- AND IANG ABOVE. THE COMPOSITE VALUES ARE SIMPLE AVERAGES OVER THE NUMBER OF WAVELENGTHS.

THIS FILE WILL CONTAIN THE FOLLOWING INFORMATION:

1) ANGLES (65 MAX) - FORMAT(11(F6.2,1X))

2) NBR OF ANGLES, PHASE FUNCTION IDENTIFIER (=0= IMPLIED USER INPUT IN EOSAEL), WAVELENGTH (UM), SINGLE SCATTERING ALBEDO, EXTINCTION AND SCATTERING COEFFICIENTS IN INVERSE KM -

FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))

3) PHASE FUNCTION AT ANGLES SPECIFIED ABOVE. N.B. THE PHASE FUNCTION AS WRITTEN OUT HERE IS NORMALIZED TO 4 PI OMEGA ZERO; THE ROUTINE IN EOSAEL WILL RENORMALIZE THE PHASE FUNCTION TO ONE.

FORMAT (6(E12.6,1X))

NEOU= UNIT NUMBER UPON WHICH EOSAEL PHASE FUNCTION IS TO BE STORED.

AGX00020
AGX00030
AGX00040
AGX00050
AGX00060
AGX00070
AGX00080
AGX00090
AGX00100
AGX00110
AGX00120
AGX00130
AGX00140
AGX00150
AGX00160
AGX00170
AGX00180
AGX00190
AGX00200
AGX00210
AGX00220
AGX00230
AGX00240
AGX00250
AGX00260
AGX00270
AGX00280
AGX00290
AGX00300
AGX00310
AGX00320
AGX00330
AGX00340
AGX00350
AGX00360
AGX00370
AGX00380
AGX00390
AGX00400
AGX00410
AGX00420
AGX00430
AGX00440
AGX00450
AGX00460
AGX00470
AGX00480
AGX00490
AGX00500
AGX00510
AGX00520
AGX00530
AGX00540
AGX00550
AGX00560
AGX00570
AGX00580
AGX00590
AGX00600
AGX00610
AGX00620
AGX00630
AGX00640
AGX00650
AGX00660
AGX00670
AGX00680
AGX00690
AGX00700


```

C      INITIALIZE QUANTITIES USED IN SUMMATIONS
      DO 2 I=1,IT
      OLT(I)=0.
2     PSUMT(I)=0.E0
      WAVAVG=0.
      ALBDOT=0.
      KEXTT=0.E0
      KSCAT=0.E0
      KBAKT=0.E0
      CATTN=0.E0
      ITT=IT-1
      PI=3.1415926535898E+00
      IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) GO TO 3
      IF ((IANG.EQ.1).OR.(IANG.EQ.2)) GO TO 3
C      WHEN IANG=0 ROUTINE GUSET IS CALLED TO SET-UP THE ABSCISSAE AND
C      WEIGHTS USED FOR CALCULATING THE PHASE-FUNCTION AT -IT- POINTS
C      USED FOR NUMERICAL INTEGRATION VIA GAUSS-LEGENDRE QUADRATURE AND
C      THE PHASE FUNCTION EXPANSION COEFS, OL( ).
C      THE WEIGHTS ARE PLACED IN THE ARRAY W( ), AND THE COSINES OF THE
C      ANGLES ARE PLACED IN THE ARRAY C( ).
      CALL GUSET(IT)
      IF (ITT.LT.3) ITT=3
      GO TO 7
3     CALL ANGLE (PI,IANG,IT)
C      SUBROUTINE ANGLE IS CALLED WHEN IANG=1 OR 2 TO SET UP THE
C      ANGLES AT WHICH PHASE FUNCTIONS WILL BE CALCULATED. ANGLES
C      GO INTO ARRAY W( ) AND COSINES IN C( ).
      CONTINUE
      WRITE ANGLES FOR EOSAEL DATA FILE
      IF (IEO.LE.0) GO TO 21
      DO 22 I=1,IT
22    C(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI
      ITP1=IT+1
      IF (ITP1.GT.JDIMCK(1)) ITP1=IT
      IF (IT.LT.65.AND.JDIMCK(1).GT.65) C(ITP1)=999.99
      WRITE (NEOU,125) (C(I),I=1,ITP1)
      DO 23 I=1,IT
23    C(I)=COS(C(I)*PI/180.)
      CONTINUE
21    DETERMINE DETAILS OF AEROSOL SIZE-DISTRIBUTION VIA AGXP1
      CALL AGXP1(DENS,FSUM,VOL,JDIMCK)
      IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS
      DRYVOL IS THE AVERAGE VOLUME OF THE DRY AEROSOL PARTICLES IN
      CUBIC MICROMETERS.
      DRYVOL=VOL
      *** READ INPUT PARAMETERS ***
      READ (IOIN,105) WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA
      IF (NWAVE.EQ.1) DWAVE=0.E0
      WRITE (IOUT,106) WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA
      IF (NINDEX.GT.1) WRITE (IOUT,107) NINDEX
      IF ((DWAVE.LT.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,108) NWAVE
      IF ((DWAVE.GE.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,109) NWAVE
      IF (DENSH.LT.1E-04) WRITE (IOUT,110)
      ENWAY=FLOAT(NWAVE)
      IF (DWAVE.LT.1.E-4) GO TO 8
      WAVE=WAVE-DWAVE
8     DO 9 NWV=1,NWAVE
      IF (DWAVE.GT.1.E-4) GO TO 10
      IF (NWV.EQ.1) GO TO 11
      READ (IOIN,105) RELHUM,TEMP
      GO TO 11
10    WAVE=WAVE+DWAVE
11    VOL=DRYVOL
C      DETERMINE WHETHER THE USER SUPPLIED PARTICLE NUMBER DENSITY DENSH
C      SHOULD BE OVERRIDDEN BECAUSE THE CHOSEN IDSTP CASE HAS FIXED
C      PARAMETERS, AND/OR IF NUMBER DENSITIES ARE TO BE CALCULATED LATER
C      FROM THE AVG PARTICLE VOLUME, MASS DENSITY, AND MASS CONCENTRATION
      LLLL=0
      IF (IDSTP.EQ.6) GO TO 12
      IF (IDSTP.EQ.3.OR.IDSTP.GE.7) LLLL=1

```

```

AGX020880
AGX020900
AGX021000
AGX021100
AGX021200
AGX021300
AGX021400
AGX021500
AGX021600
AGX021700
AGX021800
AGX021900
AGX022000
AGX022100
AGX022200
AGX022300
AGX022400
AGX022500
AGX022600
AGX022700
AGX022800
AGX022900
AGX023000
AGX023100
AGX023200
AGX023300
AGX023400
AGX023500
AGX023600
AGX023700
AGX023800
AGX023900
AGX024000
AGX024100
AGX024200
AGX024300
AGX024400
AGX024500
AGX024600
AGX024700
AGX024800
AGX024900
AGX025000
AGX025100
AGX025200
AGX025300
AGX025400
AGX025500
AGX025600
AGX025700
AGX025800
AGX025900
AGX026000
AGX026100
AGX026200
AGX026300
AGX026400
AGX026500
AGX026600
AGX026700
AGX026800
AGX026900
AGX027000
AGX027100
AGX027200
AGX027300
AGX027400
AGX027500
AGX027600
AGX027700

```

```

IF (LLLL.EQ.1) GO TO 12
IF (DENSH.LE.1.E-4) GO TO 12
LLLL=1
DENS=DENSH
12 CONTINUE
C RESTRICT RELATIVE HUMIDITY TO MAX OF 99 PERCENT.
IF (RELHUM.GE.99.E+00) RELHUM=99.0E+00
WRITE (IOUT,111) RELHUM,WAVE
IF (DENS.EQ.0.0) DENS=1.0E+00
GNU=1.0E+04/WAVE
IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) DENS=ELWC
C DENS IS USED AS AN ALIAS TO PASS ELWC TO ROUTINE AGXP2.
CALL AGXP2(RELHUM,CTSUM,CSSUM,CRSUM,VOL,TMASS,DENS,WATTN,TEMP,
1 DELTA,NINDX,IM,OLSTAR,OM2,LLLL,IT,WAVE,EM,CAY,EMM,MQRTE,PFNZRO)
LMAX=3*IFIX(2.E+0*PI*EMM*R(NRADI)/WAVE)
IF (LMAX.GT.IT) WRITE (IOUT,112) LMAX,IT
13 CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,WAVE,EM,CAY,EMM,IT,0,
+ IANG)
C SUM QUANTITIES OVER INDEX NWV.
DO 14 IK=1,IT
OLT(IK)=OLT(IK)+OL(IK)
14 PSUMT(IK)=PSUMT(IK)+PSUM(IK)
C ALBDO BECOMES THE TOTAL SINGLE SCATTERING ALBEDO
C KEXTT BECOMES THE TOTAL EXTINCTION COEF. (PER KILOMETER)
C KSCAT BECOMES THE TOTAL SCATTERING COEF. (PER KM)
C KBAKT BECOMES THE TOTAL BACK-SCATTERING COEF (PER KM)
C ARRAY OUT(,) HOLDS SOME QUANTITIES FOR LATER PRINTOUTS
ALBDO=CSSUM/CTSUM
ALBDOT=ALBDOT+ALBDO
KEXTT=KEXTT+CTSUM
KSCAT=KSCAT+CSSUM
KBAKT=KBAKT+CRSUM
CATTN=CATTN+WATTN
WAVAVG=WAVAVG+WAVE
OUT(NWV,1)=WAVE
OUT(NWV,2)=RELHUM
OUT(NWV,3)=TMASS*1.E5
OUT(NWV,4)=CTSUM
IF ((NWAVE.GT.1).AND.(DWAVE.GE.1.E-04)) WRITE (IOUT,113) NWV
IF ((NWAVE.GT.1).AND.(DWAVE.LT.1.E-04)) WRITE (IOUT,114) NWV
C EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH, SINGLE SCATTERING
C ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR
C INDIVIDUAL WAVELENGTHS.
IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,127)
+ IT,WAVE,ALBDO,CTSUM,CSSUM
C EOSAEL OPTION: WRITE PHASE FUNCTION FOR INDIVIDUAL WAVELENGTHS.
IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,128)
+ (PSUM(I),I=1,IT)
9 CONTINUE
C END OF NWAVE LOOP
IF (NWAVE.LE.1) GO TO 19
C DIVIDE BY NBR OF VALUES OF NWV TO GET AVERAGED RESULTS
DO 16 I=1,IT
OLT(I)=OLT(I)/ENWAV
PSUM(I)=PSUMT(I)/ENWAV
16 CONTINUE
ALBDOT=ALBDOT/ENWAV
KEXTT=KEXTT/ENWAV
KSCAT=KSCAT/ENWAV
KBAKT=KBAKT/ENWAV
CATTN=CATTN/ENWAV
WAVAVG=WAVAVG/ENWAV
WRITE (IOUT,117) NWAVE
WRITE (IOUT,118)
DO 18 J=1,NWAVE
18 WRITE (IOUT,119) (OUT(J,J),JJ=1,4)
WRITE (IOUT,123) KEXTT,KSCAT,KBAKT,CATTN,ALBDOT
CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,WAVE,EM,CAY,EMM,IT,1,
+ IANG)
C EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH ,SINGLE SCATTERING

```

```

C   ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR          AGX03480
C   COMPOSITE VALUES                                                  AGX03490
  IF (IEO.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,127)              AGX03500
  +   IT,WAVAVG,ALBDOT,KEXTT,KSCAT                                     AGX03510
C   EOSAEL OPTION; WRITE COMPOSITE PHASE FUNCTION.                    AGX03520
  IF (IEO.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,128)            AGX03530
  +   (PSUM(I),I=1,IT)                                               AGX03540
  GO TO 19                                                             AGX03550
1   WRITE (IOUT,120) IDSTP                                           AGX03560
19  WRITE (IOUT,126)                                                 AGX03570
   STOP                                                                AGX03580
20  WRITE (IOUT,124)                                                 AGX03590
   STOP                                                                AGX03600
103 FORMAT (10I5)                                                    AGX03610
104 FORMAT(1H,78HINTEGER CONTROL PARAMETERS: NWAVE NINDX IW IDSTP NRAAGX03620
  +DI IT MORTE IANG IEO NEOU/,1X,29X,2(I2,4X),I1,3X,I2,
  +3X,I3,1X,I3,I5,1X,I5,2(2X,I2))
105 FORMAT (6E12.6)                                                  AGX03650
106 FORMAT(1H,/,17H INPUT PARAMETERS/,1X,6X,9HWAVE = ,E12.6,8H MICRAGX03660
  +ONS/,1X,6X,9HDWAVE = ,E12.6,8H MICRONS/,1X,6X,9HRELHUM = ,E12.6, AGX03670
  +3H PERCENT/,1X,6X,9HDENSH = ,E12.6,13H PARTICLES/CC/,1X,6X,
  +9HTEMP = ,E12.6,6H DEG C/,1X,6X,19HDELTA (CONVERGENCE ,
  +13HCRITERION) = ,E12.6)
107 FORMAT (/,1H,29HLOOPING OPTION IN EFFECT FOR ,I2,
  +19H AEROSOL COMPONENTS)
108 FORMAT (/,1H,39HRELATIVE HUMIDITY OPTION IN EFFECT FOR ,I2,
  +7H VALUES)
109 FORMAT (/,1H,40HWAVELENGTH LOOPING OPTION IN EFFECT FOR ,I2,
  +12H WAVELENGTHS)
110 FORMAT(1H,52H*** PARTICLE NUMBER DENSITY WILL BE CALCULATED FROMAGX03770
  +,41H MASS DENSITY AND MASS CONCENTRATION ***)
111 FORMAT (1H1,/,1X,33HRELATIVE HUMIDITY FOR THIS RUN = ,F6.2,
  +25H PERCENT, WAVELENGTH = ,F10.3,8H MICRONS,/)
112 FORMAT (</49H *** WARNING *** OPTIMAL PF EXPANSION ORDER OF ,I3,
  + 22H EXCEEDS INPUT IT = ,I3,24H. PF VALUES SHOULD BE ,
  + 15HUSED CAUTIOUSLY/)
113 FORMAT (1H,/,1X,40(1H*),3X,31HEND OF WAVELENGTH CYCLE NUMBER ,I3,
  +3X,40(1H*))
114 FORMAT (1H,/,1X,40(1H*),3X,38HEND OF RELATIVE HUMIDITY CYCLE NUMBAGX03860
  +ER ,I3,3X,40(1H*))
117 FORMAT(1H1,/,47H SUMMARY OF RESULTS FOR THIS RUN AVERAGED OVER ,
  +12,30H WAVELENGTH(S) ARE AS FOLLOWS:/)
118 FORMAT(1H,4X,48HWAVELENGTH REL.HUMIDITY AEROSOL MASS K(,AGX03900
  + 11HEXTINCTION),/1X,29H (MICROMETERS) (PERCENT)
  + 6H (GM,25H/(SQ.CM-KM) (PER KM)/)
119 FORMAT (2F15.6,1P2E16.5)
120 FORMAT (</13H *** IDSTP = ,I5,35H IS ILLEGAL. EXECUTION TERMINATEDAGX03940
  + 2H*/)
122 FORMAT (</,1X,23H*** WATER ONLY CASE ***/)
123 FORMAT (</,20H EXTINCTION COEF. = ,5X,1PE13.7,9H (PER KM),/,
  120H SCATTERING COEF. = ,8X,1PE13.7,9H (PER KM),/,
  225H BACK-SCATTERING COEF. = ,3X,1PE13.7,9H (PER KM),/,
  321H ATTENUATION COEF. = ,7X,1PE13.7,13H SQ-METERS/MG,/,
  429H SINGLE SCATTERING ALBEDO = ,1PE13.7,/)
124 FORMAT(1H,59H*** MORE THAN 65 ANGLES FOR EOSAEL OPTION - PGM TERMAGX04020
  +INATED)
125 FORMAT (<11(F6.2,1X))
126 FORMAT(1H1)
127 FORMAT(12,1X,2H00,1X,F5.2,1X,F8.6,1X,2(E12.6,1X))
128 FORMAT(6(E12.6,1X))
129 FORMAT(1H,23H*** AGAUS WARNING ****,/,1X,
  + 37HTHE ARRAY W IS ASSIGNED 65 VALUES IN,/,1X,
  + 43HBLOCK DATA WHICH IS LARGER THAN ARRAY SIZE,/,1X,
  + 47HYOU MAY BE CLOBBERING INSTRUCTIONS AND/OR DATA./)
  END
AGX04110
AGX04120

```



```

SUBROUTINE AGXP1 (DENS,FSUM,VOL,JDIMCK)
COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
+ ,NRADI,PI, IDSTP,NKG,NHALV,NI
COMMON /IO/ IOIN, IOUT, NUNIT, IEO, NEOU
DIMENSION JDIMCK(3)
EXTERNAL GAMMA
WRITE (IOUT,2)
C CHOOSE AND SET UP PARTICLE SIZE DISTRIBUTION
IF (IDSTP.NE.0) GO TO (12,15,16,17,19,19,17,20,21,22,24,26),IDSTP
C** TYPE 0: ARBITRARY USER-SUPPLIED DISTRIBUTION. NRADI VALUES OF
C R(J) AND FF(J) MUST BE GIVEN, ONE PER CARD, AND READ IN ORDER
C FROM SMALLEST RADIUS, RLO TO THE LARGEST.
C NRADI MUST BE LESS THAN OR EQUAL TO 1+2**JDIMCK(2)
WRITE (IOUT,7)
IF (NRADI.GT.JDIMCK(3)) CALL DIMER(2)
DO 9 J=1,NRADI
9 READ (IOIN,1) R(J),FF(J)
RLO=R(1)
DO 10 J=1,NRADI,5
JK=J+4
IF (JK.GT.NRADI) JK=NRADI
10 WRITE (IOUT,11)(R(K),FF(K)),K=J,JK)
WRITE (IOUT,11)
FF(NRADI+1)=FF(NRADI)
RR(1)=RLO
RR(2)=R(NRADI)
MIN=0
GO TO 28
C** TYPE 1: LOG-NORMAL DISTRIBUTION
12 READ (IOIN,1) RBAR,SIGMA,RLO,RHI
C SIGMA IS STANDARD DEVIATION, NOT LN(SIGMA)
SIGIN=SIGMA
SIGMA=ALOG(SIGMA)
A=ABS(1.E0/(2.5066283E0*SIGMA))
IF ((RHI-RLO).LE.1.E-4) GO TO 13
RR(1)=RLO
RR(3)=RHI
GO TO 14
13 RR(1)=RBAR*EXP(-4.E0*SIGMA)
RR(3)=RBAR*EXP(4.E0*SIGMA)
14 RR(2)=RBAR
MIN=1
WRITE (IOUT,3) RBAR,SIGIN,RLO,RHI
AVOL=4.18879E0*(RBAR**3.E0)*EXP(4.5E0*SIGMA*SIGMA)
C HERE AND ELSEWHERE, AVOL IS THE VOLUME OBTAINED VIA
C ANALYTICAL INTEGRATION OVER THE LIMITS RLO =0 TO RHI =
C INFINITY: THAT CAN ONLY BE DONE FOR A FEW IDSTP CASES.
GO TO 28
C** TYPE 2: DOUBLE EXPONENTIAL F(R)=CUE*A*EXP(-A*R)+(1-CUE)*B*EXP(-B*R).
C RESTRICTIONS: RHI.GT.RLO, B.GT.A.GE.0, 0.LE.CUE.LE.1.0.
15 READ (IOIN,1) RLO,RHI,CUE,A,B
WRITE (IOUT,4) RLO,RHI,CUE,A,B
RR(1)=RLO
RR(3)=RHI
RR(2)=0.5E0*(RLO+RHI)
MIN=1
GO TO 28
C** TYPE 3: DEIRMENDJIAN MODEL C. F(R) = 1.0, RLO.LE.R.LE.4*DELRD,
C F(R)=(4*DELRD/R)**4, R.GE.(4*DELRD)
C NRADI IS READ IN EARLIER IN THE MAIN PROGRAM.
16 DENS=1.378E+04
DELRD=0.02E0
RLO=0.02E0
RHI=RLO+DELRD* FLOAT(NRADI-1)
RR(1)=RLO
RR(3)=RHI
MIN=1
RR(2)=RLO+4.E0*DELRD
GO TO 28
C** TYPE 4 AND TYPE 7: POWER LAW. F(R) = CUE*R**-A
AGA00010
AGA00020
AGA00030
AGA00040
AGA00050
AGA00060
AGA00070
AGA00080
AGA00090
AGA00140
AGA00160
AGA00170
AGA00190
AGA00200
AGA00210
AGA00220
AGA00230
AGA00240
AGA00250
AGA00260
AGA00270
AGA00280
AGA00290
AGA00300
AGA00310
AGA00320
AGA00330
AGA00340
AGA00350
AGA00360
AGA00370
AGA00380
AGA00390
AGA00400
AGA00410
AGA00420
AGA00430
AGA00440
AGA00450
AGA00460
AGA00470
AGA00480
AGA00490
AGA00500
AGA00510
AGA00520
AGA00530
AGA00540
AGA00550
AGA00560
AGA00570
AGA00580
AGA00590
AGA00600
AGA00610
AGA00620
AGA00630
AGA00640
AGA00650
AGA00660
AGA00670
AGA00680
AGA00690
AGA00700
AGA00710

```

```

C      RLO,LE,R,LF,RHI; VIS=VISIBILITY IN KILOMETERS.
17    IF (IDSTP.EQ.4) GO TO 18
C**   TYPE 7 PRESCRIBED PARAMETER.
      READ (IOIN,1) VIS
      RLO=0.1E0
      RHI=15.E0
      CUE=30.E0
      A=4.E0
      DENS=11.E0**((5.E0-ALOG10(VIS)))
C**   TYPE 4 PRESCRIBED PARAMETERS.
18    IF (IDSTP.EQ.4) READ (IOIN,1) RLO,RHI,CUE,A
      WRITE (IOUT,5) RLO,RHI,CUE,A,VIS
      RR(1)=RLO
      RR(3)=RHI
      RR(2)=(0.5E0*(RLO**(-A)+RHI**(-A)))**(-1.E0/A)
      MIN=1
      GO TO 28
C**   TYPE 5: MODIFIED GAMMA/GENERALIZED KHIRGIAN-MAZIN
      F(R) = (R**ALF)*EXP(-ALF*((R/RC)**GAM)/GAM)
C**   RLO,LE,R,LF,RHI.
C**   TYPE 6: SPECIAL CASE FOR WATER FOGS OR CLOUDS,
C**   IN WHICH CASE ELWC IS LIQUID WATER CONTENT
C**   IN GRAMS PER CUBIC CENTIMETER:
C      ELWC IS IGNORED IF IDSTP = 5.
19    READ (IOIN,1) RLO,RHI,RC,ALF,GAM,ELWC
      IF (IDSTP.EQ.6) DENS=ELWC
      WRITE (IOUT,6) RLO,RHI,RC,ALF,GAM
      RR(1)=RLO
      RR(2)=RC
      RR(3)=RHI
      MIN=1
      B=ALF/(GAM*RC**GAM)
      AVOL=4.1888*B**(-3./GAM)*GAMMA((ALF+4.)/GAM)/GAMMA((ALF+1.)/GAM)
      GO TO 28
C**   TYPES 8,9,10: BIMODAL LOG-NORMAL DISTRIBUTIONS.
C      METHOD BELOW VALID FOR RBARC*EXP(-SGA).GT.RBARA*EXP(SGA)
C**   TYPE 8: CONTINENTAL BIMODAL.
20    FOA=4.E03
      FOC=2.1E0
      SGA=0.74E0
      SGC=0.81E0
      RBARA=0.03E0
      RBARC=0.4E0
      GO TO 23
C**   TYPE 9: MARITIME BIMODAL.
21    FOA=4.E02
      FOC=3.8E0
      SGA=0.68E0
      SGC=0.74E0
      RBARA=0.05E0
      RBARC=0.65E0
      GO TO 23
C**   TYPE 10: URBAN BIMODAL.
22    FOA=2.E04
      FOC=0.6E0
      SGA=0.63E0
      SGC=1.77E0
      RBARA=0.04E0
      RBARC=0.63E0
C      CALCULATE RADII FOR TYPES 8,9,10.
23    RR(1)=RBARA*EXP(-4.E0*ABS(SGA))
      RR(2)=RBARA
      RR(3)=RBARA*EXP(4.E0*ABS(SGA))
      RR(4)=RBARC*EXP(-4.E0*ABS(SGC))
      RR(5)=RBARC*EXP(4.E0*ABS(SGC))
      MIN=2
      DO 60 J=1,4
      DO 60 I=1,4
      IF (RR(I+1).GT.RR(I)) GO TO 60
      HH=RR(I)

```

```

AGA00720
AGA00730
AGA00740
AGA00750
AGA00760
AGA00770
AGA00780
AGA00790
AGA00800
AGA00810
AGA00820
AGA00830
AGA00840
AGA00850
AGA00860
AGA00870
AGA00880
AGA00890
AGA00900
AGA00910
AGA00920
AGA00930
AGA00940
AGA00950
AGA00960
AGA00970
AGA00980
AGA00990
AGA01000
AGA01010
AGA01020
AGA01030
AGA01040
AGA01050
AGA01060
AGA01070
AGA01080
AGA01090
AGA01100
AGA01110
AGA01120
AGA01130
AGA01140
AGA01150
AGA01160
AGA01170
AGA01180
AGA01190
AGA01200
AGA01210
AGA01220
AGA01230
AGA01240
AGA01250
AGA01260
AGA01270
AGA01280
AGA01290
AGA01300
AGA01310
AGA01320
AGA01330
AGA01340
AGA01350
AGA01360
AGA01370
AGA01380
AGA01390
AGA01400
AGA01410

```

```

RR(I)=RR(I+1)
RR(I+1)=HH
60 CONTINUE
GO TO 28
C** TYPE 11:
C USER SUPPLIED BIMODAL CASE: FOA AND FOC ARE THE NUMBER DENSITIES
C FOR THE ACCUMULATION (SMALLER RBAR) AND COARSE MODES,
C RESPECTIVELY, IN PARTICLES PER CUBIC CENTIMETER,
C SGA IS STD. DEVIATION FOR MODE A ** NOT LN(SIGMA) ***
C SGC IS STD. DEVIATION FOR MODE C ** NOT LN(SIGMA) ***
C *** NOTE, HOWEVER, THAT SGA AND SGC ARE THE LOGS OF THE
C STANDARD DEVIATIONS IN THE PRE-CODED CASES TYPE 8-10.
24 READ (IOIN,1) FOA,RBARA,SGA,FOC,RBARC,SGC
WRITE (IOUT,25) FOA,RBARA,SGA,FOC,RBARC,SGC
SGA=ABS(ALOG(SGA))
SGC=ABS(ALOG(SGC))
GO TO 23
C** TYPE 12: MARSHALL-PALMER RAIN MODEL,
C C. F. MASON, PHYSICS OF CLOUDS, CH. ON RADAR METEOROLOGY,
C INPUT PARAMETER RAIN IS RAIN RATE IN MILLIMETERS/HOUR:
C ** EMA, CAYA, AND RHOA ARE REQUIRED FOR THIS DISTRIBUTION.
26 READ (IOIN,1) RAIN
ENZERO=0.08E0
CAPL=41.E0*RAIN**(-0.21E0)
DENS=ENZERO/CAPL
AVOL=PI*(CAPL**(-3.E0))*1.E12
C CONVERT UNITS FROM CM-4 TO (CM-3)*(MICROMETERS**(-1)):
C THE FACTOR OF 2 CONVERTS THE M-P FORMULA FROM DIAMETER-DATA TO
C RADIUS BASED FORM.
ENZERO=2.E-4*ENZERO
CAPL=2.E-4*CAPL
MIN=0
RR(1)=1.E-4
RR(2)=2500.E0
WRITE (IOUT,27) RAIN,DENS
C THE NEXT BLOCK IS COMMON TO ALL DISTRIBUTIONS.
C IT SETS THE NMAX VALUES OF RADIUS, R(KK).
28 MAX=JDIMCK(2)
NHALV=MAX-MIN
NMAX=1+2**MAX
NI=2**MIN
IF (NMAX.GT.JDIMCK(3).OR.NI.GT.JDIMCK(3)) CALL DIMER(3)
NLAST=NI+1
NKG=2**NHALV
ENKG=FLOAT(NKG)
IF (IDSTP.EQ.0) GO TO 30
DO 29 I=1,NI
DR(I)=RR(I+1)-RR(I)
DO 29 K=1,NKG
KK=(I-1)*NKG+K
29 R(KK)=RR(I)+(FLOAT(K-1))*DR(I)/ENKG
R(NMAX)=RR(NLAST)
C BRANCH AGAIN CALCULATE THE DIFFERENT F(R) ON THE NMAX POINTS R(K)
GO TO (31,33,35,38,41,41,39,43,43,43,43,46),IDSTP
C** TYPE 0: ARBITRARY
C INTERPOLATE TO EQUAL INCREMENTS OVER RADII
30 DELR=(R(NRADI)-RLO)/ENKG
F(1)=FF(1)
NMAXM1=NMAX-1
DO 64 KK=1,NMAXM1
RADUS=RLO+DELR*FLOAT(KK)
DO 62 J=1,NRADI
K=J
IF (R(J).GE.RADUS) GO TO 61
62 CONTINUE
61 F(KK+1)=(RADUS-R(K-1))*(FF(K)-FF(K-1))/
1 (R(K)-R(K-1))+FF(K-1)
64 CONTINUE
DO 65 I=1,NI

```

```

AGA01420
AGA01430
AGA01440
AGA01450
AGA01460
AGA01470
AGA01480
AGA01490
AGA01500
AGA01510
AGA01520
AGA01530
AGA01540
AGA01550
AGA01560
AGA01570
AGA01580
AGA01590
AGA01600
AGA01610
AGA01620
AGA01630
AGA01640
AGA01650
AGA01660
AGA01670
AGA01680
AGA01690
AGA01700
AGA01710
AGA01720
AGA01730
AGA01740
AGA01750
AGA01760
AGA01770
AGA01780
AGA01790
AGA01800
AGA01810
AGA01820
AGA01830
AGA01840
AGA01850
AGA01860
PGA01870
AGA01880
AGA01890
AGA01900
AGA01910
AGA01920
AGA01930
AGA01940
AGA01950
AGA01960
AGA01870

```



```

DR(I)=RR(I+1)-RR(I)
DO 65 K=1,NKG
KK=(I-1)*NKG+K
65 R(KK)=RR(I)+( FLDAT(K-1))*DR(I)/ENKG
R(NMAX)=RR(NLAST)
GO TO 48
C** TYPE 1: LOG NORMAL
31 DEN=2.E0*SIGMA*SIGMA
DO 32 KK=1,NMAX
GNUM=ALOG(R(KK)/RBAR)
32 F(KK)=EXP(-GNUM*GNUM/DEN)*A/R(KK)
GO TO 48
C** TYPE 2: DOUBLE EXPONENTIAL
33 DO 34 KK=1,NMAX
FKK=(1.E0-CUE)*B*EXP(-B*R(KK))
34 F(KK)=FKK+CUE*A*EXP(-A*R(KK))
GO TO 48
C** TYPE 3: DEIRMENDJIAN MODEL C.
35 DO 36 KK=1,NMAX
36 F(KK)=1.E0
NKG1=NKG+1
DO 37 KK=NKG1,NMAX
37 F(KK)=(RR(2)/R(KK))*4.E0
GO TO 48
C** TYPES 4 AND 7: POWER LAW
38 GO TO 39
39 DO 40 KK=1,NMAX
40 F(KK)=CUE*R(KK)**(-A)
GO TO 48
C** TYPE 5 AND TYPE 6: MODIFIED GAMMA
41 DO 42 KK=1,NMAX
42 F(KK)=(EXP(-B*R(KK))*GAM)**R(KK)**ALF
GO TO 48
C** TYPES 8,9,10,11: BIMODAL LOG-NORMAL DISTRIBUTIONS
43 DENA=2.E0*SGA*SGA
DENC=2.E0*SGC*SGC
FAA=FOA/SGA
FCC=FOC/SGC
DO 44 KK=1,NMAX
GNUMA=ALOG(R(KK)/RBARA)
GNUMC=ALOG(R(KK)/RBARC)
FA=FAA*EXP(-GNUMA*GNUMA/DENA)
FC=FCC*EXP(-GNUMC*GNUMC/DENC)
44 F(KK)=(FA+FC)/R(KK)
DENS=FOA+FOC
WRITE (IOUT,45) DENS
VOLA=4.18879E0*(RBARA**3.E0)*EXP(4.5E0*SGA*SGA)*FOA
VOLC=4.18879E0*(RBARC**3.E0)*EXP(4.5E0*SGC*SGC)*FOC
AVOL=(VOLA+VOLC)/DENS
GO TO 48
C** TYPE 12: MARSHALL-PALMER RAIN MODEL
46 DO 47 KK=1,NMAX
47 F(KK)=ENZERO*EXP(-CAPL*R(KK))
C CALCULATE NORMALIZED F(KK) AND SOME DRY VOLUMES USING ALL NMAX
C VALUES OF RADII.
C (VOL=AVERAGE PARTICLE VOLUME IN A DISTRIBUTION). THE
C NORMALIZATION AND FURTHER VOLUMES ARE RECALCULATED LATER
C BY THE HALVING INTEGRATION METHOD.
48 FSUM=0.E0
IF(F(1).LT.0.E0)F(1)=0.0E0
DO 49 J=2,NMAX
IF(F(J).LT.0.E0)F(J)=0.E0
49 FSUM=FSUM+0.5E0*(F(J)+F(J-1))*(R(J)-R(J-1))
DO 50 J=1,NMAX
50 F(J)=F(J)/FSUM
WRITE (IOUT,8) FSUM
NRADI =NMAX
IF (IDSTP.EQ.1.OR.IDSTP.EQ.5.OR.IDSTP.GE.8) WRITE (IOUT,51) AVOL
VOL=0.E0
DO 52 J=2,NMAX

```

```

AGA01880
AGA01890
AGA01900
AGA01910
AGA01920
AGA02010
AGA02020
AGA02030
AGA02040
AGA02050
AGA02060
AGA02070
AGA02080
AGA02090
AGA02100
AGA02110
AGA02120
AGA02130
AGA02140
AGA02150
AGA02160
AGA02170
AGA02180
AGA02190
AGA02200
AGA02210
AGA02220
AGA02230
AGA02240
AGA02250
AGA02260
AGA02270
AGA02280
AGA02290
AGA02300
AGA02310
AGA02320
AGA02330
AGA02340
AGA02350
AGA02360
AGA02370
AGA02380
AGA02390
AGA02400
AGA02410
AGA02420
AGA02430
AGA02440
AGA02450
AGA02460
AGA02470
AGA02480
AGA02490
AGA02500
AGA02510
AGA02520
AGA02530
AGA02540
AGA02550
AGA02560
AGA02570
AGA02580
AGA02590
AGA02600
AGA02610
AGA02620
AGA02630
AGA02640
AGA02650

```

```

52 VOL=VOL+2.0944E0*(F(J)*R(J)**3.E0+F(J-1)*R(J-1)**3.E0)*(R(J)-R(J-1))
1
WRITE (IOUT,53) VOL
C THE VOLUME PER PARTICLE CALCULATED HERE IS OBTAINED USING
C ALL AVAILABLE (NMAX VALUES) VALUES FOR THE PARTICLE RADII.
WRITE (IOUT,54)
DO 56 INT=1,NI
INF=INT+1
56 WRITE (IOUT,55) INT,RR(INT),RR(INF)
1 FORMAT (6E12.6,I3)
2 FORMAT (1H, //24H AEROSOL PARAMETERS ARE )
3 FORMAT (1H, 24X,6HRBAR=,E12.6,5X,7HSIGMA=,E12.6,7H RLO =,E12.6, AGA02770
+ 7H RHI =,E12.6/) AGA02780
4 FORMAT (1H, 24X,5HRLO=,E10.4,1X,5HRHI=,E10.4,1X,5HCUE=,E10.4, AGA02790
+ 1X,3HA=,E10.4,1X,3HB=,E10.4/) AGA02800
5 FORMAT (1H, 24X,5HRLO=,E10.4,1X,5HRHI=,E10.4,1X,5HCUE=,E10.4, AGA02810
+ 1X,3HA=,E10.4,1X,4HVIS=,E10.4/) AGA02820
6 FORMAT (1H, 24X,5HRLO=,E10.4,1X,5HRHI=,E10.4,1X,4HRC=,E10.4,1X, AGA02830
+ 5HALF=,E10.4,1X,5HGAM=,E10.4/) AGA02840
7 FORMAT (1H, 5(26H RADIUS RELATIVE NO. ))) AGA02850
8 FORMAT (1H, 46H NORMALIZATION FACTOR FOR SIZE DISTRIBUTION =,E14.7) AGA02860
11 FORMAT (1X,10(1PE12.6,1X)) AGA02870
25 FORMAT (1X,7HN(A) =,E12.6,2X,9H RBARA =,E12.6,2X,12H SIGMA(A) =,E12.6,2X,
+ 12H SIGMA(C) =,E12.6/) AGA02890
27 FORMAT (1X,42H MARSHALL-PALMER RAIN MODEL : RAIN RATE =,1PE10.3, AGA02910
+ 21H MM PER HOUR, DENS =,1PE12.6,8H PART/CC) AGA02920
45 FORMAT (1H,50H*** BIMODAL DISTRIBUTION...EQUIVALENT DENSITY IS,1PE13.6,18H
+ 18H PARTICLES PER CC,/) AGA02930
51 FORMAT (1H,45H AVERAGE ANALYTIC DRY VOLUME PER PARTICLE IS,3X,
+ 1PE12.6,18H CUBIC MICROMETERS) AGA02940
53 FORMAT (1H,47H AVERAGE NUMERICAL DRY VOLUME IS
+ 1PE12.6,18H CUBIC MICROMETERS/) AGA02970
54 FORMAT (1X,10X,35H SIZE-INTERVALS USED ARE AS FOLLOWS/) AGA02980
55 FORMAT (1H,14H INTERVAL NO. ,I3,5X,7HRMIN =,F11.5,5X,8H RMAX =
+ ,F11.5) AGA03000
RETURN AGA03010
END AGA03020
AGA03030

```

```

SUBROUTINE AGXP2(RELHUM, CTSUM, CSSUM, CRSUM, TVOL, TMASS, DENS, CATTN,
1 TEMP, DELTA, NINDX, IW, OLSTAR, OM2, LLLL, IT, WAVE, EM, CAY, EMM, MORTE,
2 PFNZRO)
REAL KEXT, KEXTT, KEXOLD
N.B. FFF IS AN ALIS FOR ARRAY FF
COMMON /PT1/ F(513), R(513), DR(8), RR(9), FFF(514),
+NRADI, PI, IDSTP, NKG, NHALV, NI
COMMON /PT2/ PHH(65), PSUMTT(65), PGG(65), PSUM(65), PSUMT(65),
1P(65)
COMMON /IO/ IOIN, IOOUT, NUNIT, IEO, NEQU
COMMON /AGXM/ C(65), W(65), OLT(65), JDIMCK(3)
IN THIS SUBROUTINE THE FOLLOWING CONVENTIONS ARE USED IN
PREFIXING VARIABLE NAMES:
THE LETTER C IS USED FOR CROSS-SECTIONS
THE LETTER Q IS USED FOR EFFICIENCY FACTORS
THE LETTER K IS USED FOR EXTINCTION COEF. PER UNIT PATH (KM)
THE LETTER T IS A SUFFIX FOR TOTAL VALUES
THE LETTER O IS A PREFIX FOR OMEGA SUB 1 AND 2 CALCULATIONS
FOR THE IDSTP=6 AND 12 CASES, DENS IS USED TO TRANSFER THE
LIQUID WATER CONTENT FROM THE MAIN PROGRAM TO THIS SUBROUTINE;
ELWC IS USED AS THE AEROSOL CONCENTRATION FOR THOSE CASES.
IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS
PZRSMT=0.
OLSTAR=0.0E0
OM2=0.0E0
CTSUMT=0.0E+00
CSSUMT=0.0E+00
DENST=0.0E0
CRSUMT=0.0E+00
EMM=1.0E0
NLINES=0
BH=1.056E-3
FACTORS BH AND CH ARE USED IN SIZE ADJUSTMENTS
FH IS THE SATURATION RATIO
FH=RELHUM/100.0E0
CH=FH/(1.0E0-FH)
CONCT=0.0E0
KEXTT=0.0E0
CONVERT VOL PER PARTICLE RECEIVED FROM MAIN PROGRAM VIA VARIABLE
TVOL TO DRY VOLUME PER PARTICLE IN CUBIC CENTIMETERS
DRYVOL=TVOL*1.0E-12
TVOL=0.0E0
TMASS=0.0E0
DO 6 J=1, IT
PSUMTT(J)=0.0E0
PHH(J)=0.0E0
PGG(J)=0.0E0
6 CONVERT TEMP. TO KELVIN FOR SUBROUTINE WATER USAGE
TEMK=TEMP+273.16E0
C SKIP SUBROUTINE WATER FOR THE IDSTP = 12 CASE, AND READ THE
C OPTICAL DATA FOR RAIN AS EMW, CAYA, ETC...NEEDED BECAUSE CASE
C IDSTP=12 MAY BE AT WAVELENGTHS LONGER THAN FOUND IN ROUTINE
C WATER.
IF (IDSTP.EQ.12) GO TO 8
SUBROUTINE WATER RETURNS INTERPOLATED VALUES FOR EMW, CAYW AND
RHOW AT WAVELENGTH = WAVE AND AT TEMPERATURE = TEMK (DEG K).
EMW IS REAL PART OF INDEX OF REFR FOR PURE WATER AT TEMP(DEG C).
CAYW IS IMAG. PART OF INDEX OF REFR. FOR PURE WATER;
CAYW, HERE IS POSITIVE, BUT TREATED AS NEGATIVE IN MIE-ROUTINE.
RHOW IS MASS DENSITY(GM/CC) AT TEMPERATURE = TEMP (DEG C).
CALL WATER(WAVE, EMW, CAYW, TEMK, RHOW)
WRITE (IOOUT, 9) EMW, CAYW, TEMK, RHOW
C BEGIN LOOP OVER AEROSOL COMPONENTS INDEXED BY NK
9 DO 32 NK=1, NINDX
C BYPASS READ OF EMA, CAYA, ETC. FOR IDSTP=6 CASE..USE WATER DATA
IF (IDSTP.NE.6) GO TO 10
EMA=EMW
CAYA=CAYW
RHOA=RHOW
CONC=ELWC

```

```

AGB00010
AGB00020
AGB00030
AGB00040
AGB00050
AGB00060
AGB00070
AGB00080
AGB00090
AGB00100
AGB00110
AGB00120
AGB00130
AGB00140
AGB00150
AGB00160
AGB00170
AGB00180
AGB00190
AGB00200
AGB00210
AGB00220
AGB00230
AGB00240
AGB00250
AGB00260
AGB00270
AGB00280
AGB00290
AGB00300
AGB00310
AGB00320
AGB00330
AGB00340
AGB00350
AGB00360
AGB00370
AGB00380
AGB00390
AGB00400
AGB00410
AGB00420
AGB00430
AGB00440
AGB00450
AGB00460
AGB00470
AGB00480
AGB00490
AGB00500
AGB00510
AGB00520
AGB00530
AGB00540
AGB00550
AGB00560
AGB00570
AGB00580
AGB00590
AGB00600
AGB00610
AGB00620
AGB00630
AGB00640
AGB00650
AGB00660
AGB00670
AGB00680
AGB00690
AGB00700

```

```

EMUA=0.0E0
GO TO 11
C *** READ OPTICAL AND PHYSICAL DATA ***
10 READ (IOIN,2) EMA,CAYA,EMUA,RHOA,CONC
IF (IW.EQ.0) EMA=EMW
IF (IW.EQ.0) CAYA=CAYW
IF (IDSTP.NE.12) GO TO 11
EMUA=0.0E0
RHOW=RHOA
EMW=EMA
CAYW=CAYA
11 IF (RHOA.LE.0.E0) RHOA=1.E0
WRITE (IOUT,3) NK,EMA,CAYA,RHOA,EMUA,CONC
IF (EMA.LT.1.E-30) GO TO 44
BHT=BH*(298.E0/TEMK)
IF (EMUA.LE.0.01) CH=0.0
BC=BHT*CH
A=1.E0+((RHOA/RHOW)*EMUA*CH)
AC=A**((1.E0/3.E0)
C ADJUST EM,RHO AND CAY PER G. HANEL/ADVANCES IN GEOPHYS/1976
RHO=RHOW+(RHOA-RHOW)/A
EM=EMW+(EMA-EMW)/A
CAY=CAYW+(CAYA-CAYW)/A
CAY=CAY/EM
C INITIALIZE QUANTITIES USED TO HOLD RUNNING SUMMATIONS OVER
C RADII FOR THE CURRENT COMPONENT
CTSUM=0.E0
CSSUM=0.E0
CRSUM=0.0E0
VOL=0.0E0
OL1SUM=0.0E0
OL2SUM=0.E0
PZKSUM=0.
DO 13 J=1,IT
13 PSUM(J)=0.0E0
C PRINT HEADER IF DETAILED MIE RESULTS ARE TO BE PRINTED
IF (MORTE.EQ.12345) WRITE (IOUT,5)
C BEGIN ACTUAL LOOP OVER RADIUS INTERVALS FOR THE CURRENT NK VALUE
C THIS LOOP IS THE ONE IN WHICH THE MIE CALCULATIONS ARE CALLED
C INTERVALS ARE INDEXED BY I. THERE ARE NI SUCH INTERVALS.
DO 26 I=1,NI
NRADI=2
D=RR(I+1)-RR(I)
C RIT IS THE ADJUSTED RADIUS FOR THE RELATIVE HUMIDITY TO BE USED
C IN THIS PARTICULAR RUN OR PASS
RIT=RR(I)*AC -(BC/AC)
IF (RIT.LT.RR(I).OR.RR(I).LT.0.04E0) RIT=RR(I)
ALPHA=2.E0*PI*RIT/WAVE
C ROUTINE MIEGX DOES THE ACTUAL MIE CALCULATIONS.
C NOTE THAT THE IMAG. PART OF THE REFRACTIVE INDEX (CAY) HAS BEEN
C NORMALIZED THROUGH DIVISION BY THE REAL PART (EM) BEFORE ITS
C VALUE IS PASSED TO THE MIE-ROUTINE.
C MIEGX RETURNS THE EXTINCTION EFFICIENCY FACTOR AS QT
C MIEGX RETURNS THE SCATTERING EFFICIENCY FACTOR AS QS
C MIEGX RETURNS THE BACK-SCATTERING (RADAR) EFFIC. FACTOR AS QR
C MIEGX RETURNS THE AVERAGE INTENSITY (I1+I2)/2 IN THE ARRAY P( )
C AT ANGLES = ARCCOS( C( ) ), WHERE C( ) IS SET-UP BY
C SUBROUTINE GUSET OR ANGLE
C MIEGX ALSO RETURNS THE 2-ND AND 3-RD LEGENDRE EXPANSION COEF.
C (OMEGA SUB 1 AND OMEGA SUB 2) AS O1STAR AND O2STAR.
EMD=(EM)
CAYD=(CAY)
ALPHAD=(ALPHA)
CALL MIEGX(EMD,CAYD,ALPHAD,QT,QS,QR,P,O1STRD,O2STRD,
+C,IT,PFNZRO)
EM=(EMD)
CAY=(CAYD)
ALPHA=(ALPHAD)
QT=(QTD)
QS=(QSD)
AGB00710
AGR00720
AGL00730
AGB00740
AGB00750
AGB00760
AGB00770
AGB00780
AGB00790
AGB00800
AGB00810
AGB00820
AGB00830
AGB00840
AGB00850
AGB00860
AGB00870
AGB00880
AGB00890
AGB00900
AGB00910
AGB00920
AGB00930
AGB00940
AGB00950
AGB00960
AGB00970
AGB00980
AGB00990
AGB01000
AGB01010
AGB01020
AGB01030
AGB01040
AGB01050
AGB01060
AGB01070
AGB01080
AGB01090
AGB01100
AGB01110
AGB01120
AGB01130
AGB01140
AGB01150
AGB01160
AGB01170
AGB01180
AGB01190
AGB01200
AGB01210
AGB01220
AGB01230
AGB01240
AGB01250
AGB01260
AGB01270
AGB01280
AGB01290
AGB01300
AGB01310
AGB01320
AGB01330
AGB01340
AGB01350
AGB01360
AGB01370
AGB01380
AGB01390
AGB01400

```

```

QR=(QRD)
O1STAR=(O1STRD)
O2STAR=(O2STRD)
KK=1+(I-1)*NKG
IF (MORTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I),F(KK),ALPHA,QT,QS,QR
FKK=F(KK)
FKKA=FKK*PI*RIT**2.E0
VOLHH=4.1888*FKK*RIT**3.E0
OL1HH=O1STAR*FKKA*QT
OL2HH=O2STAR*FKKA*QT
CTHH=QT*FKKA
CSHH=QS*FKKA
CRHH=QR*FKKA
DO 14 J=1,IT
PHK(J)=P(J)*FKK
14 CONTINUE
PFNZER=PFNZRO*FKK
RIT=RR(I+1)*AC-(BC/AC)
IF (RIT.LT.RR(I+1).OR.RR(I+1).LT.0.04E0) RIT=RR(I+1)
ALPHA=2.E0*PI*RIT/WAVE
EMD=(EM)
CAYD=(CAY)
ALPHAD=(ALPHA)
CALL MIEGX(EMD,CAYD,ALPHAD,QT,QSD,QRD,P,O1STRD,O2STRD,
+C,IT,PFNZRO)
EM=(EMD)
CAY=(CAYD)
ALPHA=(ALPHAD)
QT=(QTD)
QS=(QSD)
QR=(QRD)
O1STAR=(O1STRD)
O2STAR=(O2STRD)
KK1=1+NKG*1
FKK1=F(KK1)
IF (MORTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I+1),FKK1,ALPHA,QT,QS,QR
FKK1A=FKK1*PI*RIT**2.E0
VOLHH=(VOLHH+4.188E0*FKK1*RIT**3.E0)*D*0.5E0
OL1HH=(OL1HH+FKK1A*QT*O1STAR)*D*0.5E0
OL2HH=(OL2HH+FKK1A*QT*O2STAR)*D*0.5E0
CTHH=(CTHH+QT*FKK1A)*D*0.5E0
CSHH=(CSHH+QS*FKK1A)*D*0.5E0
CRHH=(CRHH+QR*FKK1A)*D*0.5E0
DO 15 J=1,IT
PHK(J)=(PHK(J)+P(J)*FKK1)*D*0.5E0
15 CONTINUE
PFNZER=(PFNZER+PFNZRO*FKK1)*D*0.5
FF=0.5E0*D*(FKK+FKK1)
NT=1
NJ=1
16 NJ=NT
NT=2*NT
D=0.5E0*D
VOLGG=0.0E0
OL1GG=0.0E0
OL2GG=0.0E0
CTGG=0.0E0
CSGG=0.0E0
CRGG=0.0E0
FT=0.0E0
DO 17 J=1,IT
PGG(J)=0.0E0
PZRTMP=0.
17 NEXT LOOP HANDLES INTERMEDIATE PARTICLE SIZES..THOSE LYING BETWEEN
C RMIN AND RMAX FOR THE CURRENT INTERVAL WHOSE INDEX IS I,
DO 19 JG=1,NJ
KK=1+(I-1)*NKG+(2*JG-1)*(NKG/NT)
RIT=R(KK)*AC-(BC/AC)
IF (RIT.LT.R(KK).OR.RR(KK).LT.0.04E0) RIT=R(KK)
ALPHA=2.E0*PI*RIT/WAVE

```

```

AGB001410
AGB001420
AGB001430
AGB001440
AGB001450
AGB001460
AGB001470
AGB001480
AGB001490
AGB001500
AGB001510
AGB001520
AGB001530
AGB001540
AGB001550
AGB001560
AGB001570
AGB001580
AGB001590
AGB001600
AGB001610
AGB001620
AGB001630
AGB001640
AGB001650
AGB001660
AGB001670
AGB001680
AGB001690
AGB001700
AGB001710
AGB001720
AGB001730
AGB001740
AGB001750
AGB001760
AGB001770
AGB001780
AGB001790
AGB001800
AGB001810
AGB001820
AGB001830
AGB001840
AGB001850
AGB001860
AGB001870
AGB001880
AGB001890
AGB001900
AGB001910
AGB001920
AGB001930
AGB001940
AGB001950
AGB001960
AGB001970
AGB001980
AGB001990
AGB002000
AGB002010
AGB002020
AGB002030
AGB002040
AGB002050
AGB002060
AGB002070
AGB002080
AGB002090
AGB002100

```

```

EMD= (EM)
CAYD= (CAY)
ALPHAD= (ALPHA)
CALL MIEGX(EMD, CAYD, ALPHAD, QTD, QSD, QRD, P, O1STRD, O2STRD,
+C, IT, PFNZRO)
EM=(EMD)
CAY=(CAYD)
ALPHA=(ALPHAD)
QT=(QTD)
QS=(QSD)
QR=(QRD)
O1STAR=(O1STRD)
O2STAR=(O2STRD)
IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,R(KK),F(KK),ALPHA,QT,QS,QR
NRADI=NRADI+1
FKK=F(KK)
FKKA=FKK*PI*RIT**2.E0
VOLGG=4.1888E0*FKK*RIT**3.OE0+VOLGG
OL1GG=OL1GG+O1STAR*FKKA*QT
OL2GG=OL2GG+O2STAR*FKKA*QT
CTGG=CTGG+QT*FKKA
CSGG=CSGG+QS*FKKA
CRGG=CRGG+QR*FKKA
DO 18 J=1, IT
PGG(J)=PGG(J)+P(J)*FKK
18 CONTINUE
PZRTMP=PZRTMP+PFNZRO*FKK
19 FT=FT+FKK
C ADD RESULTS ACCUMULATED DURING PREVIOUS HALVINGS TO THOSE FOUND
C FOR THE NEW RADII TREATED WITHIN THE LOOP OVER INDEX JG
VOLHHT=0.5E0*VOLHH+D*VOLGG
OL1HHT=0.5E0*OL1HH+D*OL1GG
OL2HHT=0.5E0*OL2HH+D*OL2GG
CTHHT=0.5E0*CTHH+D*CTGG
CSHHT=0.5E0*CSHH+D*CSGG
CRHHT=0.5E0*CRHH+D*CRGG
DO 20 J=1, IT
20 PHH(J)=.5E0*PHH(J)+D*PGG(J)
PFNZER=.5E0*PFNZER+D*PZRTMP
FFT=0.5E0*FF+D*FT
IF (CTHHT.LT.1.E-30) GO TO 22
DEL=ABS(VOLHHT-VOLHH)/ABS(VOLHHT)
IF (DEL.LE.DELTA) GO TO 21
GO TO 22
21 IF (N.GT.2) GO TO 24
C DO NOT ALLOW DEL LESS THAN DELTA EXIT UNLESS AT LEAST TWO
C HAVING BEEN DONE
22 IF (N.EQ.NHALV) GO TO 24
C MUST EXIT WHEN NHALV HALVINGS HAVE BEEN DONE EVEN IF THE DELTA
C CRITERION HAS NOT BEEN SATISFIED..SINCE NO MORE VALUES OF RADII
C ARE AVAILABLE.
FF=FFT
CRHH=CRHHT
OL1HH=OL1HHT
OL2HH=OL2HHT
CSHH=CSHHT
CTHH=CTHHT
VOLHH=VOLHHT
N=N+1
GO TO 16
24 CONTINUE
IF(N.EQ.NHALV) WRITE(IOUT,124) I
C SUM QUANTITIES OVER ALL INTERVALS TREATED UP UNTIL NOW
CTSUM=CTSUM+CTHHT
CSSUM=CSSUM+CSHHT
CRSUM=CRSUM+CRHHT
VOL=VOL+VOLHHT
OL1SUM=OL1SUM+OL1HHT
OL2SUM=OL2SUM+OL2HHT
C AT THIS POINT, PSUM( ) IS THE RUNNING SUM OF THE AVG. INTENSITY

```

```

AGB02110
AGB02120
AGB02130
AGB02140
AGB02150
AGB02160
AGB02170
AGB02180
AGB02190
AGB02200
AGB02210
AGB02220
AGB02230
AGB02240
AGB02250
AGB02260
AGB02270
AGB02280
AGB02290
AGB02300
AGB02310
AGB02320
AGB02330
AGB02340
AGB02350
AGB02360
AGB02370
AGB02380
AGB02390
AGB02400
AGB02410
AGB02420
AGB02430
AGB02440
AGB02450
AGB02460
AGB02470
AGB02480
AGB02490
AGB02500
AGB02510
AGB02520
AGB02530
AGB02540
AGB02550
AGB02560
AGB02570
AGB02580
AGB02590
AGB02600
AGB02610
AGB02620
AGB02630
AGB02640
AGB02650
AGB02660
AGB02670
AGB02680
AGB02690
AGB02700
AGB02710
AGB02720
AGB02730
AGB02740
AGB02750
AGB02760
AGB02770
AGB02780
AGB02790
AGB02800

```

```

C AS SUMMED OVER SIZES
DO 25 J=1,IT
25 PSUM(J)=PSUM(J)+PHH(J)
PZRSUM=PZRSUM+PFNZER
WRITE (IOUT,27) NK,I,NRADI,CTHHT
NLLINES=NLLINES+NRADI
C END LOOP OVER HALVING INTERVALS INDEXED BY I
26 CONTINUE
CALCULATE PARTICLE NUMBER DENSITY<NO. PER CC> AS DENSC
DENSC=CONC/(RHOA*DRYVOL)
C OVERRIDE CALCULATED VALUE OF DENSC WITH DENS IF LLLL = 1
IF (LLLL.EQ.1) DENSC=DENS
C RECALCULATE CONC FROM OTHER INPUT DATA IF LLLL=1
IF (LLLL.EQ.1) CONC=DENS*RHOA*DRYVOL
C REPLACE DENS BY DENSC FOR LATER USE BY AGXP3
DENS=DENSC
C WEIGHT CTSUM,ETC. BY NUMBER DENSITIES (DENSC) FOR THIS COMPONENT
CTSUM=CTSUM+DENSC
CSSUM=CSSUM+DENSC
CRSUM=CRSUM+DENSC
VOL=VOL+DENSC
OL1SUM=OL1SUM+DENSC
OL2SUM=OL2SUM+DENSC
DO 29 J=1,IT
29 PSUM(J)=PSUM(J)+DENSC
PZRSUM=PZRSUM+DENSC
C NOW, SUM OVER COMPONENTS INDEXED BY NK
C CONCT IS THE TOTAL DRY-AEROSOL CONCENTRATION IN MG PER CC
CONCT=CONCT+1.E3*CONC
DENST=DENST+DENSC
OLSTAR=OL1SUM+OLSTAR
OM2=OL2SUM+OM2
CTSUMT=CTSUMT+CTSUM
C AT THIS POINT, CTSUMT IS THE TOTAL EXTINCTION CROSS SECTION
C (IN SQ. MICRONS) AS SUMMED OVER ALL COMPONENTS WHICH
C HAVE BEEN DEALT WITH THUS FAR
CSSUMT=CSSUMT+CSSUM
CRSUMT=CRSUMT+CRSUM
DO 30 J=1,IT
30 PSUMTT(J)=PSUM(J)+PSUMTT(J)
PZRSMT=PZRSMT+PZRSUM
VOL=VOL*1.E-12
C TVOL IS THE TOTAL VOLUME (IN CM**3) OCCUPIED BY THE AEROSOL
C PARTICLES. TVOL IS NOT ACTUALLY USED IN THIS VERSION OF
C PROGRAM AGAUS.
TVOL=VOL+TVOL
EMASS=VOL*RHO
TMASS=TMASS+EMASS
KEXOLD=KEXTT
KEXTT=CTSUMT*1.E-3
KEXT=KEXTT-KEXOLD
WRITE (IOUT,31) NK,VOL,EMASS,KEXT
C VPF(VOL) IS THE VOLUME PACKING FRACTION; THAT IS, THE FRACTION
C OF EACH CC OF SPACE WHICH IS FILLED BY AEROSOL MATERIAL BELONGING
C TO THE CURRENT COMPONENT NK.
C TMASS IS THE TOTAL MASS OF AEROSOL FOUND IN 1 CC OF SPACE.
C EMASS IS THE MASS OF AEROSOL MATERIAL PER CC ASSOCIATED WITH
C THE CURRENT COMPONENT NK.
C KEXT IS THE EXTINCTION COEF.(PER KM) WHICH IS ASSOCIATED WITH
C THE CURRENT COMPONENT--AS IF IT ALONE WERE PRESENT.
C KEXTT IS THE SUM OF THE KEXT'S OVER ALL COMPONENTS.
C END LOOP OVER AEROSOL COMPONENTS INDEXED BY NK.
IF (NINDX.GT.1) WRITE (IOUT,42) NK
IF (MORTE.EQ.12345) WRITE (IOUT,43)
32 CONTINUE
IF (NINDX.GT.1) WRITE (IOUT,33) TMASS,KEXTT
WRITE (IOUT,34) NLLINES
DENS=DENST
NRADI=NLLINES
C NOW, PERFORM THE FINAL RENORMALIZATIONS TO OBTAIN CTSUM, ETC.
AGB02810
AGB02820
AGB02830
AGB02840
AGB02850
AGB02860
AGB02870
AGB02880
AGB02890
AGB02900
AGB02910
AGB02920
AGB02930
AGB02940
AGB02950
AGB02960
AGB02970
AGB02980
AGB02990
AGB03000
AGB03010
AGB03020
AGB03030
AGB03040
AGB03050
AGB03060
AGB03070
AGB03080
AGB03090
AGB03100
AGB03110
AGB03120
AGB03130
AGB03140
AGB03150
AGB03160
AGB03170
AGB03180
AGB03190
AGB03200
AGB03210
AGB03220
AGB03230
AGB03240
AGB03250
AGB03260
AGB03270
AGB03280
AGB03290
AGB03300
AGB03310
AGB03320
AGB03330
AGB03340
AGB03350
AGB03360
AGB03370
AGB03380
AGB03390
AGB03400
AGB03410
AGB03420
AGB03430
AGB03440
AGB03450
AGB03460
AGB03470
AGB03480
AGB03490
AGB03500

```

```

C      VALUES REPRESENTATIVE OF A SINGLE AVERAGE PARTICLE.
C      CTSUM BECOMES THE EXTINCTION CROSSSECTION IN SQ. MICROMETERS PER
C      AVERAGE PARTICLE. THE OTHER QUANTITIES CARRY SIMILAR MEANINGS.
      DO 35 J=1,IT
35     PSUM(J)=PSUMTT(J)/DENST
      OLSTAR=OLSTAR/CTSUMT
      OM2=OM2/CTSUMT
      CTSUM=CTSUMT/DENST
      CSSUM=CSSUMT/DENST
      CRSUM=CRSUMT/DENST
C      ... AND CONVERT ANG INTENS AT ZERO DEGREES INTO PHASE FUNCT.
C      WITH NORMALIZATION
      PFNZRO=PZRSMT/DENST*(WAVE*WAVE/(PI*CTSUM*EMM*EMM))
C      WRITE(IOUT,36) OLSTAR,OM2,PFNZRO
C      CALCULATE ATTENUATION COEFS. IN SQ.METERS PER MILLIGRAM
      CATTN=CTSUMT*1.E-12/CONCT
      CATTNW=1.E-12*KEXTT/TMASS
      WRITE(IOUT,37) CATTN
      IF(RELHUM.GT.1.0)WRITE(IOUT,38) CATTNW
      GO TO 41
44     WRITE(IOUT,45) EMA
      STOP
      2 FORMAT(4F10.6,E15.7)
      3 FORMAT(1H,6H INDX=,I3,4H M=,F10.6,6H K = -,F10.6,9HI. MASS ,
+          10HDENSITY =,F8.6,
+          17H GROWTH FACTOR =,F8.4,9H. CONC =,1PE12.5,7H GM/CC/)
      4 FORMAT(1X,F10.5,6(2X,1PE11.5))
      5 FORMAT(//54H R(MICRONS) DRY RADIUS N(R) MIE SIZE
+          36HQ (EXT) Q(SCA) Q(RADAR)?)
      9 FORMAT(1H,39INDEX OF REFRACTION FOR PURE WATER IS: ,F8.6,
+          3H -,F8.6,1HI//,1X,25HMASS DENSITY OF WATER AT ,F6.2,
+          11H DEG C IS: ,F8.2,6H GM/CC,/)
124    FORMAT(/52H *** CONVERGENCE LEVEL NOT REACHED FOR INTERVAL NO. ,
+          I3,4H ***/)
      27 FORMAT(1H,19H FOR COMPONENT NO. ,I3,15H INTERVAL NO. ,I3,1H ,I4,
+          43H RADII WERE USED. CONTRIBUTION TO CTSUM = ,1PE12.6)
      31 FORMAT(1H,/,20H FOR COMPONENT NO. ,I2,12H : VPF = ,1PE12.5,
+          7H PER CC,24H MASS CONCENTRATION = ,E12.5,21H GM/CC. KEXT =
+          E12.5,7H PER KM)
      33 FORMAT(1H,/,29H TOTAL MASS CONCENTRATION = ,1PE12.5,7H GM/CC;,
+          15H TOTAL KEXT = ,E12.5,7H PER KM)
      34 FORMAT(/1X,32H TOTAL NUMBER OF RADII USED WAS ,I5)
      36 FORMAT(/1X,19H ANALYTIC SOLUTIONS,/,
+          16H OMEGA SUB 1 = ,1PE14.7/,16H OMEGA SUB 2 = ,1PE14.7/
+          16H PFN AT ZERO = ,1PE14.7,/)
      37 FORMAT(1H,21H ATTENUATION COEF. = ,1PE12.5,12H SQ-METERS/,
+          33HMILLIGRAM OF DRY AEROSOL MATERIAL)
      38 FORMAT(1H,21H ATTENUATION COEF. = ,1PE12.5,12H SQ-METERS/,
+          33HMILLIGRAM OF WET AEROSOL MATERIAL/)
      42 FORMAT(1H,/,1X,10X,30(1H*),33H END OF AEROSOL COMPONENT CYCLE ,
+          7HNUMBER ,I3,2X,30(1H*)//)
      43 FORMAT(1H)
      45 FORMAT(///,1X,11H***** EMA (,F10.6,20H) IS EITHER ZERO OR ,
+          35HNEGATIVE - PROGRAM TERMINATED *****)
41     RETURN
      END

```

```

AGB03510
AGB03520
AGB03530
AGB03540
AGB03550
AGB03560
AGB03570
AGB03580
AGB03590
AGB03600
AGB03610
AGB03620
AGB03630
AGB03640
AGB03650
AGB03660
AGB03670
AGB03680
AGB03690
AGB03700
AGB03710
AGB03720
AGB03730
AGB03740
AGB03750
AGB03760
AGB03770
AGB03780
AGB03790
AGB03800
AGB03810
AGB03820
AGB03830
AGB03840
AGB03850
AGB03860
AGB03870
AGB03880
AGB03890
AGB03900
AGB03910
AGB03920
AGB03930
AGB03940
AGB03950
AGB03960
AGB03970
AGB03980
AGB03990
AGB04000
AGB04010
AGB04020
AGB04030
AGB04040
AGB04050
AGB04060

```



```

SUBROUTINE AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,
+ WAVE,EM,CAY,EMM,IT,IEND,IANG)
COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)
COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
+ NRADI,PI,IDSTP,NKG,NHALV,NI
C IEND=1 WHEN THE COMPOSITE PHASE FUNCTION IS BEING WRITTEN
C IF (IEND.EQ.1) GO TO 6
ALBDO=CSSUM/CTSUM
C PFACT IS USED TO CONVERT AVG. INTENSITY PSUM( ) INTO PHASE-
C FUNCTIONS. SFACIT IS USED TO CONVERT PSUM INTO SCATTERING
C FRACTIONS, NORMALIZED PER SOM. THE INTEGRAL OF SCAT OVER SOLID
C ANGLE SHOULD YIELD THE TOTAL SCATTERING CROSS-SECTION IN SQ. M.
SFACIT=WAVE*WAVE*DENS*1.E-6/(4.*PI*PI)
PFACT=WAVE*WAVE/(PI*CTSUM*EMM*EMM)
DO 15 J=1,IT
SCAT=PSUM(J)*SFACIT
PSUM(J)=PSUM(J)*PFACT
C UNCOMMENT THE FOLLOWING STMT IF U WANT SCAT FRACT,COSINES AND A
C COUNTER WRITTEN ON NUNIT. NUNIT IS SET IN THE BLOCK DATA ROUTINE.
C WRITE (NUNIT,3) SCAT,C(J),J
15 CONTINUE
IF (NINDX.GE.2) WRITE (IOUT,12)
WRITE (IOUT,4) IDSTP,WAVE,EM,CAY,CTSUM,CSSUM,ALBDO
C CONVERT AVG. CROSS-SECTIONS TO COEFFICIENTS (PER KM)
CTSUM=CTSUM*1.0E-3*DENS
CSSUM=CSSUM*1.0E-3*DENS
CRSUM=CRSUM*1.0E-3*DENS
WRITE (IOUT,13) CTSUM,CSSUM,CRSUM
WRITE (IOUT,14) GNU,DENS
IF (IT.LT.2) GO TO 21
C WRITE PHASE FUNCTION AT SPECIFIED ANGLES
6 WRITE (IOUT,1)
WRITE (IOUT,5)
C FIND ANGLES FROM COSINES
DO 2 I=1,IT
2 FF(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI
DO 19 J=1,IT,4
K=J+3
IF (K.GT.IT) K=IT
19 WRITE (IOUT,8) (C(I),FF(I),PSUM(I),I=J,K)
IF (IANG.NE.0) RETURN
C ROUTINE GAUS GENERATES AND PRINTS THE LEGENDRE
C EXPANSION COEFS (OMEGAS) FOR THE PHASE FUNCTION.
CALL GAUS(IT)
C CHECK TO SEE IF SNG. SCAT. ALBEDO (ALBDO) COMPUTED DIRECTLY
C FROM CROSS-SECTIONS AGREES WITH THAT FOUND FROM THE LEGENDRE
C EXPANSION OF PHASE-FUNCTION.
IF ((ABS(OL(1)-ALBDO)/ALBDO.GT.5.E-3).AND.(IEND.NE.1))
1 WRITE (IOUT,20)
1 FORMAT (//1H ,50X,14H PHASE FUNCTION/,1X,42X,31H(NORMALIZED TO 4 PI
+ OMEGA ZERO)///)
3 FORMAT (2(E13.7,1X),I3)
4 FORMAT (1H1,/,41H DISTRIBUTION WAVELENGTH REFRACTIVE,9X,
+ 20HEXTINCTION X SECTION,8X,20HSCATTERING X SECTION,12X,5HALBDO/,
+ 1H ,6X, 4HTYPE,6X,9H(MICRONS),8X,5HINDEX,16X,12H(SQ MICRONS),13X,
+ 12H(SQ MICRONS)/1H ,19,4X,F11.4,F10.4,3H(1-,F7.4,2H),
+ 7X,1PE14.7,11X,1PE14.7,12X,1PE14.7,/)
5 FORMAT (1H ,3X,4(5H MU,2X, ANGLE ,17H PHASE FUNCTION ))
8 FORMAT (1H ,F9.5,F7.2,E12.5,3(3X,F9.5,F7.2,E12.5))
12 FORMAT (52H THIS IS A MIXED CASE * SUBSEQUENT REFRACTIVE INDEX ,
+ 34HPRINT-OUTS ARE NOT GENERALLY VALID./)
13 FORMAT (1H ,10H K(EXT) = ,1PE13.7,11H; K(SCA) = ,E13.7,
+ 11H; K(RAD) = ,E13.7,11H ALL PER KM./)
14 FORMAT (//14H WAVENUMBER = ,1PE12.6,5H CM-1,5X,10HDENSITY = ,E12.6,
+ 17H PARTICLES PER CC./)
20 FORMAT (//12H *** VALUES
+ ,55HOF ALBDO AND OL(1) DISAGREE BY MORE THAN 0.5 PERCENT **
+ ,34HLARGER VALUE OF 'IT' IS NEEDED ***/)

```

21

RETURN
END

AGC00710
AGC00720

C	SUBROUTINE ANGLE(PI, IANG, IT)	ANG00010
C	THIS ROUTINE IS TO BE USED TO REPLACE GUSSET FOR THE PURPOSE	ANG00020
C	OF USING AGAUS TO DO PHASE FUNCTION CALCULATIONS AT -IT-	ANG00030
C	ANGLES BETWEEN 0 AND 180 DEGREES, RATHER THAN AT THE G-L	ANG00040
C	QUADRATURE ABSCISSA VALUES. IT ALSO READS THE INPUT ANGLES	ANG00050
C	IF IANG=2.	ANG00060
	COMMON /IO/ IOIN, IOOUT, NUNIT, IEQ, NEOU	ANG00070
	COMMON /AGXM/ C(65), W(65), OLT(65), JDIMCK(3)	ANG00080
	RADS=PI/180.	ANG00090
	DEL=180./FLOAT(IT-1)	ANG00100
	IF (IEQ.EQ.1.OR.IEQ.EQ.2.OR.IEQ.EQ.5) GO TO 4	ANG00110
	IF (IANG.EQ.2) GO TO 2	ANG00120
	DO 1 I=1, IT	ANG00130
	W(I)=DEL*FLOAT(I-1)	ANG00140
1	C(I)=COS(W(I)*RADS)	ANG00150
	RETURN	ANG00160
2	READ(5, 100) (W(I), I=1, IT)	ANG00170
4	DO 3 I=1, IT	ANG00180
3	C(I)=COS(W(I)*RADS)	ANG00190
	IANG=1	ANG00200
	RETURN	ANG00210
100	FORMAT(16F5.1)	ANG00220
	END	ANG00230

C	FUNCTION GAMMA(X)	GMA00010
C	GAMMA FUNCTION; TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS,	GMA00020
C	ABRAMOWITZ AND STEGUN, NOV 1964, PP 256-257. RECURRENCE FORMULA	GMA00030
C	6.1.16, POLYNOMIAL APPROXIMATION 6.1.35.	GMA00040
	DATA A1,A2,A3,A4,A5	GMA00050
	+ /-.5748646,.9512363,-.6998588,.4245549,-.1010678/	GMA00060
C	COMPUTER AND GAMMA FUNCTION LIMITS	GMA00070
	IF (X.GT.34..OR.X.LT.0.) GO TO 3	GMA00080
	GSUM=1	GMA00090
	N=IFIX(X+.00001)	GMA00100
C	FIND Z .LE. 1.	GMA00110
	Z=X-FLOAT(N)	GMA00120
C	OK FOR Z BEING INTEGER	GMA00130
	IF (Z.LT.1.E-04) N=N-1	GMA00140
	IF (Z.LT.1.E-04) Z=1.	GMA00150
C	COMPENSATE FOR N-1 IN FORMULA	GMA00160
	N=N-1	GMA00170
C	IF Z .LE. 1, SKIP LOOP	GMA00180
	IF (N.LE.0) GO TO 2	GMA00190
C	RECURRENCE RELATION: G(N+Z)=(N-1+Z)*(N-2+Z)...(1+Z)*G(1+Z)	GMA00200
	DO 1 I=1,N	GMA00210
	VALUE=FLOAT(I)+Z	GMA00220
	GSUM=GSUM*VALUE	GMA00230
1	POLYNOMIAL APPROXIMATION; Z.LE.1	GMA00240
C	GAMMA=1.+A1*Z+A2*Z*Z+A3*Z*Z*Z+A4*Z*Z*Z*Z+A5*Z*Z*Z*Z*Z	GMA00250
2	GAMMA=GAMMA*GSUM	GMA00260
	RETURN	GMA00270
3	WRITE (1,100) X	GMA00280
100	FORMAT(1H, ' ***** THE VALUE OF X (',2PE11.4,') IS EITHER ',	GMA00290
	+ 'OUTSIDE COMPUTER LIMITS',/, ' OR NEGATIVE - PGM STOPPED *****')	GMA00300
	STOP	GMA00310
	END	GMA00320

	SUBROUTINE GUSET(IT)	GUSET010
	THIS ROUTINE CALCULATES THE ABSCISSAE C()	GUSET020
	AND GAUSS-LEGENDRE WEIGHTS W()	GUSET030
	VIA GAUSS-LEGENDRE QUADRATURE OF ORDER N	GUSET040
	COMMON /IO/ IOIN, IOOUT, NUNIT, IEO, NEOU	GUSET050
	COMMON /AGXM/ C(65), W(65), OLT(65), JDIMCK(3)	GUSET060
	COMMON /PT1/ F(513), R(513), DR(8), RR(9), FF(514)	GUSET070
	+ ,NRADI, PI, IDSTP, NKG, NHALV, NI	GUSET080
	N=IT	GUSET090
	TOL=1.0E-06	GUSET100
	AA=2.0E+00/PI**2.0E+00	GUSET110
	AB=-62.E+00/(3.0E+00*PI**4.0E+00)	GUSET120
	AC=15116.0E+00/(15.0E+00*PI*6.0E+00)	GUSET130
	AD=-12554474.0E+00/(105.0E+00*PI*8.0E+00)	GUSET140
	F(1)=1.0E+00	GUSET150
	EN= FLOAT(N)	GUSET160
	NP1=N+1	GUSET170
	U=1.0E+00-(2.0E+00/PI)**2.0E+00	GUSET180
	D=1.0E+00/SQRT((EN+0.5E+00)**2.E+00+U/4.0E+00)	GUSET190
	DO 1 I=1,N	GUSET200
	S= FLOAT(I)	GUSET210
	A=4.0E+00*S-1.0E+00	GUSET220
	AE=AA/A	GUSET230
	AF=AB/A**3.0E+00	GUSET240
	AG=AC/A**5.0E+00	GUSET250
	AH=AD/A**7.0E+00	GUSET260
1	R(I)=PI*(A+AE+AF+AG+AH)/4.E+00	GUSET270
	DO 6 K=1,N	GUSET280
	X=COS(R(K)*D)	GUSET290
2	F(2)=X	GUSET300
	DO 3 NN=3, NP1	GUSET310
	ENN= FLOAT(NN-1)	GUSET320
	F(NN)=((2.0E+00*ENN-1.E+00)*X*F(NN-1)-(ENN-1.E+00)*F(NN-2))/ENN	GUSET330
3	IF (ABS(F(NN)).GT.1E+35) F(NN)=SIGN(1.E+35,F(NN))	GUSET340
	PNP=EN*(F(N)-X*F(NP1))/(1.0E+00-X*X)	GUSET350
	XI=X-F(NP1)/PNP	GUSET360
	XD= ABS(XI-X)	GUSET370
	XDD=XD-TOL	GUSET380
	IF (XDD) 5,5,4	GUSET390
4	X=XI	GUSET400
	GO TO 2	GUSET410
5	C(K)=X	GUSET420
6	W(K)=2.0E+00*(1.0E+00-X*X)/(EN*F(N)*EN*F(N))	GUSET430
	DO 7 I=1,N	GUSET440
	R(I)=0.00	GUSET450
7	F(I)=0.00	GUSET460
	RETURN	GUSET470
	END	GUSET480

```

SUBROUTINE MIEGX(EMD,CAYD,ALPHAD,QTQ,QSD,QRD,P,O1STRD,O2STRD,
+C,IT,PFNZRO)
THIS ROUTINE IS CURRENTLY SINGLE PRECISION COMPLEX
CHANGE THE VALUE OF NDIM IF YOU CHANGE THE DIM OF A IN NEXT LINE
COMPLEX A(600),ACAPN,ZNUM,ZDEN,ZPOT,ZRPDT,ZAN,ZANP,Y,RF,RRF,
1 RRF,X,WM1,FNA,FNB,TC1,FNAP,FNBP,FNAPP,FNBPP,TC2,WFN(2)
COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
DIMENSION P(65),C(65)
DIMENSION T(4),TA(4),TB(2),TC(2),TD(2),TE(2),TF(2),TG(2)
DIMENSION ELTRMX(4,76),PI(3,76),TAU(3,76)
EQUIVALENCE (WFN(1),TA(1)),(FNA,TB(1)),(FNB,TC(1)),(FNAP,TD(1))
EQUIVALENCE (FNBP,TE(1)),(FNAPP,TF(1)),(FNBPP,TG(1))
C THESE EQUIVALENCES ALLOW USE OF REAL AND IMAG PARTS INDIVIDUALLY
TOL = 1.E-06
ITT=IT
X=(ALPHAD)
CAY=(CAYD)
EM=(EMD)
CAYE=CAY*EM
QRT=0.0
S=1.0
RF=CMPLX(EM,-CAYE)
NMX=IFIX(X*(EM+CAYE))+9
RRF=1.0/RF
RX=1.0/X
RRFX=RRF*RX
C LOOP POINT FOR CALCULATING PFN AT ZERO DEGREES
IAPXCT=0
21 CONTINUE
C THESE ARE THE PI AND TAU FUNCTIONS
DO 1 J=1,IT
PI(1,J)=0.0
PI(2,J)=1.0
TAU(1,J)=0.0
TAU(2,J)=C(J)
1 CONTINUE
T(1)=COS(X)
T(2)=SIN(X)
WM1=CMPLX(T(1),-T(2))
WFN(1)=CMPLX(T(2),T(1))
WFN(2)=RX*WFN(1)-WM1
T(1)=CAYE*X
N=1
C NDIM MUST EQUAL THE DIMENSION OF A( ).
NDIM=600
IF (NMX.LT.NDIM) NDELTA=NMX
IF (NMX.GT.NDIM) NDELTA=NDIM
NMX=0
IF (N.EQ.1) GO TO 4
EN=FLOAT(N)
T(1)=2.0*EN-1.0
T(2)=EN-1.0
T(3)=2.0*EN+1.0
DO 3 J=1,IT
PI1J=PI(1,J)
PI2J=PI(2,J)
CJ=C(J)
C SWITCH FOR CALCULATING PFN AT ZERO DEGREES
IF (IAPXCT.EQ.1) CJ=1.0
S2T=(1.0-CJ*CJ)
PI(3,J)=(T(1)*PI2J*CJ-EN*PI1J)/T(2)
TAU(3,J)=CJ*(PI(3,J)-PI1J)-T(1)*S2T*PI2J+TAU(1,J)
3 CONTINUE
WM1=WFN(1)
WFN(1)=WFN(2)
WFN(2)=T(1)*RX*WFN(1)-WM1
4 CONTINUE
C CALCULATE RATIO OF BESSEL FNS OF CONSECUTIVE ORDER
IF (N.LT.(NMX+1)) GO TO 9
NMX=NMX+NDELTA
MIEGX010
MIEGX020
MIEGX030
MIEGX040
MIEGX050
MIEGX060
MIEGX070
MIEGX080
MIEGX090
MIEGX100
MIEGX110
MIEGX120
MIEGX130
MIEGX140
MIEGX150
MIEGX160
MIEGX170
MIEGX180
MIEGX190
MIEGX200
MIEGX210
MIEGX220
MIEGX230
MIEGX240
MIEGX250
MIEGX260
MIEGX270
MIEGX280
MIEGX290
MIEGX300
MIEGX310
MIEGX320
MIEGX330
MIEGX340
MIEGX350
MIEGX360
MIEGX370
MIEGX380
MIEGX390
MIEGX400
MIEGX410
MIEGX420
MIEGX430
MIEGX440
MIEGX450
MIEGX460
MIEGX470
MIEGX480
MIEGX490
MIEGX500
MIEGX510
MIEGX520
MIEGX530
MIEGX540
MIEGX550
MIEGX560
MIEGX570
MIEGX580
MIEGX590
MIEGX600
MIEGX610
MIEGX620
MIEGX630
MIEGX640
MIEGX650
MIEGX660
MIEGX670
MIEGX680
MIEGX690
MIEGX700

```

```

NMIN=NMX+1-NDELTA
V=FLOAT(NMX)+0.50
Y=RF*X
ZANP=2.0/Y
ZNUM=ZANP*Y
ZPDT=ZNUM
V=V+1.0
ZDEN=ZANP*V
ZNUM=ZDEN-1.0/ZNUM
5 ZRPDT=ZNUM/ZDEN
ZPDT=ZRPDT*ZPDT
IF (ABS(REAL(ZRPDT)-1.0),LT. TOL) GO TO 7
IF (V.LT.20000.0) GO TO 6
WRITE (IOUT,1000) X,EM,CAYE
STOP
6 V=V+1.0
ZAN=ZANP*V
ZNUM=ZAN-1.0/ZNUM
ZDEN=ZAN-1.0/ZDEN
GO TO 5
7 CONTINUE
J=NMX
8 JJ=J-NMX+NDELTA
A(JJ)=- (FLOAT(J))/Y+ZPDT
J=J-1
IF (J.LT.NMIN) GO TO 9
ZPDT=(2.0*FLOAT(J)+1.0)/Y-1.0/ZPDT
GO TO 8
9 CONTINUE
J=N-NMX+NDELTA
ACAPN=A(J)
IF (N.GT.1) GO TO 11
C THIS PART FOR N EQUAL 1 ONLY
TC1=ACAPN*RRF+RX
TC2=ACAPN*RF+RX
C SEE EQUIVALENCE STMTS FOR EXPANATION OF TAX ), ETC.
FNA=(TC1*TA(3)-TA(1))/(TC1*WFN(2)-WFN(1))
FNB=(TC2*TA(3)-TA(1))/(TC2*WFN(2)-WFN(1))
FNAP=FNA
FNBP=FNB
T(1)=1.50
TB(1)=T(1)*TB(1)
TB(2)=T(1)*TB(2)
TC(1)=T(1)*TC(1)
TC(2)=T(1)*TC(2)
DO 10 J=1,IT
TAU2J=TAU(2,J)
ELTRMX(1,J)=TB(1)+TC(1)*TAU2J
ELTRMX(2,J)=TB(2)+TC(2)*TAU2J
ELTRMX(3,J)=TC(1)+TB(1)*TAU2J
ELTRMX(4,J)=TC(2)+TB(2)*TAU2J
10 CONTINUE
QEXT=2.0*(TB(1)+TC(1))
QSCAT=(TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2))/0.750
O1STAR=0.0
O2STAR=0.0
SUMRR=2.0*(TB(1)-TC(1))
SUMRI=2.0*(TB(2)-TC(2))
N=2
GO TO 2
11 CONTINUE
TC1=ACAPN*RRF+EN*RX
TC2=ACAPN*RF+EN*RX
C SEE EQUIVALENCE STMTS FOR EXPLANATION OF TAX ), ETC.
FNA=(TC1*TA(3)-TA(1))/(TC1*WFN(2)-WFN(1))
FNB=(TC2*TA(3)-TA(1))/(TC2*WFN(2)-WFN(1))
T(4)=T(1)/(EN*T(2))
T(2)=(T(2)*(EN+1.0))/EN
S=-S
SUMRR=SUMRR+S*T(3)*(TB(1)-TC(1))
MIEGX710
MIEGX720
MIEGX730
MIEGX740
MIEGX750
MIEGX760
MIEGX770
MIEGX780
MIEGX790
MIEGX800
MIEGX810
MIEGX820
MIEGX830
MIEGX840
MIEGX850
MIEGX860
MIEGX870
MIEGX880
MIEGX890
MIEGX900
MIEGX910
MIEGX920
MIEGX930
MIEGX940
MIEGX950
MIEGX960
MIEGX970
MIEGX980
MIEGX990
MIEGX000
MIEGX010
MIEGX020
MIEGX030
MIEGX040
MIEGX050
MIEGX060
MIEGX070
MIEGX080
MIEGX090
MIEGX100
MIEGX110
MIEGX120
MIEGX130
MIEGX140
MIEGX150
MIEGX160
MIEGX170
MIEGX180
MIEGX190
MIEGX200
MIEGX210
MIEGX220
MIEGX230
MIEGX240
MIEGX250
MIEGX260
MIEGX270
MIEGX280
MIEGX290
MIEGX300
MIEGX310
MIEGX320
MIEGX330
MIEGX340
MIEGX350
MIEGX360
MIEGX370
MIEGX380
MIEGX390
MIEGX400

```

	SUMRI=SUMRI+S*T(3)*(TB(2)-TC(2))	MIEGX410
C	SEE LATER COMMENTS ABOUT FOLLOWING STATEMENTS	MIEGX420
	QRTL1=QRT	MIEGX430
	QRT=SUMRR*SUMRR+SUMRI*SUMRI	MIEGX440
C	O1STAR CALCULATION	MIEGX450
	O1STAR=O1STAR+(TB(1)*TD(1)+TB(2)*TD(2)+TC(1)*TE(1)+TC(2)*TE(2))	MIEGX460
	1 *T(2)*4.0+4.0*T(4)*(TD(1)*TE(1)+TD(2)*TE(2))	MIEGX470
	IF (N.LT.3) GO TO 12	MIEGX480
C	O2STAR CALCULATION	MIEGX490
	F1=TF(1)*TF(1)+TF(2)*TF(2)+TG(1)*TG(1)+TG(2)*TG(2)	MIEGX500
	F2=TB(1)*TF(1)+TB(2)*TF(2)+TC(1)*TG(1)+TC(2)*TG(2)	MIEGX510
	F3=TD(1)*TG(1)+TD(2)*TG(2)+TE(1)*TF(1)+TE(2)*TF(2)	MIEGX520
	ENL1=EN-1.0	MIEGX530
	COF1=2.50*((EN-2.0)*ENL1-3.0)*((EN-2.0)*ENL1-3.0)*(2.0*EN-3.0)/	MIEGX540
	1 ((EN-2.0)*ENL1*(2.0*EN-1.0)*(2.0*EN-5.0))	MIEGX550
	COF2=7.50*(EN-2.0)*((EN+1.0)/(2.0*EN-1.0))	MIEGX560
	COF3=15.0/ENL1	MIEGX570
	O2STAR=O2STAR+COF1*F1+COF2*F2+COF3*F3	MIEGX580
12	CONTINUE	MIEGX590
	QEXT=QEXT+T(3)*(TB(1)+TC(1))	MIEGX600
	T(4)=TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2)	MIEGX610
	QSCAT=QSCAT+T(3)*T(4)	MIEGX620
	T(2)=EN*(EN+1.0)	MIEGX630
	T(1)=T(3)/T(2)	MIEGX640
	DO 13 J=1,IT	MIEGX650
	PI3J=PI(3,J)	MIEGX660
	TAU3J=TAU(3,J)	MIEGX670
	ELTRMX(1,J)=ELTRMX(1,J)+T(1)*(TB(1)*PI3J+TC(1)*TAU3J)	MIEGX680
	ELTRMX(2,J)=ELTRMX(2,J)+T(1)*(TB(2)*PI3J+TC(2)*TAU3J)	MIEGX690
	ELTRMX(3,J)=ELTRMX(3,J)+T(1)*(TC(1)*PI3J+TB(1)*TAU3J)	MIEGX700
	ELTRMX(4,J)=ELTRMX(4,J)+T(1)*(TC(2)*PI3J+TB(2)*TAU3J)	MIEGX710
13	CONTINUE	MIEGX720
	IF (N.LT.5) GO TO 14	MIEGX730
	QRTR=ABS((QRT-QRTL1)/QRT)	MIEGX740
C	TEST FOR CONVERGENCE ON QEXT, QSCAT, AND QRTR	MIEGX750
	IF ((T(4).LT. TOL).AND.(QRTR.LT. TOL)) GO TO 16	MIEGX760
14	N=N+1	MIEGX770
	DO 15 J=1,IT	MIEGX780
	PI(1,J)=PI(2,J)	MIEGX790
	PI(2,J)=PI(3,J)	MIEGX800
	TAU(1,J)=TAU(2,J)	MIEGX810
	TAU(2,J)=TAU(3,J)	MIEGX820
15	CONTINUE	MIEGX830
	FNAPP=FNAP	MIEGX840
	FNBPP=FBNP	MIEGX850
	FNAP=FNA	MIEGX860
	FNBP=FNB	MIEGX870
	GO TO 2	MIEGX880
16	CONTINUE	MIEGX890
	DO 18 J=1,IT	MIEGX900
	DO 17 I=1,4	MIEGX910
	T(I)=ELTRMX(I,J)	MIEGX920
17	CONTINUE	MIEGX930
	ELTRMX(1,J)=T(3)*T(3)+T(4)*T(4)	MIEGX940
	ELTRMX(2,J)=T(1)*T(1)+T(2)*T(2)	MIEGX950
	ELTRMX(3,J)=T(1)*T(3)+T(2)*T(4)	MIEGX960
	ELTRMX(4,J)=(T(2)*T(3)-T(4)*T(1))	MIEGX970
	PFNZRO=(ELTRMX(1,J)+ELTRMX(2,J))/2.0	MIEGX980
	IF (IAPXCT.EQ.1) GO TO 20	MIEGX990
	P(J)=PFNZRO	MIEGX000
18	CONTINUE	MIEGX010
C	ELTRMX(2,J) IS THE VERTICAL COMPONENT SCATTERING I1 (EYE1)	MIEGX020
C	ELTRMX(1,J) IS THE HORIZONTAL COMPONENT SCATTERING I2 (EYE2)	MIEGX030
C	ELTRMX(3,J) IS EQUIVALENT TO EYE3	MIEGX040
C	ELTRMX(4,J) IS EQUIVALENT TO -1.0*EYE4	MIEGX050
	T(1)=2.0*RX*RX	MIEGX060
	SGT=QEXT*T(1)	MIEGX070
	SGS=QSCAT*T(1)	MIEGX080
	O1STAR=3.0*O1STAR/(X*X*SGT)	MIEGX090
	O2STAR=4.0*O2STAR/(X*X*SGT)	MIEGX100

SGR=(SUMRR+SUMRR+SUMRI*SUMRI)*RX*RX	MIEGX110
QTD=(SGT)	MIEGX120
QSD=(SGS)	MIEGX130
QRD=(SGR)	MIEGX140
O1STRD=(O1STAR)	MIEGX150
O2STRD=(O2STAR)	MIEGX160
C LOOP FOR CALCULATION OF PFN AT ZERO DEGREES - FOR GPHASX	MIEGX170
IAPXCT=1	MIEGX180
IT=1	MIEGX190
GO TO 21	MIEGX200
20 IT=ITT	MIEGX210
RETURN	MIEGX220
1000 FORMAT (52H V GT 20000 ERROR IN CONTINUED FRACTIONS MIE ROUTINE,	MIEGX230
1 11H ** ALPHA =,E12.6,6H EM = ,E12.6,7H CAY = ,E12.6,	MIEGX240
2 1X,54H IT IS SUGGESTED THAT TOL=1.E-06 FOR SINGLE PRECISION.)	MIEGX250
END	MIEGX260


```
9 RHODEN=DENS(P)
10 CONTINUE
GO TO 13
11 WRITE (IOUT,12) TMCHUR,WAVE
12 FORMAT (10H TEMP. OF .E12.6,14H OR WAVEL. OF .E12.6,14H BEYOND RAN
1GE .24HOF DATA IN WATER ROUTINE/22H EXECUTION TERMINATED )
STOP
13 RETURN
END
```

WAT 01410
WAT 01420
WAT 01430
WAT 01440
WAT 01450
WAT 01460
WAT 01470
WAT 01480
WAT 01490

	SUBROUTINE GAUS(IT)	GAUS0010
	COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)	GAUS0020
	COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU	GAUS0030
	COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)	GAUS0040
	WRITE (IOUT,11)	GAUS0050
C	INITIALIZE ARRAY PC() USED FOR RUNNING SUMMATION	GAUS0060
	DO 1 I=1,IT	GAUS0070
	OL(I)=0.	GAUS0080
1	PC(I)=0.E+00	GAUS0090
C	LOOPS 2 AND 3 CALCULATE EXPANSION COEFS. FOR FUNCTION PSUM()	GAUS0100
C	VIA GAUSS-LEGENDRE QUADRATURE. THE COEFS. GO INTO ARRAY OL()	GAUS0110
	DO 3 I=1,IT	GAUS0120
	COF=W(I)*PSUM(I)	GAUS0130
	PLM1=C(I)	GAUS0140
	PLM2=1.	GAUS0150
	DO 2 LL=1,IT	GAUS0160
	L=LL-1	GAUS0170
	PL=PLM2	GAUS0180
	IF (LL.EQ.2) PL=PLM1	GAUS0190
	IF (LL.LE.2) GO TO 2	GAUS0200
	PL=2.*C(I)*PLM1-PLM2-(C(I)*PLM1-PLM2)/FLOAT(L)	GAUS0210
	PLM2=PLM1	GAUS0220
	PLM1=PL	GAUS0230
2	OL(LL)=OL(LL)+COF*PL*(FLOAT(L)+.5)	GAUS0240
3	CONTINUE	GAUS0250
	DO 7 I=1,IT	GAUS0260
	II=I-1	GAUS0270
7	WRITE (IOUT,8) II,OL(I)	GAUS0280
8	FORMAT (1H ,20X,16,20X,1PE14.7)	GAUS0290
11	FORMAT (1H /,1X,25X,1HL,20X,16HL-TH COEFFICIENT)	GAUS0300
	RETURN	GAUS0310
	END	GAUS0320

	SUBROUTINE DIMER(NGO)	DIM00010
	COMMON /IO/ IOIN, IOUT, NUNIT, IEO, NEOU	DIM00020
	GO TO (1,2,3), NGO	DIM00030
1	WRITE(IOUT,101)	DIM00040
101	FORMAT(1H, '**** THE INPUT VALUE OF IT IS GREATER	DIM00050
	1, ' THAN THE ARRAY DIMENSIONS', //, ' CHANGE THE DIMENSIONS OF THE ',	DIM00060
	2, ' FOLLOWING ARRAYS IN SUBS AND COMMON', //, ' COMMON BLOCK', 10X,	DIM00070
	3, ' ARRAY(S)', //, ' AGXM', 10X, ' C, H, OLT', //, ' PT2', 11X,	DIM00080
	4, ' PC, OL, RMS, PSUM, PSUMT, P', //, ' ALSO CHANGE THE VALUE OF JIMCK(1) '	DIM00090
	5, ' IN THE DATA STMT TO AGREE WITH THE NEW', //, ' DIMENSION LIMITS',	DIM00100
	6, //, 1H, '55H THE SECOND INDEX ON ARRAYS PI, TAU, AND ELTRMX IN MIEGX	DIM00110
	7, 1H, '56H MUST ALSO BE CHANGED AND ARRAYS P AND C CHANGED AS WELL. >	DIM00120
	STOP	DIM00130
2	WRITE (IOUT,102)	DIM00140
102	FORMAT(1H, '**** TOO MANY PARTICLE RADII FOR DIMENSION LIMITS: ',	DIM00150
	1, ' IN SUBS AND COMMON CHANGE THE FOLLOWING', //,	DIM00160
	2, ' ARRAYS', //, ' COMMON BLOCK', 10X, ' ARRAY(S)', //, ' PT1', 11X, ' F, R, FF',	DIM00170
	3, ' ARRAYS F AND R MUST BE CONSISTENT WITH THE FOLLOWING: ARRAY ',	DIM00180
	4, ' SIZE = 1 + 2**JDIMCK(2)', //, ' ARRAY FF MUST BE DIMENSIONED TO ONE',	DIM00190
	5, ' MORE THAN ARRAYS F AND R', //, ' ALSO CHANGE THE VALUE OF JDIMCK(2)',	DIM00200
	6, ' IN THE DATA STATEMENT')	DIM00210
	STOP	DIM00220
3	WRITE (IOUT,103)	DIM00230
103	FORMAT (1H, ' THE DIMENSIONS OF F AND '	DIM00240
	1, ' R DO NOT AGREE WITH THE FOLLOWING: ', //, ' SIZE=1+2**JDIMCK(2) ',	DIM00250
	2, ' WHERE JDIMCK(2) APPEARS IN THE DATA STATEMENT', //, ' ALSO DIMENSION	DIM00260
	3, ' ARRAY FF TO BE ONE MORE THAN ARRAYS F AND R')	DIM00270
	STOP	DIM00280
	END	DIM00290

```

C      BLOCK DATA                                BLK00010
C      IF YOU CHANGE THE DIMENSIONS MAKE SURE THAT YOU ALSO CHANGE THE    BLK00020
C      DATA STATEMENT CONTAINING JDIMCK( ):                               BLK00030
C      JDIMCK(1)=ORDER OF QUADRATURE (INPUT IT), I.E. MAX SIZE OF DIMENSION BLK00040
C      JDIMCK(2):USED IN COMPUTATION OF SIZE OF ARRAYS F AND R,          BLK00050
C      SIZE=1+2**JDIMCK(2), ARRAY FF SHOULD BE ONE MORE THAN F AND R.   BLK00060
C      JIMCK(3) WILL BE CALCULATED.                                       BLK00070
C      THE OTHER ARRAYS ARE REUSED, SO                                     BLK00080
C      CHECK THE SUBROUTINE CALLS AND COMMON BLOCKS TO SEE IF            BLK00090
C      ARRAYS HAVE BEEN RENAMED WHEN U CHANGE DIMENSIONS.                BLK00100
C      COMMON /AQXM/ C(65),W(65),OLT(65),JDIMCK(3)                        BLK00110
C      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)                    BLK00120
C      +,NRADI,PI, IDSTP,NKG,NHALV,NI                                     BLK00130
C      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)      BLK00140
C      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                               BLK00150
C      DATA IOIN,IOUT,NUNIT,IEO,NEOU /5,6,3*0/                          BLK00160
C      DATA JDIMCK /65,9,0/                                             BLK00170
C      EOSAEL OPTION: 65 PREDETERMINED ANGLES                            BLK00180
C      DATA /w                                                            BLK00190
+      0.0, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 8.0, 10.0,            BLK00200
+      12.0, 14.0, 16.0, 18.0, 20.0, 24.0, 28.0, 32.0, 36.0, 40.0,    BLK00210
+      44.0, 48.0, 52.0, 56.0, 60.0, 64.0, 68.0, 72.0, 76.0, 80.0,    BLK00220
+      84.0, 88.0, 92.0, 96.0, 100.0, 104.0, 108.0, 112.0, 116.0, 120.0, BLK00230
+      124.0, 128.0, 132.0, 136.0, 140.0, 142.0, 144.0, 146.0, 148.0, 150.0, BLK00240
+      152.0, 154.0, 156.0, 158.0, 160.0, 162.0, 164.0, 166.0, 168.0, 170.0, BLK00250
+      172.0, 174.0, 176.0, 178.0, 180.0/                               BLK00260
C      END                                                                    BLK00270

```

```

PROGRAM FLASH1
COMMON/IOUNT/IOIN, IOOUT
COMMON/CONST/PI
COMMON/IDATA/ROBS(3), RTAR(3), TARDEG, RSRC(3), TIMGUN, FOV, WAVE,
*          RADSEE, RADLOC, TARLEN, TARWID, TARHGT, PCTSEE, PCTLOC,
*          TIME, ISRC, RCTSEE, RCTLOC
PI=3.14159
IOIN=5
IOOUT=6
C**** NOTE ICHK=-1 DEFAULTS USETR INPUT TO WAVE1
ICHK=0
WAVE1=.55
CALL FLASH(WAVE1, ICHK)
WRITE (IOOUT, 100) WAVE1, ICHK
100  FORMAT (5X, 13HEOSAEI OUTPUT, /,
*5X, 13HWAVELENGTH = ,F6.1, 1X, 11HMICROMETERS, /,
*5X, 7HICK = ,I4)
STOP
END

```

```

DRIV0010
DRIV0020
DRIV0030
DRIV0040
DRIV0050
DRIV0060
DRIV0070
DRIV0080
DRIV0090
DRIV0100
DRIV0110
DRIV0120
DRIV0130
DRIV0140
DRIV0150
DRIV0160
DRIV0170
DRIV0180

```



```

SUBROUTINE FLASH(WAVE1, ICHK)
C*****
C*          PROGRAM FLASH
C*          EUSAEL80
C*****
DIMENSION FLASH1(3), FLASH2(3), IC1(3), IC2(3), IGUN(3)
COMMON/IOUNT/IOIN, IOOUT
COMMON/CONST/PI
COMMON/IDATA/ROBS(3), RTAR(3), TARDEG, RSRC(3), TIMGUN, FOV, WAVE,
*          RADSEE, RADLOC, TARLEN, TARWID, TARHGT, PCTSEE, PCTLQC,
*          TIME, ISRC, RCTSEE, RCTLOC
DATA FLASH1/3.85, 2.55, 0.0/
DATA FLASH2/0.15, 0.15, 0.0/
DATA IC1/2HMO, 2HT-, 2HUN/
DATA IC2/2HW, 2H55, 2HKN/
DATA IGUN/105, 100, 0/
98 FORMAT(/, 21X, 30H*****PROGRAM FLASH OUTPUT*****, /, 21X, 40(2H--))
99 FORMAT(21X, 20H*****END OF RUN*****, /, 21X, 40(2H--))
100 FORMAT(1H1, 20X, 40(2H**), /, 21X, 1H*, 34X, 13HPROGRAM FLASH, 31X, 1H*, /,
*21X, 1H*, 37X, 8HEUSAEL80, 33X, 1H*, /, 21X, 40(2H**))
101 FORMAT(/, 21X, 15H*****INPUT*****, /, 21X, 40(2H--))
102 FORMAT(21X, 14HSCENARIO DATA: /, 21X, 14HREFERENCE TIME, 1X, F8.3,
*1X, 3HSEC)
103 FORMAT(21X, 9HOBSERVER: /, 18X, 7HTARGET: /, 20X, 7HSOURCE: /)
104 FORMAT(24X, 6HX(OBS), 3X, F6.1, 2H M, 10X, 6HX(TAR), 3X, F6.1, 2H M, 10X,
*6HX(SRC), 3X, F6.1, 2H M, /,
*24X, 6HY(OBS), 3X, F6.1, 2H M, 10X, 6HY(TAR), 3X, F6.1, 2H M, 10X,
*6HY(SRC), 3X, F6.1, 2H M, /,
*24X, 6HZ(OBS), 3X, F6.1, 2H M, 10X, 6HZ(TAR), 3X, F6.1, 2H M, 10X,
*6HZ(SRC), 3X, F6.1, 2H M)
105 FORMAT(48X, 11HORIENTATION, 1X, F6.1, 1X, 3HDEG, 8X,
*10HEVENT TIME, 1X, F6.3, 1X, 3HSEC, /, 67X,
*10HCCW X-AXIS)
106 FORMAT(21X, 25HDETECTOR CHARACTERISTICS: /, /,
*24X, 13HFIELD OF VIEW, 12X, F6.1, 1X, 7HDEGREES, /,
*24X, 10HWAVELENGTH, 15X, F6.1, 1X, 11HMICROMETERS, /,
*24X, 21HRESOLUTION CRITERIA-- /,
*26X, 17H(A) FOR DETECTION, 6X, F6.3, 1X, 12HMILLIRADIANS, /,
*26X, 15H(B) FOR LOCK ON, 8X, F6.3, 1X, 12HMILLIRADIANS, /,
*24X, 24HRECOVERY TIME (R=100M)-- /,
*26X, 17H(A) FOR DETECTION, 6X, F6.1, 1X, 7HSECONDS, /,
*26X, 15H(B) FOR LOCK ON, 8X, F6.1, 1X, 7HSECONDS)
107 FORMAT(21X, 23HTARGET CHARACTERISTICS: /, /,
*24X, 6HLENGTH, 19X, F6.1, 1X, 6HMETERS, /,
*24X, 5HWIDTH, 20X, F6.1, 1X, 6HMETERS, /,
*24X, 6HHEIGHT, 19X, F6.1, 1X, 6HMETERS, /,
*24X, 19HEXPOSURE CRITERIA-- /,
*26X, 17H(A) FOR DETECTION, 6X, F6.1, 1X, 7HPERCENT, /,
*26X, 15H(B) FOR LOCK ON, 8X, F6.1, 1X, 7HPERCENT)
108 FORMAT(21X, 23HSOURCE CHARACTERISTICS: /, /,
*24X, 4HTYPE, 22X, I4, 2HMM, 1X, 2A2, /,
*24X, 22HFLASH (VISIBLE) RADIUS, 4X, F6.3, 1X, 6HMETERS, /,
*24X, 24HFLASH (VISIBLE) DURATION, 2X, F6.3, 1X, 7HSECONDS)
109 FORMAT(24X, 27H*****PROGRAM FLASH END*****, /, 21X, 40(2H--), 1H1)
IWRIT=1
IFLAG=0
C*****READ IN DATA
1 WRITE(IOOUT, 100)
CALL DATRD(IWRIT, IFLAG)
IF(ICHK.EQ.-1)WAVE=WAVE1
WAVE1=WAVE
IF(IFLAG.EQ.4)GO TO 9999
JSRC=ISRC
IF(ISRC.LT.1.OR.ISRC.GT.2)JSRC=3
FLASHR=FLASH1(JSRC)
FLASHT=FLASH2(JSRC)
WRITE(IOOUT, 101)
WRITE(IOOUT, 102)TIME
WRITE(IOOUT, 103)
WRITE(IOOUT, 104)(ROBS(I), RTAR(I), RSRC(I), I=1, 3)
FLASH010
FLASH020
*FLASH030
*FLASH040
*FLASH050
FLASH060
FLASH070
FLASH080
FLASH090
FLASH100
FLASH110
FLASH120
FLASH130
FLASH140
FLASH150
FLASH160
FLASH170
FLASH180
FLASH190
FLASH200
FLASH210
FLASH220
FLASH230
FLASH240
FLASH250
FLASH260
FLASH270
FLASH280
FLASH290
FLASH300
FLASH310
FLASH320
FLASH330
FLASH340
FLASH350
FLASH360
FLASH370
FLASH380
FLASH390
FLASH400
FLASH410
FLASH420
FLASH430
FLASH440
FLASH450
FLASH460
FLASH470
FLASH480
FLASH490
FLASH500
FLASH510
FLASH520
FLASH530
FLASH540
FLASH550
FLASH560
FLASH570
FLASH580
FLASH590
FLASH600
FLASH610
FLASH620
FLASH630
FLASH640
FLASH650
FLASH660
FLASH670
FLASH680
FLASH690
FLASH700

```

```

WRITE<IOOUT,105>TARDEC,TIMGUN
WRITE<IOOUT,106>FOV,WAVE,RADSEE,RADLOC,RCTSEE,RCTLOC
WRITE<IOOUT,107>TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC
WRITE<IOOUT,108>IGUN<JSRC>,IC1<JSRC>,IC2<JSRC>,FLASHR,FLASHT
CALL GETIM<FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL>
WRITE<IOOUT,98>
CALL DATWT<ISTOP,ISEE,ILOC,TIMLEF,TIMNOL>
WRITE<IOOUT,99>
GO TO 1
9999 WRITE<IOOUT,109>
C*****DEFINE EOSAEL OUTPUT
      ICHK=ISEE+1
      STOP
      END
FLASH710
FLASH720
FLASH730
FLASH740
FLASH750
FLASH760
FLASH770
FLASH780
FLASH790
FLASH800
FLASH810
FLASH820
FLASH830
FLASH840

```

```

SUBROUTINE DATRD(IWRIT,IFLAG) DTRD0010
C/***** DTRD0020
C/***** SUBROUTINE DATRD */DTRD0030
C/***** FLASH MODULE */DTRD0040
C/***** EOSAEL80 */DTRD0050
C/***** DTRD0060
C/***** THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS DTRD0070
C/***** THE SMOKE(EOSAEL) AND GRNAD(EOSAEL) MODULES DTRD0080
C/***** INPUTS DTRD0090
C/***** EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4, DTRD0100
C/***** FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL. DTRD0110
C/***** PER FIELD BEGINNING IN COL 11. THE CARDS ARE NOT ORDER DTRD0120
C/***** DEPENDENT. DTRD0130
C/***** NAME IGNORED DTRD0140
C/***** SCEN DTRD0150
C/***** TIME SCENARIO REFERENCE TIME(SEC) DTRD0160
C/***** ISRC SOURCE TYPE CODE (1=105MM 2=100MM) DTRD0170
C/***** OBSC ROBS(3) OBSERVER COORDINATES (X,Y,Z METERS) DTRD0180
C/***** TARC RTAR(3) TARGET COORDINATES (X,Y,Z METERS) DTRD0190
C/***** TARDEG TARGET ORIENTATION COUNTERCLOCKWISE TO DTRD0200
C/***** POSITIVE X AXIS DTRD0210
C/***** SRCC RSRC(3) COORDINATES OF FLASH CENTER (X,Y,Z METERS) DTRD0220
C/***** TIMGUN TIME OF INIATION OF GUNFLASH DTRD0230
C/***** DCHR FOV DETECTOR FIELD OF VIEW (DEGREES) DTRD0240
C/***** WAVE DETECTOR WAVELENGTH (MICROMETERS) DTRD0250
C/***** RADSEE ANGULAR RESOLUTION NEEDED TO DETECT DTRD0260
C/***** (MILLIRADIANS) DTRD0270
C/***** RADLOC ANGULAR RESOLUTION NEEDED TO LOCK ON DTRD0280
C/***** (MILLIRADIANS) DTRD0290
C/***** RCTSEE RECOVERY TIME AT 100 METERS FOR DTRD0300
C/***** DETECTION (SECONDS) DTRD0310
C/***** RCTLOC RECOVERY TIME AT 100 METERS FOR DTRD0320
C/***** LOCK ON (SECONDS) DTRD0330
C/***** TCHR TARLEN TARGET LENGTH (METERS) DTRD0340
C/***** TARWID TARGET WIDTH (METERS) DTRD0350
C/***** TARHGT TARGET HEIGHT (METERS) DTRD0360
C/***** PCTSEE FRACTION OF EXPOSURE NEEDED FOR DETECTION DTRD0370
C/***** (PERCENT) DTRD0380
C/***** PCTLOC FRACTION OF EXPOSURE NEEDED FOR LOCK ON DTRD0390
C/***** (PERCENT) DTRD0400
C/***** GO SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT DTRD0410
C/***** DONE END OF JOB. DTRD0420
C/***** DTRD0430
C/***** DTRD0440
C/***** DTRD0450
C/***** DTRD0460
C/***** DTRD0470
C/***** DTRD0480
C/***** DTRD0490
C/***** COMMON/IOUNT/IOIN,IOOUT DTRD0500
C/***** COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE, DTRD0510
C/***** * RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC, DTRD0520
C/***** * TIME,ISRC,RCTSEE,RCTLOC DTRD0530
C/***** DIMENSION IR(18),IR1(2),R1(7),INAME(35) DTRD0540
C/***** DATA IR/2HNA,2HME,2HSC,2HEN,2HOB,2HSC,2HTA,2HRC,2HSR,2HCC, DTRD0550
C/***** * 2HDC,2HHR,2HTC,2HHR,2HGO,2H 2HDO,2HNE/ DTRD0560
C/***** 100 FORMAT(21X,20H*****CARD INPUT*****,/,21X,40(2H--)) DTRD0570
C/***** 101 FORMAT(2A2,6X,7F10.3) DTRD0580
C/***** 102 FORMAT(21X,2A2,6X,7F10.3) DTRD0590
C/***** 103 FORMAT(2A2,6X,35A2) DTRD0600
C/***** 104 FORMAT(21X,2A2,6X,35A2) DTRD0610
C/***** DTRD0620
C/***** BEGINNING OF READ LOOP DTRD0630
C/***** DTRD0640
C/***** IF(IWRIT.EQ.0)GO TO 6 DTRD0650
C/***** WRITE(IOOUT,100) DTRD0660
C/***** DO 10 I=1,9 DTRD0670
C/***** IF(I.EQ.9)GO TO 90 DTRD0680
C/***** IF(IFLAG.GT.0)GO TO 4 DTRD0690
C/***** IFLAG=1 DTRD0700
C/***** READ(IOIN,103)IR1(1),IR1(2),INAME(J),J=1,35) DTRD0710

```

```

IF(IWRIT.EQ.0)GO TO 4
WRITE(IOOUT,104)IR1(1),IR1(2),(INAME(J),J=1,35)
4 READ(IOIN,101) IR1(1),IR1(2),(R1(J),J=1,7)
IF(IWRIT.EQ.0) GO TO 5
WRITE(IOOUT,102) IR1(1),IR1(2),(R1(J),J=1,7)
5 IF(IR1(1).EQ.IR(17).AND.IR1(2).EQ.IR(18)) GO TO 998
C*****
C*** RELATING INPUT DATA TO VARIABLE NAMES.
C*****
IF(IR1(1).EQ.IR(1).AND.IR1(2).EQ.IR(2)) GO TO 10
IF(IR1(1).EQ.IR(3).AND.IR1(2).EQ.IR(4)) GO TO 20
IF(IR1(1).EQ.IR(5).AND.IR1(2).EQ.IR(6)) GO TO 30
IF(IR1(1).EQ.IR(7).AND.IR1(2).EQ.IR(8)) GO TO 40
IF(IR1(1).EQ.IR(9).AND.IR1(2).EQ.IR(10)) GO TO 50
IF(IR1(1).EQ.IR(11).AND.IR1(2).EQ.IR(12)) GO TO 60
IF(IR1(1).EQ.IR(13).AND.IR1(2).EQ.IR(14)) GO TO 70
IF(IR1(1).EQ.IR(15).AND.IR1(2).EQ.IR(16)) GO TO 9999
C*****
C ERROR CAUTION FOR INVALID DATA CARD
C*****
IFLAG=2
WRITE(IOOUT,105)
105 FORMAT(21X,35H*****CAUTION***** INVALID DATA CARD)
GO TO 10
20 TIME=R1(1)
ISRC=IFIX(R1(2))
GO TO 10
30 ROBS(1)=R1(1)
ROBS(2)=R1(2)
ROBS(3)=R1(3)
GO TO 10
40 RTAR(1)=R1(1)
RTAR(2)=R1(2)
RTAR(3)=R1(3)
TARDEG=R1(4)
GO TO 10
50 RSRC(1)=R1(1)
RSRC(2)=R1(2)
RSRC(3)=R1(3)
TIMGUN=R1(4)
GO TO 10
60 FOV=R1(1)
WAVE=R1(2)
RADSEE=R1(3)
RADLOC=R1(4)
RCTSEE=R1(5)
RCTLOC=R1(6)
GO TO 10
70 TARLEN=R1(1)
TARWID=R1(2)
TARHGT=R1(3)
PCTSEE=R1(4)
PCTLOC=R1(5)
10 CONTINUE
GO TO 9999
C*****
C*****CAUTION FOR TOO MANY CARDS
C*****
90 WRITE(IOOUT,106)
IFLAG=3
106 FORMAT(21X,17H*****CAUTION*****
*21X,56HMORE THAN 10 DATA CARDS ENTERED---REMAINING CARDS IGNORED)
GO TO 9999
998 IFLAG=4
9999 RETURN
END
DTRD0710
DTRD0720
DTRD0730
DTRD0740
DTRD0750
DTRD0760
DTRD0770
DTRD0780
DTRD0790
DTRD0800
DTRD0810
DTRD0820
DTRD0830
DTRD0840
DTRD0850
DTRD0860
DTRD0870
DTRD0880
DTRD0890
DTRD0900
DTRD0910
DTRD0920
DTRD0930
DTRD0940
DTRD0950
DTRD0960
DTRD0970
DTRD0980
DTRD0990
DTRD1000
DTRD1010
DTRD1020
DTRD1030
DTRD1040
DTRD1050
DTRD1060
DTRD1070
DTRD1080
DTRD1090
DTRD1100
DTRD1110
DTRD1120
DTRD1130
DTRD1140
DTRD1150
DTRD1160
DTRD1170
DTRD1180
DTRD1190
DTRD1200
DTRD1210
DTRD1220
DTRD1230
DTRD1240
DTRD1250
DTRD1260
DTRD1270
DTRD1280
DTRD1290
DTRD1300
DTRD1310
DTRD1320
DTRD1330
DTRD1340
DTRD1350
DTRD1360

```

```

SUBROUTINE DATWT(ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)
C*****
C* SUBROUTINE DATWT
C* FLASH MODULE
C* EOSAEL80
C*****
COMMON/IOUNT/IOIN,IOOUT
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
* RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
* TIME,ISRC,RCTSEE,RCTLOC
DIMENSION I1(3),I2(3)
DATA I1/2HNO,2HYE,2HPA/
DATA I2/2H ,2HS ,2HRT/
IANS1=ISEE+1
IF(ISEE.EQ.3)IANS1=3
IF(ILOC.EQ.0)IANS2=2
IF(ILOC.EQ.1)IANS2=1
IF(ILOC.EQ.2)IANS2=3
IF(ILOC.EQ.0)IANS2=2
IF(ILOC.EQ.3)IANS2=3
82 FORMAT(21X,36HDETECTION EXPOSURE CRITERIA DEFEATED)
83 FORMAT(21X,38HDETECTION RESOLUTION CRITERIA DEFEATED)
92 FORMAT(21X,34HLOCK ON EXPOSURE CRITERIA DEFEATED)
93 FORMAT(21X,36HLOCK ON RESOLUTION CRITERIA DEFEATED)
97 FORMAT(21X,17HTARGET OBSCURED ?,1X,2A2,/,
* 21X,15HTARGET LOCKED ?,1X,2A2)
98 FORMAT(21X,10HTIME LEFT:,4X,F8.3,1X,7HSECONDS,/,
* 21X,13HTIME NO LOCK:,1X,F8.3,1X,7HSECONDS)
99 FORMAT(21X,25HPROGRAM FLASH--STOP CODE:,1X,3I1)
100 FORMAT(21X,11HSOURCE TYPE,1X,I2,1X,12HUNIDENTIFIED)
101 FORMAT(21X,17HINPUT WAVELENGTH:,1X,F6.1,1X,11HMICROMETERS,
*25H IS OUT OF RANGE OF MODEL)
102 FORMAT(21X,37HGUNFLASH HAS NOT OCCURED YET: TIME = ,1X,F8.3,/,42X,
*16HTIME OF FLASH = ,1X,F8.3)
103 FORMAT(21X,38HFLASH IS NOT IN DETECTOR FIELD OF VIEW)
104 FORMAT(21X,24HFLASH IS BEHIND TARGET--,
*15H NO OBSCURATION)
105 FORMAT(21X,24HNORMAL PROGRAM EXECUTION)
WRITE(IOOUT,99)ISTOP,ISEE,ILOC
IF(ISTOP.GT.1)GO TO 1
IF(ISTOP.GT.0)GO TO 10
WRITE(IOOUT,100)ISRC
GO TO 9999
10 WRITE(IOOUT,101)WAVE
GO TO 9999
1 WRITE(IOOUT,97)I1(IANS1),I2(IANS1),I1(IANS2),I2(IANS2)
IF(ISEE.LT.2.AND.ILOC.LT.2)GO TO 11
IF(ISEE.EQ.2)WRITE(IOOUT,82)
IF(ISEE.EQ.3)WRITE(IOOUT,83)
IF(ILOC.EQ.2)WRITE(IOOUT,92)
IF(ILOC.EQ.3)WRITE(IOOUT,93)
GO TO 9998
11 IGO=ISTOP-1
GO TO (2,3,4,5,6,7,8,9),IGO
2 WRITE(IOOUT,102)TIME,TIMGUN
GO TO 9999
3 WRITE(IOOUT,103)
GO TO 9999
4 WRITE(IOOUT,103)
GO TO 9999
5 WRITE(IOOUT,104)
GO TO 9999
6 WRITE(IOOUT,103)
GO TO 9999
7 WRITE(IOOUT,103)
GO TO 9999
8 WRITE(IOOUT,103)
GO TO 9999
9 WRITE(IOOUT,105)
9998 WRITE(IOOUT,98)TIMLEF,TIMNOL

```

```

DAT00010
DAT00020
DAT00070
DAT00040
DAT00050
DAT00060
DAT00070
DAT00080
DAT00090
DAT00100
DAT00110
DAT00120
DAT00130
DAT00140
DAT00150
DAT00160
DAT00170
DAT00180
DAT00190
DAT00200
DAT00210
DAT00220
DAT00230
DAT00240
DAT00250
DAT00260
DAT00270
DAT00280
DAT00290
DAT00300
DAT00310
DAT00320
DAT00330
DAT00340
DAT00350
DAT00360
DAT00370
DAT00380
DAT00390
DAT00400
DAT00410
DAT00420
DAT00430
DAT00440
DAT00450
DAT00460
DAT00470
DAT00480
DAT00490
DAT00500
DAT00510
DAT00520
DAT00530
DAT00540
DAT00550
DAT00560
DAT00570
DAT00580
DAT00590
DAT00600
DAT00610
DAT00620
DAT00630
DAT00640
DAT00650
DAT00660
DAT00670
DAT00680
DAT00690
DAT00700

```

9999 RETURN
END

DAT00710
DAT00720

```

SUBROUTINE GETIM(FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)      GTM00010
C*****GTM00020
C*      SUBROUTINE GETIM      *GTM00030
C*      FLASH MODULE      *GTM00040
C*      EGSSEL80      *GTM00050
C*****GTM00060
REAL LOCST0,LOCFRA      GTM00070
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
*      RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTL0C,
*      TIME,ISRC,RCTSEE,RCTL0C      GTM00100
COMMON/CONST/PI      GTM00110
C*****ALL DISTANCES IN METERS ANGLES CONVERTED TO RADIANS(CCW A-AXES
DIMENSION DOTVEC(3),RHTVEC(3),RHFVEC(3),RFLASH(3)      GTM00120
C*****SCALAR(A,B)=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)      GTM00130
SCALAR(A1,A2,A3,B1,B2,B3)=A1*B1+A2*B2+A3*B3      GTM00140
C*****DEFAULT AND CONVERT INPUT      GTM00150
ANGGUN=0.0      GTM00160
ELEGUN=0.0      GTM00170
GUNHT=0.0      GTM00180
GUNLEN=0.0      GTM00190
TARHHT=TARHGT/2.0      GTM00200
ANGTAR=TARDEG*(PI/180.0)      GTM00210
HAFFOV=(FOV/2.0)*(PI/180.0)      GTM00220
SEESTO=RADSEE/1000.0      GTM00230
LOCSTO=RADLOC/1000.0      GTM00240
SEEFRA=PCTSEE/100.0      GTM00250
LOCFRA=PCTL0C/100.0      GTM00260
RADFL=FLASHR      GTM00270
DURTIM=FLASHT      GTM00280
DURSEE=10.0*RCTSEE      GTM00290
DURLOC=10.0*RCTL0C      GTM00300
REACFA=SEEFRA-LOCFRA      GTM00310
C*****INITIALIZE FLAGS *****      GTM00320
AGINSE=0.0      GTM00330
AGINLO=0.0      GTM00340
TIMLEF=0.0      GTM00350
TIMNOL=0.0      GTM00360
TIMGUN=TIME-TIMGUN      GTM00370
ISEE=0      GTM00380
ILOC=0      GTM00390
ISTOP=0      GTM00400
IF(ISRC.LT.1.OR.ISRC.GT.2)GO TO 9999      GTM00410
ISTOP=1      GTM00420
IWAVE=0      GTM00430
IF(WAVE.GE.0.40.AND.WAVE.LE.0.70)IWAVE=1      GTM00440
IF(WAVE.GE.8.00.AND.WAVE.LE.12.0)IWAVE=2      GTM00450
IF(IWAVE.EQ.0)GO TO 9999      GTM00460
ISTOP=2      GTM00470
IF(TIMGUN.LT.0.)GO TO 9999      GTM00480
TIMLEF=DURTIM-TIMGUN      GTM00490
C*****TIMLEF IS DURATION LEFT OF FLASH      GTM00500
TIMNOL =TIMLEF+REACFA*DURTIM      GTM00510
C*****TIMNOL IS DURATION OF LOCK      GTM00520
FLSHIF=RADFL      GTM00530
C*****CALCULATE FLASH COORDINATES FROM GIVEN COORDINATES      GTM00540
RFLASH(1)=RSRC(1)+FLSHIF*COS(ANGGUN)*COS(ELEGUN)      GTM00550
RFLASH(2)=RSRC(2)+FLSHIF*SIN(ANGGUN)*COS(ELEGUN)      GTM00560
RFLASH(3)=RSRC(3)+FLSHIF*SIN(ELEGUN) + GUNHT      GTM00570
C*****DEFINE DIRECTION OF TARGET UNIT VECTOR      GTM00580
DOTVEC(1)=COS(ANGTAR)      GTM00590
DOTVEC(2)=SIN(ANGTAR)      GTM00600
DOTVEC(3)=0      GTM00610
C*****DEFINE OBSERVER-TARGET,OBSERVER-FLASH VECTORS      GTM00620
DO 10 I=1,3      GTM00630
RHTVEC(I)=RTAR(I)-ROBS(I)      GTM00640
RHFVEC(I)=RFLASH(I)-ROBS(I)      GTM00650
10 CONTINUE      GTM00660
C*****ADD HALF TARGET HEIGHT      GTM00670
RHTVEC(3)=RHTVEC(3)+TARHHT      GTM00680
C*****FIND LENGTHS      GTM00690

```

```

A1=RHTVEC<1>
A2=RHTVEC<2>
A3=RHTVEC<3>
B1=RHFVEC<1>
B2=RHFVEC<2>
B3=RHFVEC<3>
RHT=SQRT<SCALAR<A1,A2,A3,A1,A2,A3>>
RHF=SQRT<SCALAR<B1,B2,B3,B1,B2,B3>>
C*****FIND COSINES BETWEEN VECTORS
C1=DOTVEC<1>
C2=DOTVEC<2>
C3=DOTVEC<3>
GAMMAD=SCALAR<C1,C2,C3,B1,B2,B3>/RHF
GAMMAT=SCALAR<A1,A2,A3,B1,B2,B3>/<RHT*RHF>
W=SCALAR<C1,C2,C3,A1,A2,A3>/RHT
C*****FIND TRIG FCNS OF THETA=HALF-ANGLE OF FLASH CONE
SINTH=RADFL/RHF
COSTH=SQRT<1.0-SINTH*SINTH>
COSSQ=COSTH*COSTH
COSINE=ABS<W>
SINE=SQRT<1.0-COSINE**2>
C*****CALCULATE CYLINDRICAL TARGET LONGEST DIMENSION AS SEEN IN
C*****PLANE PERPENDICULAR TO RHFVEC
TARDIM=TARLEN*SINE+TARWID*COSINE
HAFDIM=TARDIM/2.
THETA=ATAN2<SQRT<1.0-COSTH**2>,COSTH>
THETAT=ATAN2<SQRT<1.0-GAMMAT**2>,GAMMAT>
DELTH=THETAT-THETA
ISTOP=3
IF<DELTH.GT.HAFFOV>GO TO 9999
C*****IF OUT OF FIELD OF VIEW,RETURN.
DISPLC=RHT*SIN<DELTH>
PROJRH=RHT*GAMMAT
ISTOP=4
IF<DISPLC.GT.HAFDIM>GO TO 99
ISTOP=5
IF<PROJRH.LT.RHF.AND.IWAVE.LT.2>GO TO 99
C*****NO OBSCURATION IF TARGET IN FRONT OF FLASH SO RETURN
C*****FROM HERE ON, GAMMAT NECESSARILY POSITIVE
C*****FIND INTERSECTIONS OF FLASH CONE WITH DOTVEC EXTENDED FROM TARGET
C*****GET COEFFICIENT OF QUADRATIC EQUATION FOR DISTANCE ALONG DOTVEC
C*****FROM TARGET TO FLASH CONE/RHT
A=COSSQ-GAMMAD**2
BD2=COSSQ*W-GAMMAD*GAMMAT
C=COSSQ-GAMMAT**2
B=BD2*2.
C*****IF DISCRIMINANT NEGATIVE, NO INTERSECTION
DISCRM=BD2*BD2-A*C
ISTOP=6
IF<DISCRM.LT.0.>GO TO 99
C*****IF A=0, QUADRATIC FORMULA BLOWS UP. ACTUALLY HAVE LINEAR EQN.
IF<ABS<A>.GT.1.E-30>GOTO20
IF<ABS<B>.LT.1.E-30>B=SIGN<1.E-30,B>
SPL=-C/B*RHT
ZPL=SPL*GAMMAD+PROJRH
ISTOP=7
IF<ZPL.LE.0.>GO TO 99
C*****REJECT IF SOLE INTERSEC IS W NEG CONE SHEET.MEANINGLESS
DPL=ABS<SPL>*SINE
IF<GAMMAT-COSTH>14,14,12
C*****IF TAR OUTSIDE FLASH CONE GO TO 14, OTHERWISE 12
12 OBSCUR=HAFDIM+AMIN1<DPL,HAFDIM>
GO TO 30
14 OBSCUR=AMAX1<0.,HAFDIM-DPL>
GO TO 30
20 ROOT=SQRT<DISCRM>
ROOT=SIGN<ROOT,A>
C*****ABOVE NOT REALLY NEEDED BUT NICER TO HAVE SPL.GT.SMI-SEE BELOW
SPL=(-BD2+ROOT)/A *RHT
SMI=(-BD2-ROOT)/A *RHT

```

```

GTM00710
GTM00720
GTM00730
GTM00740
GTM00750
GTM00760
GTM00770
GTM00780
GTM00790
GTM00800
GTM00810
GTM00820
GTM00830
GTM00840
GTM00850
GTM00860
GTM00870
GTM00880
GTM00890
GTM00900
GTM00910
GTM00920
GTM00930
GTM00940
GTM00950
GTM00960
GTM00970
GTM00980
GTM00990
GTM01000
GTM01010
GTM01020
GTM01030
GTM01040
GTM01050
GTM01060
GTM01070
GTM01080
GTM01090
GTM01100
GTM01110
GTM01120
GTM01130
GTM01140
GTM01150
GTM01160
GTM01170
GTM01180
GTM01190
GTM01200
GTM01210
GTM01220
GTM01230
GTM01240
GTM01250
GTM01260
GTM01270
GTM01280
GTM01290
GTM01300
GTM01310
GTM01320
GTM01330
GTM01340
GTM01350
GTM01360
GTM01370
GTM01380
GTM01390
GTM01400

```


C*****SPL,SMI ARE 2 LENGTHS ALONG DOTVEC OF INTERSEC PTS W FLASH CONE	GTM01410
C*****ZPL,ZMI ARE 2 COORDINATES OF INTERSECTIONS<ON FLASH CONE AXIS>	GTM01420
ZPL=SPL*GAMMAD+PROJRH	GTM01430
ZMI=SMI*GAMMAD+PROJRH	GTM01440
ISTOP=8	GTM01450
IF<ZPL.LE.0. AND,ZMI.LE.0.>GO TO 99	GTM01460
C*****REJECT IF BOTH INTERCEPTS IN NEGATIVE CONE	GTM01470
C*****DPL,DMI ARE PROJECNS OF SPL,SMI PERPEN TO RHTVEC	GTM01480
C*****THEIR MAGNITUDES ARE LIMITED TO HAFDIM	GTM01490
DPL=SPL*SINE	GTM01500
DPL=AMINI<DPL,HAFDIM>	GTM01510
DPL=AMAX1<DPL,-HAFDIM>	GTM01520
DMI=SMI*SINE	GTM01530
DMI=AMINI<DMI,HAFDIM>	GTM01540
DMI=AMAX1<DMI,-HAFDIM>	GTM01550
OBSCUR=ABS<DPL-DMI>	GTM01560
IF<ZPL*ZMI.GT.0.>GOTO30	GTM01570
C*****SKIP AROUND UNLESS BOTH SHEETS OF CONE INVOLVED	GTM01580
IF<ZPL.LE.0.>DPL=SIGN<HAFDIM,DPL>	GTM01590
IF<ZMI.LE.0.>DMI=SIGN<HAFDIM,DMI>	GTM01600
OBSCUR=TARDIM-ABS<DPL-DMI>	GTM01610
C*****ABOVE BRANCH RARE INTERSEC W BOTH CONE SHEETS.NEG SHEET IGNORE.	GTM01620
30 CONTINUE	GTM01630
SEEN=TARDIM-OBSCUR	GTM01640
ISTOP=9	GTM01650
98 IF<IWAVE.EQ.2>GO TO 1	GTM01660
CALL VSBLC<ISEE,ILOC,SEEN,TARDIM,RHT,TIMLEF,SEESTO,LOCSTO,	GTM01670
* SEEFRA,LOCFRA>	GTM01680
GO TO 99	GTM01690
1 CALL IRBLC<RHF,ISEE,ILOC,AGINSE,AGINLO,DURSEE,DURLOC,TIMGON>	GTM01700
99 CONTINUE	GTM01710
TIMLEF=AMAX1<TIMLEF,AGINSE>	GTM01720
TIMNOL=AMAX1<TIMNOL,AGINLO>	GTM01730
9999 RETURN	GTM01740
END	GTM01750

```

SUBROUTINE IRBLC(RHF, ISEE, ILOC, AGINSE, AGINLO, DURSEE, DURLOC, TIMGON) IRBLC010
C***** IRBLC020
C*          SUBROUTINE IRBLC * IRBLC030
C*          FLASH MODULE * IRBLC040
C*          EOSAEL80 * IRBLC050
C***** IRBLC060
      INTEGER LOCIRM IRBLC070
      SEEIRM=4.0 IRBLC080
      LOCIRM=5.0 IRBLC090
      SQRTRH=SQRT(RHF) IRBLC100
      AGINSE=DURSEE/SQRTRH IRBLC110
      AGINLO=DURLOC/SQRTRH IRBLC120
C***** ABOVE WHOLLY EMPIRICAL FROM FIT TO TV TAPES AT 2 RANGES IRBLC130
      IF(AGINSE.GT.SEEIRM) AGINSE=SEEIRM IRBLC140
      IF(AGINLO.GT.LOCIRM) AGINLO=LOCIRM IRBLC150
      IF(AGINSE.GT.TIMGON) ISEE=1 IRBLC160
      IF(AGINLO.GT.TIMGON) ILOC=1 IRBLC170
      RETURN IRBLC180
      END IRBLC190

```

```

SUBROUTINE VSBLC(ISEE, ILOC, SEEN, TARDIM, RHT, TIMLEF, SEESTO, LOCSTO, VSBLC010
*          SEEFRA, LOCFRA) VSBLC020
C***** VSBLC030
C*          SUBROUTINE VSBLC * VSBLC040
C*          FLASH MODULE * VSBLC050
C*          EOSAEL80 * VSBLC060
C***** VSBLC070
REAL LOCCRI, LOCFRA, LOCSTO VSBLC080
C***** SEEFRA IS FRACTION THAT MUST BE SEEN, LIKEWISE FOR LOCK VSBLC090
C***** SEESTO IS MIN ANGLE THAT MUST BE SEEN, LIKEWISE FOR LOCK VSBLC100
SEEANG=SEEN/RHT VSBLC110
TARANG=TARDIM/RHT VSBLC120
C***** DEFINE SEEN ANGLE AND TARGET ANGLE VSBLC130
SEECRI=AMIN1(TARANG, SEESTO) VSBLC140
LOCCRI=AMIN1(TARANG, LOCSTO) VSBLC150
IF(TIMLEF.LE.0.)GO TO 9999 VSBLC160
IF(SEEANG.LT.SEEFRA*TARANG)GO TO 48 VSBLC170
ISEE=2 VSBLC180
GO TO 50 VSBLC190
48 IF(SEEANG.LT.SEECRI)GO TO 49 VSBLC200
ISEE=1 VSBLC210
GO TO 50 VSBLC220
49 CONTINUE VSBLC230
ISEE=3 VSBLC240
50 CONTINUE VSBLC250
IF(SEEANG.LT.LOCFRA*TARANG)GO TO 98 VSBLC260
ILOC=2 VSBLC270
GO TO 9999 VSBLC280
98 IF(SEEANG.LT.LOCCRI)GO TO 99 VSBLC290
ILOC=1 VSBLC300
GO TO 9999 VSBLC310
99 CONTINUE VSBLC320
ILOC=3 VSBLC330
9999 RETURN VSBLC340
END VSBLC350

```

EORUN	4.0									
WAVL	10.6	10.6								
VIS	7.0									
TURB										
XSCALE	3.0									
SMOKE										
DRTRAN										
LZTRAN										
CLTRAN	3.0									
SCREEN										
UVRCST	2.0									
CLIMAT	1.0	1.0	5.0	6.0	-1.0	1.0				
GRNADE										
GO										
PARM	0.1016	1.3E-04	0.0	400.0	5.0	512.0				
CN1	1.0	6.0E-13	1.0E-14	1.3E-14	2.7E-14	5.0E-14				
V1	1.0	0.93	0.93	0.93	0.93	0.93				
DVRV	0.400	500.0								
CN2	1.0	6.0E-13	1.0E-14	1.3E-14	2.7E-14	5.0E-14				
V2	1.0	0.93	0.93	0.93	0.93	0.93				
GO										
END										
FOG	1.									
HORZ	0.4									
GO										
FOG	2.									
SLNH	0.133,56.3									
GO										
FOG	3.									
HORZ	0.4									
GO										
END										
MUNC	0.0	-50.0	0.0							
OBSC	200.0	0.0	2.0							
TARC	200.0	0.0	0.0							
BART	5.	180.	0.0	90.						
OUTP	0.	0.	0.0	0.						
BURN	10.	0.	0.0	0.	0.					
GO										
BURN	4.	0.	0.	0.	0.					
BART	5.	250.	5.	90.	0.					
GO										
DONE										
END										
MET1	4.	2.	286.0	2.	2.0	0.0				
MET2	1.	53.								
SOIL	2.	0.0	0.15							
CHAR	3.	6.8	0.0							
EXPL	1.	1.	1.	0.0	0.0	0.0	0.0	0.0	0.0	0.0
OBSC	200.0	0.0	2.0							
TRNC	-200.0	0.0	2.0							
RECC	200.0	0.0	2.0							
TIMS	1.0	71.0	2.0							
GO	1.									
DONE										
END										
END	8.55	15.	0.4							
SEEK,	0.2,	0.0,	0.6							
TARG,	-0.2,	0.0,	0.002							
CLST,	0.20,	0.40,								
GO										
SEEK,	0.133,	0.0,	0.5							
GO										
SEEK,	0.0,	0.0,	0.3							
GO										
END										
1 1 1										
TARV	1.0	2.0	0.24	2.3	81.	2.	.000			
SENS	.99	8.	1.	0.	0.	1.				

GO							
SENS	.90	8.	1.	0.	0.	1.	
GO							
SENS	.75	8.	1.	0.	0.	1.	
GO							
SENS	.50	8.	1.	0.	0.	1.	
GO							
SENS	.10	8.	1.	0.	0.	1.	
GO							
DONE							
SCRN	15.	400.	0.4	0.	90.	74.	2.
DONE							
END							
OPOS,	-0.0667,	0.0,	0.2				
CLDS,	0.2,	200.0,	40.0,	0.7,	1.0		
SPOS,	-0.2,	0.0,	0.002				
BKGR,	50.0						
GRND,	50.0						
TEMP,	9.0						
GO							
UPOS,	-0.1333,	0.0,	0.1				
GO							
END							
NAME							
UTP							
OBSC	-200.0	0.0	2.0				
MUNC	-200.0	0.0	2.0	95.0	100.0	10.0	
TARC	-200.0	+40.0	50.0				
BART	5.0	400.0	50.0	90.0			
MUNT	1.0	0.793	14.3	1.0	0.0	4.7	0.07
METR	50.0	2.0	220.0	4.0	20.0	0.0	0.0
EXTC							
BURN							
MISC							
GO							
DONE							
END							
MAVL	1.06	1.06	4.0				
VIS	5.0	5.0					
BASCAT							
F CLOUD							
GO							
PART,	1.	5000.	1.				
SORC,	-0.2,	0.	-0.098,	90.,	0.,	50.	
DETR,	0.2,	0.	-0.098,	90.,	180.,	1.,	1.
CLDS,	.1,	.2,	.1,	0.,	0.,	0.,	5.
GRND,	-0.1,	0.5,					
PULS,	.33,	0.					
GO							
END							
OPOS,	0.0,	0.0,	0.1				
RPOS,	-0.2,	0.0,	0.002				
SPOS,	-0.2,	0.0,	0.002				
AXES,	0.1,	0.0,	0.1				
CLDS,	5.0,	0.95,	4.0,	1.0,	9.8		
ATMO,	2.0,	50.0					
BKGR,	0.5,	0.0					
SANG,	80.0,	0.0					
LUND,	0.0						
GO							
END							
FREQ	35.0	35.0					
NMMW	2.0						
GO							
PATH,	0.4						
ATMO,	15.0	1013.25	6.44				
FOGD,	0.5						
RAIN,	5.0						
GO							
ATMO,	-1.0	1015.2	7.7				

RAIN,		0.0								
SNOW,		5.0								
GO										
END										
WVNUM		2010.	2710.	2.						
LT4M										
RESF										
SPTCT										
GO										
ENVR		3.	2.	2.	4.	1.	1.			
EMIS		.100+01	.283+03	.950+00	.295+03					
ATM		.650+02	0.0	0						
TRHG		.400+00	.450+02	.900+02	.450+02					
REFL		0.5	0.5	0	.500-01	.0	.0			
SENS		.200-02	.900+02	.270+03	.100+01					
GO										
12										
	5		0.70							
	4		0.83							
	4		0.87							
	4		0.92							
	4		0.98							
	4		0.97							
	4		0.96							
	4		0.95							
	4		0.94							
	4		0.93							
	4		0.93							
END										
4	2	1	0	1	0	0	0	0.000	0.000	
4	0.002			0.002			0.000	0.400	0.000	1.000
4	2	1	0	1	0	0	0	1	0.000	0.000
0										
END										

```

*****
** ELECTRO-OPTICAL SYSTEMS **
** ATMOSPHERIC EFFECTS **
** LIBRARY **
*****

```

INDIVIDUAL MODULES SELECTED

```

VOLUME
SPECTRUM
PROGRAM
LIBRARY
SCREEN
OVERCAST
GRNADE
CLIMATE

```

	BEGINNING	ENDING
WAVENUMBER(CH***)	943.396	943.396
WAVELENGTH(MICRONS)	10.600	10.600
FREQUENCY(GHZ)	28301.886	28301.886

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

1	=	FOG,	HAZE	AND	MIST	WITH	VIS	LT	1	KM.		
2	=	FOG,	HAZE	AND	MIST	WITH	3	LE	VIS	LT	3	KM.
3	=	FOG,	HAZE	AND	MIST	WITH	VIS	GE	7	KM.		
4	=	FOG,	HAZE	AND	MIST	WITH	VIS	GE	7	KM.		
5	=	DUST	WITH	VIS	LT	1	KM.					
6	=	DUST	WITH	VIS	LT	3	KM.					
7	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	LT	1	KM.		
8	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	3	LE	VIS	LT	3	KM.
9	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
10	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
11	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
12	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
13	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
14	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
15	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
16	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
17	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
18	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
19	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
20	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
21	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		
22	=	DRIZZLE,	RAIN	AND	ISTMS	WITH	VIS	GE	7	KM.		

EDSABEL CLIMATOLOGY FOR EUROPEAN LOWLANDS														DURING SPRING AT 03-09 (ALIST)													
CLASS NO.	FREQCY CLASS	MEAN TEMP (C)	MEAN DP (C)	MEAN WIND (CM/QU. H)	MEAN RAH	MEAN RH (CM/QU. H)	MEAN VIS (KM)	PRES (MB)	MEAN WIND VELOC (MPS)	MEAN SIDE WIND VELOC (KMS)	MEAN CLOUDY CLDHT (KMS)	MEAN CLOUDY CLDVR	FREQCY CLASS	FREQCY	FREQCY CLASS	FREQCY	FREQCY CLASS	FREQCY	FREQCY CLASS	FREQCY	FREQCY CLASS	FREQCY	FREQCY CLASS	FREQCY			
1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10		
2	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20		
3	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30		
4	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40		
5	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50		
6	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60		
7	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70		
8	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	
9	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	
10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	

9.30 KM
VISIBILITY

TUPB LASER MODULE

CALCULATION OF POWER SPECTRUM AND TURBULENCE INDUCED POINTING JITTER OF A LASER TARGET DESIGNATOR AND SEEKER

LASER WAVELENGTH (MICROMETERS)	10.6000
DESIG. APERTURE DIAMETER (METERS)	.101600
BEAMSPREAD ANGLE (RADIAN)	.000130
SEEKER APERTURE DIAMETER (METERS)	.400000
RANGE FROM TARGET TO SEEKER (METERS)	500.00
BEAM SLUE RATE (RAD/SEC)	.000000
DESIGNATION RANGE (METERS)	400.00
DURATION OF TEST (SECONDS)	5.0000
TOTAL DESIGNATOR PATH SEGMENTS	5
TOTAL SEEKER PATH SEGMENTS	5
TOTAL FREQUENCIES FOR WHICH POWER SPECTRUM IS TO BE CALCULATED	512

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN DESIGNATOR PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-013	.93
3	.130000-013	.93
4	.270000-013	.93
5	.500000-013	.93

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN SEEKER PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-013	.93
3	.130000-013	.93
4	.270000-013	.93
5	.500000-013	.93

DESIGNATOR TO TARGET

VIRTUAL POINT SOURCE TO APERTURE DISTANCE 781.53845 (METERS)
 DISTANCE FROM VIRTUAL POINT SOURCE TO TARGET 1181.53845 (METERS)
 INTEGRATED COHERENCE LENGTH .335957 (METERS)
 DIAMETER/INTEGRATED COHERENCE LENGTH .302419
 TRANSMITTER-INDUCED BEAM SPREAD .13000-003 (RADIAN) .052000 (METERS)
 DIFFRACTION-LIMITED BEAM SPREAD .11769-003 (RADIAN) .047074 (METERS)
 DIFFRACTION AND TURBULENCE BEAM SPREAD .11907-003 (RADIAN) .047629 (METERS)
 TOTAL EFFECTIVE BEAM SIZE .17629-003 (RADIAN) .172116 (METERS)

SEGMENT NO. COHERENCE LENGTH REFERENCE FREQUENCY (HERTZ)

SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY (HERTZ)
1	5.741660	56.285500
2	2.154930	19.267600
3	2.588230	11.826440
4	13.658743	8.325389

TARGET TO SEEKER

INTEGRATED COHERENCE LENGTH .934983 (METERS)
 DIAMETER/INTEGRATED COHERENCE LENGTH .184085
 TRANSMITTER-INDUCED BEAM SPREAD .065000 (METERS)
 DIFFRACTION-LIMITED BEAM SPREAD .13000-003 (RADIAN)S
 DIFFRACTION AND TURBULENCE BEAM SPREAD .69469-004 (RADIAN)S
 .69828-004 (RADIAN)S
 TOTAL EFFECTIVE BEAM SIZE .14757-003 (RADIAN)S
 .245900 (METERS)

SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY (HERTZ)
1	2.689629	7.400705
2	10.458227	2.466902
3	5.360999	1.489141
4	2.469819	1.05244
5	1.327262	.822301

THE VARIANCE OF THE POWER SPECTRUM IS .8090-011
 OUTPUT FOR DESIGNATOR TO TARGET PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .92021-007, MEAN OF IMAG PART = .00000
 VAR. OF REAL PART = .83284-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = -.52602-011, MEAN OF IMAG PART = .19630-016
 VAR. OF REAL PART = .79425-017, VAR. OF IMAG PART = .13359-031

OUTPUT FOR TARGET TO SEEKER PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .17265-006, MEAN OF IMAG PART = .00000
 VAR. OF REAL PART = .91505-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = .24531-011, MEAN OF IMAG PART = .26733-016
 VAR. OF REAL PART = .87592-017, VAR. OF IMAG PART = .12492-031

XSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE

OPTIONS CHOSEN
MARITIME ARTIC
HORIZONTAL PATH

EXTINCTION FROM 8.0 TO 12.0 MICRONS	DISTANCE	TRANSMISSION
KM**-1 .178	KM .400	.931+000

OPTIONS CHOSEN
MARITIME POLAR
SLANT PATH FOR 10.600 MICRONS

WARNING FROM SLANT
THE VERTICAL DISTANCE EXCEEDS THE 500 METER UPPER LIMIT, OR
IS NOT AN INTEGER MULTIPLE OF 20 METERS
SLANT DISTANCE CHANGED FROM .2397 TO .2238 KM

SLANT EXTINCTION AT 10.60 MICRONS	DISTANCE	TRANSMISSION	ANGLE
KM**-1 1.295	KM .224	.748+000	56.30

OPTIONS CHOSEN
CONTINENTAL POLAR
HORIZONTAL PATH

EXTINCTION FROM 8.0 TO 12.0 MICRONS	DISTANCE	TRANSMISSION
KM**-1 .008	KM .400	.997+000

SMOKE MODEL MODULE

 * SMOKE *

EXECUTION 1

SMOKE MUNITIONS
 WHITE PHOSPHORUS (WP)
 NO. ROUNDS 1
 FILL WEIGHT 15.600 LB
 BURN TIME 1.0 SEC
 EFFICIENCY 100.0 PERCENT
 YIELD FACTOR 5.84

METEOROLOGICAL CONDITIONS
 WINDSPEED 3.6 M/S
 WIND DIRECTION (USUAL)
 MET CONVENTION AZIMUTH) 225.0 DEGREES
 RELATIVE HUMIDITY 87.1 PERCENT
 PASQUILL CATEGORY 0
 AIR TEMPERATURE 5.3 DEGREE C
 TEMP. GRADIENT .00 C DEG./M

EXTINCTION COEFFICIENTS
 0.4-0.7 MICROMETERS 4.304 M**2/GM
 0.7-1.2 MICROMETERS 2.166 M**2/GM
 1.06 MICROMETERS 1.541 M**2/GM
 3.0-5.0 MICROMETERS .358 M**2/GM
 8.0-12. MICROMETERS .364 M**2/GM
 10.6 MICROMETERS .001 M**2/GM
 94.0 GHZ .001 M**2/GM

BURN RATE PROFILE = 1.0000

FIELD COORDINATES
 MUNITION COORDINATES = (X) .00 (Y) -50.00 (Z) .00 METERS
 OBSERVER COORDINATES = 200.00 .00 2.00 METERS
 TARGET COORDINATES = -200.00 .00 2.00 METERS
 ANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = 90.00 DEG.
 ROTATED COORD. (WIND X-AXIS, MUNITION ORIGIN)
 (XW) (YW) (ZW)
 .00 .00 .00 METERS
 176.78 -106.07 2.00 METERS
 -106.07 176.78 2.00 METERS

TIME (SEC)	LENGTH (METERS)	WIDTH (METERS)	HEIGHT (METERS)	PATHLENGTH (METERS)	CL (GM/M**2)	TRANSMISSION SPECTRAL BANDS (MICROMETERS)						
						0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94.0GHZ
5.	19.	13.	22.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.	36.	16.	34.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15.	54.	19.	44.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20.	72.	22.	53.	15.	4.00	.000	.000	.002	.245	.258	.232	.996
25.	90.	25.	61.	30.	4.89	.000	.000	.001	.181	.191	.163	.995
30.	108.	28.	69.	45.	5.78	.000	.000	.001	.115	.125	.107	.994
35.	126.	31.	76.	60.	6.67	.000	.000	.001	.048	.058	.050	.993
40.	144.	34.	83.	75.	7.56	.000	.000	.001	.013	.013	.010	.992
45.	162.	37.	90.	90.	8.45	.000	.000	.001	.004	.004	.003	.991
50.	180.	40.	95.	105.	9.34	.000	.000	.001	.001	.001	.001	.990
55.	198.	43.	101.	120.	10.23	.000	.000	.001	.000	.000	.000	.989
60.	216.	46.	106.	135.	11.12	.000	.000	.001	.000	.000	.000	.988
65.	234.	49.	111.	150.	12.01	.000	.000	.001	.000	.000	.000	.987
70.	252.	52.	115.	165.	12.90	.000	.000	.001	.000	.000	.000	.986
75.	270.	55.	120.	180.	13.79	.000	.000	.001	.000	.000	.000	.985
80.	288.	58.	124.	195.	14.68	.000	.000	.001	.000	.000	.000	.984
85.	306.	61.	129.	210.	15.57	.000	.000	.001	.000	.000	.000	.983
90.	324.	64.	133.	225.	16.46	.000	.000	.001	.000	.000	.000	.982
95.	342.	67.	137.	240.	17.35	.000	.000	.001	.000	.000	.000	.981
100.	360.	70.	141.	255.	18.24	.000	.000	.001	.000	.000	.000	.980
105.	378.	73.	144.	270.	19.13	.000	.000	.001	.000	.000	.000	.979
110.	396.	76.	147.	285.	20.02	.000	.000	.001	.000	.000	.000	.978
115.	414.	79.	150.	300.	20.91	.000	.000	.001	.000	.000	.000	.977
120.	432.	82.	153.	315.	21.80	.000	.000	.001	.000	.000	.000	.976
125.	450.	85.	156.	330.	22.69	.000	.000	.001	.000	.000	.000	.975
130.	468.	88.	159.	345.	23.58	.000	.000	.001	.000	.000	.000	.974
135.	486.	91.	162.	360.	24.47	.000	.000	.001	.000	.000	.000	.973
140.	504.	94.	165.	375.	25.36	.000	.000	.001	.000	.000	.000	.972
145.	522.	97.	168.	390.	26.25	.000	.000	.001	.000	.000	.000	.971
150.	540.	100.	171.	405.	27.14	.000	.000	.001	.000	.000	.000	.970
155.	558.	103.	174.	420.	28.03	.000	.000	.001	.000	.000	.000	.969
160.	576.	106.	177.	435.	28.92	.000	.000	.001	.000	.000	.000	.968
165.	594.	109.	180.	450.	29.81	.000	.000	.001	.000	.000	.000	.967
170.	612.	112.	183.	465.	30.70	.000	.000	.001	.000	.000	.000	.966
175.	630.	115.	186.	480.	31.59	.000	.000	.001	.000	.000	.000	.965
180.	648.	118.	189.	495.	32.48	.000	.000	.001	.000	.000	.000	.964
185.	666.	121.	192.	510.	33.37	.000	.000	.001	.000	.000	.000	.963
190.	684.	124.	195.	525.	34.26	.000	.000	.001	.000	.000	.000	.962
195.	702.	127.	198.	540.	35.15	.000	.000	.001	.000	.000	.000	.961
200.	720.	130.	201.	555.	36.04	.000	.000	.001	.000	.000	.000	.960
205.	738.	133.	204.	570.	36.93	.000	.000	.001	.000	.000	.000	.959
210.	756.	136.	207.	585.	37.82	.000	.000	.001	.000	.000	.000	.958
215.	774.	139.	210.	600.	38.71	.000	.000	.001	.000	.000	.000	.957
220.	792.	142.	213.	615.	39.60	.000	.000	.001	.000	.000	.000	.956
225.	810.	145.	216.	630.	40.49	.000	.000	.001	.000	.000	.000	.955
230.	828.	148.	219.	645.	41.38	.000	.000	.001	.000	.000	.000	.954
235.	846.	151.	222.	660.	42.27	.000	.000	.001	.000	.000	.000	.953
240.	864.	154.	225.	675.	43.16	.000	.000	.001	.000	.000	.000	.952
245.	882.	157.	228.	690.	44.05	.000	.000	.001	.000	.000	.000	.951
250.	900.	160.	231.	705.	44.94	.000	.000	.001	.000	.000	.000	.950
255.	918.	163.	234.	720.	45.83	.000	.000	.001	.000	.000	.000	.949
260.	936.	166.	237.	735.	46.72	.000	.000	.001	.000	.000	.000	.948
265.	954.	169.	240.	750.	47.61	.000	.000	.001	.000	.000	.000	.947
270.	972.	172.	243.	765.	48.50	.000	.000	.001	.000	.000	.000	.946
275.	990.	175.	246.	780.	49.39	.000	.000	.001	.000	.000	.000	.945
280.	1008.	178.	249.	795.	50.28	.000	.000	.001	.000	.000	.000	.944
285.	1026.	181.	252.	810.	51.17	.000	.000	.001	.000	.000	.000	.943
290.	1044.	184.	255.	825.	52.06	.000	.000	.001	.000	.000	.000	.942
295.	1062.	187.	258.	840.	52.95	.000	.000	.001	.000	.000	.000	.941
300.	1080.	190.	261.	855.	53.84	.000	.000	.001	.000	.000	.000	.940
305.	1098.	193.	264.	870.	54.73	.000	.000	.001	.000	.000	.000	.939
310.	1116.	196.	267.	885.	55.62	.000	.000	.001	.000	.000	.000	.938
315.	1134.	199.	270.	900.	56.51	.000	.000	.001	.000	.000	.000	.937
320.	1152.	202.	273.	915.	57.40	.000	.000	.001	.000	.000	.000	.936
325.	1170.	205.	276.	930.	58.29	.000	.000	.001	.000	.000	.000	.935
330.	1188.	208.	279.	945.	59.18	.000	.000	.001	.000	.000	.000	.934
335.	1206.	211.	282.	960.	60.07	.000	.000	.001	.000	.000	.000	.933
340.	1224.	214.	285.	975.	60.96	.000	.000	.001	.000	.000	.000	.932
345.	1242.	217.	288.	990.	61.85	.000	.000	.001	.000	.000	.000	.931
350.	1260.	220.	291.	1005.	62.74	.000	.000	.001	.000	.000	.000	.930
355.	1278.	223.	294.	1020.	63.63	.000	.000	.001	.000	.000	.000	.929
360.	1296.	226.	297.	1035.	64.52	.000	.000	.001	.000	.000	.000	.928
365.	1314.	229.	300.	1050.	65.41	.000	.000	.001	.000	.000	.000	.927
370.	1332.	232.	303.	1065.	66.30	.000	.000	.001	.000	.000	.000	.926
375.	1350.	235.	306.	1080.	67.19	.000	.000	.001	.000	.000	.000	.925
380.	1368.	238.	309.	1095.	68.08	.000	.000	.001	.000	.000	.000	.924
385.	1386.	241.	312.	1110.	68.97	.000	.000	.001	.000	.000	.000	.923
390.	1404.	244.	315.	1125.	69.86	.000	.000	.001	.000	.000	.000	.922
395.	1422.	247.	318.	1140.	70.75	.000	.000	.001	.000	.000	.000	.921
400.	1440.	250.	321.	1155.	71.64	.000	.000	.001	.000	.000	.000	.920
405.	1458.	253.	324.	1170.	72.53	.000	.000	.001	.000	.000	.000	.919
410.	1476.	256.	327.	1185.	73.42	.000	.000	.001	.000	.000	.000	.918
415.	1494.	259.	330.	1200.	74.31	.000	.000	.001	.000	.000	.000	.917
420.	1512.	262.	333.	1215.	75.20	.000	.000	.001	.000	.000	.000	.916
425.	1530.	265.	336.	1230.	76.09	.000	.000	.001	.000	.000	.000	.915
430.	1548.	268.	339.	1245.	76.98	.000	.000	.001	.000	.000	.000	.914
435.	1566.	271.	342.	1260.	77.87	.000	.000	.001	.000	.000	.000	.913
440.	1584.	274.	345.	1275.	78.76	.000	.000	.001	.000	.000	.000	.912
445.	1602.	277.	348.	1290.	79.65	.000	.000	.001	.000	.000	.000	.911
450.	1620.	280.	351.	1305.	80.54	.000	.000	.001	.000	.000	.000	.910
455.	1638.	283.	354.	1320.	81.43	.000	.000	.001	.000	.000	.000	.909
460.	1656.	286.	357.	1335.	82.32	.000	.000	.001	.000	.000	.000	.908
465.	1674.	289.	360.	1350.	83.21	.000	.000	.001	.000	.000	.000	.907
470.	1692.	292.	363.	1365.	84.10	.000	.000	.001	.000	.000	.000	.906
475.	1710.	295.	366.	1380.	84.99	.000	.000	.001	.000	.000	.000	.905
480.	1728.	298.	369.	1395.	85.88	.000	.000	.001	.000	.000	.000	.904
485.	1746.	301.	372.	1410.	86.77	.000	.000	.001	.000	.000	.000	.903
490.	1764.	304.	375.	1425.	87.66	.000	.000	.001	.000	.000	.000	.902
495.	1782.	307.	378.	1440.	88.55	.000	.000	.001	.000	.000	.000	.901
500.	1800.	310.	381.	1455.	89.44	.000	.000	.001	.000	.000	.000	.900

 * SMOKE *

EXECUTION 2

SMOKE MUNITIONS
 HL SMOKE
 NO ROUNDS 1
 FILL WEIGHT 17.190 LB
 BURN TIME 100.0 SEC
 EFFICIENCY 40.0 PERCENT
 YIELD FACTOR 2.39

METEOROLOGICAL CONDITIONS
 WIND SPEED 3.6 M/S
 WIND DIRECTION (USUAL)
 MET CONVECTION AZIMUTH) 225.0 DEGREES
 RELATIVE HUMIDITY 87.1 PERCENT
 PASQUILL CATEGORY 5
 AIR TEMPERATURE 5.3 DEGREE C
 TEMP. GRADIENT .00 C/DEC./M

EXTINCTION COEFFICIENTS
 0.4-0.7 MICROMETER 4.573 M+M+M+M
 0.7-1 MICROMETER 1.183 M+M+M+M
 1-5 MICROMETER 1.040 M+M+M+M
 5-10 MICROMETER .190 M+M+M+M
 10-12 MICROMETER .051 M+M+M+M
 10.6 MICROMETER .051 M+M+M+M
 34.0 GHZ .001 M+M+M+M

BURN RATE PROFILE = .5370+ .4760 (T/BURN) + 4.7790 (T/BURN)**2 + -5.4720 (T/BURN)**3

FIELD COORDINATES (X) (Y) (Z) ROTATED COORD. (WIND X-AXIS, MUNITION ORIGIN)
 (XW) (YW) (ZW)
 OBSERVER COORDINATES = 200.00 -50.00 2.00 METER METER METER
 TARGET COORDINATES = -200.00 .00 2.00 METER METER METER
 ANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = 90.00 DEG.
 EVENT TIME = .0 SEC

TIME (SEC)	LENGTH (METERS)	WIDTH (METERS)	HEIGHT (METERS)	PATHLENGTH (METERS)	CL (GM/M**2)	TRANSMISSION SPECTRAL BANDS (MICROMETERS)						
						0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	34.0GHZ
0.0	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.1	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.2	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.3	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.4	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.5	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.6	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.7	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.8	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.9	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.1	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.2	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.3	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.4	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.5	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.6	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.7	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.8	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.9	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.0	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.1	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.2	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.3	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.4	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.5	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.6	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.7	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.8	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.9	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.0	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.1	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.2	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.3	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.4	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.5	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.6	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.7	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.8	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.9	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.0	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.1	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.2	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.3	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.4	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.5	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.6	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.7	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.8	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.9	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5.0	0.0	0.0	0.0	0.0	0.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000

TIME (SEC)	CL (G/M**2)	COMBINED EFFECT OF 2 EXECUTIONS IN SMOKE:						
		0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94.0HT
5.	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15.	.01	.949	.975	.977	.998	.999	.999	1.000
20.	.17	.000	.000	.000	.374	.764	.768	.995
25.	.00	.000	.000	.000	.270	.699	.704	.995
30.	.00	.000	.000	.000	.333	.740	.745	.995
35.	.01	.000	.000	.000	.371	.763	.768	.995
40.	.00	.000	.000	.000	.385	.772	.776	.995
45.	.00	.000	.000	.000	.392	.774	.778	.995
50.	.00	.000	.000	.000	.385	.770	.774	.995
55.	.00	.000	.000	.000	.365	.764	.768	.995
60.	.00	.000	.000	.000	.360	.756	.760	.995
65.	.00	.000	.000	.000	.346	.748	.752	.994
70.	.00	.000	.000	.000	.333	.741	.745	.994
75.	.00	.000	.000	.000	.322	.736	.741	.994
80.	.00	.000	.000	.000	.312	.734	.738	.994
85.	.00	.000	.000	.000	.303	.734	.738	.994
90.	.00	.000	.000	.000	.303	.734	.738	.994
95.	.00	.000	.000	.000	.303	.734	.738	.994
100.	.00	.000	.000	.000	.303	.734	.738	.994
105.	.00	.000	.000	.000	.303	.734	.738	.994
110.	.00	.000	.000	.000	.303	.734	.738	.994
115.	.00	.000	.000	.000	.303	.734	.738	.994
120.	.00	.000	.000	.000	.303	.734	.738	.994
125.	.00	.000	.000	.000	.303	.734	.738	.994
130.	.00	.000	.000	.000	.303	.734	.738	.994
135.	.00	.000	.000	.000	.303	.734	.738	.994
140.	.00	.000	.000	.000	.303	.734	.738	.994
145.	.00	.000	.000	.000	.303	.734	.738	.994
150.	.00	.000	.000	.000	.303	.734	.738	.994
155.	.00	.000	.000	.000	.303	.734	.738	.994
160.	.00	.000	.000	.000	.303	.734	.738	.994
165.	.00	.000	.000	.000	.303	.734	.738	.994
170.	.00	.000	.000	.000	.303	.734	.738	.994
175.	.00	.000	.000	.000	.303	.734	.738	.994
180.	.00	.000	.000	.000	.303	.734	.738	.994
185.	.00	.000	.000	.000	.303	.734	.738	.994
190.	.00	.000	.000	.000	.303	.734	.738	.994
195.	.00	.000	.000	.000	.303	.734	.738	.994
200.	.00	.000	.000	.000	.303	.734	.738	.994
205.	.00	.000	.000	.000	.303	.734	.738	.994
210.	.00	.000	.000	.000	.303	.734	.738	.994
215.	.00	.000	.000	.000	.303	.734	.738	.994
220.	.00	.000	.000	.000	.303	.734	.738	.994
225.	.00	.000	.000	.000	.303	.734	.738	.994
230.	.00	.000	.000	.000	.303	.734	.738	.994
235.	.00	.000	.000	.000	.303	.734	.738	.994
240.	.00	.000	.000	.000	.303	.734	.738	.994
245.	.00	.000	.000	.000	.303	.734	.738	.994
250.	.02	.896	.949	.952	.995	.999	.999	1.000

***TRANSMISSION RETURNED TO MAIN FOR WAVELENGTH OF 10.600 MICROMETERS IS .999 AT TIME 250.

DIRT TRANSMISSION MODULE

DIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION CALCULATION

*** NOTE -- ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED ***

PASQUILL CATEGORY D
 HT 2.00 TEMP 278.30 HT 2.00 WIND 3.60
 WIND DIRECTION 45.00

LATITUDE 53.00
 THE INVERSION LAYER HEIGHT IS GROWING

SOIL-2
 SILT CONTENT .15
 DEPTH OF SOO .00

30 DEGREE TILTED TIP AT 0.3 METER DEPTH
 WEIGHT OF CHARGE 6.80 KG.
 DETONATION DEPTH .00

SIMULTANEOUS BURST, UNIFORMLY DISTRIBUTED CHARGES IN A PARALLELOGRAM
 TOTAL NUMBER OF CHARGES IS 1 WITH REFERENCE CHARGE AT (.00, .00) .00
 1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (.00, .00)
 1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (.00, .00)

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 1.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .767-002

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 19.74 METERS
 THE CENTROID COORDINATES ARE 1.34 10.55
 THE WIDTH AT THE CENTROID IS 20.62 METERS
 THE WIDTH AT 2.00 METERS IS 18.56 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 -5.670 2.000
 -2.973 10.550
 1.337 19.745
 1.337 19.745
 11.646 10.550
 10.693 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 3.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .626+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 21.89 METERS
 THE CENTROID COORDINATES ARE 8.02 10.55
 THE WIDTH AT THE CENTROID IS 31.26 METERS
 THE WIDTH AT 2.00 METERS IS 21.88 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 -2.947 2.000
 -7.608 10.550

9.020 21.800
 23.649 21.800
 18.928 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 5.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .940+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 23.50 .00
 THE HEIGHT OF THE CLOUD IS 23.50 METERS
 THE CENTROID COORDINATES ARE 14.70 10.50
 THE WIDTH AT THE CENTROID IS 38.28 METERS
 THE WIDTH AT 2.00 METERS IS 23.37 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 .224 2.000
 -4.436 10.550
 14.704 23.558
 14.704 23.558
 33.844 10.550
 29.151 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 7.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .987+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 24.95 .00
 THE HEIGHT OF THE CLOUD IS 24.95 METERS
 THE CENTROID COORDINATES ARE 21.39 10.55
 THE WIDTH AT THE CENTROID IS 43.76 METERS
 THE WIDTH AT 2.00 METERS IS 35.94 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 3.436 2.000
 -4.91 10.550
 21.398 24.948
 21.398 24.948
 43.267 10.550
 39.374 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 9.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .997+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 26.15 .00
 THE HEIGHT OF THE CLOUD IS 26.15 METERS
 THE CENTROID COORDINATES ARE 28.07 10.55
 THE WIDTH AT THE CENTROID IS 48.33 METERS
 THE WIDTH AT 2.00 METERS IS 41.56 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 7.097 2.000

60.904	10.550
57.944	26.146
54.755	10.550
41.439	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 11.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .999+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 27.21 .00
 THE HEIGHT OF THE CLOUD IS 27.21 METERS
 THE CENTROID COORDINATES ARE 34.76 10.55
 THE WIDTH AT THE CENTROID IS 52.30 METERS
 THE WIDTH AT 2.00 METERS IS 46.56 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 11.38 2.000
 38.607 10.550
 34.755 27.213
 60.904 10.550
 57.944 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 13.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 28.17 .00
 THE HEIGHT OF THE CLOUD IS 28.17 METERS
 THE CENTROID COORDINATES ARE 41.44 10.55
 THE WIDTH AT THE CENTROID IS 55.82 METERS
 THE WIDTH AT 2.00 METERS IS 50.94 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 15.867 2.000
 13.529 10.550
 41.439 28.173
 41.439 28.173
 69.349 10.550
 66.605 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 15.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 29.05 .00
 THE HEIGHT OF THE CLOUD IS 29.05 METERS
 THE CENTROID COORDINATES ARE 48.12 10.55
 THE WIDTH AT THE CENTROID IS 59.00 METERS
 THE WIDTH AT 2.00 METERS IS 55.00 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

20.265	2.000
18.625	10.550
48.102	29.050
48.102	3.050
77.624	10.550
78.265	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 17.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	200.00	.00
RECEIVER COORDINATES	200.00	.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.100+001	2.00

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	29.86	.00
THE HEIGHT OF THE CLOUD IS		54.81	10.55
THE CENTROID COORDINATES ARE		61.91	METERS
THE WIDTH AT THE CENTROID IS		59.38	METERS
THE WIDTH AT 2.00 METERS IS		6 CONTOUR POINTS HAVE BEEN DETERMINED	
	24.551	2.000	
	23.859	10.550	
	54.807	29.859	
	85.763	10.550	
	83.926	2.000	

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 19.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	200.00	.00
RECEIVER COORDINATES	200.00	.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.100+001	2.00

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	30.61	.00
THE HEIGHT OF THE CLOUD IS		61.49	10.55
THE CENTROID COORDINATES ARE		64.60	METERS
THE WIDTH AT THE CENTROID IS		63.12	METERS
THE WIDTH AT 2.00 METERS IS		6 CONTOUR POINTS HAVE BEEN DETERMINED	
	29.148	2.000	
	29.190	10.550	
	61.490	30.611	
	61.490	30.611	
	93.791	10.550	
	92.273	2.000	

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 21.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	200.00	.00
RECEIVER COORDINATES	200.00	.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.100+001	2.00

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	31.32	.00
THE HEIGHT OF THE CLOUD IS		68.17	10.55
THE CENTROID COORDINATES ARE		67.10	METERS
THE WIDTH AT THE CENTROID IS		66.87	METERS
THE WIDTH AT 2.00 METERS IS			

6 CONTOUR POINTS HAVE BEEN DETERMINED

33.434	2.000
34.623	10.550
68.174	31.315
68.174	31.315
101.725	10.550
100.309	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 23.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 31.98 METERS
THE CENTROID COORDINATES ARE 74.86 10.55
THE WIDTH AT THE CENTROID IS 63.44 METERS
THE WIDTH AT 2.00 METERS IS 70.31 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
38.032 2.000
40.135 10.550
74.858 31.977
74.858 31.977
109.580 10.550
108.344 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 25.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 32.60 METERS
THE CENTROID COORDINATES ARE 81.54 10.55
THE WIDTH AT THE CENTROID IS 71.65 METERS
THE WIDTH AT 2.00 METERS IS 73.75 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
42.630 2.000
45.717 10.550
81.541 32.603
81.541 32.603
117.366 10.550
116.380 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 27.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 33.20 METERS
THE CENTROID COORDINATES ARE 68.53 10.55
THE WIDTH AT THE CENTROID IS 73.73 METERS

THE WIDTH AT 2.00 METERS IS 77.19 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

47.227	2.000
51.055	10.550
88.000	33.196
124.415	33.196
	10.550
	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 29.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 33.76 METERS
 THE CENTROID COORDINATES ARE 94.91 10.55
 THE WIDTH AT THE CENTROID IS 75.71 METERS
 THE WIDTH AT 2.00 METERS IS 80.94 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

51.513	2.000
57.055	10.550
94.909	33.761
94.909	33.761
132.763	10.550
132.450	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 31.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 34.30 METERS
 THE CENTROID COORDINATES ARE 101.59 10.55
 THE WIDTH AT THE CENTROID IS 77.59 METERS
 THE WIDTH AT 2.00 METERS IS 84.06 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

56.111	2.000
62.799	10.550
101.593	34.299
101.593	34.299
140.387	10.550
140.173	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 33.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 34.81 METERS
 THE CENTROID COORDINATES ARE 108.28 10.55

THE WIDTH AT THE CENTROID IS 79.38 METERS
 THE WIDTH AT 2.00 METERS IS 37.19 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 60 700 2.000
 108 350 10.550
 108 370 34.814
 147 368 34.814
 147 368 10.550
 147 896 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 35.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 35.31 METERS
 THE CENTROID COORDINATES ARE 114.96 10.55
 THE WIDTH AT THE CENTROID IS 81.10 METERS
 THE WIDTH AT 2.00 METERS IS 90.31 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 65.306 2.000
 74.410 10.550
 114.960 35.308
 114.960 35.308
 155.510 10.550
 155.619 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 37.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 35.78 METERS
 THE CENTROID COORDINATES ARE 121.64 10.55
 THE WIDTH AT THE CENTROID IS 92.75 METERS
 THE WIDTH AT 2.00 METERS IS 93.75 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 69.592 2.000
 80.271 10.550
 121.644 35.782
 121.644 35.782
 163.017 10.550
 163.342 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 39.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 36.24 METERS

THE CENTROID COORDINATES ARE 128.35 10.55
 THE WIDTH AT THE CENTROID IS 84.33 METERS
 THE WIDTH AT 2.00 METERS IS 96.88 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 74.190 2.000
 86.164 10.550
 128.328 36.638
 128.328 36.638
 170.492 10.550
 171.065 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 41.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 36.68 METERS
 THE CENTROID COORDINATES ARE 135.01 10.55
 THE WIDTH AT THE CENTROID IS 85.85 METERS
 THE WIDTH AT 2.00 METERS IS 100.00 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 78.787 2.000
 92.086 10.550
 135.011 36.677
 135.011 36.677
 177.936 10.550
 178.787 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 43.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 37.10 METERS
 THE CENTROID COORDINATES ARE 141.70 10.55
 THE WIDTH AT THE CENTROID IS 87.32 METERS
 THE WIDTH AT 2.00 METERS IS 102.81 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 83.385 2.000
 98.037 10.550
 141.695 37.102
 141.695 37.102
 185.353 10.550
 186.198 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 45.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00

THE HEIGHT OF THE CLOUD IS 37.51 METERS
 THE CENTROID COORDINATES ARE 148.38 10.55
 THE WIDTH AT THE CENTROID IS 88.73 METERS
 THE WIDTH AT 2.00 METERS IS 105.94 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 87.983 2.000
 104.013 10.550
 148.379 37.512
 148.379 37.512
 132.745 10.550
 133.921 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 47.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 37.91 METERS
 THE CENTROID COORDINATES ARE 155.06 10.55
 THE WIDTH AT THE CENTROID IS 90.10 METERS
 THE WIDTH AT 2.00 METERS IS 108.75 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 92.581 2.000
 110.012 10.550
 155.063 37.909
 155.063 37.909
 200.113 10.550
 201.331 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 49.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 38.29 METERS
 THE CENTROID COORDINATES ARE 161.75 10.55
 THE WIDTH AT THE CENTROID IS 91.43 METERS
 THE WIDTH AT 2.00 METERS IS 111.87 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 97.175 2.000
 116.034 10.550
 161.746 38.293
 161.746 38.293
 207.459 10.550
 209.054 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 51.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 38.67 METERS
 THE CENTROID COORDINATES ARE 168.43 10.55
 THE WIDTH AT THE CENTROID IS 92.71 METERS
 THE WIDTH AT 2.00 METERS IS 114.69 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 101.777 2.000
 122.076 10.550
 168.430 38.665
 168.430 38.665
 214.784 10.550
 216.464 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 53.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 39.03 METERS
 THE CENTROID COORDINATES ARE 175.11 10.55
 THE WIDTH AT THE CENTROID IS 93.95 METERS
 THE WIDTH AT 2.00 METERS IS 117.50 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 106.375 2.000
 128.138 10.550
 175.114 39.027
 175.114 39.027
 222.090 10.550
 223.875 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 55.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 39.38 METERS
 THE CENTROID COORDINATES ARE 181.80 10.55
 THE WIDTH AT THE CENTROID IS 95.16 METERS
 THE WIDTH AT 2.00 METERS IS 120.00 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 111.285 2.000
 134.217 10.550
 181.797 39.378
 181.797 39.378
 229.377 10.550
 231.285 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 57.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 39.72 METERS
 THE CENTROID COORDINATES ARE 188.48 10.55
 THE WIDTH AT THE CENTROID IS 96.33 METERS
 THE WIDTH AT 2.00 METERS IS 122.81 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 115.883 2.000
 140.315 10.550
 188.481 39.720
 188.481 39.720
 230.648 10.550
 238.695 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 59.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 40.05 METERS
 THE CENTROID COORDINATES ARE 195.16 10.55
 THE WIDTH AT THE CENTROID IS 97.47 METERS
 THE WIDTH AT 2.00 METERS IS 125.62 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 120.481 2.000
 146.428 10.550
 195.165 40.052
 195.165 40.052
 243.902 10.550
 246.106 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 61.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 40.38 METERS
 THE CENTROID COORDINATES ARE 201.85 10.55
 THE WIDTH AT THE CENTROID IS 98.58 METERS
 THE WIDTH AT 2.00 METERS IS 128.12 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 125.391 2.000
 152.557 10.550
 201.849 40.376
 201.849 40.376
 251.140 10.550
 253.516 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 63.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 40.69 METERS
 THE CENTROID COORDINATES ARE 208.53 10.55
 THE WIDTH AT THE CENTROID IS 99.66 METERS
 THE WIDTH AT 2.00 METERS IS 130.62 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 130.301 2.000
 156.700 10.550
 208.532 40.691
 208.532 40.691
 258.364 10.550
 260.926 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 65.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 41.00 METERS
 THE CENTROID COORDINATES ARE 215.22 10.55
 THE WIDTH AT THE CENTROID IS 100.72 METERS
 THE WIDTH AT 2.00 METERS IS 133.12 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 135.212 2.000
 164.857 10.550
 215.216 40.998
 215.216 40.998
 265.575 10.550
 268.337 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 67.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 41.30 METERS
 THE CENTROID COORDINATES ARE 221.90 10.55
 THE WIDTH AT THE CENTROID IS 101.74 METERS
 THE WIDTH AT 2.00 METERS IS 135.00 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 140.435 2.000
 171.028 10.550
 221.900 41.298
 221.900 41.298
 272.772 10.550
 275.435 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 69.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 41.59 METERS
 THE CENTROID COORDINATES ARE 228.58 10.55
 THE WIDTH AT THE CENTROID IS 103.78 METERS
 THE WIDTH AT 2.00 METERS IS 137.50 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 145.345 2.000
 177.211 10.550
 228.584 41.591
 228.584 41.591
 279.956 10.550
 282.845 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 71.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 41.88 METERS
 THE CENTROID COORDINATES ARE 235.27 10.55
 THE WIDTH AT THE CENTROID IS 103.72 METERS
 THE WIDTH AT 2.00 METERS IS 139.69 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 150.568 2.000
 183.406 10.550
 235.267 41.877
 235.267 41.877
 287.129 10.550
 290.255 2.000

LASER TRANSMITTANCE MODULE

WAVELENGTH (MICRONS)	H2O PRESSURE (TORR)	TEMPERATURE (ABS)	ABSORPTION COEFFICIENT (KM-1)	LINE	PATHLENGTH (KM)	TRANSMISSION
10.591	5.821	278.46	.111+000	P<20>	.4000+000	.9566+000

*** WARNING INPUT WAVELENGTH 10.600 CHANGED TO 10.591 NEAREST STANDARD WAVELENGTH ***

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .719 KM
 TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .481 KM

TOTAL OPTICAL DEPTH = 25.11

TRANSMITTANCE ALONG LINE-OF-SIGHT = .12449-010

SEEKER COORDINATES (KM) TARGET COORDINATES (KM)
 XSEEKER YSEEKER ZSEEKER XTARGET YTARGET ZTARGET

 .200 .000 .600 -.200 .000 .002

CLOUD TYPE LINE-OF-SIGHT INTERSECTION COORDINATES (KM)
 /ID NUMBER XUPPER YUPPER ZUPPER XLOWER YLOWER ZLOWER

 ST/ 1 .200 .000 .600 -.068 .000 .200

CLOUD TYPE THICKNESS RADIUS OF CLOUD OPTICAL DEPTH TRANSMITTANCE
 /ID NUMBER (KM) (KM) ALONG L-O-S

 ST/ 1 .200 .400 .000 25.11 .12449-010

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .599 KM
 TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .361 KM

TOTAL OPTICAL DEPTH = 16.57

TRANSMITTANCE ALONG LINE-OF-SIGHT = .63594-007

SEEKER COORDINATES (KM) TARGET COORDINATES (KM)
 XSEEKER YSEEKER ZSEEKER XTARGET YTARGET ZTARGET

 .133 .000 .500 -.200 .000 .002

CLOUD TYPE LINE-OF-SIGHT INTERSECTION COORDINATES (KM)
 /ID NUMBER XUPPER YUPPER ZUPPER XLOWER YLOWER ZLOWER

 ST/ 1 .133 .000 .500 -.068 .000 .200

CLOUD TYPE HEIGHT OF BASE THICKNESS RADIUS OF CLOUD OPTICAL DEPTH TRANSMITTANCE
 /ID NUMBER (KM) (KM) (KM) ALONG L-O-S

 ST/ 1 .200 .400 .000 16.57 .63594-007

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .359 KM
 TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .120 KM

TOTAL OPTICAL DEPTH = 3.02

TRANSMITTANCE ALONG LINE-OF-SIGHT = .48846-001

SEEKER COORDINATES (KM) TARGET COORDINATES (KM)
 XSEEKER YSEEKER ZSEEKER XTARGET YTARGET ZTARGET

.000 .000 .300 -.200 .000 .002

CLOUD TYPE LINE-OF-SIGHT INTERSECTION COORDINATES (KM)
 /ID NUMBER XUPPER YUPPER ZUPPER XLOWER YLOWER ZLOWER

ST/ 1 .000 .000 .300 -.067 .000 .200

CLOUD TYPE HEIGHT OF BASE THICKNESS RADIUS OF CLOUD OPTICAL DEPTH TRANSMITTANCE
 /ID NUMBER (KM) (KM) (KM) ALONG L-O-S ALONG L-O-S

ST/ 1 .200 .400 .000 3.02 .48846-001

CWIC MUNITION EXPENDITURES / INVERSE STATIC TARGET DETECTION MODULE

 ** INVERSE STATIC TARGET DETECTION MODEL **

DEVICE NUMBER 8
 FOV TYPE - WIDE
 FOV (DEG) 10.620
 MAGNIFICATION 1.000 (ARBITRARY DEFAULT)
 TARGET_INTRINSIC CONTRAST OR TEMPERATURE DIFFERENCE 1.000
 MINIMUM TARGET DIMENSION (M) 2.300
 ACQUISITION LEVEL (50 PCNT) 1.000
 SEARCH ZONE (DEGREES**2) 81.000

FOR NO GREATER
 DETECTION
 PROB. TIME(SEC)
 .990 1.32
 .900 1.58
 .750 2.16
 .500 3.89

UNDER INPUT AND INPUT AT
 AMB ILLUM. SKY/GROUND RANGE INPUT
 (FT CDLS) RATIO (KM) (KM)
 2.000 2.000 .24
 10.000000 2.000 .24
 10.000000 2.000 .24
 10.000000 2.000 .24

REQUIRES (TO DEFEAT DEVICE) AT MOST
 COMPUTED CONTRAST OR COMPUTED
 RESOLVABLE TEMP. DIFF. TOTAL PATH
 CYCLES, RC AT DEVICE TRANSMITTANCE
 .773 .037 .037 .037

NOTE- INPUT DETECTION PROB-
 ILITY REQUIRES CONTRAST (OR
 TEMP. DIFF.) BELOW 99 PERCENT
 VALUES ASSUMED ARE 99 PERCENT
 OF THRESHOLD. ADDITIONAL OBS-
 CURANT WILL NOT DECREASE
 DETECTION PROBABILITY.

NOTE- INPUT DETECTION PROB-
 ILITY REQUIRES CONTRAST (OR
 TEMP. DIFF.) BELOW THRESHOLD.
 VALUES ASSUMED ARE 99 PERCENT
 OF THRESHOLD. ADDITIONAL OBS-
 CURANT WILL NOT DECREASE
 DETECTION PROBABILITY.

NOTE- INPUT DETECTION PROB-
 ILITY REQUIRES CONTRAST (OR
 TEMP. DIFF.) BELOW 99 PERCENT
 VALUES ASSUMED ARE 99 PERCENT
 OF THRESHOLD. ADDITIONAL OBS-
 CURANT WILL NOT DECREASE
 DETECTION PROBABILITY.

NOTE- INPUT DETECTION PROB-
 ILITY REQUIRES CONTRAST (OR
 TEMP. DIFF.) BELOW THRESHOLD.
 VALUES ASSUMED ARE 99 PERCENT
 OF THRESHOLD. ADDITIONAL OBS-
 CURANT WILL NOT DECREASE
 DETECTION PROBABILITY.

** CONTINUED ON NEXT PAGE **

 ** INVERSE STATIC TARGET DETECTION MODEL **

DEVICE NUMBER	8	TARGET INTRINSIC CONTRAST OR TEMPERATURE DIFFERENCE	1.000
FOV TYPE	WIDE	MINIMUM TARGET DIMENSION (M)	2.300
FOV (DEG)	10.620	ACQUISITION LEVEL (<50 PCNT)	1.000
MAGNIFICATION	1.000 (ARBITRARY DEFAULT)	SEARCH ZONE (DEGREES**2)	81.000

FOR NO GREATER INPUT DETECTION PROB. TIME (SEC)	.100	24.64
UNDER INPUT AMB. ILLUM. (<FT CDLS)	10.000000	
AND INPUT SKY/GROUND RATIO (<KM)	2.000	
AT INPUT RANGE (<KM)	.24	
COMPUTED RESOLVABLE CYCLES, RC	.059	
CONTRAST OF TEMP. DIFF. AT DEVICE	.037	
COMPUTED TOTAL PATH TRANSMITTANCE	.037	

COMMENTS

NOTE - INPUT DETECTION PROBABILITY REQUIRES COMPARISON OF TEMP. DIFF. WITH BELOW TABLED VALUES ASSUMED FOR PERCENT OF THE RESOLUTION RANGE OBSERVED. THIS IS A FUNCTION OF RANGE AND DETECTION PROBABILITY.

*** FINAL TOTAL TRANSMISSION FROM ITAM = .037

 * * CMIC MUNITION EXPENDITURES * *
 * *

MUNITION EXPENDITURES
 FOR HC AND WP SMOKE

SLANT RANGE OBS-TGT	- KM	=	.400
ELEVATION OF TARGET	- DEG	=	90.00
AZIMUTH OF TARGET	- DEG	=	90.00
AVG ROUGHNESS ELEMENT	- CM	=	74.0
ATMOSPHERIC EXTINGUISHMENT CORRECTIONS			
CORRECTED FOR VISIBILITY	-		YES
MARITIME ARCTIC AIR MASS	-		YES
CONTINENTAL POLAR AIR MASS	-		NO
CORRECTED FOR RAIN	-		NO
CORRECTED FOR SNOW	-		NO
METEOROLOGICAL INPUTS			
WINDSPEED	- M/SEC	=	2.60
WIND DIRECTION	- DEG	=	225.00
WIND GULLY	-		0
WIND CATEGORY	-		9.302
RELATIVE HUMIDITY	- PERCENT	=	87.1
TRANSMISSION THRESHOLDS			
TOTAL			SMOKE
VISIBILITY			.043
NEAR IR			.041
MID IR			.044
FAR IR			.039

----- VISIBLE: -----		----- NEAR IR: -----	
SCREEN	LENGTH METERS 400.	DURATION MINUTES 15.00	LENGTH METERS 400.
HC SMOKE SCREEN			
105MM HOWITZER			
VOLLEY	GUNS RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	8.	51.	60.
SUSTAINING:	.5	31.	173.
155MM HOWITZER			
VOLLEY	GUNS RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	3.	162.	16.
SUSTAINING:	.5	328.	30.
UP SMOKE SCREEN			
105MM HOWITZER			
VOLLEY	GUNS RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	6.	82.	129.
SUSTAINING:	.8	66.	172.
155MM HOWITZER			
VOLLEY	GUNS RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	3.	246.	31.
SUSTAINING:	.8	182.	41.

--- MID IR: ---
 LENGTH METERS 400.
 DURATION MINUTES 15.00
 SCREEN
 UP SMOKE SCREEN
 ROUNDS/ RATE/ TOTAL
 60 METERS MINUTE ROUNDS
 105MM: 3. 1. 435.
 155MM: 3. 1. 435.

--- FAR IR: ---
 LENGTH METERS 400.
 DURATION MINUTES 15.00
 SCREEN
 UP SMOKE SCREEN
 ROUNDS/ RATE/ TOTAL
 60 METERS MINUTE ROUNDS
 105MM: 2. 1. 330.
 155MM: 2. 1. 330.

OVERCAST SKY RADIATIVE TRANSFER MODULE

```

-- RADIATION UNDER OVERCAST SKY --
XO = .067 (KM)      XT = -.200 (KM)
ZO = .200          ZT = .000
ZC = .200          LC = 2.0000+002 (W/M2-SR-MU)
LAMBDA = 10.600 (MU)  LG = 5.0000+001
TEMP = 9.8 (DEG.C)  LB0 = 5.0000+001
KAPPA = 4.0000+001 (KM-1)  W0 = 1.000
ETA = .700
  
```

**THERMAL CALCULATION OF PATH RADIANCE

BBTEMP= 7.4082+000 W/M2-SR-MU

```

+-----+
PATH LENGTH (KM)  TRANSMITTANCE (W/M2-SR-MU)  PATH RADIANCE (W/M2-SR-MU)  CONTRAST TRANSMITTANCE
+-----+-----+-----+-----+
.239             .00007          2.409+001          .00015
  
```

```

-- RADIATION UNDER OVERCAST SKY --
XO = .133 (KM)      XT = -.200 (KM)
ZO = .200          ZT = .000
ZC = .200          LC = 2.0000+002 (W/M2-SR-MU)
LAMBDA = 10.600 (MU)  LG = 5.0000+001
TEMP = 9.8 (DEG.C)  LB0 = 5.0000+001
KAPPA = 4.0000+001 (KM-1)  W0 = 1.000
ETA = .700
  
```

**THERMAL CALCULATION OF PATH RADIANCE

BBTEMP= 7.4082+000 W/M2-SR-MU

```

+-----+
PATH LENGTH (KM)  TRANSMITTANCE (W/M2-SR-MU)  PATH RADIANCE (W/M2-SR-MU)  CONTRAST TRANSMITTANCE
+-----+-----+-----+-----+
.119             .00872          9.826-001          .30742
  
```


385.0	.277	.204	.519	.659	.908	.911	.904	1.000
390.0	.437	.299	.624	.827	.909	.912	.905	1.000
395.0	.570	.432	.800	.960	.910	.913	.902	1.000
400.0	.666	.518	.902	.984	.911	.914	.908	1.000

*****CARD INPUT*****

DONE .000 .000 .000 .000 .000 .000 .000 .000 .000

*****PROGRAM CRNADE END*****

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .2331-019

RUN NUMBER 2

INDIVIDUAL MODULES SELECTED
BASCHI
PCL00B

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	9433.962	9433.962
WAVELENGTH(MICRONS)	1.060	1.060
FREQUENCY(GHZ)	283018.863	283018.863

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

```

1 == FOG, HAZE AND MIST WITH VIS LT 1 KM. 3 KM.
2 == FOG, HAZE AND MIST WITH 3 LE VIS LT 3 KM.
3 == FOG, HAZE AND MIST WITH VIS GE 7 KM.
4 == DUST WITH VIS GE 3 KM.
5 == DUST WITH VIS LT 3 KM.
6 == DRIZZLE, RAIN AND TSUNGS WITH VIS LT 1 KM.
7 == DRIZZLE, RAIN AND TSUNGS WITH 3 LE VIS LT 3 KM.
8 == DRIZZLE, RAIN AND TSUNGS WITH VIS GE 7 KM.
9 == DRIZZLE, RAIN AND TSUNGS WITH VIS LT 1 KM.
10 == SNOW WITH VIS LT 3 KM.
11 == SNOW WITH 3 LE VIS LT 3 KM.
12 == SNOW WITH VIS GE 7 KM.
13 == NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CM CU M.
14 == NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CM CU M.
15 == VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
16 == VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
17 == VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
18 == VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
19 == VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
20 == VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
21 == ALL CONDITIONS COMBINED.
22 == ALL CONDITIONS COMBINED.

```


BASCAT LASER SCATTERING MODULE

 ** MONTE CARLO MULTIPLE SCATTERING **
 ** AEROSOL SCATTERING **

PARAMETERS FOR THIS RUN

WHITE PHOSPHORUS
 WAVELENGTH= 1.060 MICROMETERS ALBEDO= .999
 AEROSOL EXTINCTION COEFFICIENT= .4000+001 KM**⁻¹
 ELLIPSOIDAL AEROSOL CLOUD
 COORDINATE ORIGIN AT CENTER OF CLOUD
 Z-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH
 SOURCE PARAMETERS
 SOURCE XYZ COORDINATES(KM)= -.2000 .0000 -.0980
 SOURCE AXIS POLAR ANGLE = 90.000 DEGREES
 SOURCE AXIS AZIMUTH ANGLE = .000 DEGREES
 SOURCE APERTURE RADIUS(MM)= 50.000
 SOURCE BEAM SPREAD ANGLE = .1293-004 RADIAN
 DETECTOR PARAMETERS
 CONE OF VIEW HALF-ANGLE = 1.000 DEGREES
 DETECTOR APERTURE RADIUS = 1.000 CM
 DETECTOR XYZ COORDINATES(KM)= .2000 .0000 -.0980
 DETECTOR AXIS POLAR ANGLE = 90.000 DEGREES
 DETECTOR AXIS AZIMUTH ANGLE = 180.000 DEGREES
 GROUND PLANE PARAMETERS
 ISOTROPIC REFLECTIVITY GROUND PLANE
 GROUND PLANE ALBEDO, ALBG, = .500
 CLOUD PARAMETERS
 ELLIPSOID PRINCIPAL XYZ HALF-AXES(KM) = 1.000 .2000 .1000
 EULER ANGLES THE, PHI, PSI OF ELLIPSOID(DEC) = .0000 .0000 .0000
 OPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES = .8000 1.6000 .8000

STEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE POWER

ORDER	STEADY STATE POWER	NUMBER OF PHOTONS
0	.27973-001	.50000+004
1	.66398-009	.50000+004
2	.90441-011	.49999+004
3	.18337-012	.50000+004
4	.12166-013	.50000+004
5	.58910-015	.50000+004
TOTAL	.27973-001	

POWER INTO DETECTOR FOR 1 PULSE(S) OF DIFFERENT LENGTH

PULSE NUMBER 1 HAS LENGTH 2667+000 MICROSECONDS
 DETECTOR RESPONSE CUTOFF TIME FOR PULSE NUMBER 1 IS .7067+001 MICROSECONDS

DETECTOR RESPONSE, POWER AS A FUNCTION OF TIME, FOR UNIT PULSE POWER

TIME	POWER FROM EACH ORDER					
	0	1	2	3	4	5
.0000	.0000	.0000	.0000	.0000	.0000	.0000
.1333+000	.2797-001	.6640-009	.9044-011	.1838-012	.1188-013	.0900-015
.2667+000	.2797-001	.6640-009	.9044-011	.1838-012	.1205-013	.0900-015
.4000+000	.0000	.0000	.1887-015	.5633-016	.2372-017	.73253-018
.5333+000	.0000	.0000	.7290-017	.4168-018	.22226-019	.56036-016
.6667+000	.0000	.0000	.0000	.1108-016	.1301-018	.13334-016
.8000+000	.0000	.0000	.0000	.0000	.0000	.0000

FINITE CLOUD RADIATIVE TRANSFER MODULE

-- RADIATIVE TRANSFER THROUGH FINITE CLOUD --

(XC, YC, ZC) = (.0000, .0000, .1000) KILOMETERS
 (XE, YE, ZE) = (.1000, .2000, .1000)
 (XB, YB, ZB) = (.2000, .0000, .0020)
 (XS, YS, ZS) = (-.2000, .0000, .0020)
 INDEXP = 5
 LAMBDA = 1.060 (MU)
 OMEGA_0 = 1.000
 THETA_0 = 80.0
 RHO = .500
 TMPA = 9.8 (DEG.C)
 LD = 0
 ETA = .950
 KAPPA = 4.000+000 (KM-1)
 TAUBAR = 2.000+000
 PHIG = 0 (DEGREES)
 LBU = 50.000 (W/M2-SR-MU)
 TMPC =

**THERMAL CALCULATION OF PATH RADIANCE
 BB(LAMBDA, TMPC) = 1.3071-013 W/M2-SR-MU
 BB(LAMBDA, TMPC) = 1.3071-013 W/M2-SR-MU

PATH LENGTH TRANSMITTANCE PATH RADIANCE CONTRAST
 (IN CLOUD) (W/M2-SR-MU) TRANSMITTANCE

.040 8.523-001 1.129-014 1.000+000

RUN NUMBER 3

**** EDSEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED

	BEGINNING	ENDING
WAVENUMBER<CH**-1>	1.167	1.167
WAVELENGTH<MICRONS>	8571.428	8571.428
FREQUENCY<GHZ>	35.000	35.000

ISRAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS										DURING SPRING AT 03-09 (LST)									
CLASS NO.	FREQY CLAS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/24 H)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (MB)	MEAN WINDVEL (KMH)	MEAN/STDEV WINDVEL (%)	MEAN CLDHT (%)	MEAN/STDEV CLDCLR (%)	FREQY A (%)	FREQY B (%)	FREQY C (%)	FREQY D (%)	FREQY E (%)	FREQY F (%)		
1	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
2	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
3	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
4	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
5	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
6	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
7	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
8	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
9	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
10	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
11	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
12	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
13	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
14	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
15	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
16	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
17	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
18	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
19	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
20	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
21	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
22	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
23	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
24	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
25	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
26	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
27	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
28	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
29	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
30	10	15.0	10.0	1.0	80	10	1010	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		

VISIBILITY
9.30 KM

NEAR MILLIMETER WAVE MODULE

TEMPERATURE	5.300	DEGREES C
PRESSURE	1015.000	MB
ABSOLUTE HUMIDITY	6.300	G/M**3
FOG DENSITY	5.000	M/M**3
RAIN RATE	5.000	MM/HR
SNOW RATE	35.000	CM/HR
WIND SPEED	.400	KM
PATH LENGTH	.693	DB/KM
GAS ABSORPTION	.455	DB/KM
FOG EXTINCTION	1.252	DB/KM
RAIN EXTINCTION	.000	DB/KM
SNOW EXTINCTION	.843	DB/KM
TRANSMISSION	.572	M**2/M**3
FOG BACKSCATTER	1.026	M**2/M**3
RAIN BACKSCATTER	.000	M**2/M**3
SNOW BACKSCATTER	.000	M**2/M**3
TOTAL BACKSCATTER	1.026	M**2/M**3

TEMPERATURE	5.300	DEGREES C
PRESSURE	1015.000	MB
ABSOLUTE HUMIDITY	6.300	G/M**3
FOG DENSITY	5.000	M/M**3
RAIN RATE	5.000	MM/HR
SNOW RATE	35.000	CM/HR
WIND SPEED	.400	KM
PATH LENGTH	.693	DB/KM
GAS ABSORPTION	.455	DB/KM
FOG EXTINCTION	1.252	DB/KM
RAIN EXTINCTION	.000	DB/KM
SNOW EXTINCTION	.479	DB/KM
TRANSMISSION	.617	M**2/M**3
FOG BACKSCATTER	.522	M**2/M**3
RAIN BACKSCATTER	.000	M**2/M**3
SNOW BACKSCATTER	.327	M**2/M**3
TOTAL BACKSCATTER	.327	M**2/M**3

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .5195+000

RUN NUMBER 4

**** EOSAEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED
SPOT
LT*H

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	2010.000	2710.000
WAVELENGTH(MICRONS)	3.690	4.975
FREQUENCY(GHZ)	60300.000	81300.000

SPOT CONTRAST MODULE

SPOT CONTROL CARDS READ FOR THIS RUN:

ENVR 3000+001 2000+001 2000+001 4000+001 1000+001 1000+001
 ENVS 1000+001 2830+003 9500+000 2950+005 0000 0000
 ENVA 6500+002 0000 0000 0000 0000 0000
 TARG 4000+000 4500+002 9000+002 4500+002 0000 0000
 REFL 5000+000 5000+000 0000 5000-001 0000 0000
 SENS 2000-002 9000+002 2700+003 1000+001 0000 0000
 CG 0000 0000 0000 0000 0000 0000
 DIVISION LIMITS CHANGED FROM 40.000 TO 60.000
 INPUT RESPONSE FUNCTION
 WAVELENGTH R FUNCTION
 .3500+001 .7800+000
 .3600+001 .8000+000
 .3700+001 .8200+000
 .3800+001 .8400+000
 .3900+001 .8600+000
 .4000+001 .8800+000
 .4100+001 .9000+000
 .4200+001 .9200+000
 .4300+001 .9400+000
 .4400+001 .9600+000
 .4500+001 .9800+000
 .4600+001 1.0000+000
 .4700+001 1.0200+000
 .4800+001 1.0400+000
 .4900+001 1.0600+000
 .5000+001 1.0800+000

SPOT DIAGNOSTIC MESSAGES FOLLOW:

1. NO DIRECT SUNLIGHT INCIDENT WITHIN RECEIVER'S FIELD OF VIEW.

 DEFINITION OF CONTROL PARAMETERS FOLLOWS:
 PARAMETER VALUE DESCRIPTION
 ISORC 3 SUNLIGHT AND EMISSION
 ITARG 2 TARGET REFLECTANCE / EMISSION
 MODEL 4 SUBARCTIC SUMMER
 IHAZE 2
 CLOUD BOTTOM HEIGHT = .000 KM

SOURCE INTENSITIES

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	SUNLIGHT SOURCE STRENGTH (WATTS M-2 MICRON-1)	TARGET SOURCE STRENGTH (WATTS M-2 MICRON-1 SR-1)	GROUND SOURCE STRENGTH (WATTS M-2 MICRON-1 SR-1)
4.9751+000	2010	8787+000	2.0519+000	.0000
4.3370+000	2270	7260+000	1.7239+000	.0000
4.3398+000	2269	4894+000	1.5271+000	.0000
4.4562+000	2230	4551+000	1.3957+000	.0000
4.4443+000	2250	6741+000	1.1828+000	.0000
4.3270+000	2310	5742+000	9.5289+001	.0000
4.1192+000	2370	7482+000	8.8231+001	.0000
4.0161+000	2430	8090+000	5.7691+001	.0000
3.9311+000	2450	9096+000	4.8232+001	.0000
3.7453+000	2670	1.1966+001	3.9940+001	.0000

COMPONENTS FOR RADIANCE FROM TARGET

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET EMISSION	TARGET REFLECTANCE	PARTIAL ATMOSPHERIC EMISSION	PARTIAL PATH RADIANCE	TOTAL TARGET RADIANCE
4.9751+000	2010	8.367-001	3.550-003	6.955-001	2.645-004	1.532+000
4.8309+000	2070	7.088-001	1.422-003	1.132-001	4.047-004	1.322+000
4.6948+000	2130	6.894-001	1.630-003	2.727-001	4.879-004	1.121+000
4.5662+000	2190	6.804-001	3.183-003	2.173-001	4.889-004	1.630+001
4.4434+000	2250	6.725-001	6.095-006	3.320-001	4.629-004	3.457+001
4.3213+000	2310	6.651-001	1.937-015	2.146-001	1.048-024	2.182-001
4.2112+000	2370	6.582-001	4.489-011	3.985-002	1.238-016	1.822-001
4.1021+000	2430	6.513-001	9.717-006	5.252-002	1.274-016	1.922-001
4.0035+000	2490	6.444-001	7.271-005	5.252-002	1.274-016	1.922-001
3.9157+000	2550	6.375-001	3.645-002	2.727-001	1.274-016	1.922-001
3.8387+000	2610	6.306-001	1.640-002	2.727-001	1.274-016	1.922-001
3.7725+000	2670	6.237-001	1.640-002	2.727-001	1.274-016	1.922-001

COMPONENTS FOR BACKGROUND RADIANCE

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	GROUND EMISSION	GROUND REFLECTANCE	TOTAL ATMOSPHERIC EMISSION	TOTAL PATH RADIANCE	TOTAL BACKGROUND RADIANCE
4.9751+000	2010	.0000	.0000	1.433+000	1.350-004	1.6447+000
4.8309+000	2070	.0000	.0000	1.4031+000	1.2953+004	1.4094+000
4.6548+000	2130	.0000	.0000	1.2032+000	1.2355+003	1.2059+000
4.4444+000	2250	.0000	.0000	1.0271+001	4.0020-006	1.0274+000
4.4290+000	2270	.0000	.0000	8.5216-001	1.0520-024	9.7317+001
4.4219+000	2370	.0000	.0000	6.1611-001	1.0555-016	6.5161+001
4.1152+000	2430	.0000	.0000	5.1673-001	1.1337-011	5.2016+001
4.0161+000	2480	.0000	.0000	3.216-001	1.1559-002	3.5723+001
3.9216+000	2550	.0000	.0000	3.9335-001	2.6321-002	3.5723+001
3.8314+000	2610	.0000	.0000	2.9849-001	4.11007-002	3.4650+001
3.7453+000	2670	.0000	.0000	4.849-001	4.11007-002	3.8950+001

DIRECT SUNLIGHT

 (WATTS M-2 MICRON-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	SUNLIGHT SOURCE STRENGTH	SUNLIGHT FLUX
4.9751+000	20110	7.927+000	.0000
4.9795+000	20170	7.740+000	.0000
4.9839+000	20230	4.9846+000	.0000
4.9883+000	20290	5.4957+000	.0000
4.9927+000	20350	6.741+000	.0000
4.9971+000	20410	6.742+000	.0000
5.0015+000	20470	7.462+000	.0000
5.0059+000	20530	8.288+000	.0000
5.0103+000	20590	9.089+000	.0000
5.0147+000	20650	9.916+000	.0000
5.0191+000	20710	1.0960+001	.0000
5.0235+000	20770	1.1996+001	.0000

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(11)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

6-6

6-82

END
DATE
FILMED
6-82
DTIC

TOTAL RADIANCE

< WATTS M-2 MICRON-1 SR-1 >

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET	BACKGROUND	CONTRAST RATIO
4.9751+000	2010	1.5362+000	1.6447+000	-6.5959-002
4.8309+000	2070	1.3232+000	1.4094+000	-6.1179-002
4.6949+000	2130	1.1211+000	1.2059+000	-7.0270-002
4.5662+000	2190	9.6050-001	1.0274+000	-6.5111-002
4.4444+000	2250	7.4575-001	8.0671E-001	-2.4691-002
4.3290+000	2310	5.3211-001	5.3321E-001	-7.7684-005
4.2194+000	2370	3.1542-001	3.1611E-001	-1.1165-003
4.1152+000	2430	1.1886-001	1.0301E-001	-2.1316-002
4.0161+000	2490	6.102E-001	4.067E-001	9.4883E-003
3.9216+000	2550	1.435E-001	4.77E-001	7.4395E-003
3.8314+000	2610	7.419E-001	6.65E-001	7.3745E-003
3.7453+000	2670	2.256E-001	6.655E-001	1.1420E-001

DETECTOR-RESPONSE WAVELENGTH-INTEGRATED

(WATTS M-2 SR-1)

TARGET EMISSION	4.9117-002
TARGET REFLECTANCE	1.8157-003
PARTIAL ATMOSPHERIC EMISSION	4.3029-002
PARTIAL PATH RADIANCE	7.7395-005
TOTAL TARGET RADIANCE	9.4039-002
GROUND EMISSION	.0000
GROUND REFLECTANCE	.0000
TOTAL ATMOSPHERIC EMISSION	9.7352-002
TOTAL PATH RADIANCE	9.1328-004
TOTAL BACKGROUND RADIANCE	9.8265-002

CONTRAST -4.3004-002

DIRECT SUNLIGHT .0000
(WATTS M-2)

FREQ	WAVELENGTH	H2O	CO2+	OZONE	N2	C	H2O C	HOL S	NITRIC	SO2	HN03	NO2	INTEGRATED	TOTAL	AEROSOL
CM-1	MICRONS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	ABSORPTION	TRANS	TRANS
2010	4.9731	.6090	.9998	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	12.6982	.5767	.94820
2070	4.83309	.76088	.9996	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	38.8084	.5648	.94820
2130	4.6944	.86649	.9990	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	54.9334	.7313	.94820
2190	4.5662	.93398	.9983	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	70.8646	.7349	.94820
2250	4.4444	.9367	.9973	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	113.9851	.2813	.94820
2310	4.3290	.9769	.9961	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	173.9270	.0010	.94820
2370	4.2194	.9925	.9941	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	233.0067	.0153	.94820
2430	4.1152	.9977	.9941	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	298.9729	.9006	.94820
2490	4.0161	.9986	.9941	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	343.8880	.3181	.94820
2550	3.9216	.9903	.9976	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	349.4692	.9070	.94820
2610	3.8314	.9663	.9989	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	355.0312	.9073	.94820
2670	3.7453	.9383	.9981	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	262.2208	.8802	.94820

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

INTEGRATED ABSORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358

FR(CM-1)	WVL(MICRON)	RADIANCE(WATTS/CM2-STER-XXX)		INTEGRAL	TRANS	AERO TRAN	
		PER CM-1	PER MICRON			EXTN	TRAN
2010.0	4.975124	.17216-006	.69555-004	.51648-005	.576727	.94820	.98940
2070.0	4.830918	.14311-006	.61320-004	.13751-004	.564830	.94820	.98940
2130.0	4.694836	.71276-007	.32337-004	.18028-004	.731250	.94820	.98940
2190.0	4.566210	.56657-007	.27173-004	.21427-004	.734479	.94820	.98940
2250.0	4.444444	.12310-006	.62320-004	.28813-004	.281320	.94820	.98940
2310.0	4.329004	.13709-006	.73146-004	.37038-004	.009967	.94820	.98940
2370.0	4.219409	.10801-006	.60666-004	.43518-004	.015339	.94820	.98940
2430.0	4.115226	.87026-008	.51388-005	.44040-004	.900563	.94820	.98940
2490.0	4.016064	.57098-008	.35402-005	.44383-004	.919083	.94820	.98940
2550.0	3.921569	.51549-008	.33520-005	.44692-004	.906379	.94820	.98940
2610.0	3.831418	.40774-008	.27776-005	.44937-004	.907301	.94820	.98940
2670.0	3.745318	.41768-008	.29776-005	.45188-004	.880173	.94820	.98940

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

RADMIN 2610.000 .40774-008
 RADMAX 2010.000 .17216-006

INTEGRATED ABSORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358
 INTEGRATED RADIANCE = .45188-004 WATT CM -2 SR

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .4711-001

END EOSAEL RUN

ELECTRO-OPTICS DISTRIBUTION LIST

Commander

US Army Aviation School
Fort Rucker, AL 36362

Commander

US Army Aviation Center
ATTN: ATZQ-D-MA (Mr. Oliver N. Heath)
Fort Rucker, AL 36362

Commander

US Army Aviation Center
ATTN: ATZQ-D-MS (Mr. Donald Wagner)
Fort Rucker, AL 36362

NASA/Marshall Space Flight Center
ATTN: ES-83 (Otha H. Vaughan, Jr.)
Huntsville, AL 35812

NASA/Marshall Space Flight Center
Atmospheric Sciences Division
ATTN: Code ES-81 (Dr. William W. Vaughan)
Huntsville, AL 35812

Nichols Research Corporation
ATTN: Dr. Lary W. Pinkley
4040 South Memorial Parkway
Huntsville, AL 35802

John M. Hobbie
c/o Kentron International
2003 Byrd Spring Road
Huntsville, AL 35802

Mr. Ray Baker
Lockheed-Missile & Space Company
4800 Bradford Blvd
Huntsville, AL 35807

Commander

US Army Missile Command
ATTN: DRSMI-OG (Mr. Donald R. Peterson)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-OGA (Dr. Bruce W. Fowler)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-REL (Dr. George Emmons)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-REO (Huey F. Anderson)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-REO (Mr. Maxwell W. Harper)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-REO (Mr. Gene Widenhofer)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-RHC (Dr. Julius Q. Lilly)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
Redstone Scientific Information Center
ATTN: DRSMI-RPRD (Documents Section)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-RRA (Dr. Oskar Essenwanger)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-RRO (Mr. Charles Christensen)
Redstone Arsenal, AL 35809

Commander

US Army Missile Command
ATTN: DRSMI-RRO (Dr. George A. Tanton)
Redstone Arsenal, AL 35809

Commander
US Army Communications Command
ATTN: CC-OPS-PP
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-CS (Mr. Richard G. Cundy)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-MD (Mr. Harry Wilder)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CS-C (2LT Coffman)
Fort Huachuca, AZ 85613

Commander
US Army Yuma Proving Ground
ATTN: STEYP-MSA-TL
Bldg 2105
Yuma, AZ 85364

Northrop Corporation
Electro-Mechanical Division
ATTN: Dr. Richard D. Tooley
500 East Orangethorpe Avenue
Anaheim, CA 92801

Commander
Naval Weapons Center
ATTN: Code 3918 (Dr. Alexis Shlanta)
China Lake, CA 93555

Hughes Helicopters
Army Advanced Attack Helicopter Weapons
ATTN: Mr. Charles R. Hill
Centinela and Teale Streets
Bldg 305, MS T-73A
Culter City, CA 90230

Commander
US Army Combat Developments
Experimentation Command
ATTN: ATEC-PL-M (Mr. Gary G. Love)
Fort Ord, CA 93941

SRI International
ATTN: K2060/Dr. Edward E. Uthe
333 Ravenswood Avenue
Menlo Park, CA 94025

SRI International
ATTN: Mr. J. E. Van der Laan
333 Ravenswood Avenue
Menlo Park, CA 94025

Joane May
Naval Environmental Prediction
Research Facility (NEPRF)
ATTN: Library
Monterey, CA 93940

Sylvania Systems Group,
Western Division
GTE Products Corporation
ATTN: Technical Reports Library
P.O. Box 205
Mountain View, CA 94042

Sylvania Systems Group
Western Division
GTE Products Corporation
ATTN: Mr. Lee W. Carrier
P.O. Box 188
Mountain View, CA 94042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3250-3 (R. de Violini)
Point Mugu, CA 93042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3253 (Terry E. Battalino)
Point Mugu, CA 93042

Effects Technology Inc.
ATTN: Mr. John D. Carlyle
5383 Hollister Avenue
Santa Barbara, CA 93111

Commander
Naval Ocean Systems Center
ATTN: Code 532 (Dr. Juergen Richter)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 5322 (Mr. Herbert G. Hughes)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 4473 (Tech Library)
San Diego, CA 92152

The RAND Corporation
ATTN: Ralph Huschke
1700 Main Street
Santa Monica, CA 90406

Particle Measuring Systems, Inc.
ATTN: Dr. Robert G. Knollenberg
1855 South 57th Court
Boulder, CO 80301

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: Library, R-51, Technical Reports
325 Broadway
Boulder, CO 80303

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: R45X3 (Dr. Vernon E. Derr)
Boulder, CO 80303

US Department of Commerce
National Telecommunications and
Information Administration
Institute for Telecommunication Sciences
ATTN: Code 1-3426 (Dr. Hans J. Liebe)
Boulder, CO 80303

AFATL/DLODL
Technical Library
Eglin AFB, FL 32542

Commanding Officer
Naval Training Equipment Center
ATTN: Technical Information Center
Orlando, FL 32813

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. Robert W. McMillan
Atlanta, GA 30332

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. James C. Wiltse
Atlanta, GA 30332

Commandant
US Army Infantry Center
ATTN: ATSH-CD-MS-E (Mr. Robert McKenna)
Fort Benning, GA 31805

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-CS
Fort Gordon, GA 30905

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-O
Fort Gordon, GA 30905

USAFETAC/DNE
ATTN: Mr. Charles Glauber
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DNDP (LTC Kit G. Cottrell)
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DOOF (MAJ Robert Wright)
Scott AFB, IL 62225

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-CAA-Q (Mr. H. Kent Pickett)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Robert DeKinder, Jr.)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Mr. Kent I. Johnson)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-WE (LTC Darrell Holland)
Fort Leavenworth, KS 66027

President
USAARENBD
ATTN: ATZK-AE-TA (Dr. Charles R. Leake)
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-MS
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-SD
Fort Knox, KY 40121

Aerodyne Research Inc.
ATTN: Dr. John F. Ebersole
Crosby Drive
Bedford, MA 01730

Commander
Air Force Geophysics Laboratory
ATTN: OPA (Dr. Robert W. Fenn)
Hanscom AFB, MA 01731

Commander
Air Force Geophysics Laboratory
ATTN: OPI (Dr. Robert A. McClatchey)
Hanscom AFB, MA 01731

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. T. J. Goblick, B-370
P.O. Box 73
Lexington, MA 02173

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. Michael Gruber
P.O. Box 73
Lexington, MA 02173

Raytheon Company
Equipment Division
ATTN: Dr. Charles M. Sonnenschein
430 Boston Post Road
Wayland, MA 01778

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-BLB (Mr. Richard McGee)
Aberdeen Proving Ground, MD 21005

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Dr. Edward Stuebing)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Mr. Joseph Vervier)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLY-A (Mr. Ronald Pennsyle)
Aberdeen Proving Ground, MD 21010

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-TSB-S (STINFO)
Aberdeen Proving Ground, MD 21005

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CCM (W. H. Pepper)
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CG/DRDEL-DC/DRDEL-CS
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CT
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-PAO (M. Singleton)
2800 Powder Mill Road
Adelphi, MD 20783

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK
(Dr. Anthony Van de Wal, Jr.)
Aberdeen Proving Ground, MD 21005

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK-T (Mr. Sidney Gerard)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSTE-AD-M (Mr. Warren M. Baity)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSTE-AD-M (Dr. Norman E. Pentz)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-AAM (Mr. William Smith)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-CS (Mr. Philip H. Beavers)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GB (Wilbur L. Warfield)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GP (Mr. Fred Campbell)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GP (H. Stamper)
Aberdeen Proving Grounds, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GS
(Mr. Michael Starks/Mr. Julian Chernick)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-J (Mr. James F. O'Bryon)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-LM (Mr. Robert M. Marchetti)
Aberdeen Proving Ground, MD 21005

Commander
Harry Diamond Laboratories
ATTN: Dr. William W. Carter
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM (Mr. Robert McCoskey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Robert Humphrey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Z. G. Sztankay)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Joseph Nemarich)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Air Force Systems Command
ATTN: WER (Mr. Richard F. Picanso)
Andrews AFB, MD 20334

Martin Marietta Laboratories
ATTN: Jar Mo Chen
1450 South Rolling Road
Baltimore, MD 21227

Commander
US Army Concepts Analysis Agency
ATTN: CSCA-SMC (Mr. Hal E. Hock)
8120 Woodmont Avenue
Bethesda, MD 20014

Director
National Security Agency
ATTN: R52/Dr. Douglas Woods
Fort George G. Meade, MD 20755

Chief
Intelligence Materiel Development
& Support Office
US Army Electronic Warfare Laboratory
ATTN: DELEW-I (LTC Kenneth E. Thomas)
Fort George G. Meade, MD 20755

The Johns Hopkins University
Applied Physics Laboratory
ATTN: Dr. Michael J. Lun
John Hopkins Road
Laurell, MD 20810

Dr. Stephen T. Hanley
1720 Rhodesia Avenue
Oxon Hill, MD 20022

Science Applications Inc.
ATTN: Mr. G. D. Currie
15 Research Drive
Ann Arbor, MI 48103

Science Applications Inc.
ATTN: Dr. Robert E. Turner
15 Research Drive
Ann Arbor, MI 48103

Commander
US Army Tank-Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Harry Young)
Warren, MI 48090

Commander
US Army Tank Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Wallace Mick, Jr.)
Warren, MI 48090

Dr. A. D. Belmont
Research Division
Control Data Corporation
P.O. Box 1249
Minneapolis, MN 55440

Director
US Army Engr Waterways Experiment Station
ATTN: WESEN (Mr. James Mason)
P.O. Box 631
Vicksburg, MS 39180

Dr. Jerry Davis
Department of Marine, Earth
and Atmospheric Sciences
North Carolina State University
Raleigh, NC 27650

Commander
US Army Research Office
ATTN: DRXRO-GS (Dr. Leo Alpert)
P.O. Box 12211
Research Triangle Park, NC 27709

Commander
US Army Research Office
ATTN: DRXRO-PP (Brenda Mann)
P.O. Box 12211
Research Triangle Park, NC 27709

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RD (Dr. K. F. Sterrett)
Hanover, NH 03755

Commander/Director
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. George Aitken)
Hanover, NH 03755

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. Roger H. Berger)
Hanover, NH 03755

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-AC (Mr. James Greenfield)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-TSS (Bldg #59)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. Peteris Jansons)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. G. H. Waldron)
Dover, NJ 07801

Deputy Joint Project Manager
for Navy/USMC SAL GP
ATTN: DRCPM-CAWS-NV (CPT Joseph Miceli)
Dover, NJ 07801

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-I (Mr. David Longinotti)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-PE (Mr. Ben A. Di Campi)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-R-S (Mr. Donald L. Fofani)
Fort Monmouth, NJ 07703

Director
US Army Electronics Technology &
Devices Laboratory
ATTN: DELET-DD (S. Danko)
Fort Monmouth, NJ 07703

Project Manager
FIREFINDER/REMBASS
ATTN: DRCPM-FFR-TM (Mr. John M. Bialo)
Fort Monmouth, NJ 07703

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-SA (Dr. Walter S. McAfee)
Fort Monmouth, NJ 07703

OLA, 2WS (MAC)
Holloman AFB, NM 88330
Commander
Air Force Weapons Laboratory
ATTN: AFWL/WE (MAJ John R. Elrick)
Kirtland, AFB, NM 87117

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL (Dolores Anguiano)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. Louie Dominguez)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. William J. Leach)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TGP (Mr. Roger F. Willis)
White Sands Missile Range, NM 88002

Director
Office of Missile Electronic Warfare
ATTN: DELEW-M-STO (Dr. Steven Kovel)
White Sands Missile Range, NM 88002

Office of the Test Director
Joint Services EO GW CM Test Program
ATTN: DRXDE-TD (Mr. Weldon Findley)
White Sands Missile Range, NM 88002

Commander
US Army White Sands Missile Range
ATTN: STEWS-PT-AL (Laurel B. Saunders)
White Sands Missile Range, NM 88002

Commander
US Army R&D Coordinator
US Embassy - Bonn
Box 165
APO New York 09080

Grumman Aerospace Corporation
Research Department - MS A08-35
ATTN: John E. A. Selby
Bethpage, NY 11714

Rome Air Development Center
ATTN: Documents Library
TSLD (Bette Smith)
Griffiss AFB, NY 13441

Dr. Roberto Vaglio-Laurin
Faculty of Arts and Science
Dept. of Applied Science
26-36 Stuyvesant Street
New York, NY 10003

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (Mr. Harold Geltmacher)
Wright-Patterson AFB, OH 45433

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (CPT William C. Smith)
Wright-Patterson AFB, OH 45433

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R (CPT James M. Watson)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CD-MS
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: NOAA Liaison Officer
(CDR Jeffrey G. Carlen)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
Morris Swett Library
ATTN: Reference Librarian
Fort Sill, OK 73503

Commander
Naval Air Development Center
ATTN: Code 301 (Mr. George F. Eck)
Warminster, PA 18974

The University of Texas at El Paso
Electrical Engineering Department
ATTN: Dr. Joseph H. Pierluissi
El Paso, TX 79968

Commandant
US Army Air Defense School
ATTN: ATSA-CD-SC-A (CPT Charles T. Thorn)
Fort Bliss, TX 79916

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-OP-Q (CPT Henry C. Cobb, Jr.)
Fort Hood, TX 76544

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-SCI (Dr. Darrell W. Collier)
Fort Hood, TX 76544

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-L
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-M (Mr. Paul E. Carlson)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. John Trethewey)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. William Peterson)
Dugway, UT 84022

Defense Documentation Center
ATTN: DDC-TCA
Cameron Station Bldg 5
Alexandria, VA 22314
12

Ballistic Missile Defense Program Office
ATTN: DACS-BMT (Colonel Harry F. Ennis)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Technical Information Center
ATTN: DDA-2 (Mr. James E. Shafer)
Cameron Station, Bldg 5
Alexandria, VA 22314

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCBSI-EE (Mr. Albert Giambalvo)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCLDC (Mr. James Bender)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Advanced Rsch Projects Agency
ATTN: Steve Zakanyez
1400 Wilson Blvd
Arlington, VA 22209

Defense Advanced Rsch Projects Agency
ATTN: Dr. James Tegnalia
1400 Wilson Blvd
Arlington, VA 22209

Institute for Defense Analyses
ATTN: Mr. Lucien M. Biberman
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Ernest Bauer
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Hans G. Wolfhard
400 Army-Navy Drive
Arlington, VA 22202

System Planning Corporation
ATTN: Mr. Daniel Friedman
1500 Wilson Boulevard
Arlington, VA 22209

System Planning Corporation
ATTN: COL Hank Shelton
1500 Wilson Boulevard
Arlington, VA 22209

US Army Intelligence & Security Command
ATTN: Edwin Speakman, Scientific Advisor
Arlington Hall Station
Arlington, VA 22212

Commander
US Army Operational Test
& Evaluation Agency
ATTN: CSTE-ED (Mr. Floyd I. Hill)
5600 Columbia Pike
Falls Church, VA 22041

Commander and Director
US Army Engineer Topographic Laboratories
ATTN: ETL-GS-A (Mr. Thomas Neidringhaus)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Rudolf G. Buser)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Robert S. Rodhe)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Joseph R. Moulton)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Luanne P. Obert)
Fort Belvoir, VA 22060

Director
US Army Night Vision
& Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Thomas W. Cassidy)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Richard J. Bergemann)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Dr. James A. Ratches)
Fort Belvoir, VA 22060

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN-M
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-F-A (Mr. Chris O'Connor, Jr.)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-IE-R (Mr. David M. Ingram)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-M-I/ATCD-M-A
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATDOC-TA (Dr. Marvin P. Pastel)
Fort Monroe, VA 23651

Department of the Air Force
OL-I, AWS
Fort Monroe, VA 23651

Department of the Air Force
HQS 5 Weather Wing (MAC)
ATTN: 5 WW/DN
Langley Air Force Base, VA 23655

Commander
US Army INSCOM/Quest Research Corporation
ATTN: Mr. Donald Wilmot
6845 Elm Street, Suite 407
McLean, VA 22101

General Research Corporation
ATTN: Dr. Ralph Zirkind
7655 Old Springhouse Road
McLean, VA 22102

Science Applications, Inc.
8400 Westpark Drive
ATTN: Dr. John E. Cockayne
McLean, VA 22102

US Army Nuclear & Chemical Agency
ATTN: MONA-WE (Dr. John A. Berberet)
7500 Backlick Road, Bldg 2073
Springfield, VA 22150

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-EA (Mr. Douglas Harkleroad)
Vint Hill Farms Station
Warrenton, VA 22186

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-OS (Dr. Royal H. Burkhardt)
Vint Hill Farms Station
Warrenton, VA 22186

Commander
US Army Cold Regions Test Center
ATTN: STECR-TD (Mr. Jerold Barger)
APO Seattle, WA 98733

HQDA (SAUS-OR/Hunter M. Woodall, Jr./
Dr. Herbert K. Fallin)
Rm 2E 614, Pentagon
Washington, DC 20301

COL Elbert W. Friday, Jr.
OUSDRE
Rm 3D 129, Pentagon
Washington, DC 20301

Defense Communications Agency
Technical Library Center
Code 222
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: Technical Library (Mrs. Betty Fox)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: RAAE (Dr. Carl Fitz)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: SPAS (Mr. Donald J. Kohler)
Washington, DC 20305

Defense Intelligence Agency
ATTN: DT/AC (LTC Robert Poplawski)
Washington, DC 20301

HQDA (DAMA-ARZ-D/Dr. Verderame)
Washington, DC 20310

HQDA (DAMI-ISP/Mr. Beck)
Washington, DC 20310

Department of the Army
Deputy Chief of Staff for
Operations and Plans
ATTN: DAMO-RQ
Washington, DC 20310

Department of the Army
Director of Telecommunications and
Command and Control
ATTN: DAMO-TCZ
Washington, DC 20310

Department of the Army
Assistant Chief of Staff for Intelligence
ATTN: DAMI-TS
Washington, DC 20310

HQDA (DAEN-RDM/Dr. de Percin)
Casimir Pulaski Building
20 Massachusetts Avenue
Room 6203
Washington, DC 20314

National Science Foundation
Division of Atmospheric Sciences
ATTN: Dr. Eugene W. Bierly
1800 G. Street, N.W.
Washington, DC 20550

Director
Naval Research Laboratory
ATTN: Code 4320 (Dr. Lothar H. Ruhnke)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6009 (Dr. John MacCallum, Jr.)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6530 (Mr. Raymond A. Patten)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6533 (Dr. James A. Dowling)
Washington, DC 20375

ATMOSPHERIC SCIENCES RESEARCH REPORTS

1. Lindberg, J. D. "An Improvement to a Method for Measuring the Absorption Coefficient of Atmospheric Dust and other Strongly Absorbing Powders," ECOM-5565, July 1975.
2. Avara, Elton P., "Mesoscale Wind Shears Derived from Thermal Winds," ECOM-5566, July 1975.
3. Gomez, Richard B., and Joseph H. Pierluissi, "Incomplete Gamma Function Approximation for King's Strong-Line Transmittance Model," ECOM-5567, July 1975.
4. Blanco, A. J., and B. F. Engebos, "Ballistic Wind Weighting Functions for Tank Projectiles," ECOM-5568, August 1975.
5. Taylor, Fredrick J., Jack Smith, and Thomas H. Pries, "Crosswind Measurements through Pattern Recognition Techniques," ECOM-5569, July 1975.
6. Walters, D. L., "Crosswind Weighting Functions for Direct-Fire Projectiles," ECOM-5570, August 1975.
7. Duncan, Louis D., "An Improved Algorithm for the Iterated Minimal Information Solution for Remote Sounding of Temperature," ECOM-5571, August 1975.
8. Robbiani, Raymond L., "Tactical Field Demonstration of Mobile Weather Radar Set AN/TPS-41 at Fort Rucker, Alabama," ECOM-5572, August 1975.
9. Miers, B., G. Blackman, D. Langer, and N. Lorimier, "Analysis of SMS/GOES Film Data," ECOM-5573, September 1975.
10. Manquero, Carlos, Louis Duncan, and Rufus Bruce, "An Indication from Satellite Measurements of Atmospheric CO₂ Variability," ECOM-5574, September 1975.
11. Petracca, Carmine, and James D. Lindberg, "Installation and Operation of an Atmospheric Particulate Collector," ECOM-5575, September 1975.
12. Avara, Elton P., and George Alexander, "Empirical Investigation of Three Iterative Methods for Inverting the Radiative Transfer Equation," ECOM-5576, October 1975.
13. Alexander, George D., "A Digital Data Acquisition Interface for the SMS Direct Readout Ground Station - Concept and Preliminary Design," ECOM-5577, October 1975.
14. Cantor, Israel, "Enhancement of Point Source Thermal Radiation Under Clouds in a Nonattenuating Medium," ECOM-5578, October 1975.

15. Norton, Colburn, and Glenn Hoidale, "The Diurnal Variation of Mixing Height by Month over White Sands Missile Range, NM," ECOM-5579, November 1975.
16. Avara, Elton P., "On the Spectrum Analysis of Binary Data," ECOM-5580, November 1975.
17. Taylor, Fredrick J., Thomas H. Pries, and Chao-Huan Huang, "Optimal Wind Velocity Estimation," ECOM-5581, December 1975.
18. Avara, Elton P., "Some Effects of Autocorrelated and Cross-Correlated Noise on the Analysis of Variance," ECOM-5582, December 1975.
19. Gillespie, Patti S., R. L. Armstrong, and Kenneth O. White, "The Spectral Characteristics and Atmospheric CO₂ Absorption of the Ho³:YLF Laser at 2.05 μ m," ECOM-5583, December 1975.
20. Novlan, David J., "An Empirical Method of Forecasting Thunderstorms for the White Sands Missile Range," ECOM-5584, February 1976.
21. Avara, Elton P., "Randomization Effects in Hypothesis Testing with Autocorrelated Noise," ECOM-5585, February 1976.
22. Watkins, Wendell R., "Improvements in Long Path Absorption Cell Measurement," ECOM-5586, March 1976.
23. Thomas, Joe, George D. Alexander, and Marvin Dubbin, "SATTEL - An Army Dedicated Meteorological Telemetry System," ECOM-5587, March 1976.
24. Kennedy, Bruce W., and Delbert Bynum, "Army User Test Program for the RDT&E-XM-75 Meteorological Rocket," ECOM-5588, April 1976.
25. Barnett, Kenneth M., "A Description of the Artillery Meteorological Comparisons at White Sands Missile Range, October 1974 - December 1974 ('PASS' - Prototype Artillery [Meteorological] Subsystem)," ECOM-5589, April 1976.
26. Miller, Walter B., "Preliminary Analysis of Fall-of-Shot From Project 'PASS'," ECOM-5590, April 1976.
27. Avara, Elton P., "Error Analysis of Minimum Information and Smith's Direct Methods for Inverting the Radiative Transfer Equation," ECOM-5591, April 1976.
28. Yee, Young P., James D. Horn, and George Alexander, "Synoptic Thermal Wind Calculations from Radiosonde Observations Over the Southwestern United States," ECOM-5592, May 1976.

29. Duncan, Louis D., and Mary Ann Seagraves, "Applications of Empirical Corrections to NOAA-4 VTPR Observations," ECOM-5593, May 1976.
30. Miers, Bruce T., and Steve Weaver, "Applications of Meteorological Satellite Data to Weather Sensitive Army Operations," ECOM-5594, May 1976.
31. Sharenow, Moses, "Redesign and Improvement of Balloon ML-566," ECOM-5595, June 1976.
32. Hansen, Frank V., "The Depth of the Surface Boundary Layer," ECOM-5596, June 1976.
33. Pinnick, R. G., and E. B. Stenmark, "Response Calculations for a Commercial Light-Scattering Aerosol Counter," ECOM-5597, July 1976.
34. Mason, J., and G. B. Hoidale, "Visibility as an Estimator of Infrared Transmittance," ECOM-5598, July 1976.
35. Bruce, Rufus E., Louis D. Duncan, and Joseph H. Pierluissi, "Experimental Study of the Relationship Between Radiosonde Temperatures and Radiometric-Area Temperatures," ECOM-5599, August 1976.
36. Duncan, Louis D., "Stratospheric Wind Shear Computed from Satellite Thermal Sounder Measurements," ECOM-5800, September 1976.
37. Taylor, F., P. Mohan, P. Joseph, and T. Pries, "An All Digital Automated Wind Measurement System," ECOM-5801, September 1976.
38. Bruce, Charles, "Development of Spectrophones for CW and Pulsed Radiation Sources," ECOM-5802, September 1976.
39. Duncan, Louis D., and Mary Ann Seagraves, "Another Method for Estimating Clear Column Radiances," ECOM-5803, October 1976.
40. Blanco, Abel J., and Larry E. Taylor, "Artillery Meteorological Analysis of Project Pass," ECOM-5804, October 1976.
41. Miller, Walter, and Bernard Engebos, "A Mathematical Structure for Refinement of Sound Ranging Estimates," ECOM-5805, November 1976.
42. Gillespie, James B., and James D. Lindberg, "A Method to Obtain Diffuse Reflectance Measurements from 1.0 and 3.0 μ m Using a Cary 17I Spectrophotometer," ECOM-5806, November 1976.
43. Rubio, Roberto, and Robert O. Olsen, "A Study of the Effects of Temperature Variations on Radio Wave Absorption," ECOM-5807, November 1976.

44. Ballard, Harold N., "Temperature Measurements in the Stratosphere from Balloon-Borne Instrument Platforms, 1968-1975," ECOM-5808, December 1976.
45. Monahan, H. H., "An Approach to the Short-Range Prediction of Early Morning Radiation Fog," ECOM-5809, January 1977.
46. Engebos, Bernard Francis, "Introduction to Multiple State Multiple Action Decision Theory and Its Relation to Mixing Structures," ECOM-5810, January 1977.
47. Low, Richard D. H., "Effects of Cloud Particles on Remote Sensing from Space in the 10-Micrometer Infrared Region," ECOM-5811, January 1977.
48. Bonner, Robert S., and R. Newton, "Application of the AN/GVS-5 Laser Rangefinder to Cloud Base Height Measurements," ECOM-5812, February 1977.
49. Rubio, Roberto, "Lidar Detection of Subvisible Reentry Vehicle Erosive Atmospheric Material," ECOM-5813, March 1977.
50. Low, Richard D. H., and J. D. Horn, "Mesoscale Determination of Cloud-Top Height: Problems and Solutions," ECOM-5814, March 1977.
51. Duncan, Louis D., and Mary Ann Seagraves, "Evaluation of the NOAA-4 VTPR Thermal Winds for Nuclear Fallout Predictions," ECOM-5815, March 1977.
52. Randhawa, Jagir S., M. Izquierdo, Carlos McDonald, and Zvi Salpeter, "Stratospheric Ozone Density as Measured by a Chemiluminescent Sensor During the Stratcom VI-A Flight," ECOM-5816, April 1977.
53. Rubio, Roberto, and Mike Izquierdo, "Measurements of Net Atmospheric Irradiance in the 0.7- to 2.8-Micrometer Infrared Region," ECOM-5817, May 1977.
54. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, Consultant for Chemical Kinetics, "Calculation of Selected Atmospheric Composition Parameters for the Mid-Latitude, September Stratosphere," ECOM-5818, May 1977.
55. Mitchell, J. D., R. S. Sagar, and R. O. Olsen, "Positive Ions in the Middle Atmosphere During Sunrise Conditions," ECOM-5819, May 1977.
56. White, Kenneth O., Wendell R. Watkins, Stuart A. Schleusener, and Ronald L. Johnson, "Solid-State Laser Wavelength Identification Using a Reference Absorber," ECOM-5820, June 1977.
57. Watkins, Wendell R., and Richard G. Dixon, "Automation of Long-Path Absorption Cell Measurements," ECOM-5821, June 1977.

58. Taylor, S. E., J. M. Davis, and J. B. Mason, "Analysis of Observed Soil Skin Moisture Effects on Reflectance," ECOM-5822, June 1977.
59. Duncan, Louis D., and Mary Ann Seagraves, "Fallout Predictions Computed from Satellite Derived Winds," ECOM-5823, June 1977.
60. Snider, D. E., D. G. Murcray, F. H. Murcray, and W. J. Williams, "Investigation of High-Altitude Enhanced Infrared Background Emissions," (U), SECRET, ECOM-5824, June 1977.
61. Dubbin, Marvin H., and Dennis Hall, "Synchronous Meteorological Satellite Direct Readout Ground System Digital Video Electronics," ECOM-5825, June 1977.
62. Miller, W., and B. Engebos, "A Preliminary Analysis of Two Sound Ranging Algorithms," ECOM-5826, July 1977.
63. Kennedy, Bruce W., and James K. Luers, "Ballistic Sphere Techniques for Measuring Atmospheric Parameters," ECOM-5827, July 1977.
64. Duncan, Louis D., "Zenith Angle Variation of Satellite Thermal Sounder Measurements," ECOM-5828, August 1977.
65. Hansen, Frank V., "The Critical Richardson Number," ECOM-5829, September 1977.
66. Ballard, Harold N., and Frank P. Hudson (Compilers), "Stratospheric Composition Balloon-Borne Experiment," ECOM-5830, October 1977.
67. Barr, William C., and Arnold C. Peterson, "Wind Measuring Accuracy Test of Meteorological Systems," ECOM-5831, November 1977.
68. Ethridge, G. A., and F. V. Hansen, "Atmospheric Diffusion: Similarity Theory and Empirical Derivations for Use in Boundary Layer Diffusion Problems," ECOM-5832, November 1977.
69. Low, Richard D. H., "The Internal Cloud Radiation Field and a Technique for Determining Cloud Blackness," ECOM-5833, December 1977.
70. Watkins, Wendell R., Kenneth O. White, Charles W. Bruce, Donald L. Walters, and James D. Lindberg, "Measurements Required for Prediction of High Energy Laser Transmission," ECOM-5834, December 1977.
71. Rubio, Robert, "Investigation of Abrupt Decreases in Atmospherically Backscattered Laser Energy," ECOM-5835, December 1977.
72. Monahan, H. H., and R. M. Cionco, "An Interpretative Review of Existing Capabilities for Measuring and Forecasting Selected Weather Variables (Emphasizing Remote Means)," ASL-TR-0001, January 1978.

73. Heaps, Melvin G., "The 1979 Solar Eclipse and Validation of D-Region Models," ASL-TR-0002, March 1978.
74. Jennings, S. G., and J. B. Gillespie, "M.I.E. Theory Sensitivity Studies - The Effects of Aerosol Complex Refractive Index and Size Distribution Variations on Extinction and Absorption Coefficients, Part II: Analysis of the Computational Results," ASL-TR-0003, March 1978.
75. White, Kenneth O., et al, "Water Vapor Continuum Absorption in the 3.5 μ m to 4.0 μ m Region," ASL-TR-0004, March 1978.
76. Olsen, Robert O., and Bruce W. Kennedy, "ABRES Pretest Atmospheric Measurements," ASL-TR-0005, April 1978.
77. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, "Calculation of Atmospheric Composition in the High Latitude September Stratosphere," ASL-TR-0006, May 1978.
78. Watkins, Wendell R., et al, "Water Vapor Absorption Coefficients at HF Laser Wavelengths," ASL-TR-0007, May 1978.
79. Hansen, Frank V., "The Growth and Prediction of Nocturnal Inversions," ASL-TR-0008, May 1978.
80. Samuel, Christine, Charles Bruce, and Ralph Brewer, "Spectrophone Analysis of Gas Samples Obtained at Field Site," ASL-TR-0009, June 1978.
81. Pinnick, R. G., et al., "Vertical Structure in Atmospheric Fog and Haze and its Effects on IR Extinction," ASL-TR-0010, July 1978.
82. Low, Richard D. H., Louis D. Duncan, and Richard B. Gomez, "The Microphysical Basis of Fog Optical Characterization," ASL-TR-0011, August 1978.
83. Heaps, Melvin G., "The Effect of a Solar Proton Event on the Minor Neutral Constituents of the Summer Polar Mesosphere," ASL-TR-0012, August 1978.
84. Mason, James B., "Light Attenuation in Falling Snow," ASL-TR-0013, August 1978.
85. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' Meteorological Application," ASL-TR-0014, September 1978.
86. Heaps, M. G., and F. E. Niles, "Modeling of Ion Chemistry of the D-Region: A Case Study Based Upon the 1966 Total Solar Eclipse," ASL-TR-0015, September 1978.

87. Jennings, S. G., and R. G. Pinnick, "Effects of Particulate Complex Refractive Index and Particle Size Distribution Variations on Atmospheric Extinction and Absorption for Visible Through Middle-Infrared Wavelengths," ASL-TR-0016, September 1978.
88. Watkins, Wendell R., Kenneth O. White, Lanny R. Bower, and Brian Z. Sojka, "Pressure Dependence of the Water Vapor Continuum Absorption in the 3.5- to 4.0-Micrometer Region," ASL-TR-0017, September 1978.
89. Miller, W. B., and B. F. Engebos, "Behavior of Four Sound Ranging Techniques in an Idealized Physical Environment," ASL-TR-0018, September 1978.
90. Gomez, Richard G., "Effectiveness Studies of the CBU-88/B Bomb, Cluster, Smoke Weapon," (U), CONFIDENTIAL ASL-TR-0019, September 1978.
91. Miller, August, Richard C. Shirkey, and Mary Ann Seagraves, "Calculation of Thermal Emission from Aerosols Using the Doubling Technique," ASL-TR-0020, November 1978.
92. Lindberg, James D., et al, "Measured Effects of Battlefield Dust and Smoke on Visible, Infrared, and Millimeter Wavelengths Propagation: A Preliminary Report on Dusty Infrared Test-I (DIRT-I)," ASL-TR-0021, January 1979.
93. Kennedy, Bruce W., Arthur Kinghorn, and B. R. Hixon, "Engineering Flight Tests of Range Meteorological Sounding System Radiosonde," ASL-TR-0022, February 1979.
94. Rubio, Roberto, and Don Hoock, "Microwave Effective Earth Radius Factor Variability at Wiesbaden and Balboa," ASL-TR-0023, February 1979.
95. Low, Richard D. H., "A Theoretical Investigation of Cloud/Fog Optical Properties and Their Spectral Correlations," ASL-TR-0024, February 1979.
96. Pinnick, R. G., and H. J. Auvermann, "Response Characteristics of Knollenberg Light-Scattering Aerosol Counters," ASL-TR-0025, February 1979.
97. Heaps, Melvin G., Robert O. Olsen, and Warren W. Berning, "Solar Eclipse 1979, Atmospheric Sciences Laboratory Program Overview," ASL-TR-0026, February 1979.
98. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' GR-8 Sound Ranging Data," ASL-TR-0027, March 1979.
99. Kennedy, Bruce W., and Jose M. Serna, "Meteorological Rocket Network System Reliability," ASL-TR-0028, March 1979.

100. Swingle, Donald M., "Effects of Arrival Time Errors in Weighted Range Equation Solutions for Linear Base Sound Ranging," ASL-TR-0029, April 1979.
101. Umstead, Robert K., Ricardo Pena, and Frank V. Hansen, "KWIK: An Algorithm for Calculating Munition Expenditures for Smoke Screening/Obscuration in Tactical Situations," ASL-TR-0030, April 1979.
102. D'Arcy, Edward M., "Accuracy Validation of the Modified Nike Hercules Radar," ASL-TR-0031, May 1979.
103. Rodriguez, Ruben, "Evaluation of the Passive Remote Crosswind Sensor," ASL-TR-0032, May 1979.
104. Barber, T. L., and R. Rodriguez, "Transit Time Lidar Measurement of Near-Surface Winds in the Atmosphere," ASL-TR-0033, May 1979.
105. Low, Richard D. H., Louis D. Duncan, and Y. Y. Roger R. Hsiao, "Microphysical and Optical Properties of California Coastal Fogs at Fort Ord," ASL-TR-0034, June 1979.
106. Rodriguez, Ruben, and William J. Vechione, "Evaluation of the Saturation Resistant Crosswind Sensor," ASL-TR-0035, July 1979.
107. Ohmstede, William D., "The Dynamics of Material Layers," ASL-TR-0036, July 1979.
108. Pinnick, R. G., S. G. Jennings, Petr Chylek, and H. J. Auvermann, "Relationships between IR Extinction Absorption, and Liquid Water Content of Fogs," ASL-TR-0037, August 1979.
109. Rodriguez, Ruben, and William J. Vechione, "Performance Evaluation of the Optical Crosswind Profiler," ASL-TR-0038, August 1979.
110. Miers, Bruce T., "Precipitation Estimation Using Satellite Data," ASL-TR-0039, September 1979.
111. Dickson, David H., and Charles M. Sonnenschein, "Helicopter Remote Wind Sensor System Description," ASL-TR-0040, September 1979.
112. Heaps, Melvin G., and Joseph M. Heimerl, "Validation of the Dairchem Code, I: Quiet Midlatitude Conditions," ASL-TR-0041, September 1979.
113. Bonner, Robert S., and William J. Lentz, "The Visioceilometer: A Portable Cloud Height and Visibility Indicator," ASL-TR-0042, October 1979.
114. Cohn, Stephen L., "The Role of Atmospheric Sulfates in Battlefield Obscurations," ASL-TR-0043, October 1979.

115. Fawbush, E. J., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF), White Sands Missile Range, New Mexico, Part I, 24 March to 8 April 1977," ASL-TR-0044, November 1979.
116. Barber, Ted L., "Short-Time Mass Variation in Natural Atmospheric Dust," ASL-TR-0045, November 1979.
117. Low, Richard D. H., "Fog Evolution in the Visible and Infrared Spectral Regions and its Meaning in Optical Modeling," ASL-TR-0046, December 1979.
118. Duncan, Louis D., et al, "The Electro-Optical Systems Atmospheric Effects Library, Volume I: Technical Documentation," ASL-TR-0047, December 1979.
119. Shirkey, R. C., et al, "Interim E-0 SAEL, Volume II, Users Manual," ASL-TR-0048, December 1979.
120. Kobayashi, H. K., "Atmospheric Effects on Millimeter Radio Waves," ASL-TR-0049, January 1980.
121. Seagraves, Mary Ann, and Louis D. Duncan, "An Analysis of Transmittances Measured Through Battlefield Dust Clouds," ASL-TR-0050, February 1980.
122. Dickson, David H., and Jon E. Ottesen, "Helicopter Remote Wind Sensor Flight Test," ASL-TR-0051, February 1980.
123. Pinnick, R. G., and S. G. Jennings, "Relationships Between Radiative Properties and Mass Content of Phosphoric Acid, HC, Petroleum Oil, and Sulfuric Acid Military Smokes," ASL-TR-0052, April 1980.
124. Hinds, B. D., and J. B. Gillespie, "Optical Characterization of Atmospheric Particulates on San Nicolas Island, California," ASL-TR-0053, April 1980.
125. Miers, Bruce T., "Precipitation Estimation for Military Hydrology," ASL-TR-0054, April 1980.
126. Stenmark, Ernest B., "Objective Quality Control of Artillery Computer Meteorological Messages," ASL-TR-0055, April 1980.
127. Duncan, Louis D., and Richard D. H. Low, "Bimodal Size Distribution Models for Fogs at Meppen, Germany," ASL-TR-0056, April 1980.
128. Olsen, Robert O., and Jagir S. Randhawa, "The Influence of Atmospheric Dynamics on Ozone and Temperature Structure," ASL-TR-0057, May 1980.

129. Kennedy, Bruce W., et al, "Dusty Infrared Test-II (DIRT-II) Program," ASL-TR-0058, May 1980.
130. Heaps, Melvin G., Robert O. Olsen, Warren Berning, John Cross, and Arthur Gilcrease, "1979 Solar Eclipse, Part I - Atmospheric Sciences Laboratory Field Program Summary," ASL-TR-0059, May 1980
131. Miller, Walter B., "User's Guide for Passive Target Acquisition Program Two (PTAP-2)," ASL-TR-0060, June 1980.
132. Holt, E. H., editor, "Atmospheric Data Requirements for Battlefield Obscuration Applications," ASL-TR-0061, June 1980.
133. Shirkey, Richard C., August Miller, George H. Goedecke, and Yugal Behl, "Single Scattering Code AGAUSX: Theory, Applications, Comparisons, and Listing," ASL-TR-0062, July 1980.
134. Sojka, Brian Z., and Kenneth O. White, "Evaluation of Specialized Photoacoustic Absorption Chambers for Near-Millimeter Wave (NMMW) Propagation Measurements," ASL-TR-0063, August 1980.
135. Bruce, Charles W., Young Paul Yee, and S. G. Jennings, "In Situ Measurement of the Ratio of Aerosol Absorption to Extinction Coefficient," ASL-TR-0064, August 1980.
136. Yee, Young Paul, Charles W. Bruce, and Ralph J. Brewer, "Gaseous/Particulate Absorption Studies at WSMR using Laser Sourced Spectrophones," ASL-TR-0065, June 1980.
137. Lindberg, James D., Radon B. Loveland, Melvin Heaps, James B. Gillespie, and Andrew F. Lewis, "Battlefield Dust and Atmospheric Characterization Measurements During West German Summertime Conditions in Support of Grafenwohr Tests," ASL-TR-0066, September 1980.
138. Vechione, W. J., "Evaluation of the Environmental Instruments, Incorporated Series 200 Dual Component Wind Set," ASL-TR-0067, September 1980.
139. Bruce, C. W., Y. P. Yee, B. D. Hinds, R. G. Pinnick, R. J. Brewer, and J. Minjares, "Initial Field Measurements of Atmospheric Absorption at 9 μ m to 11 μ m Wavelengths," ASL-TR-0068, October 1980.
140. Heaps, M. G., R. O. Olsen, K. D. Baker, D. A. Burt, L. C. Howlett, L. L. Jensen, E. F. Pound, and G. D. Allred, "1979 Solar Eclipse: Part II Initial Results for Ionization Sources, Electron Density, and Minor Neutral Constituents," ASL-TR-0069, October 1980.
141. Low, Richard D. H., "One-Dimensional Cloud Microphysical Models for Central Europe and their Optical Properties," ASL-TR-0070, October 1980.

142. Duncan, Louis D., James D. Lindberg, and Radon B. Loveland, "An Empirical Model of the Vertical Structure of German Fogs," ASL-TR-0071, November 1980.
143. Duncan, Louis D., "EOSAEL 80, Volume I, Technical Documentation," ASL-TR-0072, January 1981.
144. Shirkey, R. C., and S. G. O'Brien, "EOSAEL 80, Volume II, Users Manual," ASL-TR-0073, January 1981.
145. Bruce, C. W., "Characterization of Aerosol Nonlinear Effects on a High-Power CO₂ Laser Beam," ASL-TR-0074, February 1981.
146. Duncan, Louis D., and James D. Lindberg, "Air Mass Considerations in Fog Optical Modeling," ASL-TR-0075, February 1981.
147. Kunkel, Kenneth E., "Evaluation of a Tethered Kite Anemometer," ASL-TR-0076, February 1981.
148. Kunkel, K. E., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF) White Sands Missile Range, New Mexico, August 1977 to October 1978, Part II, Optical Turbulence, Wind, Water Vapor Pressure, Temperature," ASL-TR-0077, February 1981.
149. Miers, Bruce T., "Weather Scenarios for Central Germany," ASL-TR-0078, February 1981.
150. Cogan, James L., "Sensitivity Analysis of a Mesoscale Moisture Model," ASL-TR-0079, March 1981.
151. Brewer, R. J., C. W. Bruce, and J. L. Mater, "Optoacoustic Spectroscopy of C₂H₂ at the 9 μ m and 10 μ m C¹⁸O₂ Laser Wavelengths," ASL-TR-0080, March 1981.
152. Swingle, Donald M., "Reducible Errors in the Artillery Sound Ranging Solution, Part I: The Curvature Correction" (U), SECRET, ASL-TR-0081, April 1981.
153. Miller, Walter B., "The Existence and Implications of a Fundamental System of Linear Equations in Sound Ranging" (U), SECRET, ASL-TR-0082, April 1981.
154. Bruce, Dorothy, Charles W. Bruce, and Young Paul Yee, "Experimentally Determined Relationship Between Extinction and Liquid Water Content," ASL-TR-0083, April 1981.
155. Seagraves, Mary Ann, "Visible and Infrared Obscuration Effects of Ice Fog," ASL-TR-0084, May 1981.

156. Watkins, Wendell R., and Kenneth O. White, "Wedge Absorption Remote Sensor," ASL-TR-0085, May 1981.
157. Watkins, Wendell R., Kenneth O. White, and Laura J. Crow, "Turbulence Effects on Open Air Multipaths," ASL-TR-0086, May 1981.
158. Blanco, Abel J., "Extending Application of the Artillery Computer Meteorological Message," ASL-TR-0087, May 1981.
159. Heaps, M. G., D. W. Hooch, R. O. Olsen, B. F. Engebos, and R. Rubio, "High Frequency Position Location: An Assessment of Limitations and Potential Improvements," ASL-TR-0088, May 1981.
160. Watkins, Wendell R., and Kenneth O. White, "Laboratory Facility for Measurement of Hot Gaseous Plume Radiative Transfer," ASL-TR-0089, June 1981.
161. Heaps, M. G., "Dust Cloud Models: Sensitivity of Calculated Transmittances to Variations in Input Parameters," ASL-TR-0090, June 1981.
162. Seagraves, Mary Ann, "Some Optical Properties of Blowing Snow," ASL-TR-0091, June 1981.
163. Kobayashi, Herbert K., "Effect of Hail, Snow, and Melting Hydrometeors on Millimeter Radio Waves," ASL-TR-0092, July 1981.
164. Cogan, James L., "Techniques for the Computation of Wind, Ceiling, and Extinction Coefficient Using Currently Acquired RPV Data," ASL-TR-0093, July 1981.
165. Miller, Walter B., and Bernard F. Engebos, "On the Possibility of Improved Estimates for Effective Wind and Temperature," (U), SECRET, ASL-TR-0094, August 1981.
166. Heaps, Melvin G., "The Effect of Ionospheric Variability on the Accuracy of High Frequency Position Location," ASL-TR-0095, August 1981.
167. Sutherland, Robert A., Donald W. Hooch, and Richard B. Gomez, "An Objective Summary of US Army Electro-Optical Modeling and Field Testing in an Obscuring Environment," ASL-TR-0096, October 1981.
168. Pinnick, R. G., et al, "Backscatter and Extinction in Water Clouds," ASL-TR-0097, October 1981.
169. Cole, Henry P., and Melvin G. Heaps, "Properties of Dust as an Electron and Ion Attachment Site for Use in D Region Ion Chemistry," ASL-TR-0098, October 1981.

170. Spellicy, Robert L., Laura J. Crow, and Kenneth O. White, "Water Vapor Absorption Coefficients at HF Laser Wavelengths Part II: Development of the Measurement System and Measurements at Simulated Altitudes to 10 KM," ASL-TR-0099, November 1981.
171. Cohn, Stephen L., "Transport and Diffusion Solutions for Obscuration Using the XM-825 Smoke Munition," ASL-TR-0100, November 1981.
172. Pinnick, R. G., D. M. Garvey, and L. D. Duncan, "Calibration of Knollenberg FSSP Light-Scattering Counters for Measurement of Cloud Droplets," ASL-TR-0101, December 1981.
173. Cohn, Stephen L. and Ricardo Pena, "Munition Expenditure Model Verification: KWIK Phase I," ASL-TR-0102, December 1981.
174. Blackman, George R., "Cloud Geometry Analysis of the Smoke Week III Obscuration Trials," ASL-TR-0103, January 1982.
175. Sutherland, R. A., and D. W. Hooch, "An Improved Smoke Obscuration Model ACT II: Part 1 Theory," ASL-TR-0104, January 1982.
176. Lentz, W. J., "The Visioceilometer: A Portable Visibility and Cloud Ceiling Height Lidar," ASL-TR-0105, January 1982.
177. Kennedy, Bruce W., "An Analysis of Craters Produced by Artillery Munitions During Dusty Infrared Test Series," ASL-TR-0106, January 1982.
178. Steinhoff, R. G., "Program Listings for EOSAEL 80-B and Ancillary Codes AGAUS and FLASH," ASL-TR-0107, February 1982.

DATE
ILME