CR115522

STANFORD RSL Technical Report 71-2

The National Aeronautics and Space Administration
Research Contract NAS 9-7313

FINAL REPORT (A) -- PHASE IV

SOFTWARE (COMPUTER PROGRAMMING)

1970/71 STANFORD SPECTRAL DATA MANAGEMENT PROGRAMS

BY

A. A. Marshall
Stanford Remote Sensing Laboratory

PRIME OFFICE OF RESPONSIBILITY

TF

OFFICE OF PRIME RESPONSIBILITY

TF

Report Prepared Under
NASA Contract NAS 9-7313
"Infrared Spectrometry Studies"

REMOTE SENSING LABORATORY
SCHOOL OF EARTH SCIENCES
STANFORD UNIVERSITY • STANFORD CALIFORNIA

CAT. 13

STANFORD RSL Technical Report 71-2

The National Aeronautics and Space Administration
Research Contract NAS9-7313

FINAL REPORT (A) -- PHASE IV

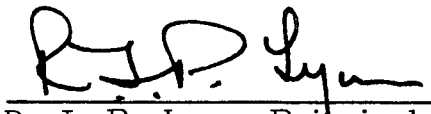
SOFTWARE (COMPUTER PROGRAMMING)

1970/71 STANFORD SPECTRAL DATA MANAGEMENT PROGRAMS

by

A. A. Marshall
Stanford Remote Sensing Laboratory

Report Prepared Under
NASA Contract NAS9-7313
"Infrared Spectrometry Studies"

Approved: 

R. J. P. Lyon, Principal Investigator
Department of Mineral Engineering
Stanford University
Stanford, California 94305

June 1971

STANFORD RSL TECHNICAL REPORT SERIES

67-1 "Field Infrared Analysis of Terrain-Spectral Correlation Program" Part I (by R.J.P. Lyon)

67-2 "Field Infrared Analysis of Terrain-Spectral Correlation Program" Part II and Part III (by R.J.P. Lyon)

67-3 "Statistical Analysis of IR Spectra--Stanford Programs Applied to USGS Spectra in Tech Letter #13" (by R.J.P. Lyon)

67-4 "Computer Reduction and Analysis of an Infrared Image" (by Keenan Lee)

68-1 "Infrared Exploration for Coastal and Shoreline Springs" (by Keenan Lee)

68-2 "Re-evaluation of the Normative Minerals of Sonora Pass Rock Standards-University of Nevada Reports #7 and #12" (by Attila Kilinc)

68-3 "Nearest Neighbor--A New Non-parametric Test Used for Classifying Spectral Data" (by Paul Switzer)

68-4 Final Report "Field Analysis of Terrain"--NGR-05-020-115 (contains all the drawings and logic diagrams for Stanford CVF Spectrometer and Digital Data Recording System (by R.J.P. Lyon)

69-1 "Mission 78--Flights 1 and 2 Ninety-Day Report" (by R.J.P. Lyon)

69-2 "Quantitative Geological Analysis of Multiband Photography from the Mono Craters Area, California" (by G.I. Ballew)

69-3 "The Stanford Infrared Spectra Processing Package" (by John Moore with subsequent modifications by Michael Heathman)

69-4 "Thermal Dynamics at the Earth-Air Interface: The Implications for Remote Sensing of the Geologic Environment" (by Rossman Smith)

69-5 "Infrared Spectrometry Studies: Final Report Phase I" (by R.J.P. Lyon)

69-6 "Infrared Spectrometry Studies: Final Report Phase II" (by R. J. P. Lyon and Attila Kilinc)

69-7 "Infrared Exploration for Shoreline Springs at Mono Lake, California, Test Site" (by Keenan Lee)

70-1 "Detailed Ground Study of 8-13 μm Infrared Imagery, Carrizo Plains, California" (by R. Campbell, Jr., L. D. Hoover and F. Querol-Suñé)

70-2 "Terms in Geologic Remote Sensing: A Condensed Glossary" (by R. J. P. Lyon)

70-3 "Geologic Interpretation of Airborne Infrared Thermal Imagery of Goldfield, Nevada" (by I. A. Kilinc and R. J. P. Lyon)

70-4 "Surface Heat Flow Studies for Remote Sensing of Geothermal Resources" (by H. Hase, Geological Survey of Japan)

70-5 "Geophysical Analysis of Thermal Infrared Imagery near Goldfield Nevada: Experimental and Correlation Analysis" (by W. B. Ervine)

70-6 "The Multiband Approach to Geologic Mapping from Orbiting Satellites: Is it Redundant or Vital?" (by R. J. P. Lyon), now published in Remote Sensing of Environment, 1, (4), 237-244

70-7 "Airborne Geological Mapping Using Infrared Emission Spectra" (by R. J. P. Lyon and J. Patterson), now published in Proc. of the 6th Symposium on Remote Sensing of Environment, 1, 527-552

70-8 "Pseudo-Radar: Very High Contrast Aerial Photography at Low Sun Angles" (by R. J. P. Lyon, Jose Mercado and Robert Campbell, Jr.), now published in Photogrammetric Engineering, 36, (12), 1257-1261

70-9 "Remote Sensing in Exploration for Mineral Deposits" (by R. J. P. Lyon and Keenan Lee), now published in Economic Geology, 65, 785-800

70-10 "Phenomena and Properties of Geologic Materials Affecting Microwaves - A Review" (by D. Oberste-Lehn)

70-11 "1969/70 Stanford Spectral Data Management System" (by Michael Heathman)

71-1 "Operational Calibration of an Airborne Infrared Spectrometer Over Geologically-Significant Terrains", (by R. J. P. Lyon and A. A. Marshall)

71-2 "1970/71 Stanford Spectral Data Management Programs" (by A. A. Marshall) Final Report (A) -- Phase IV (Software - Computer Programming).

71-3 "Stanford Digital Data System." Final Report (B) -- Phase IV.

71-4 "Comparison of Airborne Infrared Spectral Emittance and Radar Scatterometer Data from Pisgah Crater Lava Flows," (Abstr.) Paper presented at 7th Int. Symp. on Rem. Sens. of Environ., Ann Arbor, Mich., May 17, 1971.

71-5 "Infrared Spectral Emittance in Geological Mapping: Airborne Spectrometer Data from Pisgah Crater, California." Paper submitted to Science, August 1971. pp. 14.

71-6 "Spectral Data from Flights 1 and 3, Mission 108." Final Report (C) -- Phase IV (IR Spectral Emittance Data - Airborne).

I. Introduction

This is a report on the data management programs used by the Stanford Remote Sensing Laboratory to access, modify, and reduce the data obtained from both the NASA IR airborne spectrometer, and Stanford's SG-4 field spectrometer. Many details covered in previous reports are not repeated here. References are provided below.

These programs are written in Fortran IV and S/360 Assembler Language, and are currently running on a S/360 model 67 (operating under OS/MFT) at the Stanford Computation Center Campus Facility.

References

1) "1969/70 Stanford Spectral Data Management System", RSL Tech #70-11 by Michael Heathman.

2) "The Stanford Infrared Spectra Processing Package", RSL Tech #69-3 by John R. Moore.

3) "Mission 78 - Flights 1 and 2 Ninety Day Report", RSL Tech #69-1 by R.J.P. Lyon and Attila Kilinc.

II. Program Descriptions

1. Program Cal

Cal computes instrument calibration functions using NASA spectra. The required function is computed for each member of a group of spectra, and the mean and standard deviation over the group are printed and plotted. Currently the functions computed by Cal are not used for any further processing within the system.

If the option INSTRANS is specified, the instrument response correction function is computed. This function may be used to correct for the non linear response of the spectrometer. It is computed by ratioing a theoretical blackbody spectrum to an observed blackbody spectrum. Since the spectrometer measures the radiation difference between the outside world and an internal reference, the theoretical blackbody mentioned above is the difference between two absolute radiance curves, one calculated using the target temperature, and the other calculated using the internal reference temperature.

If the option AIRPATH is specified, the airpath absorbion function is computed. Airborne blackbody spectra (from lakes, oceans etc.) are corrected for the instrument response and for the reflectance of water. The ratio of these corrected spectra to a theoretical blackbody spectrum is the airpath absorbsion function. This function descibes the effect of the air mass on the radiance levels seen by the spectrometer.

If the option EMITT is specified, the ground rock emittance spectrum is computed. Each ground rock spectrum is corrected for instrument response and subtracted from a theoretical blackbody spectrum calculated at the internal spectrometer reference temperature. This gives an estimate of the absolute radiance of the target. The ratio of this to an absolute blackbody radiance curve gives the emittance spectrum for the rock.

Cal uses Splot for line printer plotting, Irrad for theoretical radiance calculations, Tcalc to estimate target temperature if unknown, and Sigma to compute standard deviations.

2. Program Prep

Prep is used to access and save small groups of spectra within the NASA data base. The spectra are time coded in increasing order. The program reads sequentially though the data base until the group is found. Any spectra within the group whose temperature variance is above a given tolerance is rejected. The spectra alternate between up ramp (6.8-13.4 microns) and down ramp (13.4-6.8 microns) recording, but the output file contains only spectra of a given ramp code. The group average spectrum and standard deviation is printed and plotted for each group processed.

Prep uses Splot for plotting, Table for data listing, Xlate to convert time, Rdnasa to read the data base, Unpack to unpack identification bytes, and Dater to provide the date and time for the printed output.

3. Program Proc

Proc is used to process spectral groups produced by Prep. Since the data saved by Prep is contained in individual datasets, Proc finds groups by dataset name alone. The standard processing steps are as follows: the raw spectra are ratioed to a blackbody spectrum; the tails of the spectra are clipped since they contain little useful information; the ratioed spectra are smoothed to minimize the effects of random noise; and finally they are each normalized so the mean "radiance" of each spectrum is zero, with a standard deviation of one, allowing valid comparison of spectra with different mean intensities.

The processed spectra are output onto a single file in card image format so that they may be read by classification programs such as BMD07M. The group average spectrum for each group is saved on a separate file so that the individuals may be further processed by program Discard.

Proc uses Splot for plotting, Table for data listing, Dater for date and time, Norm for normalizing spectra, and Sm for smoothing spectra.

4. Program Discard

Discard is used to delete from spectral groups spectra which vary greatly from the group mean. The program reads the output produced by Proc and computes for each member of a group the distance in Euclidean space from the group mean. If this distance is greater than a given tolerance, the spectrum is deleted from the group. There is no firm reason to think the information about a group is any better after this processing, but it has been found that spectra rejected by this method correspond well with the spectra which the classification programs cannot identify correctly.

Discard uses no subroutines.

5. Program Trkload

Trkload is used to copy ground based ("truck") spectra tapes to disk. The organization of the disk file is different from the NASA data base in that individual spectra may be accessed directly. An index with pointers to the raw spectra is created which may be searched by later programs in order to find spectral groups. A program which does this searching and saves the groups in a format compatible with Proc has not been written yet, since the ground system is not fully operational.

Trkload uses Rdtrk to read and convert the raw data tapes, and Daload to create the direct access file. Daload is used to bypass the formatting of direct access files which the FORTRAN direct access routines must do.

III. Program Examples

1. Program Cal

```
//CAL JOB (J032,332,,10),MARSHALL
//JOBLIB DD DSN=J032.PROGLIB,DISP=SHR
//WHYNOT EXEC PGM=CAL
//FT20F001 DD DSN=J032.PRE40,DISP=SHR
//FT30F001 DD DSN=J032.SHALL,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
108-1
INSTRANS 60.     40.     20     MX108-1 PREFLIGHT BB
AIRPATH  60.     0.0     30     MX108-1 SHALLOW LAKE
/*
```

In this example, the dataset J032.PRE40 is used to calculate the instrument response correction function, and the dataset J032.SHALL is used to calculate the airpath absorbsion function.

2. Program Prep

```
//PREP JOB (J032,332,,10),MARSHALL
//JOBLIB DD DSN=J032.PROGLIB,DISP=SHR
//WHYNOT EXEC PGM=PREP
//NASA DD DSN=J032.FLIGHT1,DISP=SHR,DCB=OPTCD=C
//FT20F001 DD DSN=J032.ROCKA,VOL=SER=USER07,UNIT=2314,
//  DISP=(,CATLG),SPACE=(TRK,5,RLSE),DCB=(RECFM=VBS,
//  BLKSIZE=7294,LRECL=400)
//FT30F001 DD DSN=J032.ROCKB,VOL=SER=USER07,UNIT=2314,
//  DISP=(,CATLG),SPACE=(TRK,5,RLSE),DCB=*.FT20F001
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
 &PARMS TEMP=150., &END
00 20 15 15 12345 15 15 23456 MX108-1 ROCKA
00 30 15 16 12345 15 15 23456 MX108-1 ROCKB
/*
```

In this example, J032.FLIGHT1 contains raw time-coded spectra from which the datasets J032.ROCKA and J032.ROCKB are created. The data cards contain the ramp code, logical unit number for output, start and stop times, and some identication for each group of spectra.

3. Program Proc

```
//PROC JOB (J032,332,,10),MARSHALL
//JOBLIB DD DSN=J032.PROGLIB,DISP=SHR
//WHYNOT EXEC PGM=PROC
//FT99F001 DD DSN=J032.SHALLAVG,DISP=SHR
//FT20F001 DD DSN=J032.ROCKA,DISP=SHR
//FT30F001 DD DSN=J032.ROCKB,DISP=SHR
//FT07F001 DD DSN=J032.ROCKLIBI,DISP=MOD
//FT08F001 DD DSN=J032.ROCKLIBA,DISP=MOD
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
 &PARMS SMOOTH=T, CARD=5, &END
THESE SPECTRA HAVE BEEN RATIOED, SMOOTHED, AND NORMALIZED.
00 20 15 15 12345 15 15 23456 MX108-1 ROCKA
00 30 15 16 12345 15 16 23456 MX108-1 ROCKB
/*
```

In this example, the two datasets created in the previous example are processed and saved in J032.ROCKLIBI (for individual spectra) and in J032.ROCKLIBA (for the average of each group). Each spectrum is ratioed to an averaged blackbody spectrum contained in J032.SHALLAVG, smoothed, and normalized. Note that the control cards are the same as those used above.

4. Program Discard

```
//RSL JOB (J032,332,,10),MARSHALL
//JOBLIB DD DSN=J032.PROGLIB,DISP=SHR
//WHYNOT EXEC PGM=DISCARD
//FT10F001 DD DSN=J032.ROCKLIBD,DISP=MOD
//FT03F001 DD DSN=J032.ROCKLIBI,DISP=SHR
//FT04F001 DD DSN=J032.ROCKLIBA,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
 &PARMS LIMIT=30, &END
 &PARMS LIMIT=30, &END
          .
          .
          .
 &PARMS LIMIT=30, &END
```

In this example, data read from J032.ROCKLIBI are copied to J032.ROCKLIBD rejecting any spectrum whose distance from the group mean is greater than a given tolerance. The group means are contained in J032.ROCKLIBA, and the tolerance for each group is specified using the &PARMS namelist.

5. Program Trkload

```
//RSL JOB (J032,332,,10),MARSHALL
//JOBLIB DD DSN=J032.PROGLIB,DISP=SHR
//WHYNOT EXEC PGM=TRKLOAD
//DIRECT DD DSN=J032.TROCKS,VOL=SER=USER07,UNIT=2314,
//   SPACE=(CYL,10,RLSE),DISP=(,CATLG),
//   DCB=(DSORG=DA,KEYLEN=0,BLKSIZE=204)
//SPECTAPE DD UNIT=0C0,VOL=SER=TRUCK,LABEL=(,BLP),DISP=SHR
//FT10F001 DD DSN=J032.TINDEX,VOL=SER=USER07,UNIT=2314,
//   SPACE=(TRK,10,RLSE),DISP=(,CATLG),
//   DCB=(RECFM=FB,LRECL=40,BLKSIZE=3520)
//FT04F001 DD SYSOUT=A,DCB=(RECFM=FA,BLKSIZE=133)
//FT06F001 DD SYSOUT=A,DCB=*.FT04F001
//FT05F001 DD *
 &PARMS LIST=T, TERR=T, &END
```

In this example, a truck tape called TRUCK, is copied to disk. The spectra are saved in J032.TROCKS, and the identification information is saved in J032.TINDEX. The namelist input specifies that the individual spectra are to be listed, and that rereads should be supressed in case of an error while reading the tape.

IV. Program Listings

1.1 Program Cal

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C            PROGRAM CAL -- DECEMBER 1970 VERSION                      C
C                                                                      C
C                                                                      C
C    COMPUTES INSTRUMENT CALIBRATION FOR IR SPECTROMETER               C
C            STANFORD REMOTE SENSING LABORATORY                        C
C                                                                      C
C                  PROGRAM  OPTIONS                                    C
C                                                                      C
C    INSTRANS -- COMPUTE INSTRUMENT RESPONSE CORRECTION FUNCTION.      C
C                                                                      C
C    AIRPATH  -- COMPUTE AIRPATH ABSORBSION SPECTRUM FOR AIRBORNE      C
C                BLACKBODY SPECTRA.                                    C
C                                                                      C
C    EMITT    -- COMPUTE TARGET RADIANCE AND EMITTANCE SPECTRUM        C
C                FOR KNOWN ROCK TYPES.                                 C
C                                                                      C
C    SAVETRAN -- READ/WRITE INSTRUMENT RESPONSE FUNCTION.              C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      INTEGER DISK, CARD/5/, PRINT/6/, NMAX/30/
      INTEGER INTRN/'INST'/,EMITT/'EMIT'/,AIRPT/'AIRP'/,SAVET/'SAVE'/
      INTEGER GET/'GET'/, PUT /'PUT'/, FLT, NAME(8), CNT(88)
      REAL  RAW (88), DSK (92), IRAD(88), PLNK(88), ASP (88),
     *      AINS(88), SINS(88), AAIR(88), SAIR(88), ARAD(88),
     *      AEM (88), SEM (88), SSP (88), SRAD(88)
      REAL  ZERO(88) /88*0.0/, FACT/0.0/
C
C     DEFINE REFLECTANCE CORRECTION FUNCTION
      REAL REFLT(88)
     * / .9797,  .9800,  .9805,  .9808,  .9812,  .9818,  .9819,
     *   .9820,  .9821,  .9821,  .9822,  .9822,  .9823,  .9823,
     *   .9824,  .9826,  .9828,  .9830,  .9831,  .9833,  .9834,
     *   .9836,  .9837,  .9839,  .9841,  .9843,  .9845,  .9847,
     *   .9848,  .9850,  .9853,  .9855,  .9858,  .9861,  .9863,
     *   .9865,  .9869,  .9873,  .9877,  .9881,  .9885,  .9888,
     *   .9891,  .9895,  .9899,  .9903,  .9907,  .9910,  .9913,
     *   .9917,  .9920,  .9923,  .9926,  .9929,  .9930,  .9932,
     *   .9933,  .9932,  .9931,  .9930,  .9926,  .9922,  .9918,
     *   .9912,  .9906,  .9900,  .9897,  .9893,  .9890,  .9887,
     *   .9984,  .9980,  .9974,  .9968,  .9962,  .9857,  .9851,
     *   .9844,  .9835,  .9824,  .9810,  .9794,  .9778,  .9762,
     *   .9748,  .9732,  .9715,  .9702
     * /
```

```
C     IGNORE 4 WORD HEADER ON DATA READS.
      EQUIVALENCE (RAW(1), DSK(5))
C
C
C     READ MISSION AND FLIGHT IDENTIFICATION.
      READ (CARD,54) MISS, FLT
C
C     INITIALIZE SEQUENCE NUMBER ARRAY
      DO 2 I = 91,178
2     CNT(I-90) = I
C
C     READ PROGRAM OPTIONS
10    READ (CARD,51,END=99) IOPT, REFT, BBT, DISK, NAME
C
      IF (IOPT .EQ. SAVET) GOTO 50
C
C     COMPUTE TARGET TEMPERATURE IF UNSPECIFIED.
C
      IF (BBT .NE. 0.0) GOTO 11
      IF (FACT .NE. 0.0) GOTO 12
C
C     ERROR IN AUTOMATIC TEMPERATURE CALCULATION
      WRITE (PRINT,97)
      STOP
C
12    BBT = TCALC (REFT, FACT, DISK)
C
C     BRANCH TO SPECIFIED ROUTINE
11    IF (IOPT .EQ. AIRPT) GOTO 30
      IF (IOPT .EQ. EMITT) GOTO 40
      IF (IOPT .EQ. INTRN) GOTO 20
C
C     ERROR IN OPTION CODE, STOP.
      WRITE (PRINT,61)
      STOP
C
C     READ/WRITE RESPONSE FUNCTION
50    IF (NAME(1) .EQ. PUT) WRITE (DISK) AINS
      IF (NAME(1) .EQ. GET) READ (DISK) AINS
      GOTO 10
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C            INSTRUMENT RESPONSE CORRECTION FUNCTION                    C
C                                                                       C
C                                                                       C
C     THE RESPONSE FUNCTION IS THE RATIO OF A CALCULATED                C
C     IRRADIANCE CURVE TO AN OBSERVED BLACK BODY SPECTRUM               C
C     AT A GIVEN TEMPERATURE.  IRRADIANCE IS THE DIFFERENCE             C
C     BETEEN TWO BLACKBODY RADIATORS, ONE AT THE INTERNAL               C
C     SPECTROMETER REFERENCE TEMPERATURE, AND ONE AT THE EXTERNAL       C
C     TARGET TEMPERATURE.                                               C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
20     CONTINUE
C
C      COMPUTE THEORETICAL NET IRRADIANCE
       CALL IRRAD (IRAD, REFT, BBT)
C
C      READ IN GROUND BLACKBODY SPECTRA AND COMPUTE RESPONSE OF EACH
C
       DO 26 I = 1,88
       ASP(I) = 0.0
       SSP(I) = 0.0
       AINS(I) = 0.0
       AINS(I) = 0.0
       SINS(I) = 0.0
26     CONTINUE
C
       DO 22 I = 1,NMAX
       READ (DISK,END=23) DSK
C
       DO 22 J = 1,88
       ASP(J) = ASP(J) + RAW(J)
       SSP(J) = SSP(J) + RAW(J) ** 2
       AINS(J) = AINS(J) + IRAD(J)/RAW(J)
       SINS(J) = SINS(J) + (IRAD(J)/RAW(J)) ** 2
22     CONTINUE
C
C      FIND STANDARD DEVIATIONS
C
23     NSPEC = I - 1
       EN = NSPEC
       CALL SIGMA (ASP, SSP, EN, 88)
       CALL SIGMA (AINS, SINS, EN, 88)
```

```
C     FIND OVERALL AVERAGES
C
      AIRAD = AVER (IRAD, 88)
      AASP = AVER (ASP, 88)
      ASSP = AVER (SSP, 88)
      AAINS = AVER (AINS, 88)
      ASINS = AVER (SINS, 88)
C
C
C     PRINT AND PLOT RESULTS
C
      WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
      WRITE (PRINT,67) (CNT(I), IRAD(I), ASP(I),
     *                  SSP(I), AINS(I), SINS(I), I = 1,88)
      WRITE (PRINT,66) AIRAD, AASP, ASSP, AAINS, ASINS
C
      WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
      CALL SPLOT (IRAD, ZERO, 0.0, 0.0, PRINT, 88, 91)
      WRITE (PRINT,62)
C
      WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
      CALL SPLOT (ASP, SSP, 0.0, 0.0, PRINT, 88, 91)
      WRITE (PRINT,63)
C
      WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
      CALL SPLOT (AINS, SINS, 0.0, 0.0, PRINT, 88, 91)
      WRITE (PRINT,64)
C
C
C     COMPUTE TEMPERATURE CONVERSION TABLE BASED ON LINEAR
C     INTERPOLATION THROUGH (0,REFT) AND (AASP,BBT)
C
      FACT = (BBT - REFT) / AASP
      RR = 0.0
      WRITE (PRINT,93) MISS, FLT, REFT, BBT, NSPEC, NAME
C
      DO 28 I = 1,1101,25
      TT = REFT + FACT * RR
      WRITE (PRINT,94) RR, TT
      RR = RR + 25.0
28    CONTINUE
C
      GOTO 10
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                    AIRPATH ABSORBSION FUNCTION                       C
C                                                                      C
C                                                                      C
C       AIRPATH ABSORBSION IS THE RATIO OF THE THEORETICAL             C
C       NET IRRADANCE OF A BLACKBODY AT A GIVEN TEMPERATURE TO         C
C       AN AIRBORNE BLACKBODY (IE LAKES, OCEAN) AT THE SAME            C
C       TEMPERATURE, MULTIPLIED BY THE INSTRUMENT RESPONSE             C
C       CORRECTION FUNCTION.                                           C
C       MOD -- WATER REFLECTANCE CORRECTION ADDED.                     C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
30     CONTINUE
C
C      COMPUTE IRRADIANCE
       CALL IRRAD (IRAD, REFT, BBT)
C
C      READ IN AIRBORNE BLACKBODY SPECTRA AND FIND AIRPATH OF EACH
C
       DO 31 I = 1,88
       ASP (I) = 0.0
       SSP (I) = 0.0
       AAIR(I) = 0.0
       SAIR(I) = 0.0
31     CONTINUE
C
       DO 33 I = 1,NMAX
       READ(DISK,END=38) DSK
C
       DO 33 J = 1,88
       ASP(J) =ASP(J) + RAW(J)
       SSP(J) = SSP(J) + RAW(J) ** 2
       T = IRAD(J) * REFLT(J) / RAW(J) / AINS(J)
       AAIR(J) = AAIR(J) + T
       SAIR(J) = SAIR(J) + T*T
33     CONTINUE
```

```
C       FIND STANDARD DEVIATIONS
C
38      NSPEC = I - 1
        EN = NSPEC
        CALL SIGMA (ASP, SSP, EN, 88)
        CALL SIGMA (AAIR, SAIR, EN, 88)
C
C       FIND OVERALL AVERAGES
C
        AASP = AVER (ASP, 88)
        ASSP = AVER (SSP, 88)
        AAAIR = AVER (AAIR, 88)
        ASAIR = AVER (SAIR, 88)
C
C
C       PRINT AND PLOT RESULTS
C
        WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
        WRITE (PRINT,65) (CNT(I), ASP(I),
       *                  SSP(I), AAIR(I), SAIR(I), I = 1,88)
        WRITE (PRINT,66) AASP, ASSP, AAAIR, ASAIR
C
        WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
        CALL SPLOT (ASP, SSP, 0.0, 0.0, PRINT, 88, 91)
        WRITE (PRINT,86)
C
        WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
        CALL SPLOT (AAIR, SAIR, 0.0, 0.0, PRINT, 88, 91)
        WRITE (PRINT,87)
C
        GOTO 10
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                  GROUND ROCK EMITTANCE SPECTRUM                      C
C                                                                      C
C                                                                      C
C       EMITTANCE IS FOUND BY RATIOING THE TARGET RADIANCE TO          C
C       A CALCULATED BLACKBODY RADIATOR AT THE SAME TEMPERATURE.       C
C       THE TARGET RADIANCE IS FOUND BY SUBTRACTING FROM THE           C
C       INTERNAL REFERENCE SPECTRUM AN OBSERVED ROCK SPECTRUM          C
C       MULTIPLIED BY THE INSTRUMENT RESPONSE FUNCTION.                C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
40     CONTINUE
C      COMPUTE ABSOLUTE INTERNAL RADIANCE
       CALL ABSL (PLNK, REFT)
C
C      COMPUTE RADIANCE FOR BLACKBODY AT TARGET TEMPERATURE
       CALL ABSL (IRAD, BBT)
C
C      READ IN SPECTRA AND COMPUTE TARGET RADIANCE
C
       DO 41 I = 1,88
       ASP (I) = 0.0
       SSP (I) = 0.0
       ARAD(I) = 0.0
       SRAD(I) = 0.0
       AEM (I) = 0.0
       SEM (I) = 0.0
41     CONTINUE
C
       DO 42 I = 1,NMAX
       READ (DISK,END=45) DSK
C
       DO 42 J = 1,88
       SSP(J) = SSP(J) + RAW(J) ** 2
       ASP(J) =ASP(J) + RAW(J)
C
C      COMPUTE TARGET RADIANCE (T)
       T = PLNK(J) - RAW(J) * AINS(J)
       ARAD(J) = ARAD(J) + T
       SRAD(J) = SRAD(J) + T*T
```

```
C       COMPUTE EMITTANCE (TT)
        TT = T/IRAD(J)
        AEM(J) = AEM(J) + TT
        SEM(J) = SEM(J) + TT ** 2
42      CONTINUE
C
45      NSPEC = I - 1
        EN = NSPEC
C
C       FIND STANDARD DEVIATIONS
C
        CALL SIGMA (ASP, SSP, EN, 88)
        CALL SIGMA (ARAD, SRAD, EN, 88)
        CALL SIGMA (AEM, SEM, EN, 88)
C
C       FIND OVERALL AVERAGES
C
        AASP = AVER (ASP, 88)
        ASSP = AVER (SSP, 88)
        AARAD = AVER (ARAD, 88)
        ASRAD = AVER (SRAD, 88)
        ASEM = AVER (SEM, 88)
        AAEM = AVER (AEM, 88)
C
C       PRINT AND PLOT RESULTS
C
        WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
        WRITE (PRINT,95) (CNT(I), ASP(I), SSP(I),
       *            ARAD(I), SRAD(I), AEM(I), SEM(I), I = 1,88)
        WRITE (PRINT,96) AASP, ASSP, AARAD, ASRAD, AAEM, ASEM
C
        WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
        CALL SPLOT (ASP, SSP, 0.0, 0.0, PRINT, 88, 91)
        WRITE (PRINT,81)
C
        WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
        CALL SPLOT (ARAD, SRAD, 0.0, 0.0, PRINT, 88, 91)
        WRITE (PRINT,82)
C
        WRITE (PRINT,68) MISS, FLT, REFT, BBT, NSPEC, NAME
        CALL SPLOT (AEM, SEM, 0.0, 0.0, PRINT, 88, 91)
        WRITE (PRINT,83)
C
        GOTO 10
```

```
C     END OF FILE READ
99    WRITE (PRINT,69)
      STOP
C
C
51    FORMAT(A4,T10,2F10.5,I2,T40,8A4)
54    FORMAT(I3,1X,I1)
55    FORMAT(A4,I2)
61    FORMAT(///' RSL0501  OPTION CODE INVALID')
62    FORMAT(//T54,'THEORETICAL NET IRRADIANCE')
63    FORMAT(//T50,'AVERAGED GROUND BLACKBODY SPECTRUM')
64    FORMAT(//T50,'INSTRUMENT RESPONSE CORRECTION FUNCTION')
65    FORMAT( T20,'AVER SPECTRUM',T40,'STANDARD DEV',T60,'AVER ',
     *   'AIRPATH',T80,'STANDARD DEV'//(I10,4E20.3))
66    FORMAT(//T11,6E20.3)
67    FORMAT(T23,'DIFFRAD',T40,'AVER SPECTRUM',T60,'STANDARD DEV',
     *   T80,'INSTRANS',T100,'STANDARD DEV'//(I10,5E20.3))
68    FORMAT('1',T10,'MISSION',I4,' FLIGHT ',I1,' CALIBRATION.'//
     *   T10,'INTERNAL REFERENCE TEMPERATURE IS',F4.0,' DEGREES '
     *   ,'CENTIGRADE.'/T10,'EXTERNAL TEMPERATURE IS',F4.0,
     *   ' DEGREES CENTIGRADE.'/T10,'USED',I3,' SPECTRA -- ',8A4///
     *  )
69    FORMAT('1RSL0011  NORMAL END OF RUN')
81    FORMAT(//T54,'AVERAGED ROCK SPECTRUM')
82    FORMAT(//T60,'TARGET RADIANCE')
83    FORMAT(//T60,'EMITTANCE SPECTRUM')
86    FORMAT(//T47,'AVERAGED AIRBORNE BLACKBODY SPECTRUM')
87    FORMAT(//T51,'AIRPATH ABSORBSION SPECTRUM')
93    FORMAT('1',T10,'MISSION',I4,' FLIGHT ',I1,
     *   ' TEMPERATURE CONVERSION TABLE.'//
     *   T10,'INTERNAL REFERENCE TEMPERATURE IS',F4.0,' DEGREES '
     *   ,'CENTIGRADE.'/T10,'EXTERNAL TEMPERATURE IS',F4.0,
     *   ' DEGREES CENTIGRADE.'/T10,'USED',I3,' SPECTRA -- ',8A4///
     *   T13,'READING',T33,'TEMPERATURE')
94    FORMAT(F17.0,F23.1)
95    FORMAT(T10,'AVER SPECTRUM',T30,'STANDARD DEV',T50,'TARGET RAD',
     *   T70,'STANDARD DEV',T90,'EMITTANCE',T110,'STANDARD DEV'//
     *   (I4,E16.3,5E20.3))
96    FORMAT(//4X,E16.3,5E20.3)
97    FORMAT('0RSL0521  CANNOT COMPUTE TARGET RADIANCE')
      END
```

1.2 Subroutine Splot

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C              SPLOT -- SPECTRUM PLOT PROGRAM                          C
C                                                                      C
C   MEAN     -- ARRAY CONTAINING AVERAGE SPECTRUM                      C
C   SD       -- ARRAY CONTAINING STANDARD DEVIATION OF SPECTRUM        C
C   MIN      -- PLOT LEFT BOUND                                        C
C   MAX      -- PLOT RIGHT BOUND                                       C
C   LOG      -- UNIT ON WHICH PLOT IS WRITTEN (LRECL > 130)            C
C   NPT      -- NUMBER OF POINTS IN ARRAYS (SIZE OF PLOT)              C
C   COUNT    -- INITIAL SEQUENCE NUMBER                                C
C                                                                      C
C   N.B. IF MIN=MAX THE PROGRAM WILL FIND BOUNDS TO FIT THE DATA       C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE SPLOT (MEAN ,SD, MIN, MAX, LOG, NPT, COUNT)
      INTEGER COUNT
      REAL MIN, MAX, MEAN(NPT), SD(NPT)
      REAL GRAPH(101), YCORD(11), XXXX/'XXXX'/, STAR/'****'/, MARK,
     *     PLUS/'++++'/, MINUS/'----'/, DOT/'....'/, BLANK/'    '/
C
      XMIN = MIN
      XMAX = MAX
      NPTS = NPT
      ICNT = COUNT
C
      IF (XMIN .LT. XMAX) GOTO 10
C
C     FIND XMIN AND XMAX
C
      XMIN = MEAN(1) - SD(1)
      XMAX = MEAN(1) + SD(1)
      DO 20 I = 2,NPTS
      XMIN = AMIN1 (XMIN,MEAN(I)-SD(I))
      XMAX = AMAX1 (XMAX,MEAN(I)+SD(I))
20    CONTINUE
      FUDGE = .02 * (XMAX - XMIN)
      XMAX = XMAX + FUDGE
      XMIN = XMIN - FUDGE
C
10    DELTA = (XMAX - XMIN) / 100.
      WRITE (LOG,54) XMIN, XMAX, DELTA
C
C     COMPUTE AND PRINT Y COORDINATES
C
      YCORD(1) = XMIN
      DO 30 I = 2,11
      YCORD(I) = YCORD(I-1) + DELTA * 10.
30    CONTINUE
```

```
      WRITE (LOG,51) YCORD
      WRITE (LOG,52)
C
      DO 50 I = 1,NPTS
C     INITIALIZE GRAPH LINE.
C
      MARK = BLANK
      IF (MOD(I,10) .EQ. 0) MARK = DOT
      DO 40 J = 1,101
      GRAPH(J) = MARK
40    CONTINUE
C
      DO 35 J = 1,101,10
      GRAPH(J) = DOT
35    CONTINUE
C
C     COMPUTE *,-,+ POSITIONS.
C
      SM = MEAN(I) - XMIN
      ISM = SM/DELTA + 0.5
      ISD = SD(I) / DELTA
      ISL = ISM - ISD
      ISH = ISM + ISD
      IF (ISM .LT. 2) GOTO 45
      LL = MIN0 (ISM-1, 101)
      DO 60 J = 1,LL
      GRAPH(J) = XXXX
60    CONTINUE
C
45    IF (ISH .GE. 1 .AND. ISH .LE. 101) GRAPH(ISH) = PLUS
      IF (ISL .GE. 1 .AND. ISL .LE. 101) GRAPH(ISL) = MINUS
      IF (ISM .GE. 1 .AND. ISM .LE. 101) GRAPH(ISM) = STAR
C
      WRITE (LOG,53) MEAN(I), ICNT, GRAPH, ICNT
C
      ICNT = ICNT + 1
50    CONTINUE
C
      WRITE (LOG,52)
      WRITE (LOG,51) YCORD
      RETURN
C
51    FORMAT (17X,11(E9.2,1X))
52    FORMAT (19X,10('.*********'),'.')
53    FORMAT (3X,E10.2,I5,1X,101A1,I5)
54    FORMAT (/25X,'XMIN IS',E9.2,15X,'XMAX IS',E9.2,15X,'DELTA IS',
     *          E9.2/)
      END
```

1.3 Subroutines Irrad and Absl

```
      SUBROUTINE IRRAD (IRAD, REFT, BBT)
      REAL IRAD(88)
      REAL LAM1/6.8/,LAM88/13.4/,C1/37410./,C2/14338./,PI/3.141593/
C
C     DEFINE BLACK BODY RADIANCE FUNCTION
      RAD (T, W) = C1 / (PI * (EXP (C2 / (W * T)) - 1.0) * W ** 5)
C
C     COMPUTE NET IRRADIANCE
C
      TEMP1 = REFT + 273.
      TEMP2 = BBT + 273.
      DLAM = (LAM88 - LAM1) / 87.0
      WW = LAM1
      DO 21 I = 1,88
      IRAD(I) = RAD(TEMP1,WW) - RAD(TEMP2,WW)
      WW = WW + DLAM
21    CONTINUE
      RETURN
C
C     COMPUTE ABSOLUTE RADIANCE
C
      ENTRY ABSL (IRAD, TEMP)
      TEMP1 = TEMP + 273.
      DLAM = (LAM88 - LAM1) / 87.0
      WW = LAM1
      DO 22 I = 1,88
      IRAD(I) = RAD(TEMP1,WW)
      WW = WW + DLAM
22    CONTINUE
      RETURN
      END
```

1.4 Subroutines Tcalc, Aver, and Sigma

```
      REAL FUNCTION TCALC (REFT, FACT, DISK)
      INTEGER DISK
      REAL RAW(88), DSK(92)
      EQUIVALENCE (RAW(1), DSK(5))
C
      ACC = 0.0
      DO 10 I = 1,30
      READ (DISK,END=20) DSK
      DO 10 J = 1,88
      ACC = ACC + RAW(J)
10    CONTINUE
C
20    EN = FLOAT(I-1) * 88
      TCALC = REFT + FACT * (ACC/EN)
      REWIND DISK
      RETURN
      END
```

```
      REAL FUNCTION AVER (A, N)
      REAL A(N)
      S = 0.0
      DO 10 I = 1,N
10    S = S + A(I)
      AVER = S / N
      RETURN
      END
```

```
      SUBROUTINE SIGMA (MEAN, SD, EN, NPT)
      REAL MEAN(NPT), SD(NPT)
      DO 10 I = 1,NPT
      SD(I) = SQRT ((SD(I) - MEAN(I)**2/EN) / (EN-1.0))
      MEAN(I) = MEAN(I) / EN
10    CONTINUE
      RETURN
      END
```

2.1 Program Prep

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C                PROGRAM PREP  --  SPECTRUM PRE-PROCESSOR                  C
C                                                                          C
C                              PROGRAM DESCRIPTION                         C
C                                                                          C
C     PREP READS RAW SPECTRAL DATA IN 1969 NASA FORMAT AND                 C
C     OUTPUTS SPECTRA WITHIN GIVEN TIME LIMITS IN STANDARD                 C
C     FORMAT COMPATIBLE WITH PROGRAM TASK.  PRINTED OUTPUT                 C
C     CONSISTS OF THE IDENTIFICATION HEADER ASSOCIATED WITH                C
C     EACH OUTPUT SPECTRUM, THE RECORDING TIME IN THE FORM                 C
C     HH:MM:SS.MSEC, THE MINIMUM, MAXIMUM, AND AVERAGE RADIOMETER          C
C     READING, AND THE STANDARD DEVIATION FOR THE NINE                     C
C     RADIOMETER SAMPLES.  THE AVERAGE RAW SPECTRUM, ALONG WITH            C
C     THE STANDARD DEVIATION FOR EACH COUNTER POINT IS PRINTED             C
C     AND PLOTTED FOR EACH GROUP OF SPECTRA PROCESSED.                     C
C                                                                          C
C                              RECORD FORMATS                              C
C                                                                          C
C     THE FIRST TWELVE BYTES ARE IDENTICAL IN BOTH FORMATS,                C
C     THESE REPRESENT THE IDENTIFICATION PART OF THE SPECTRUM              C
C     (8 BYTES) AND THE TIME OF DAY IN ELAPSED MILLISECONDS.               C
C     NEXT COME THE SPECTROMETER DATA POINTS (88) IN HALFWORD              C
C     INTEGERS ON THE TAPE, AND FULLWORD FLOATING POINT ON THE             C
C     OUTPUT FILES. LAST COMES THE CALIBRATION HALFWORDS.  THE             C
C     FIRST NINE OF THESE ARE RADIOMETER READINGS.                         C
C                                                                          C
C                            TEMPERATURE VARIANCE                          C
C                                                                          C
C     SPECTRA WHOSE TEMPERATURE VARIANCE IS GREATER THAN A                 C
C     GIVEN LIMIT ARE NOW BYPASSED (2/9/71).  A NAMELIST                   C
C     MUST PRECEDE THE CONTROL CARDS OF THE FORM:                          C
C     &PARMS TEMP=NNN., &END, WHERE NNN IS IN MILLIVOLTS.                  C
C                                                                          C
C                              DECK SETUP                                  C
C                                                                          C
C     RR UU AA AA AAAAA ZZ ZZ ZZZZZ TTTTTTTTTTTTTTTTTTTTTTTTTTTTT         C
C                                                                          C
C     R -- RAMP CODE (00=UP, 01=DOWN)                                      C
C     U -- OUTPUT UNIT NUMBER (A DD CARD MUST BE SUPPLIED)                 C
C     A -- START TIME IN HH MM SSTTT FORMAT                                C
C     Z -- STOP TIME IN SAME FORMAT                                        C
C     T -- THE REST OF THE CARD MAY CONTAIN A TITLE FOR THE                C
C          PRINTED OUTPUT.                                                 C
C     ANY NUMBER OF INPUT CARDS MAY BE USED BUT THE TIMES                  C
C     MUST APPEAR IN INCREASING ORDER TO AVOID REREADING THE               C
C     INPUT DATASET.                                                       C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      INTEGER CARD/5/, PRINT/6/, DISK, NMAX/88/, RMAX/9/
      INTEGER AH, AM, AS, ZH, ZM, ZS, HR, MN, MS, TITLE(8)
      INTEGER BEGIN, END, LAST, TIME/0/, DUMMY/0/
      INTEGER NAME(2)/'  UP', 'DOWN'/, DATE(5)
      INTEGER*2  INBUF(150), HEADER(6), RAD(35), RAMP
      INTEGER*2 MISDAY, LINRUN, SITUNS, ERRAMP,
     *          MIS, DAY, LIN, RUN, SIT, UNS, ERR
      REAL SPECT(88), ASP(88), SSP(88), RSP(88), ZERO(88)/88*0.0/
      EQUIVALENCE (INBUF(1), HEADER(1)), (INBUF(95), RAD(1))
      EQUIVALENCE  (HEADER(1), MISDAY),
     *             (HEADER(2), LINRUN),
     *             (HEADER(3), SITUNS),
     *             (HEADER(4), ERRAMP),
     *             (HEADER(5), TIME  )
C
C     DEFINE NAMELIST
      DATA TEMP /150./
      NAMELIST /PARMS/ TEMP
C
C     DEFINE MILLISECOND CONVERSION FORMULA
      MSEC (IH, IM, IS) = 3600000*IH + 60000*IM + IS
C
C     READ NAMELIST
      READ (CARD,PARMS)
      RRMAX = RMAX
      RNMAX = NMAX
      NREAD = 0
C
C     GET DAY DATE & TIME
      CALL DATER (DATE)
C
C     READ CONTROL CARD
10    READ (CARD,51,END=99) RAMP,DISK,AH,AM,AS,ZH,ZM,ZS,TITLE
      INAME = RAMP+1
      ICNT = 1
      IF (RAMP .EQ. 0) ICNT = 91
      WRITE (PRINT,61) NAME(INAME),AH,AM,AS,ZH,ZM,ZS,TITLE,DATE
C
C     CONVERT TO MILLISECONDS
      BEGIN = MSEC (AH,AM,AS)
      END   = MSEC (ZH,ZM,ZS)
C
C     CHECK FOR ERRORS
      IF (END .GE. BEGIN .AND. END .GE. BEGIN) GOTO 19
      WRITE (PRINT,67) TITLE
      GOTO 10
C
19    DO 20 I = 1,NMAX
```

```
      ASP(I) = 0.0
      SSP(I) = 0.0
20    CONTINUE
      NSPEC = 0
      NDEL = 0
      AARAD = 0.0
      ASRAD = 0.0
      GHIGH = -1E70
      GLOW = 1E70
      WRITE (PRINT,64)
      GOTO 15
C
C
C     INPUT READ LOOP
C
30    CALL RDNASA (INBUF, IEOT)
      IF (IEOT .EQ. 1) GOTO 40
      NREAD = NREAD + 1
C
C     CHECK FOR ERROR AND WRONG RAMP AT SAME TIME
      IF (ERRAMP .NE. RAMP) GOTO 30
C
C     CHECK FOR WITHIN TIME LIMITS
15    IF (TIME .LT. BEGIN) GOTO 30
      IF (TIME .GT. END) GOTO 50
C
C     SPECTRUM FOUND WITHIN RANGE
      NSPEC = NSPEC + 1
C
C     UNPACK HEADER
      CALL UNPACK (MISDAY, MIS, DAY)
      CALL UNPACK (LINRUN, LIN, RUN)
      CALL UNPACK (SITUNS, SIT, UNS)
C
C     CONVERT TIME
      HR = TIME/3600000
      MN = MOD(TIME/60000,60)
      NS = MOD(TIME,60000)
C
C     PROCESS RADIOMETER VALUES
      ARAD = 0.0
      RHIGH = -1E70
      RLOW = 1E70
      DO 45 I = 1,RMAX
      R = RAD(I)
      RHIGH = AMAX1 (RHIGH,R)
      RLOW  = AMIN1 (RLOW,R)
      ARAD = ARAD + R
45    CONTINUE
      SRAD = RHIGH - RLOW
      GHIGH = AMAX1 (GHIGH,RHIGH)
```

```
      GLOW = AMIN1 (GLOW,RLOW)
      ARAD = ARAD/RRMAX
      AARAD = AARAD + ARAD
      ASRAD = ASRAD + SRAD
C     WRITE SPECTRUM AND RAD INFO
      WRITE (PRINT,63) NSPEC, MIS, DAY, LIN, RUN, SIT, RAMP,
     *                 HR, MN, MS, RLOW, RHIGH, ARAD, SRAD
C
C     CHECK FOR UNACCEPTABLE TEMPERATURE VARIANCE
      IF (SRAD .LE. TEMP) GOTO  31
      NDEL = NDEL + 1
      WRITE (PRINT,71)
      GOTO 30
C
C     SUM SPECTRA
31    DO 60 I = 1,NMAX
      SPECT(I) = INBUF(I+6)
      ASP(I) = ASP(I) + SPECT(I)
      SSP(I) = SSP(I) + SPECT(I) ** 2
60    CONTINUE
C
C     WRITE OUTPUT RECORD
      IF (DISK .NE. 0) WRITE (DISK) HEADER, DUMMY, SPECT, RAD
      GOTO 30
C
C     END OF READ LOOP
50    RNSP = NSPEC
      AARAD = AARAD / RNSP
      ASRAD = ASRAD / RNSP
      NSPEC = NSPEC - NDEL
      IF (NSPEC .LT. 2) WRITE (PRINT,62)
      IF (NSPEC .LT. 2) GOTO 10
      WRITE (PRINT,85)
      WRITE (PRINT,79) GLOW, GHIGH, AARAD, ASRAD
C
C     WRITE OUT AVERAGED SPECTRA, STANDARD DEV, AND REL ERROR
      WRITE (PRINT,61) NAME(INAME),AH,AM,AS,ZH,ZM,ZS,TITLE,DATE
      WRITE (PRINT,65) NSPEC
      IF (RAMP .EQ. 0) WRITE (PRINT,81)
      IF (RAMP .EQ. 1) WRITE (PRINT,82)
      CALL TABLE (ASP, SSP, NMAX, NSPEC, ICNT, PRINT, 'RELATIVE')
C
C     PLOT AVERAGED SPECTRUM
      WRITE (PRINT,61) NAME(INAME),AH,AM,AS,ZH,ZM,ZS,TITLE,DATE
      WRITE (PRINT,65) NSPEC
      IF (RAMP .EQ. 0) WRITE (PRINT,81)
      IF (RAMP .EQ. 1) WRITE (PRINT,82)
      CALL SPLOT (ASP, SSP, 0.0, 0.0, PRINT, 88, ICNT)
      WRITE (PRINT,84)
```

```
C     PLOT STANDARD DEVIATION
      WRITE (PRINT,61) NAME(INAME),AH,AM,AS,ZH,ZM,ZS,TITLE,DATE
      WRITE (PRINT,65) NSPEC
      IF (RAMP .EQ. 0) WRITE (PRINT,81)
      IF (RAMP .EQ. 1) WRITE (PRINT,82)
      CALL SPLOT (SSP, ZERO, 0.0, 100., PRINT, 88, ICNT)
      WRITE (PRINT,83)
      GOTO 10
C
C     END OF FILE EXITS
40    WRITE (PRINT,68)
      WRITE (PRINT,69) NREAD
      STOP
99    WRITE (PRINT,69) NREAD
      WRITE (PRINT,66)
      STOP
C
C
51    FORMAT(4(I2,1X),I5,1X,2(I2,1X),I5,1X,8A4)
61    FORMAT('1',A4,' RAMP SPECTRUM GROUP (',2(I2,1X),I5,' TO ',
     *       2(I2,1X),I5,') CALLED  --  ',13A4)
62    FORMAT(///' RSL020I  INSUFFICIENT RECORDS -- GROUP BYPASSED')
63    FORMAT(10X,I7,3X,6I7,5X,2I3,I6,3X,3F10.0,F10.1)
64    FORMAT(//20X,'MISSION',4X,'DAY',3X,'LINE',4X,'RUN',3X,'SITE',
     *       3X,'RAMP',7X,'TIME',9X, 2X,'LOW RAD',3X,'HIGH RAD',3X,
     *       'AVER RAD',3X,'DEL RAD'/)
65    FORMAT('   NUMBER OF SPECTRA IN GROUP:',I4)
66    FORMAT(' RSL001I  NORMAL END OF RUN')
67    FORMAT(///' RSL032I  TIMES NOT SPECIFIED IN INCREASING ORDER'/
     *          ' RSL032I  ',20A4)
68    FORMAT('0RSL030I  DATA EXHAUSTED -- END OF GROUP NOT FOUND')
69    FORMAT('1RSL000I  ',I5,' RECORDS READ')
71    FORMAT('+',T130,'<=')
79    FORMAT(/72X,'GROUP:    ',3F10.0,F10.1)
81    FORMAT('   COUNTERS RANGE FROM 6.8 TO 13.4 MICRONS')
82    FORMAT('   COUNTERS RANGE FROM 13.4 TO 6.8 MICRONS')
83    FORMAT(//T53,'PLOT OF STANDARD DEVIATION')
84    FORMAT(//T57,'PLOT OF GROUP MEAN')
85    FORMAT(//84X,'LOW RAD',3X,'HIGH RAD',3X,'AVER RAD',3X,'DEL RAD')
      END
```

2.2 Subroutine Table

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                          SUBPROGRAM TABLE                            C
C                                                                      C
C    THE ROUTINE COMPUTES AND PRINTS THE MEAN, STANDARD                C
C    DEVIATION, AND RELATIVE ERROR OF EACH COUNTER POINT IN            C
C    A GROUP OF SPECTRA.                                               C
C                                                                      C
C    AVG      -- A VECTOR CONTAINING THE SUM OF THE SPECTRA            C
C                THE SPECTRUM MEAN IS RETURNED IN AVG.                 C
C    SD       -- A VECTOR CONTAINING THE SUM OF THE SPECTRA SQUARED    C
C                THE SPECTRUM STANDARD DEV IS RETURNED IN SD.          C
C    NPT      -- THE NUMBER OF COUNTER POINTS                          C
C    NSPCT    -- THE NUMBER OF SPECTRA IN THE GROUP                    C
C                THIS NUMBER MUST MUST BE GREATER THAN 1               C
C    ICNT     -- THE INITIAL COUNTER POINT SEQUENCE NUMBER             C
C    PRINT    -- PRINTER UNIT NUMBER                                   C
C    MODE     -- 'RELATIVE' FOR RELATIVE ERROR                         C
C                'STANDARD' FOR STANDARD ERROR                         C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE TABLE (AVG, SD, NPT, NSPCT, INCT, PRINT, MODE)
      REAL AVG(NPT), SD(NPT), RE(100)
      INTEGER PRINT, MODE(2), REL /'RELA'/
C
C     COMPUTE STATISTICS
      EN = NSPCT
      SQRTN = SQRT (EN)
      DO 10 I = 1,NPT
      SD(I) = SQRT ((SD(I) - AVG(I)**2/EN) / (EN - 1.0))
      AVG(I) = AVG(I) / EN
      RE(I) = SD(I) / SQRTN
      IF (MODE(1) .EQ. REL) RE(I) = RE(I) / AVG(I)
10    CONTINUE
C
C     FIND OVERALL AVERAGES
      AAVG = AVER (AVG, NPT)
      ASD = AVER (SD, NPT)
      ARE = AVER (RE, NPT)
```

```
C     PRINT TABLE
      WRITE (PRINT,63) MODE
      IC = INCT
      DO 20 I = 1,NPT
      WRITE (PRINT,61) IC, AVG(I), SD(I), RE(I)
      IC = IC + 1
20    CONTINUE
      WRITE (PRINT,62) AAVG, ASD, ARE
C
      RETURN
C
61    FORMAT (I16,2F20.2,F20.5)
62    FORMAT (/T13,'MEAN' ,2F20.2,F20.5)
63    FORMAT (/T13,'COUNTER',T27,'AVER SPECTRUM',T48,'STANDARD DEV',
     *        T66,2A4,' ERROR')
      END
```

2.3 Subroutine Xlate

```
      SUBROUTINE XLATE (TIME, HMS)
      INTEGER TIME, HMS(3)
C
C     TRANSLATE FROM ELAPSED MSEC TO HH:MM:SS.MSEC FORMAT
C
      HMS(1) = TIME / 3600000
      HMS(2) = MOD (TIME/60000,60)
      HMS(3) = MOD (TIME,60000)
C
      RETURN
      END
```

2.4 Subroutine Rdnasa

```
         TITLE   'RDNASA -- NASA TAPE READ PROGRAM'
         MACRO                          PROVIDE STANDARD OS LINKAGE
&CSECT   LINKS   &SAVE,&BASE=12         GIVE CSECT NAME, SAVEAREA
         LCLC    &NAME                  NAME, AND GLOBAL BASE REG
&NAME    SETC    '&SAVE'                SAVE IF SPECIFIED
         AIF     ('&NAME' NE '').OK     JUMP IF SPECIFIED
&NAME    SETC    'SAVEAREA'             SET DEFAULT NAME
.OK      ANOP
&CSECT   CSECT                          DEFINE EXTERNAL SYMBOL
         STM     14,12,12(13)           SAVE CALLERS REGS
         BALR    &BASE,0                GET ADDRESSIBILITY
         USING   *,&BASE                TELL ASSEMBLER
         LR      10,13                  SAVE POINTER TO CALLERS SA
         LA      13,&NAME               POINT TO CURRENT SA
         ST      13,8(0,10)             PLANT LINK TO CURRENT SA
         ST      10,4(0,13)             PLANT LINK TO HIGHER SA
         B       *+76                   BRANCH AROUND SAVEAREA
&NAME    DC      18A(0)                 ALLOCATE SAVEAREA
         MEND
*
         MACRO                          TEST FOR SUCCESSFUL OPEN
&L       TOPEN   &DCB,&ADDR             DCB ADDR / BRANCH ADDR
&L       TM      &DCB+48,X'10'          TEST OPEN BIT
         BO      &ADDR                  TAKE BRANCH IF OPEN
         MEND
         PRINT   NOGEN
RDNASA   LINKS
*********************************************************************
*                                                                   *
*            SUBROUTINE RDNASA (DATA, IEOT)                         *
*                                                                   *
*   DATA -- OUTPUT HALFWORD ARRAY USED BY FORTRAN PROGRAMS.         *
*   IEOT -- SET TO ONE ON END OF FILE READS.                        *
*   NASA -- DDNAME FOR INPUT DATASET.                               *
*                                                                   *
*                  PROGRAM FUNCTION                                 *
*                                                                   *
*   THIS ROUTINE READS SPECTRUM DATA RECORDS IN THE 1969 NASA       *
*   FORMAT (SEE DSECT).  IT MOVES THE RAW DATA INTO THE             *
*   MAIN PROGRAM BUFFER AND CLIPS THE FIRST TWO SPECTROMETER        *
*   POINTS.  THE CLIP IS TO CORRECT FOR THE TWO COUNTER POINT       *
*   ASYMMETRY BETWEEN UP AND DOWN RAMP RECORDS.                     *
*                                                                   *
*         UP RAMP .....................................             *
*       DOWN RAMP | .....................................           *
*                 |------- LEADING DATA POINT ---------|           *
*                                                                   *
*********************************************************************
```

```
DATA     EQU   2
IEOT     EQU   3                 VALUE OF IEOT
AIEOT    EQU   4                 ADDRESS OF IEOT
         L     DATA,0(0,1)         GET BASE ADDRESS OF DATA
         L     AIEOT,4(0,1)      GET ADDRESS OF IEOT
         SR    IEOT,IEOT         SET DEFAULT ZERO
*
         TOPEN NASA,READ
         OPEN  NASA              ATTEMPT TO OPEN
         TOPEN NASA,READ
         WTO   'RSL1001  NASA DD CARD MISSING'
         ABEND 20
*
READ     GET   NASA                          LOCATE A RECORD
         LA    1,4(0,1)                      SKIP RECORD CONTROL WORD.
         MVC   0(NASAHDR,DATA),0(1)            MOVE ID HEADER
         MVC   NASAHDR(REST,DATA),NASAHDR+4(1)
*
         B     DONE
EOT      CLOSE (NASA,REREAD)
         LA    IEOT,1            SET END OF FILE INDICATOR
DONE     ST    IEOT,0(0,AIEOT) STORE IT
         L       13,4(0,13)
         RETURN (14,12)
NASA     DCB   DDNAME=NASA,DSORG=PS,RECFM=V,BLKSIZE=NASABUFL+4,
               EODAD=EOT,MACRF=GL
         TITLE   'NASA RECORD FORMAT'
NASARECD DSECT
NASARCW  DS      F                          RECORD CONTROL WORD
NASAMISS DS      X                          MISSION
NASADAY  DS      X                          DAY
NASALINE DS      X                          LINE
NASARUN  DS      X                          RUN
NASASITE DS      X                          SITE
NASAUNUS DS      X                          UNUSED
NASAERR  DS      X                          ERROR INDICATOR
NASARAMP DS      X                          RAMP CODE
NASATIME DS      F                          TIME IN ELAPSED MSEC
NASAHDR  EQU     *-NASAMISS                 HEADER LENGTH
NASASPCT DS      90H                        SPECTROMETER DATA
NASARAD  DS      9H                         RADIOMETER READING
         DS      9H                         CALIBRATION DATA
         DS      9H                         CALIBRATION DATA
         DS      2H                         CALIBRATION DATA
         DS      6H                         REMAINDER
NASALEN  EQU     *-NASAMISS                 RECORD LENGTH
NASABUFL EQU     *-NASARCW                  BLOCK LENGTH
REST     EQU   *-(NASASPCT+4)
```

```
         TITLE   'STANDARD FORMAT USED BY TASK/PREP/PROC'
STANDRD  DSECT
STDRCW   DS      F                       RECORD CONTROL WORD
STDMISS  DS      X                       MISSION
STDDAY   DS      X                       DAY
STDLINE  DS      X                       LINE
STDRUN   DS      X                       RUN
STDSITE  DS      X                       SITE
STDUNUS  DS      X                       UNUSED
STDERR   DS      X                       ERROR INDICATOR
STDRAMP  DS      X                       RAMP CODE
STDTIME  DS      F                       TIME IN ELAPSED MILLISECONDS
STDDUMMY DS      F                       TASK PROCESSING HISTORY
STDSPECT DS      88F                     FLOATING POINT SPECTRAL DATA
STDRAD   DS      35H                     INTEGER RADIOMETER DATA
STDLEN   EQU     *-STDMISS               RECORD LENGTH
         END
```

2.5 Subroutine Unpack

```
UNPACK   LINKS
         L       R2,0(0,R1)              POINT TO FIRST ARG
         L       R3,4(0,R1)              POINT TO SECOND ARG
         L       R4,8(0,R1)              POINT TO THIRD ARG
         LH      R5,0(0,R2)              PICK UP FIRST ARG
         N       R5,=X'000000FF'         GET A BYTE
         STH     R5,0(0,R4)              STORE THIRD ARG
         LH      R5,0(0,R2)              PICK UP FIRST ARG
         N       R5,=X'0000FF00'         GET A BYTE
         SRL     R5,8                    ALIGN IT
         STH     R5,0(0,R3)              STORE SECOND ARG
         L       R13,4(0,R13)            SCRAM
         RETURN  (14,12)
         COPY    REGS
         END
```

2.6 Subroutine Dater

```
         TITLE   'DATER -- ZELLER''S CONGRUENCE FOR DAY OF THE WEEK'
         PRINT   NOGEN
DATER    LINKS
*
*****************************************************************
*                                                               *
*     SUBROUTINE DATER (AREA) -- RETURNS DAY, DATE, AND TIME    *
*                                                               *
*                    AREA MUST CONTAIN 20 BYTES.                *
*                                                               *
*            WORD 1    -- THREE CHARACTER DAY OF THE WEEK       *
*            WORD 2/3  -- DATE IN THE FORM MM/DD/YY             *
*            WORD 4/5  -- TIME OF DAY IN THE FORM (HH:MM)       *
*                                                               *
*          THIS IS AN ADAPTATION OF ZELLER'S CONGRUENCE         *
*                                                               *
*****************************************************************
*
SLASH    EQU     C'/'                    CHARACTER CONSTANTS
LPAR     EQU     C'('
RPAR     EQU     C')'
COLON    EQU     C':'
BLANK    EQU     C' '
*
         L       1,0(0,1)                GET AREA ADDRESS
         LR      3,1                     SAVE AREA ADDRESS
         USING   RETURN,3                TELL ASSEMBLER
         TIME    DEC
         STM     0,1,SAVE                SAVE TIME AND DATE
*
         UNPK    RHOUR(3),TIMEHH(2)      UNPACK TIME
         UNPK    RMIN(3),TIMEMM(2)       UNPACK TIME
         MVI     RTIME,BLANK
         MVI     RLPAR,LPAR
         MVI     RCOLON,COLON
         MVI     RRPAR,RPAR
         EJECT
         XC      TIME,TIME               SET HIGHORDER BYTES TO ZERO
         CVB     5,SAVE                  CONVERT YY.DDD TO BINARY
         SR      4,4                     CLEAR FOR DIVIDE
         D       4,C1000                 R4 = DDD   R5 = YY
         EX      5,TESTLEAP              TEST FOR LEAP YEAR
         BNZ     *+10                    SKIP IF NOT LEAP YEAR
         MVC     MCONS+2(2),LEAPFEB      MODIFY FOR LEAP YEAR
*
         SR      6,6                     SET TO FIND MONTH
LOOP     SH      4,MCONS(6)              SUBTRACT UNTIL NOT PLUS
         BNP     OVER                    MONTH FOUND
         LA      6,L'MCONS(0,6)          POINT TO NEXT MONTH COUNT
         B       LOOP                    CONTINUE SEARCH
```

```
OVER     AH      4,MCONS(6)            ADJUST DAY OF MONTH NUMBER
         SRL     6,1                   MONTH NUMBER IN R6 (0-11)
         CH      6,C1                  TEST FOR JAN OR FEB
         BH      *+6                   THIS IS REQUIRED BY ZELLER
         BCTR    5,0                   DECREASE YEAR BY 1
         LR      9,5                   PUT YEAR (0-99) IN R9
         SR      8,8                   CLEAR FOR DIVIDE
         D       8,C4                  FIND FLOOR (YEAR/4)
         LR      8,6                   MONTH NUMBER IN R8
         AR      8,8                   NEED HALFWORD OFFSET
         AH      9,ZCONS(8)            ADD FIRST AND FOURTH TERM
         AR      9,4                   ADD DAY (1-31)
         AR      9,5                   ADD IN YEAR (0-99)
         AH      9,CENT1               ADD FIRST CENTURY TERM
         SH      9,CENT2
         SR      8,8
         LA      9,777(,9)             IN CASE OF NEGATIVE SUM
         D       8,C7                  FIND DAY OF THE WEEK
         AR      8,8                   NEED FULLWORD OFFSET
         AR      8,8                   DO IT THE PL/1 WAY
         LA      8,DAYNAME(8)          GET ADDRESS OF DAY NAME
         MVC     RDAYWEEK,0(8)         PLANT IN RETURN AREA
*
         UNPK    RDAY(3),DATEYEAR(2)   UNPACK YEAR (0-99)
         MVC     RYEAR,RDAY            MOVE TO CORRECT AREA
         MVI     RSLASH2,SLASH         MOVE IN SLASH
         CVD     4,SAVE                GET DAY INTO PACKED DEC
         UNPK    RDAY-1(3),SAVE+6(2)   UNPACK DAY OF THE MONTH
         OI      RDAY+1,X'F0'          STICK IN VALID ZONE
         MVI     RSLASH1,SLASH         MOVE IN SLASH
         LA      6,1(0,6)              USE ONE-ORIGIN MONTH
         CVD     6,SAVE                GET MONTH INTO PACKED DEC
         UNPK    RMONTH-1(3),SAVE+6(2) UNPACK MONTH OF THE YEAR
         OI      RMONTH+1,X'F0'        STICK IN VALID ZONE
         MVI     RMONTH-1,BLANK        CLEAR UNPK GARBAGE
*
         L       13,4(0,13)
         RETURN  (14,12),T,RC=0        RETURN TO CALLER
```

```
SAVE      DS      0D                      TIME MACRO SAVED HERE
TIME      DS      0F
TIMEHH    DS      PL1
TIMEMM    DS      PL1
TIMEREST  DS      PL2
DATE      DS      PL1                     MUST BE LOW HALF OF DBLE WD
DATEYEAR  DS      PL1
DATEDAY   DS      PL2
DAYNAME   DC      C'SUN MON TUE WED THR FRI SAT '
ZCONS     DC      AL2(28,31,2,5,7,10,12,15,18,20,23,25)
MCONS     DC      AL2(31,28,31,30,31,30,31,31,30,31,30,31)
LEAPFEB   DC      AL2(29)
CENTURY   EQU     19                      ASSUME CURRENT CENTURY
CENT1     DC      Y(CENTURY/4)            FLOOR CENTURY/4
CENT2     DC      Y(2*CENTURY)
C1        DC      Y(1)
C4        DC      A(4)
C7        DC      A(7)
C1000     DC      A(1000)
TESTLEAP  TM      BYTE,0                  EXECUTE FOR LEAP YEAR TEST
BYTE      DC      X'03'                   TEST LOW ORDER TWO BITS
*
          TITLE   'RETURN AREA FORMAT'
RETURN    DSECT                           SAT 01/16/71 (15:25)
*
RDAYWEEK  DS      CL4                     SAT
RMONTH    DS      CL2                     01
RSLASH1   DS      CL1                     /
RDAY      DS      CL2                     16
RSLASH2   DS      CL1                     /
RYEAR     DS      CL2                     71
RTIME     DS      CL1
RLPAR     DS      CL1                     (
RHOUR     DS      CL2                     15
RCOLON    DS      CL1                     :
RMIN      DS      CL2                     25
RRPAR     DS      CL1                     )
          END
```

3.1 Program Proc

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
C                    PROGRAM PROC -- SEPTEMBER 1970                    C
C                  STANFORD REMOTE SENSING LABORATORY                  C
C                                                                      C
C                                                                      C
C      THIS PROGRAM READIES SELECTED IR SPECTRA FOR STATISTICAL        C
C      ANALYSIS.  THE DATA ARE RATIOED TO A BLACKBODY, NORMALIZED      C
C      AND INVERTED.  INDIVIDUAL AND AVERAGED SPECTRA ARE SAVED        C
C      IN CARD IMAGE FORMAT ON SEPARATE FILES AFTER PROCESSING.        C
C      ALTHOUGH THERE MAY BE ANY NUMBER OF INPUT DATASETS, ALL         C
C      OUTPUT APPEARS ON TWO DATASETS, ONE FOR INDIVIDUALS AND         C
C      ONE FOR COMPOSITES.                                             C
C                                                                      C
C                                                                      C
C                         DDNAMES REQUIRED                             C
C                                                                      C
C      FT05F001 -- INPUT CONTROL CARDS.                                C
C      FT06F001 -- OUTPUT MESSAGES, TABLES, AND PLOTS.                 C
C      FT07F001 -- OUTPUT FILE FOR INDIVIDUAL SPECTRA                  C
C      FT08F001 -- OUTPUT FILE FOR AVERAGED SPECTRA                    C
C      FTNNF001 -- INPUT DATASETS                                      C
C      FT99F001 -- INPUT BLACKBODY REFERENCE SPECTRUM                  C
C                                                                      C
C                                                                      C
C                         RECORD FORMATS                               C
C                                                                      C
C   A. INPUT SPECTRA MUST BE IN "STANDARD" FORMAT.  THIS MEANS         C
C      THAT THE DATA WAS PRODUCED BY EITHER TASK OR PREP.              C
C   B. THE BLACKBODY DATA MUST BE PRE-AVERAGED AND IS USUALLY          C
C      GENERATED BY PROGRAM AVERAGE AND SAVED ON DISK.                 C
C   C. CARD OUTPUT CONSISTS OF AN IDENTIFICATION CARD,                 C
C      FOLLOWED BY THE DATA IN 8F9.4  FORMAT.                          C
C   D. PRINTED OUTPUT CONSISTS OF LISTINGS AND PLOTS OF THE            C
C      AVERAGED DATA AFTER PROCESSING.                                 C
C   E. THE CONTROL CARDS CONTAIN A UNIT NUMBER IN COLUMNS 4 AND 5      C
C      AND A DESCRIPTIVE NAME FOR THE DATA IN COLUMNS 31 - 70.         C
C      THIS FORMAT IS THE SAME AS THAT USED BY PREP.                   C
C                                                                      C
```

```
C                                                                       C
C                                                                       C
C                        NAMELIST PARAMETERS                            C
C                                                                       C
C       THESE VARIABLES MAY BE ALTERED USING THE &PARMS NAMELIST.       C
C       THE NAMELIST MUST BE PRESENT AND MUST BE THE FIRST DATA         C
C       IN THE INPUT STREAM.  IT IS FOLLOWED BY A CARD CONTAINING       C
C       A DESCRIPTION OF THE PROCESSING, WHICH APPEARS ON THE           C
C       PRINTED OUTPUT.  AFTER THIS COMES THE CONTROL CARDS.            C
C                                                                       C
C            VARIABLE    |       FUNCTION         |   DEFAULT          C
C            ------------------------------------------------           C
C             PSZ        | PLOT WIDTH             |  2.0               C
C             SMOOTH     | SMOOTHING SWITCH       |  .TRUE.            C
C             ICARDS     | INDIV CARDS SWITCH     |  .TRUE.            C
C             ACARDS     | AVERAGE CARDS SWITCH   |  .TRUE.            C
C             LIST       | DATA LISTING SWITCH    |  .TRUE.            C
C             PLOT       | PLOT SWITCH            |  .TRUE.            C
C             SMTYPE     | SMOOTHING TYPE         |  -9                C
C             BODY       | BLACKBODY UNIT NO.     |  99                C
C             NCLIP      | NO. OF POINTS CLIPPED  |  18                C
C             CNT        | INITIAL SEQUENCE NO.   |  91                C
C             CARD       | CONTROL INPUT UNIT     |   5                C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      INTEGER CARD/5/, PRINT/6/, IOUT/7/, AOUT/8/, BODY/99/, DATE(5)
      INTEGER CNT/91/, NCLIP/18/, SMTYPE/-9/, IER/0/
      INTEGER NAME(8), HEAD(2), TIME, DISK, HMS(3), HMS2(3), DES(18)
      REAL RAW(88), ASP(88), SSP(88), DSK(92), BLB(92), PSZ/2.0/
      EQUIVALENCE (DSK(1), HEAD(1)), (DSK(3), TIME)
      LOGICAL*1 SMOOTH/.TRUE./, ICARDS/.TRUE./, ACARDS/.TRUE./,
     *          LIST  /.TRUE. /, PLOT  /.TRUE./
C
C
C     DEFINE NAMELIST PARAMETERS
      NAMELIST /PARMS/ PSZ, SMOOTH, ICARDS, ACARDS, LIST, PLOT,
     *                 SMTYPE, BODY, NCLIP, CNT, CARD
C
C     READ NAMELIST PARAMETERS
      READ (CARD,PARMS)
      WRITE (PRINT,PARMS)
C
C     SET PARAMETERS
      CNT = CNT + NCLIP
      NMAX = 88 - 2*NCLIP
```

```
C       READ DESCRIPTOR
        READ (CARD,52) DES
        WRITE (PRINT,53) DES
C
C       GET DAY DATE & TIME
        CALL DATER (DATE)
C
C       READ & CLIP BLACKBODY, IGNORE FOUR WORD HEADER
        READ (BODY) BLB
        DO 12 I = 1,NMAX
12      BLB(I) = BLB(I+4+NCLIP)
C
C
C       READ CONTROL CARD
15      READ (CARD,51,END=44) DISK, NAME
C
        DO 10 I = 1,NMAX
        ASP(I) = 0.0
        SSP(I) = 0.0
10      CONTINUE
C
C       READ AND PROCESS SPECTRA
        DO 30 I = 1,10000
        READ (DISK,END=31) DSK
        IF (I .EQ. 1) ITIME = TIME
C
C       CLIP SPECTRUM, IGNORE FOUR WORD HEADER
        DO 13 J = 1,NMAX
13      RAW(J) = DSK(J+4+NCLIP)
C
C       RATIO SPECTRUM
        DO 26 J = 1,NMAX
26      RAW(J) = RAW(J) / BLB(J)
C
C       SMOOTH SPECTRA
        IF (SMOOTH) CALL SM (RAW, NMAX, IER, SMTYPE)
        IF (IER .EQ. 0) GOTO 17
        WRITE (PRINT,67)
        STOP
C
C       NORMALIZE SPECTRUM
17      CALL NORM (RAW, NMAX)
C
C       INVERT SPECTRUM
        DO 27 J = 1,NMAX
27      RAW(J) = -RAW(J)
```

```
C     SUM RESULT
      DO 25 J = 1,NMAX
      ASP(J) = ASP(J) + RAW(J)
      SSP(J) = SSP(J) + RAW(J) ** 2
25    CONTINUE
C
C     OUTPUT INDIVIDUAL SPECTRUM
      CALL XLATE (TIME, HMS)
      IF (ICARDS) WRITE (IOUT,71) HEAD,HMS,NAME,I,(RAW(J),J=1,NMAX)
C
30    CONTINUE
C
C     EOF ON SPECTRUM INPUT
31    NSPECT = I - 1
      IF (NSPECT .LT. 2) WRITE (PRINT,69) NAME
      IF (NSPECT .LT. 2) GOTO 15
      CALL XLATE (ITIME, HMS2)
C
C     PRINT RESULTS
      IF (LIST) WRITE(PRINT,62) NAME,HEAD,DATE,NSPECT,HMS2,HMS,DES
      IF (LIST) CALL TABLE (ASP, SSP, NMAX, NSPECT, CNT, PRINT,
     *                      'STANDARD')
C
C     PLOT RESULTS
      IF (PLOT) WRITE(PRINT,62) NAME,HEAD,DATE,NSPECT,HMS2,HMS,DES
      IF (PLOT) CALL SPLOT (ASP, SSP, -PSZ, PSZ, PRINT, NMAX, CNT)
C
C     OUTPUT AVERAGED SPECTRUM AND STANDARD DEVIATION
      IF (ACARDS) WRITE (AOUT,71) HEAD, HMS2, NAME, NSPECT,
     *                            (ASP(J),J=1,NMAX)
      IF (ACARDS) WRITE (AOUT,71) HEAD, HMS, NAME, NSPECT,
     *                            (SSP(J),J=1,NMAX)
C
C     READ NEXT CONTROL CARD
      GOTO 15
C
C     NO MORE CONTROL CARDS EXIT
44    WRITE (PRINT,66)
      STOP
```

```
51    FORMAT(3X,I2,T31,10A4)
52    FORMAT(20A4)
53    FORMAT('0RSL000I  ',20A4)
57    FORMAT(F10.3)
62    FORMAT(////'1UP RAMP SPECTRUM GROUP CALLED -- ',8A4,5X,2Z10,8X,
     *       5A4/' NUMBER OF SPECTRA IN GROUP:',I3/
     *       ' RECORDED FROM',2I3,I6,' TO',2I3,I6,'.'/1X,18A4/)
66    FORMAT(/////'1RSL001I  NORMAL END OF RUN')
67    FORMAT('1RSL010I  SMOOTHING PARAMETER INVALID')
69    FORMAT(/////'1RSL020I  INSUFFICIENT RECORDS -- GROUP BYPASSED'/
     *            ' RSL020I   ',8A4)
C
C     THIS IS THE CARD OUTPUT FORMAT
71    FORMAT(2Z10,1X,2I3,I6,1X,8A4,I4/(8F9.4))
      END
```

3.2 Subroutine Norm

```
      SUBROUTINE NORM (A, N)
      REAL A(N)
      SUM = 0.0
      SQS = 0.0
      DO 10 I = 1,N
      SUM = SUM + A(I)
      SQS = SQS + A(I) ** 2
10    CONTINUE
      EN = N
      SQS = SQRT ((SQS - SUM ** 2 / EN) / (EN - 1.0))
      SUM = SUM / EN
      DO 20 I = 1,N
      A(I) = (A(I) - SUM)/SQS
20    CONTINUE
      RETURN
      END
```

3.3 Subroutine Sm

```
      SUBROUTINE SM (NDATA, N, IER, NMP)
C
C    SMOOTHING SUBROUTINE WRITTEN BY J.R. MOORE
C
C      NDATA=INPUT SPECTRUM & OUTPUT SMOOTHED SPECTRUM
C      N=NUMBER OF POINTS
C      IER=ERROR MESSAGE--0 IF OK, -1 IF NOT
C      NMP=SMOOTHING TYPE
       REAL NDATA(100),MDATA(100),NP(20)
       IF(N.GT.100.OR.NMP.LT.-20.OR.NMP.GT.20) GO TO 900
       NNP=NMP
       IF(NMP.LT.0)NNP=-NMP
       NXP=NNP
       IF(NMP.EQ.-1) NNP=3
       MM=NNP-1
       M=N-MM
       DO 20 I=1,N
20     MDATA(I)=NDATA(I)
       DO 10 I=2,NNP
       J=I-1
10     NP(I)=NDATA(J)
       DO 200 I=1,M
       J=I+MM
       DO 11 K=1,MM
       KA=K+1
11     NP(K)=NP(KA)
       NP(NNP)=NDATA(J)
       IF(NMP.LT.0) GO TO 100
       GO TO (300,900,900,900,101,900,102,900,103,900,104,900,
      1401,900,900,900,402),NNP
101    SUM=17*NP(3)+12*(NP(2)+NP(4))-3*(NP(1)+NP(5))
       MDATA(I+2)=SUM/35
       GO TO 200
401    SUM=NP(1)+NP(2)+NP(3)+NP(4)+NP(5)+NP(6)+NP(7)+NP(8)+NP(9)+
      1NP(10)+NP(11)+NP(12)+NP(13)
       MDATA(I+6)=SUM/13
       GO TO 200
402    SUM=-21*(NP(1)+NP(17))-6*(NP(2)+NP(16))+7*(NP(3)+NP(15))+
      118*(NP(4)+NP(14))+27*(NP(5)+NP(13))+34*(NP(6)+NP(12))+
      139*(NP(7)+NP(11))+42*(NP(8)+NP(10))+43*NP(9)
       MDATA(I+8)=SUM/323
       GO TO 200
403    SUM=195*(NP(1)+NP(17))-195*(NP(2)+NP(16))-260*(NP(3)+NP(15))-
      1117*(NP(4)+NP(14))+135*(NP(5)+NP(13))+415*(NP(6)+NP(12))+
      2660*(NP(7)+NP(11))+825*(NP(8)+NP(10))+883*(NP(9))
```

```
      MDATA(I+8)=SUM/4199
      GO TO 200
102   SUM=-2*(NP(1)+NP(7))+3*(NP(2)+NP(6))+6*(NP(3)+NP(5))+7*NP(4)
      MDATA(I+3)=SUM/21
      GO TO 200
103   SUM=-21*(NP(1)+NP(9))+14*(NP(2)+NP(8))+39*(NP(3)+NP(7))+
     154*(NP(4)+NP(6))+59*NP(5)
      MDATA(I+4)=SUM/231
      GO TO 200
300   SUM=NP(1)+NP(2)+NP(3)
      MDATA(I+1)=SUM/3
      GO TO 200
104   SUM=-36*(NP(1)+NP(11))+9*(NP(2)+NP(10))+44*(NP(3)+NP(9))+
     160*(NP(4)+NP(8))+84*(NP(5)+NP(7))+89*NP(6)
      MDATA(I+5)=SUM/429
      GO TO 200
100   CONTINUE
      GO TO (300,900,900,900,101,900,106,900,107,900,108,
     1900,900,900,900,900,403),NXP
106   SUM=5*(NP(1)+NP(7))-30*(NP(2)+NP(6))+75*(NP(3)+NP(5))+131*NP(4)
      MDATA(I+3)=SUM/231
      GO TO 200
107   SUM=15*(NP(1)+NP(9))-55*(NP(2)+NP(8))+30*(NP(3)+NP(7))+
     1135*(NP(4)+NP(6))+179*NP(5)
      MDATA(I+4)=SUM/429
      GO TO 200
108   SUM=18*(NP(1)+NP(11))-45*(NP(2)+NP(10))-10*(NP(3)+NP(9))+
     160*(NP(4)+NP(8))+120*(NP(5)+NP(7))+143*NP(6)
      MDATA(I+5)=SUM/429
200   CONTINUE
C
C     RETURN MDATA IN NDATA
      DO 500 I = 1,N
500   NDATA(I) = MDATA(I)
      IER=0
      RETURN
900   IER=-1
      RETURN
      END
```

4.1 Program Discard

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     PROGRAM DISCARD                                                 C
C     DELETE TRAINING SPECTRA WHOSE DISTANCE FROM THE GROUP           C
C     MEAN IS UNACCEPTABLE                                            C
C                                                                     C
C                                                                     C
C     DDNAME    |          PURPOSE                 | VARIABLE NAME  C
C ------------------------------------------------------------------- C
C     FT06F001 | LISTING FILE                      | PRINT           C
C     FT05F001 | CONTROL CARD FILE                 | CARD            C
C     FT10F001 | OUTPUT SPECTRA FILE               | OUTPUT          C
C     FT03F001 | INDIVIDUAL SPECTRA INPUT FILE     | INDIV           C
C     FT04F001 | AVERAGE SPECTRUM INPUT FILE       | AVER            C
C                                                                     C
C                                                                     C
C     THE ABOVE DDNAMES MAY BE ALTERED USING THE &PARMS NAMELISTC
C     AND THE REJECTION LEVEL IS ENTERED BY GIVING "LIMIT"            C
C     A VALUE IN THE NAMELIST. A NAMELIST IS READ FOR EACH GROUPC
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER PRINT/6/, CARD/5/, OUTPUT/10/, INDIV/3/, AVER/4/
      INTEGER FLAG, NAME(8), TIME(3)
      REAL SPECT(52), ASP(52), LIMIT
      REAL*8 WORD(2) /' ','REJECTED'/
C
C     DEFINE NAMELIST
      NAMELIST /PARMS/ LIMIT, PRINT, CARD, OUTPUT, INDIV, AVER
C
C     READ NAMELIST
10    READ (CARD,PARMS,END=88)
C
C     READ NAME, NUMBER OF SPECTRA, AVERAGE SPECTRUM
      READ (AVER,51,END=88) NAME, NSPEC, ASP
      WRITE (PRINT,61) NAME
```

```
C     READ INDIVIDUAL SPECTRA
      ICNT = 0
      DO 20 I = 1,NSPEC
      READ (INDIV,52,END=77) TIME, NAME, ISEQ, SPECT
      TALLY = 0.0
      DO 15 J = 1,52
15    TALLY = TALLY + (SPECT(J) - ASP(J)) ** 2
C
      FLAG = 2
      IF (TALLY .GT. LIMIT) GOTO 17
C
C     WRITE ACCEPTABLE SPECTRUM
      WRITE (OUTPUT,52) TIME, NAME, ISEQ, SPECT
      ICNT = ICNT + 1
      FLAG = 1
C
17    WRITE (PRINT,63) ISEQ, TIME, TALLY, WORD(FLAG)
20    CONTINUE
C
C     GO READ NEXT GROUP
      WRITE (PRINT,62) LIMIT, ICNT
      GOTO 10
C
C     NORMAL END OF FILE EXIT
88    WRITE (PRINT,66)
      STOP
C
C     ERROR END OF FILE EXIT
77    WRITE (PRINT,67)
      STOP
C
51    FORMAT (T35,8A4,T68,I3/6(8F9.4/),4F9.4////////)
52    FORMAT (T22,2I3,I6,T35,8A4,T68,I3/(8F9.4))
61    FORMAT (1H1,T15,8A4,T48,'TIME',T58,'DISTANCE',T69,'DECISION'/)
62    FORMAT (///T40,'TOLERANCE',F5.0,' LEAVES',I3,' SPECTRA.')
63    FORMAT (T38,I4,')',I4,1X,I2,I6,F10.4,3X,A8)
66    FORMAT ('1RSL0011  NORMAL END OF RUN')
67    FORMAT ('1RSL1451  UNEXPECTED END OF FILE')
      END
```

5.1 Program Trkload

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C            PROGRAM TRKLOAD -- TRUCK TAPE TO DISK                     C
C                                                                      C
C   THIS PROGRAM READS TRUCK TAPES AND CREATES TWO OUTPUT FILES.       C
C   DATA RECORDS ARE STORED IN A DIRECT ACCESS DATASET, AND            C
C   IDENTIFICATION RECORDS ARE STORED IN A SEQUENTIAL DATASET          C
C   WITH POINTERS TO THE CORRESPONDING DATA.                           C
C                                                                      C
C            SPECTAPE -- DDNAME OF TRUCK TAPE.                         C
C                        NO DCB PARAMETERS REQUIRED.                   C
C            DIRECT   -- DDNAME OF DATA OUTPUT FILE.                   C
C                        DCB=(DSORG=DA,BLKSIZE=204)                    C
C            FT10F001 -- DDNAME OF IDENTIFICATION FILE.                C
C                        DCB=(RECFM=FB,LRECL=40,BLKSIZE=3520)          C
C            FT06F001 -- DDNAME OF IDENTIFICATION LISTING FILE.        C
C            FT04F001 -- DDNAME OF DATA LISTING FILE.                  C
C            FT05F001 -- DDNAME OF PARAMETER INPUT FILE.               C
C                        &PARMS LIST=F, ERRCNT=10, TERR=F, &END        C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER KEY/1/, COUNT/0/, PRINT/6/, INDEX/10/, DUMP/4/, LRECL
      INTEGER ERRCNT/10/,CARD/5/,NERR/0/,NIDS/0/,NINV/0/,NREADS/0/
      INTEGER JUMP, DATE(5), I, J, K, L, M, N
      INTEGER DTLEN, IDLEN/20/, DTSIZ/48/
      LOGICAL   LIST /.FALSE./,   TERR /.FALSE./
      DIMENSION SAVEID(6)
C
C     RDTRK COMMON DEFINITION
      COMMON /TDATA/ INPA(200),IDENT(6),SPECT(48),RADIO(48),MULT(6)
C
C     DEFINE DIGITIZED (0,1023) TO DECIVOLTS (-100,100) FORMULA
      DVOLT(RAW) = (200*RAW - 102300) / 1023
C
C     DEFINE AND READ NAMELIST
      NAMELIST /PARMS/ ERRCNT, LIST, TERR
      READ (CARD,PARMS,END=2)
2     CONTINUE
C
C     INITIALIZE
      DTLEN = DTSIZ*4 + 12
      IF (TERR) CALL NOERR
C
      CALL DATER (DATE)
      WRITE (PRINT,65) DATE
C
      ASSIGN 11 TO JUMP
```

```
11    CALL RDTRK (LRECL)
      NREADS = NREADS + 1
      IF (LRECL .EQ. IDLEN) GOTO 12
      IF (LRECL .LT. 0) GOTO 80
      IF (LRECL .EQ. 0) STOP
C
      BUFL = LRECL/2
      WRITE (PRINT,61) NREADS, LRECL, (INPA(J), J = 1,BUFL)
      NINV = NINV + 1
      GOTO 11
C
12    DO 10 I = 1,6
10    SAVEID(I) = IDENT(I)
      SAVKEY = KEY
C
C     READ INPUT TAPE
      ASSIGN 20 TO JUMP
20    CALL RDTRK (LRECL)
      NREADS = NREADS + 1
C
      IF (LRECL .EQ. DTLEN) GOTO 30
      IF (LRECL .EQ. IDLEN) GOTO 40
      IF (LRECL .EQ. 0) GOTO 50
      IF (LRECL .LT. 0) GOTO 80
C
C     BAD LRECL, IGNORE RECORD
      NINV = NINV + 1
      BUFL = LRECL/2
      WRITE (PRINT,61) NREADS, LRECL, (INPA(J), J = 1,BUFL)
      WRITE (PRINT,67)
      GOTO 20
C
C     DATA RECORD FOUND
30    DO 31 M = 1,DTSIZ
      SPECT(M) = DVOLT(SPECT(M))
31    RADIO(M) = DVOLT(RADIO(M))
      DO 32 M = 1,6
32    MULT(M)  = DVOLT(MULT(M))
C
      CALL DALOAD (SPECT, KEY)
      IF (.NOT. LIST) GOTO 49
C
      WRITE (DUMP,72) KEY, (SPECT(N), N = 1,DTSIZ)
      WRITE (DUMP,73)      (RADIO(N), N = 1,DTSIZ)
      WRITE (DUMP,74) MULT
C
49    KEY = KEY + 1
      COUNT = COUNT + 1
      GOTO 20
```

```
C       IDENTIFICATION RECORD FOUND
40      IF (COUNT .NE. 0) GOTO 47
C
C       IDENTIFICATION RECORD CONTAINS NO DATA
        WRITE (PRINT,62) SAVEID
        NINV = NINV + 1
        GOTO 45
C
C       WRITE IDENTICATION RECORD
47      NIDS = NIDS + 1
        WRITE (INDEX,66) SAVEID, SAVKEY, COUNT
        WRITE (PRINT,64) NIDS, SAVEID, SAVKEY, COUNT
        SAVKEY = KEY
        COUNT = 0
C
45      DO 46 I = 1,6
46      SAVEID(I) = IDENT(I)
        GOTO 20
C
C       END OF FILE EXIT
50      IF (COUNT .NE. 0) GOTO 48
C
C       IDENTIFICATION RECORD CONTAINS NO DATA
        WRITE (PRINT,62) SAVEID
        NINV = NINV + 1
        GOTO 60
C
C       WRITE FINAL IDENTIFICATION RECORD
48      NIDS = NIDS + 1
        WRITE (INDEX,66) SAVEID, SAVKEY, COUNT
        WRITE (PRINT,64) NIDS, SAVEID, SAVKEY, COUNT
```

```
60     NREADS = NREADS - 1
       KEY = KEY - 1
       WRITE (PRINT,63) NREADS, NIDS, KEY, NINV, NERR
       IF (LIST) WRITE (DUMP,63) NREADS, NIDS, KEY, NINV, NERR
       STOP
C
C      READ ERROR ROUTINE
80     WRITE (PRINT,69) NREADS, (INPA(J), J = 1,160)
       NERR = NERR + 1
       IF (NERR .LE. ERRCNT) GOTO JUMP, (11, 20)
C
C      TOO MANY ERRORS
       WRITE (PRINT,71) NREADS
       STOP
C
61     FORMAT(///' RSL0421  RECORD',I5,' INVALID',I4,' BYTES'//
      *          (' RSL0421',16Z6))
62     FORMAT(///' RSL0401  IDENTIFICATION RECORD CONTAINS NO DATA'//
      *          ' RSL0401  DAY IS   ',I9/
      *          ' RSL0401  TIME IS  ',3I3/
      *          ' RSL0401  SAMPLE IS',I9/
      *          ' RSL0401  SITE IS  ',I9///)
63     FORMAT('1RSL0001',I6,' RECORDS READ'/
      *       ' RSL0001',I6,' IDENTIFICATION RECORDS SAVED'/
      *       ' RSL0001',I6,' DATA RECORDS SAVED'/
      *       ' RSL0001',I6,' INVALID RECORDS FOUND'/
      *       ' RSL0001',I6,' PERMANENT READ ERRORS'//
      *       ' RSL0011        NORMAL END OF RUN')
64     FORMAT(T15,I4,')  DAY  =',I4,';  TIME  =',3I3,';  SAMPLE  =',
      *       I4,';  SITE  =',I4,';  START  =',I5,';  COUNT  =',I3)
65     FORMAT('1',T35,'IDENTIFICATION RECORDS SAVED ON ',5A4//)
66     FORMAT(8I5)
67     FORMAT(////)
69     FORMAT(///' RSL0441  RECORD NO',I5,' PERMANENT READ ERROR'//
      *       10(' RSL0441',16Z6/)///)
71     FORMAT(///' RSL0461  I/O ERROR COUNT EXCEEDED  ',I5,
      *          ' RECORDS READ')
72     FORMAT('1',T32,'RECORD NO.',I5///' SPECTROMETER DATA'//(8I10))
73     FORMAT(////' RADIOMETER DATA'//(8I10))
74     FORMAT(////' MULTIPLEXED DATA'//6I10)
       END
```

5.2 Subroutine Rdtrk

```
         TITLE   'SG-4 SPECTROMETER TAPE READ ROUTINE'
*
         MACRO                          CONVERT FROM PACKED BCD
&L       BCD     &TO,&FROM              TO HALFWORD INTEGER
&L       LH      TEM2,&FROM             PICK UP BCD HALFWORD
         SLDL    TEMP,24                SEPARATE BYTES
         SRL     TEM2,26                GET UNITS DIGIT
         LA      BCD,X'F'               LOAD MASK
         NR      BCD,TEMP               GET HUNDREDS DIGIT
         MH      BCD,=Y(100)            SCALE
         AR      BCD,TEM2               ADD UNITS DIGIT
         LH      TEM2,&FROM             PICK UP BCD HALFWORD
         SR      TEMP,TEMP              CLEAR TEMP
         SLDL    TEMP,20                GET RIGHT HALF OF TENS DIGIT
         SLL     TEM2,10                GET LEFT HALF OF TENS DIGIT
         SRL     TEM2,28                ALIGN LEFT HALF
         OR      TEMP,TEM2              PUT HALFS TOGETHER
         MH      TEMP,=Y(10)            SCALE
         AR      BCD,TEMP               SUM
         STH     BCD,&TO                STORE HALFWORD RESULT
         MEND
*
*
*
         MACRO                          CONVERT FROM SG CODE TO I*2
&L       TENBIT  &TO,&FROM              TO/FROM ARE HALFWORDS
&L       LH      TEM2,&FROM             PICK UP DATA
         SLDL    TEMP,51                GET RID OF SYNC BIT
         SRDL    TEMP,27                ALIGN
         SRL     TEM2,19                ALIGN
         OR      TEMP,TEM2              PUT TOGETHER
         STH     TEMP,&TO               STORE CONVERTED DATA
         MEND
```

```
         MACRO                                    WRITE TO OPERATOR MACRO
&L       WTOP    &ARG,&MF=,&LIMIT=10              WRITE LIMIT NUMBER OF TIMES
         LCLC    &COUNT
         AIF     ('&MF' EQ 'L').LIST              GO SET UP MESSAGE AREA
         AIF     ('&MF' EQ 'E').XEQ               GO PRINT OUT MESSAGE AREA
         AGO     .ERROR                           MACRO FORM ERROR
.XEQ     ANOP
         AIF     ('&ARG' EQ '').ERROR             MESSAGE ADDR MUST BE GIVEN
&COUNT   SETC    'CNT'.'&SYSNDX'                  COUNTER NAME SYMBOL
&L       L       1,&COUNT                         PICK UP COUNT
         BCT     1,*+8                            DECREMENT AND JUMP
         B       &COUNT+4                         IGNORE WRITE REQUEST
         ST      1,&COUNT                         RESTORE COUNTER
         LA      1,&ARG-4                         LOAD ARG LIST POINTER
         SVC     35                               ISSUE WTO SVC
         B       &COUNT+4                         JUMP COUNTER
&COUNT   DC      A(&LIMIT)                        SAVE COUNTER HERE
         MEXIT
.LIST    ANOP                                     DEFINE WTO MESSAGE AREA
         AIF     ('&L' EQ '').ERROR               LABEL SYMBOL NEEDED
         AIF     ('&ARG' EQ '').ERROR             BYTE COUNT NEEDED
         CNOP    0,4                              GET ON A FULLWORD BOUNDARY
         DC      AL2(&ARG+4)                      DEFINE MESSAGE LENGTH FOR OS
         DC      AL2(0)                           REQUIRED BY OS
&L       DC      CL(&ARG)' '                      ALLOCATE BLANK MESSAGE AREA
         MEXIT
.ERROR   MNOTE   8,'RSL2001  PARAMETER INVALID -- NO CODE GENERATED'
         MEND
*
SGNSIZ   EQU     10
SGDSIZ   EQU     48
```

```
         PRINT   NOGEN
RDTRK    LINKS
*************************************************************************
*                                                                       *
*                    SUBROUTINE RDTRK (LRECL)                           *
*                                                                       *
*                                                                       *
*      LRECL    -- SIZE IN BYTES OF CURRENT RECORD, SET TO ZERO        *
*                  ON EOF READS.                                        *
*      SPECTAPE -- DDNAME FOR INPUT DATA SET                           *
*      TDATA    -- FORTRAN COMMON, HALFWORD INTEGERS                   *
*                                                                       *
*COMMON /TDATA/ INPA(200), IDENT(6), SPECT(48), RADIO(48), MULT(6)*
*                                                                       *
*                                                                       *
*      THIS FORTRAN CALLABLE SUBROUTINE READS AND CONVERTS DATA       *
*      READ FROM 7-TRACK MAG TAPE GENERATED BY STANFORDS SG-4         *
*      SPECTROMETER SYSTEM.                                           *
*      THE RAW DATA IS CONTAINED IN TWO DIFFERENT RECORD FORMATS      *
*      EACH OF A DIFFERENT PHYSICAL LENGTH AND DATA RECORDING MODE*
*      THE IDENTIFICATION RECORD CONTAINS DATA IN A PACKED BCD        *
*      FORMAT WHERE EACH PAIR OF SIX BIT BYTES CONTAIN THREE          *
*      FOUR BIT BCD CHARACTERS.                                       *
*                                                                       *
*      TAPE DATA FORMAT:      001FGHIJ 000ABCDE                       *
*      CONVERTED FORMAT:      000000AB CDEFGHIJ                       *
*                                                                       *
*      TAPE BCD FORMAT:       00EF IJKL   00ABCD GH                   *
*      CONVERTED FORMAT:      ABCD + 10*EFGH + 100*IJKL               *
*                                                                       *
*      THE DATA IS RETURNED IN COMMON TO FORTRAN, ALL NUMBERS         *
*      ARE CONVERTED TO 16 BIT TWOS COMPLIMENT INTEGERS.              *
*                                                                       *
*************************************************************************
*
         L       LRECL,0(0,PARM)        GET ARG ADDRESS
         SR      COUNT,COUNT            SET COUNT TO ZERO
         TITLE   'OPEN, READ, CLOSE SECTION'
         TOPEN   SPECTAPE,READ
         OPEN    (SPECTAPE)
         TOPEN   SPECTAPE,READ
         WTO     'RSL1001  SPECTAPE DD CARD MISSING'
         ABEND   20,DUMP
```

```
READ     READ    DECB,SF,SPECTAPE,INPA,'S'
         CHECK   DECB
         LTR     COUNT,COUNT             DID WE GET AN ERROR?
         BM      EXIT                    IF SO EXIT
         L       CBASE,=V(TDATA)         ESTABLISH COMMON BASE REG
         USING   INPA,CBASE
         BAL     LINKR,BLKSIZE           GET BLOCK BYTE COUNT
*
         C       COUNT,=A(SGIDSIZ)       CHECK FOR IDENT RECORD
         BE      IDCONV
         C       COUNT,=A(SGDTSIZ)       CHECK FOR DATA RECORD
         BE      DATACONV
         B       EXIT                    RECORD LENGTH ERROR
*
EODAD    CLOSE   (SPECTAPE,LEAVE)        LEAVE FOR MULTIPLE FILES
EXIT     ST      COUNT,0(0,LRECL)        RETURN LRECL TO FORTRAN
         L       SAVER,4(0,SAVER)
         RETURN  (14,12)
         TITLE   'IDENTIFICATION CONVERSION ROUTINE'
IDCONV   DS      0H
         BCD     DAY,SGDAY
         BCD     TIMEH,SGTIME
         BCD     TIMEM,SGTIME+2
         BCD     SAMPLE,SGSAMPLE
         BCD     SITE,SGSITE
*
         SR      TEMP,TEMP               FIX TIME
         LH      TEM2,TIMEM              LOAD LOWEST THREE DIGITS
         D       TEMP,=F'100'            EXTRACT LOW ORDER TWO DIGITS
         STH     TEMP,TIMES              STORE SECONDS
         ST      TEM2,CSAVE              SAVE LOW ORDER MINUTES DIGIT
         LH      TEM2,TIMEH              LOAD HIGH ORDER THREE DIGITS
         SR      TEMP,TEMP               CLEAR EVEN REGISTER
         D       TEMP,=F'10'             EXTRACT HIGH ORDER TWO DIGITS
         STH     TEM2,TIMEH              SAVE HOUR DIGITS
         MH      TEMP,=H'10'             SCALE HIGH ORDER MINUTE DIGIT
         A       TEMP,CSAVE              ADD LOW ORDER MINUTE DIGIT
         STH     TEMP,TIMEM              SAVE MINUTES
         B       EXIT
CSAVE    DS      F
```

```
          TITLE   'DATA CONVERSION ROUTINE'
DATACONV  DS      0H
          LA      POINT,SGCHANLA          CONVERT SPECT/RADIO DATA
          LA      STEP,4
          LA      LIMIT,SGMULT-4
          SR      INDEX,INDEX
*
DLOOP     DS      0H
          TENBIT  SPECT(INDEX),0(0,POINT)
          TENBIT  RADIO(INDEX),2(0,POINT)
          LA      INDEX,2(0,INDEX)
          BXLE    POINT,STEP,DLOOP
*
          LA      LIMIT,6                 CONVERT MULTIPLEX DATA
          SR      INDEX,INDEX
MLOOP     DS      0H
          TENBIT  MULT(INDEX),SGMULT(INDEX)
          LA      INDEX,2(0,INDEX)
          BCT     LIMIT,MLOOP
          B       EXIT
          TITLE   'ROUTINE TO TURN OFF ERROR RETRY BITS'
          ENTRY   NOERR
*
          USING   *,15
NOERR     OI      SPECTAPE+49,X'0C'
          BR      14                      SHOULD BE CALLED BEFORE OPEN
          DROP    15
          TITLE   'INPUT BLKSIZE ROUTINE'
BLKSIZE   DS      0H
          L       POINT,DECB+16           GET POINTER TO STATUS INFO
          L       TEMP,12(0,POINT)        GET RESIDUAL COUNT
          N       TEMP,MASK               ONLY USE LOW ORDER HALFWORD
          L       COUNT,SPECTAPE+60       GET BLKSIZE FROM DCB
          N       COUNT,MASK              ONLY USE LOW ORDER HALFWORD
          SR      COUNT,TEMP              SUBTRACT REMAINDER
          BR      LINKR                   RETURN
          DS      0F
MASK      DC      X'0000FFFF'
```

```
         TITLE   'READ ERROR ROUTINE'
SYNAD    DS      0H
         SYNADAF ACSMETH=BSAM
         STM     14,1,ERRSAV             SAVE OS REGISTERS
         MVC     STATUS(27),=CL27'RSL1101  I/O ERROR INFO --'
         MVC     STATUS+27(78),50(PARM)
         WTOP    STATUS,LIMIT=20,MF=E
         L       COUNT,=F'-1'            SET ERROR FLAG
         SYNADRLS
         LM      14,1,ERRSAV             RESTORE THE REGISTERS
         BR      14                      RETURN TO CHECK MODULE
*
ERRSAV   DC      4A(0)
STATUS   WTOP    27+78,MF=L
SPECTAPE DCB   DDNAME=SPECTAPE,DSORG=PS,RECFM=U,BLKSIZE=400,MACRF=R,
               EODAD=EODAD,SYNAD=SYNAD
         LTORG
         TITLE   'COMMON DEFINITION'
TDATA    COM
INPA     DS      50D                     400 BYTE INPUT AREA
*
         ORG     INPA                    BACK TO START OF INPUT AREA
SGIDENT  EQU     *                       IDENTIFICATION RECORD FORMAT
SGNOISE  DS      (SGNSIZ)X               NOISE BYTES
SGDAY    DS      2X                      THREE DIGIT (BCD) DAY
SGTIME   DS      4X                      SIX DIGIT TIME  HH.MM.SS
SGSAMPLE DS      2X                      SAMPLE IDENTIFICATION
SGSITE   DS      2X                      SITE IDENTIFICATION
SGIDSIZ  EQU     *-SGIDENT               DEFINE SIZE OF IDENT RECORD
*
         ORG     INPA                    BACK TO START OF INPUT AREA
SGDATA   EQU     *                       DATA RECORD FORMAT
SGCHANLA DS      2X                      FIRST SPECTROMETER HALFWORD
SGCHANLB DS      2X                      FIRST RADIOMETER HALFWORD
         DS      (4*(SGDSIZ-1))X         REMAINDER OF SPECT/RAD DATA
SGMULT   DS      (2*6)X                  MULTIPLEX DATA
SGDTSIZ  EQU     *-SGDATA                DEFINE SIZE OF DATA RECORD
*
         ORG     ,                       RESET LOCATION COUNTER
```

```
IDENT     DS      6H                      ALLOCATE RESULT AREA
*
          ORG     IDENT                   TO DEFINE IDENT SUBFIELDS
DAY       DS      2X                      CONVERTED FROM ABOVE
TIMEH     DS      2X
TIMEM     DS      2X
TIMES     DS      2X
SAMPLE    DS      2X
SITE      DS      2X
*
SPECT     DS      48H                     CONVERTED SPECTRUM AREA
RADIO     DS      48H                     CONVERTED RADIOMETER AREA
MULT      DS      6H                      CONVERTED MULTIPLEX AREA
*
COMLENG   EQU     *-INPA                  LENGTH SHOULD AGREE WITH MAIN
*
          TITLE   'REGISTER DEFINITIONS'
PARM      EQU     1
TEMP      EQU     2
TEM2      EQU     TEMP+1
STEP      EQU     4
LIMIT     EQU     5
LRECL     EQU     6
BCD       EQU     7
COUNT     EQU     8
POINT     EQU     9
INDEX     EQU     10
CBASE     EQU     11
BASE      EQU     12
SAVER     EQU     13
LINKR     EQU     14
          END
```

5.3 Subroutine Daload

```
         PRINT   NOGEN
DALOAD   LINKS
*
*********************************************************************
*                                                                   *
*              SUBROUTINE DALOAD (DATA, KEY)                        *
*                                                                   *
*        DATA   -- LOCATION OF DATA TO BE WRITTEN                   *
*        KEY    -- ERROR CHECKING FEATURE, IF ZERO                  *
*                  NO ERROR CHECKING WILL BE DONE,                  *
*                  ELSE IT MUST AGREE WITH THE KEY                  *
*                  OF THE BLOCK CURRENTLY BEING WRITTEN.            *
*        DIRECT -- DDNAME OF DIRECT ACCESS DATASET.                 *
*                  BLKSIZE MUST APPEAR IN JCL.                      *
*                  DSORG=DA MUST APPEAR IN JCL.                     *
*                                                                   *
*********************************************************************
*
         L       DATA,0(0,PARM)          GET POINTERS TO ARGS
         L       KEY,4(0,PARM)
         CLC     BLKCNT,=F'0'            FIRST TIME THRU?
         BNE     OPENED
         OPEN    (DIRECT,(OUTPUT))
         TOPEN   DIRECT,OPENED
         WTO     'RSL1001  DIRECT DD CARD MISSING'
ABEND    ABEND   20,DUMP
OPENED   L       TEMP,BLKCNT
         LA      TEMP,1(0,TEMP)          INCREMENT BLOCK COUNT
         ST      TEMP,BLKCNT
         S       TEMP,0(0,KEY)           ERROR CHECK
         C       TEMP,BLKCNT             WAS KEY ZERO?
         BE      WRITE
         C       TEMP,=F'0'              WAS KEY EQUAL TO BLKCNT?
         BE      WRITE
         WTO     'RSL1201  DIRECT ACCESS KEY INVALID'
         B       ABEND
WRITE    WRITE   DECB,SF,DIRECT,(2)
         CHECK   DECB
         L       SAVER,4(0,SAVER)
         RETURN  (14,12)
         TITLE   'DATA CONTROL BLOCK'
         PRINT   GEN
DIRECT   DCB     DDNAME=DIRECT,DSORG=PS,OPTCD=C,RECFM=F,MACRF=WL
BLKCNT   DC      F'0'
         TITLE   'REGISTER DEFINITIONS.'
PARM     EQU     1
DATA     EQU     2
KEY      EQU     3
TEMP     EQU     4
TEM2     EQU     5
BASE     EQU     12
SAVER    EQU     13
LINKR    EQU     14
         END
```

DISTRIBUTION LIST

NASA - WASHINGTON	
J. DeNoyer/Arch Park.	1
NASA - HOUSTON	
T. Barnett Larry York/W. E. Hensley	4
Technical Information Dissemination Branch	4
NASA - GODDARD	
Warren Hovis	1
USGS	
W. H. Hemphill	1
K. Watson	1
S. Gawarecki	1
OTHERS	
Branner Library	1
Dave Landgrebe	1
Jack Quade	1
Keenan Lee	1
R. K. Moore	1
V. Myres	1
R. B. MacDonald	1
Files	1
	25