

ESTI File Copy

ESD ACCESSION LIST

ESD-TR-70-32

ESTI Call No. 68967
Copy No. 1 of 2 cys.

DESCRIPTION OF COMPUTER PROGRAMS FOR THE ANALYSIS AND PRESENTATION OF TRADE WINDS DATA



Jerald Schwarz

December 1969

ESD RECORD COPY

RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211

AEROSPACE INSTRUMENTATION PROGRAM OFFICE
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts 01730

This document has been
approved for public release and
sale; its distribution is
unlimited.

(Prepared under Contract No. F19628-68-C-0208 by Syracuse University Research Corporation, Merrill Lane, University Heights, Syracuse, New York 13210.)

AD702530

LEGAL NOTICE

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

DESCRIPTION OF COMPUTER PROGRAMS FOR THE ANALYSIS
AND PRESENTATION OF TRADE WINDS DATA

Jerald Schwarz

December 1969

AEROSPACE INSTRUMENTATION PROGRAM OFFICE
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts 01730

This document has been
approved for public release and
sale; its distribution is
unlimited.

(Prepared under Contract No. F19628-68-C-0209 by Syracuse University Research Corporation, Merrill Lane, University Heights, Syracuse, New York 13210.)



FOREWORD

This report is prepared for the

Aerospace Instrumentation Program Office
Electronic Systems Division
Air Force Systems Command of the United States Air Force
L. G. Hanscom Field
Bedford, Massachusetts

Air Force Program Monitor - Lt. C. Schafer, ESD/ESSIE
Project Number 6684, Task 6684.08

covering research over the period

1969 March 1 to 1969 December 15.

Prepared under Contract No. F19628-69-C-0208 by

Syracuse University Research Corporation
Merrill Lane, University Heights
Syracuse, New York.

This report was reviewed and approved.

C. Schafer, Lieutenant, USAF
Program Manager for ESD/ESSIE/6684.

ABSTRACT

An investigation of the Trade Wind Duct was carried out from 6 March through 25 March 1969 in the Northern part of the Caribbean Sea. An instrumented aircraft was used to record meteorological and radio refractivity data in digitized format for computer analysis. In addition, extensive radio-sonde data was included in the analysis to support the aircraft measurements and provide a basis for weather analysis. In order to assimilate, process and present such a large amount of data it was imperative that machine processing be used. The following report describes the various programs which were used in the analysis and presentation of the data. A ray-tracing program was also developed to analyze radio wave propagation in relation to Trade Wind Duct characteristics. This program has the advantage that horizontal changes in the Duct can be included. Most ray-tracing programs assume that the vertical variation of refractivity is spherically stratified.

TABLE OF CONTENTS

		<u>Page</u>
Section I	Introduction	1
Section II	PAPTOMAG	3
Section III	QUACK	5
	3.1 Input	6
	3.2 Output	11
Section IV	RAWCON	13
	4.1 Computations	13
	4.1.1 Basic Computations	13
	4.1.2 Input and Corrections	15
	4.2 Magnetic Tape Format	17
	4.3 Control Card Parameters	18
	4.4 Secondary Output	25
	4.5 Restriction on Indirect Method for the Computation of Refractivity	25
	4.6 Structure of RAWCON	26
	4.6.1 Input-Oriented Subroutines	26
	4.6.2 Computation-Oriented Subroutines	26
Section V	PLOT	29
	5.1 Operation Procedure Under IBSYS	29
	5.2 Plotting RAWCON Data	29
	5.3 Plotting QUACK Data	30
Section VI	TRACE	31
	6.1 Input Format	31
	6.2 Control Cards	32
	6.3 Interpolation	35
	6.4 Step Size	37
	6.5 Equations	37
Appendix I	Computer Program Listings	39

SECTION I

INTRODUCTION

This report describes the computer programs that were used in processing the data collected for this contract.

A brief summary of this processing is as follows:

PAPTOMAG:	Converts aircraft paper tape to magnetic tape.
QUACK:	Pre-processes radiosonde data.
RAWCON:	Processes output of PAPTOMAG. Converts this to atmospheric profiles.
PLOT:	Plots atmospheric profiles. Use output of QUACK RAWCON.
TRACE:	Performs ray-tracing and produces plots.

All the above except PAPTOMAG are written primarily in FORTRAN IV and run on an IBM 7094. PAPTOMAG is written in assembly language for an SDS-930. QUACK and RAWCON contain small input and utility routines written in assembly language. PLOT and TRACE make use of assembly language routines to produce plotting tapes for a Stromberg-Carlson 4020.

PAPTOMAG, PLOT, and TRACE were written specifically for this project. RAWCON is a modification of a program originally written at MITRE Corporation for an IBM 7030. QUACK was supplied by the sponsor.

SECTION II

PAPTOMAG

This program converts the paper tape containing the airplane measurements to 7-track magnetic tape. It is written in assembly language for an SDS-930 equipped with a paper tape reader and tape drive on the W buffer. The program takes three 8-bit characters from the paper tape and writes them as four 6-bit characters on the tape. The end of the paper tape roll is indicated by five consecutive blank frames. Records on magnetic tape are 1524 characters long. (The last record of a roll may be short.)

The program is organized to put multiple rolls of paper tape on a single magnetic tape. The paper tapes are organized as missions. Several rolls may make up a mission. The missions may be put on the proper tape in any order, but all rolls from a single mission must be put on together and in order. The output tape is on Unit 1.

At various times in the processing, the program will type messages and wait for a response. The only valid responses are Y or N followed by a carriage return. If any other response is given the program will request the response be re-entered.

The following are the messages and appropriate responses.

IS THIS A NEW MAG TAPE.

This message is always typed at the beginning of each program.

Responses:

- | | |
|---|--|
| Y | The program should start at the beginning of the magnetic tape. |
| N | Missions have been put on the magnetic tape previously. The program will skip to the end of the last mission already on the magnetic tape. |

ARE THERE MORE ROLLS FOR THIS MISSION.

ARE THERE MORE MISSIONS FOR THIS MAGNETIC TAPE.

ARE THERE MORE MAGNETIC TAPES.

END OF MAGNETIC TAPE. RESTART LAST MISSION.

No response is needed to this message. Since all rolls of a single mission must be on the same tape, this message is typed when an end of magnetic tape is found. The operator should restart the first roll of the mission currently being converted after readying a new tape 1.

The program will print what is being put on magnetic tape unless sense switch 2 is set. For normal processing, therefore, this sense switch should be set.

Since no identification appears on the tape, it is very important that an accurate record be kept of which paper tapes have been converted.

In case of trouble, the following is relevant for repositioning. There is 1 EOF between each roll of a mission; 2 EOFs between each mission; 3 EOFs at the end of a tape.

SECTION III

QUACK

The purpose of the program "QUACK" is to read and find tropospheric ducts from the B4 hydro tapes, obtained from the Environmental Tactical Air Command (ETAC), located at the Washington Naval Yard. These tapes contain worldwide radiosonde and pilot balloon soundings and aircraft information.

The decks and relative location in the program "QUACK" and a short explanation of their purpose are as follows:

- | | | |
|----|-------|--|
| A. | WORK | Reads the B4 hydro tape and converts the data into a useable form. It also compiles statistical information concerning the soundings for all stations for each month. |
| B. | QT | Calculates the characteristics of tropospheric ducts. If a duct exists the program outputs the location, time, height, thickness, and refractivity gradient of the duct. |
| C. | HGT | Given two readings of pressure, temperature, and dew point, calculates the height difference between these levels. |
| D. | INDX | Given the pressure, temperature, and dew point, calculates the refractivity. |
| E. | DECOD | Unpacks each word of an array into six words each containing one character. |
| F. | HELP | Reads a variable length record until it finds a record mark or until it reads a maximum of 315 words. It counts the number of words read and flags it. Finds an end of file. |
| G. | PR | Flags appropriate counters when examining the pressure levels of a sounding for a given station. The counters are for first pressure LT 850, mandatory levels only, mandatory and significant levels, and mandatory levels and surface pressure. |

3.1 Input

The program uses units B(1) and B(2) for input tape units. The program initially uses unit B(1), and if the operator desires to use another input tape, through proper use of the sense switches, the program will then use unit B(2). If other input tapes are desired, the program then transfers between unit B(1) and B(2). The format of the input data (the B4 hydro tape) follows:

A. CONTENTS AND FORMAT OF UPPER AIR DATA ON B4 HYDRO TAPE

- (1) UNCLASSIFIED
- (2) BCD MODE
- (3) 800 BPI DENSITY, IN UNPACKED FORM
- (4) BLOCKED ONE REPORT PER PHYSICAL RECORD.
LOGICAL RECORDS CONSIST OF 20 DATA WORDS
PLUS A 606060606072 WORD, PSEUDO RECORD MARK
- (5) ALL NUMERIC WORDS ARE RIGHT ADJUSTED WITH
LEAD BLANKS
- (6) THE WORD "PIBAL" REFERS TO ANY REPORT CON-
TAINING ONLY WIND INFORMATION
- (7) DATA ALTERED DURING CHECKING ARE FLAGGED
ACCORDINGLY.
 - (A) RAOB HEIGHTS OR TEMPERATURES (FLAG
LEFT ADJUSTED)
R = RECOMPUTED DATA
E = EXTRAPOLATED DATA
 - (B) WINDS, WHEN INCONSISTENT, ARE REMOVED
AND A "D" REPLACES THE DIRECTION
(RIGHT ADJUSTED).
- (8) ALL DATA IN ALPHA-NUMERIC FORMAT

B. TIME RECORD CONTAINS FOUR WORDS AND IS THE FIRST RECORD ON THE TAPE.

- (1) HOUR - BASIC DATA TIME
- (2) DAY - DAY OF MONTH
- (3) MONTH - NUMERICAL VALUE (1 JAN)
- (4) YEAR - LAST TWO DIGITS (66 = 1966)

NOTE. A "\$" INDICATES A BLANK CHARACTER

C. IDENTIFICATION. IN EACH RECORD, EXCEPT TIME RECORD, THE FIRST 8 DATA WORDS ARE USED TO IDENTIFY THE REPORT BY TYPE, TIME, AND LOCATION.

- (1) TYPE OF REPORT - RAOB\$\$, PIBAL\$\$, AGFT\$\$
- (2) BLOCK AND STATION
 - (A) 00000 IF ROVING SHIP
 - (B) NAME IF PERMANENT SHIP (4YA)
 - (C) IIII FOR LAND STATION
 - (D) 00000 AIRCRAFT WITH NON-SPOT WIND, 77 AIRCRAFT WITH SPOT WIND.
- (3) TIME - HOUR
- (4) DAY - DAY OF MONTH
- (5) MONTH - NUMERICAL (1 JAN)
- (6) LATITUDE - IN HUNDREDTHS OF DEGREES (NO DECIMAL POINT)
- (7) LONGITUDE - IN HUNDREDTHS OF DEGREES (NO DECIMAL POINT)
- (8) ELEVATION - WHOLE METERS (0 FOR ROVING SHIPS AND AIRCRAFT)

D. DATA FORMAT

- (1) RAOBS - 6 WORDS DESCRIBE EACH POINT OF A SOUNDING. ALL POINTS (SIG AND MANDATORY) ARE IN LOGICAL DESCENDING PRESSURE ORDER. THE SIX WORD FORMAT IS REPEATED AS MANY TIMES AS NEEDED FOR THE COMPLETE REPORT.
 - (A) PRESSURE - WHOLE NUMBERS
 - (B) HEIGHT - TENS OF FEET (0 IS SIG POINT)
 - (C) TEMPERATURE - SIGNED TO TENTHS OF DEGREE WITH DECIMAL POINT
 - (D) DEW POINT - SIGNED TO TENTHS OF DEGREE WITH DECIMAL POINT
 - (E) WIND DIRECTION - TO TENS OF DEGREE (0 IS SIG POINT)
 - (F) WIND SPEED - TO WHOLE KNOTS (0 IS SIG POINT)
 - (G) AFTER THE LAST POINT OF THE RAOB COMES THE TROP INFORMATION IN A FOUR WORD FORMAT
 - ((1)) TROP \$\$ - IDENTIFIER
 - ((2)) TROP PRESSURE - WHOLE MB'S
 - ((3)) TROP HEIGHT - TENS OF FEET
 - ((4)) TROP TEMPERATURE - TENTHS OF DEGREE WITH DECIMAL POINT
 - (H) NEXT IS SF (PRESSURE IN A TWO WORD FORMAT).
 - ((1)) SPRESR - IDENTIFIER
 - ((2)) SFC PRESSURE - WHOLE MB'S

(I) AND LAST IS MAXIMUM WIND INFORMATION
IN A FOUR WORD FORMAT. (ONLY WINDS AT
OR BELOW 44000 FEET ARE CONSIDERED.)

((1)) MAXWIND - IDENTIFIER

((2)) HEIGHT - TENS OF FEET

((3)) WIND DIRECTION - TO TENS OF DEGREE

((4)) WIND SPEED - TO WHOLE KNOTS

(J) WORDS 2, 3, AND 4 WILL BE 0 (ZERO) IF NO
WINDS BELOW 44000.

(2) PIBALS. NEXT 26 WORDS FOLLOWING I.D. IN-
FORMATION ARE WINDS FOR THE 13 STANDARD
LEVELS (1000, 850, 700, 500, 300, 250, 200, 150,
100, 050, 030, 010) TWO WORDS PER LEVEL.

(A) WIND DIRECTION - TENS OF DEGREES

(B) WIND SPEED - WHOLE KNOTS

AFTER THE STANDARD LEVEL WINDS ARE THE
WORDS \$TOTAL \$PIBAL. FOLLOWING THIS IS A
COMPLETE LIST OF ALL THE PIBAL WINDS IN A
TWO WORD FORMAT REPEATED AS MANY TIMES
AS NEEDED FOR THE COMPLETE RUN.

(A) HEIGHT - IN THOUSANDS OF FEET

(B) WIND - DIRECTION AND SPEED (\$DD\$FF)

((1)) LEFT 3 CHARACTERS - WIND
DIRECTION IN TENS OF DEGREES

((2)) RIGHT 3 CHARACTERS - SPEED IN
WHOLE KNOTS

(3) ACFT. WORDS AFTER I.D. AS FOLLOWS:

(A) PRESSURE LEVEL - WHOLE MB'S

- (B) HEIGHT OF STANDARD LEVEL - TENS OF FEET (FROM RECCO ONLY)
- (C) TEMPERATURE - TO TENTHS OF DEGREE WITH DECIMAL POINT (FROM RECCO ONLY)
- (D) WIND DIRECTION - TENS OF DEGREES
- (E) WIND SPEED - KNOTS
- (F) TRUE ALTITUDE - HUNDREDS OF FEET
- (G) ICING - \$000RT
R = RATE OF ICING
T = TYPE OF ICING
- (H) FLIGHT CONDITION AND TURBULENCE - \$OWFCBK
W = WEATHER
F = FLIGHT CONDITION
B = TURBULENCE INTENSITY
K = CHARACTER OF TURBULENCE
- PARAMETER REPORTED IN WORDS SEVEN AND EIGHT ARE CONVERTED TO RECCO CODE UNITS WITH EXCEPTION W - 3 DENOTES BROKEN CLOUD COVER
- (I) CLOUDS (LOWEST LAYER REPORTED \$NBBTT)
N = AMOUNT OF CLOUDS
BB = BASES
TT = TOPS
- WORDS 10, 11, AND 12 (SAME FORMAT AS 9) WILL BE USED AS NEEDED TO DESCRIBE MULTIPLE LAYERS.

IF 5 OR MORE LAYERS ARE REPORTED WORD
12 WILL BE THE HIGHEST LEVEL REPORTED.
IN WORD 7 THROUGH 12 THE "=" SYMBOL
WILL DENOTE A MISSING ELEMENT.

E. SEQUENCE OF REPORTS

- (1) ROVING SHIPS (BLOCK AND STATION = 0) BY LATITUDE AND LONGITUDE.
- (2) PERMANENT SHIPS BY NAME (4YA, 4YB, ETC.)
- (3) LAND STATION BY BLOCK AND STATION NUMBER. WHEN A STATION HAS BOTH A RAOB AND PIBAL THE RAOB PRECEDES THE PIBAL.
- (4) AIRCRAFT REPORTS BY LATITUDE AND LONGITUDE.

3.2 Output

There are several outputs produced by QUACK. Only one of them was used in this project.

For each launch processed, three logical binary records are produced on FORTRAN UNIT 11.

RECORD 1. Nine (six-character) words.

Words 1 - 7 are station number, latitude, longitude, station elevation, launch hour, launch day, and launch month. The words are BCD right adjusted with leading blanks. Words 8 - 9 are binary integers which describe the second record.

RECORD 2. 2000 words.

This record contains a 4 x 500 floating point array: each row of four numbers contains temperature ($^{\circ}$ C), dew point ($^{\circ}$ C), height (m), refractivity (M units).

Word 8 of record 1 is the row number of the first row of the array containing valid information. Word 9 of record 1 is the row number of the last row of the array containing valid information.

RECORD 3 is not relevant.

SECTION IV

RAWCON

The index of refraction at a point in the atmosphere may be obtained directly by taking a refractometer reading at that point, or it may be computed indirectly from the temperature, pressure, and humidity values.

The primary function of the program RAWCON is to accept refractometer and atmospheric data from airborne observations and to compute the refractivity from this information. The principal output is then the refractivity as a function of the altitude. The program also produces such output as potential temperature, potential index, vapor pressure, and mixing ratio. It produces both a printed listing and secondary output which can be used for further processing.

The program provides the user with the choice of computing refractivity by either the direct or the indirect method. An option for producing output on punched cards is also provided. The program accepts atmospheric input data through magnetic tape. Any information punched onto paper tape must be converted to magnetic tape before it can be used by RAWCON.

This program is based entirely on a program written at MITRE Corporation. This program is described in MITRE working paper 919¹. This description is largely a reproduction of this working paper. Sections which have not been changed are indicated by an * after the section title.

4.1 Computations

4.1.1 Basic Computations*

Rather than computing the index of refraction, n , this program will deal with the refractivity, N , which is defined by

$$N = (n - 1) 10^6$$

1. Beebe, Otto W., "REFCOL, A Data Reduction Program for the Generation of Refractivity Profiles," Mitre Corporation WP919, 9 November 1966.

The relation between the refractivity, N, and the various atmospheric parameters is given by

$$N = 77.6 \frac{P}{T} + 3.73 \times 10^5 \frac{e}{T^2} \quad (1)$$

where T = Temperature in ° K.

P = Pressure in millibars.

e = Partial pressure of water vapor in millibars.

Suppose that, for the "direct" refractometer calculation, the variation of frequency with respect to refractivity is 9.245 kc/N. Suppose further that:

h_0 = Initial height.

F_s = Reference frequency at h_0 .

N_s = Reference refractivity corresponding to F_s .

F = Frequency observed at height h.

Then the difference in refractivity (ΔN) between height h_0 and height h is given by

$$\Delta N = \frac{F - F_s}{9.245}$$

Refractivity at height h is then:

$$N = N_s - \Delta N \quad (2)$$

From (1) and (2) the vapor pressure e can be computed by

$$e = \frac{T^2 (N - 77.6 \frac{P}{T})}{3.73 \times 10^5}$$

The mixing ratio, r, is defined by

$$r = \frac{.62197e}{P - e}$$

The geopotential height (Z) is computed by the following formulas:

$$\left\{ \begin{array}{l} Z_1 = h_0 \\ Z_{k+1} = Z_k + \Delta Z_{k+1} \\ \Delta Z_{k+1} = \frac{\mu_{k+1} R}{A} \left(1 + \frac{2Z_k}{R} + \frac{\mu_{k+1}}{A} \right) \\ \mu_{k+1} = 14.645 \left[T_k \left(1 + 0.388 \frac{e_k}{P_k} \right) + T_{k+1} \left(1 + 0.388 \frac{e_{k+1}}{P_{k+1}} \right) \right] \ln \left(\frac{P_k}{P_{k+1}} \right) \\ \text{for } k = 1, 2, 3, \dots \end{array} \right.$$

where h_0 = Beginning height.

P_k = Pressure (mb) for k'th reading.

T_k = Temperature ($^{\circ}$ K) for k'th reading.

e_k = Vapor pressure (mb) for k'th reading.

A = Length of semi-major axis (km).

R = Length of semi-minor axis (km).

The potential temperature, T_P , is given by:

$$T_P = T \left(\frac{1000}{P} \right)^{2/7} - 273.16$$

The potential index K is given by:

$$K = N \left(\frac{1000}{P} \right)^{0.714}$$

4.1.2 Input and Corrections*

All parameters which are received from magnetic tape are converted by the input routines into a four digit floating point representation. In order to be of use in the computations, further scaling and corrections must be performed.

The following parameters are received as input from magnetic tape:

1. Time
2. Frequency (Kc, Refractometer No. 1).
3. Frequency (Kc, Refractometer No. 2).
4. Frequency (Kc, Refractometer No. 3).
5. Altitude (radar interval counter).
6. Event.
7. Air Speed.
8. Pressure.
9. KS4 Temperature.
10. EK Temperature.
11. Humidity.
12. Voltage (Refractometer No. 4).
13. Vortex Temperature.

The temperature, air speed and pressure values are in a linear relation with their final floating point representation and are converted by a **linear** function (TLIN). The user establishes the conversion functions which are to be employed. The user also specifies correction constants for these parameters.

The program applies a further correction to the value of the selected temperature probe. Suppose T_S is the value of the selected probe, then

$$T = (T_S + 273.16)/(1 + B_k S)$$

where

$$k = \begin{cases} 1, & \text{if KS4 probe} \\ 2, & \text{if EK probe} \\ 3, & \text{if Vortex probe} \end{cases}$$

β_k is an input parameter = speed corrections for temperature probes

$$S = \frac{SPEED^2}{P}$$

T = temperature in $^{\circ}$ K and will be used in all computations involving temperature.

The following corrections are made to the refractivity computations:

$$\Delta N = \frac{F - F_S}{9.245} + \alpha [T (1 + \beta_4 S) - T_0]$$

α = Temperature correction for cavity expressed in N units / ° C.
(This correction varies from time to time.)

where α , β_4 are correction constants supplied by user.

T_0 = Surface temperature in ° K.

The "uncorrected" value for refractivity is then given by:

$$N = N_S - \Delta N$$

The "wet" term of N is computed by:

$$N_{WET} = C_2 \quad N \frac{C_2}{C_1} - \frac{77.6P}{T}$$

where $C_1 = 1 + 3.5 \beta_5 S$ (β_5 supplied by user.)

$$C_2 = 1 + \beta_5 S$$

The vapor pressure e is then:

$$e = \frac{N_{WET} T^2}{3.73 \times 10^5}$$

and the final "corrected" value for refractivity is given by:

$$N^* = 77.6 \frac{P}{T} + N_{WET}$$

4.2 Magnetic Tape Format

The IBM 7094 has seven track tapes and a 36-bit word. The paper tape has 8-bit characters. The 8-bit characters are packed without any slack bits. Thus, nine characters are packed in two words. The magnetic tape contains no information except a representation of the paper tapes.

Within missions, paper tapes are separated by file marks. Missions are separated by double file marks, and the end of the tape is delineated by a triple file mark.

The tape is set up on IBSYS unit A(1).

The following table gives the legal 8-bit paper tape codes and their meaning. Any other codes which appear on the paper tape are considered errors.

<u>Paper Tape</u> <u>(Octal Representation)</u>	<u>Meaning</u>
001	1
002	2
023	3
004	4
025	5
026	6
007	7
010	8
031	9
040	0
200	End of Line

4.3 Control Card Parameters

Input to the program consists of groups of parameter cards separated by cards containing *END* in columns 1 - 5.

The program will process a part of the input tape according to the values of a large number of parameters. However, the program contains default settings for most of the parameters and once a parameter value is set it continues to have that value until explicitly changed. Thus, generally each group of cards need only contain values of flight parameters.

Succeeding groups of parameter cards must specify data in the same order as it is contained on the tape.

The format of the parameter cards is:

Column 1 - 6	The name of the parameter.
Column 9 - 18	The value of the parameter if the parameter is floating, integer, or a time.
Column 19 - 24	The value of the parameter if the parameter is alpha-numeric or logical.
Column 25 - 54	Comments

An integer value may be expressed either with a decimal point or else right adjusted to column 18.

A time parameter must be expressed as an integer HHMMSS (either with a decimal point or right adjusted to column 18) in which the first two digits are the hour, the next two the minute, and the last two the second.

A logical parameter must be punched as a "T" or "F" in column 19 with columns 20 - 24 blank.

If a card with "*STOP*" in columns 1 - 6 is encountered while reading parameter cards, the program terminates immediately.

The following is an example:

EXAMPLE OF RAWCON CONTROL CARDS

PAGE 1

```

$DATA
PUNCH          T
RADIUS  6370999.
KPAR      0.
R         6339971.
A         6331158.
ITPR0B    3.
IRSCT     2.
CHKFC          T
ZS         0.
ALPHA      0.
T4MIN     -10.
T4MAX      30.
T4VMIN    410.
T4VMAX    880.
EKMIN     -10.
EKMAX      30.
EKVMIN    100.
EKVMAX    650.
VXMIN     -10.
VXMAX      30.
VXVMIN    340.
VXVMAX    820.
PMIN      600.
PMAX     1050.
PVMIN     115.
PVMAX     970.
MISID
TSTART  063140.
TSTOP   095000.
Z0FS1   152.
RNM1    355.
RFS1    2710.
RKP1    294.3
CPRES    1.
*END*
MISID
TSTART  103500.
TSTOP   154000.
Z0FS1   305.
RNM1    382.
RFS1    2511.
RKP1    296.8
CPRES    2.
*END*
*STOP*
CAR001 MARCH 6,1969 KEY WEST
MARCH 6,1969 KEYWEST
MISSION 1
MARCH 6,1969 KEYWEST
MARCH 6,1969 KEYWEST
MARCH 6,1969 KEYWEST
MARCH 6,1969 KEYWEST
CAR003 MARCH 9,1969
MARCH 9,1969
MARCH 9,1969
MARCH 9,1969
MARCH 9,1969
MARCH 9,1969
CAR003

```

The following is a table of all Mission Parameters and the associated default settings:

Code	Type	Default	Description
RFS1	R	1871.0	Reference Frequency of Refractometer.
RFV1	R	0.0	Reference Voltage of Voltage Refractometer.
RNM1	R	316.0	Reference Refractivity.
RKP1	R	285.94	Surface Temperature in ° K.
ANDF1	R	0.0	Refractivity of Dry Air at Surface.
ANWF1	R	0.0	Refractivity of Wet Air at Surface.
ACMRVP	R	0.0	Height (m) above which Mix-Ratio, Vapor Pressure, and Refractivity will be corrected.
CORMR	R	0.0	Correction to Mixing Ratio.
CORVP	R	0.0	Correction to Vapor Pressure.
CORIN	R	0.0	Correction to Refractivity.
ITPROB	I	1	Selection of Temperature Probe. If ITPROB = $\begin{cases} 1, \text{ select KS4 probe.} \\ 2, \text{ select EK probe.} \\ 3, \text{ select Vortex probe.} \end{cases}$
IHUM	I	0	Selection Between Refractometer and Humidity Processing. If IHUM = $\begin{cases} 0, \text{ process refractometer input.} \\ 1, \text{ do not process refractometer but process humidity input.} \end{cases}$
IRSCT	I	1	Selection of Refractometer. If IRSCT = $\begin{cases} 1, \text{ use Refractometer No. 1} \\ 2, \text{ use Refractometer No. 2} \\ 3, \text{ use Refractometer No. 3} \\ 4, \text{ use Voltage Refractometer} \end{cases}$

Code	Type	Default	Description
PUNCH	L	.FALSE.	Option for output through Unit. If PUNCH = .FALSE., no output on Unit 1 .TRUE., no output on Unit 1
CHKFC	L	.TRUE.	Option to check input line length. If CHKFC = .TRUE., reject lines of incorrect length. .FALSE., do not check line length.
KPAR	I	1	Option to process a reading if a parity error is found. If KPAR = 0, do not process reading. 1, process readings with parity error ¹ .
BETA1	R	.0002632	β_1 correction for KS4 Temperature.
BETA2	R	-0.0002106	β_2 correction for EK Temperature.
BETA3	R	-0.0000648	β_3 correction for Vortex Temperature.
BETA4	R	0.0001316	β_4 correction for Refractometer.
BETA5	R	0.0000658	β_5 correction for Refractometer.
ALPHA	R	-0.75	α correction for ΔN .
RADIUS	R	6357000.	Radius of Earth.
R	R	6354120.	Length of Semi-Minor Axis.
A	R	6356363.	Length of Semi-Major Axis.
PROCS	L	.TRUE.	Selection to Process Data If PROCS = .TRUE., then process data .FALSE., do not process.
DUMP	L	.FALSE.	Selection to dump tape input. If DUMP = .TRUE., dump aircraft input. .FALSE., do not dump.
ZS	R	535.4117	Height of surface above Sea Level.

1. If readings with parity errors are processed, then on the printed output the reading number is followed by *.

The following parameters establish the linear conversion functions for Speed, Pressure, and Temperature input (tape).

Code	Type	Default	Description
PVMIN	R	18.	Minimum voltage of pressure probe (mv).
PVMAX	R	1017.	Maximum voltage of pressure probe (mv).
PMIN	R	600.	Pressure corresponding to PVMIN (mb).
PMAX	R	1060.	Pressure corresponding to PVMAX (mb).
SVMIN	R	691.	Minimum voltage of Air Speed probe (mv).
SVMAX	R	1060.	Maximum voltage of Air Speed probe (mv).
SMIN	R	135.	Air Speed corresponding to SVMIN (knots).
SMAX	R	195.	Air Speed corresponding to SVMAX (knots).
T4VMIN	R	190.	Minimum voltage of KS4 temperature probe.
T4VMAX	R	891.	Maximum voltage of KS4 temperature probe.
T4MIN	R	-40.	Temperature corresponding to T4VMIN.
T4MAX	R	35.9	Temperature corresponding to T4VMAX.
EKVMIN	R	278.	Minimum voltage of EK temperature probe.
EKVMAX	R	769.	Maximum voltage of EK temperature probe.
EKMIN	R	-40.	Temperature corresponding to EKVMIN.
EKMAX	R	35.9	Temperature corresponding to EKVMAX.
VXVMIN	R	241.	Minimum voltage of Vortex Temperature probe.
VXVMAX	R	1050.	Maximum voltage of Vortex temperature probe.
VXMIN	R	-40.	Temperature corresponding to VXVMIN.
VXMAX	R	32.	Temperature corresponding to VXVMAX.

The following flight parameters are those likely to change with each request.

Code	Type	Default	Description
MISID	A	None	Mission ID.
TSTART	T	0.	Flight Start Time.
TSTOP	T	235959.	Flight Stop Time.
CPRES	R	0.	Pressure Correction.
CSPEED	R	0.	Speed Correction.
CKS4T	R	0.	KS4 Temperature Correction.
CEKT	R	0.	EK Temperature Correction.
CVXT	R	0.	Vortex Temperature Correction.
ZØFS1	R	914.	Beginning Height.

In addition to these parameters the tape to be processed must be specified. This is done through various variables placed in labeled commons.

The following is a list of the tape-description parameters:

Common	Variable	Description
ZTPDNN	NMISS	Number of Missions on the Tape.
ZTPDNN	NAMES (I)	Name (Number) of the ith Physical Mission.
ZTPDNT	NTMPER (I)	Number of Time Periods in ith Mission. (New time period, if off-the-air for more than one hour.)
ZTPDHL	ISPEC (1,I) ISPEC (2,I)	Start Hour for ith Time Period (0-24). Stop Hour for ith Time Period.
ZTPDMP	MAPT (J)	The Position Number of the jth Input Parameter of a "Reading" where Time is input parameter No. 1, Refractometer 1 is input parameter No. 2, etc. The standard input sequence is the same as listed in section 2-2.
ZNFPL	NFPL	Number of parameters in a reading.

Every airborne "reading" consists of a maximum of 13 parameters. Due to frequent changes in the equipment configuration, these parameters may appear in a sequence other than the standard format. MAPT (j) allows the arbitrary ordering of input parameters on the mission tape, since it provides a mapping to the standard sequence.

4.4 Secondary Output

If parameter PUNCH is true, certain variables are output to FORTRAN Unit 1. There is one record for each "reading," and at the end of each flight a tape mark is written.

The record has the following format:

Column 1 - 6	Count which appears on listing.
Column 7 - 13	Height (in meters).
Column 14 - 20	Refractivity (N units).
Column 21 - 27	Refractivity (M units).
Column 28 - 34	Temperature (° C).
Column 35 - 41	Potential Temperature (° C).
Column 42 - 48	Water Vapor Pressure (mb).
Column 49 - 55	Air Pressure (mb).
Column 56 - 62	Mixing Ratio (g/Kg).

4.5 Restriction on Indirect Method for the Computation of Refractivity*

RAWCON provides two methods for the computation of refractivity. The "direct method" obtains the value of the refractivity directly from a refractometer, while the "indirect method" arrives at the result from various atmospheric parameters such as temperature, pressure and humidity.

The formula used in this case is again

$$N = 77.6 \frac{P}{T} + 3.73 \cdot 10^5 \frac{e}{T^2}$$

where e must be obtained through the mixing ratio and a vapor pressure table (EH20).

Since the currently used humidity probe is functioning in an unreliable fashion, the humidity input has been set to a constant 100%. Thus, any refractivity results obtained by the "indirect method" are based on a humidity parameter of 100%. If, at any future time, it is desired to use the actual observed value for humidity, a minor modification must be made to the subroutine "WET."

4.6 Structure of RAWCON*

RAWCON is a collection of individual subroutines, with each subroutine serving an integral function. This collection of subprograms separates into two categories:

1. Reading and pre-processing of input.
2. Computation and output.

4.6.1 Input-Oriented Subroutines

INPUT	This routine controls "line" input. By a "line" of input we mean one complete set of instantaneous atmospheric readings consisting of time, temperature, pressure,
KHAR	This routine reads and interprets characters.
PINT1	This routine is the mission parameter card interpreter.
RDLINE	This routine reads a "line" of data.

4.6.2 Computation-Oriented Subroutines

REFCOL	This subroutine controls all computation-oriented routines and produces the output.
PAT	Routine to compute pressure, air-speed, and temperature.
WET	This routine computes refractivity, vapor-pressure and mixing ratio by the indirect method.
REFCT	This routine computes refractivity, vapor-pressure and mixing ratio by the direct method.
HEIGHT	This routine performs the geopotential height computations.

The only linkage between the input-oriented routines and the computational routines is in the driver-program AIDA with a call to RAWCON. The only data link between the two categories is a labelled COMMON with the name /INPT/.

SECTION V

PLOT

This program processes meteorological data and produces a plotting tape for the Stromberg-Carlson 4020. The source of the meteorological data is either a tape produced by RAWCON or a tape containing radiosonde data produced by RSONDE. Card input is also used to control what data is plotted.

For each set of readings specified by control cards two frames are produced. The first frame contains refractivity. Two lines are plotted, one in N units and one in M units. The second frame contains three lines
1) Temperature (labeled T); 2) Potential Temperature (labeled θ); 3) Vapor Pressure (labeled E). All these parameters are plotted horizontally with the vertical axis being height with limits of 0m. and 4000 m.

5.1 Operational Procedure Under IBSYS

The tape to be processed must be "set up" as FORTRAN logic unit 1. There are two routines named "INPUT" in the deck. One is used for plotting RAWCON tapes, the other is used for plotting QUACK tapes. The sub-routine for plotting RAWCON data has a deckname of "XINPUT", the one for the QUACK data has a deckname of "XINRAD". Either one of these must be removed or a \$USE IBJOB control card may be used. The format of this card is either

1	16
\$USE	XINPUT (INPUT)
\$USE	XINRAD (INPUT)

The first is used for RAWCON data, the second for RSONDE data.

5.2 Plotting RAWCON Data

The output tape from RAWCON consists of a file for each set of parameters processed. Card input to PLOT consists of one card for each set of data to be plotted. The format of the card is:

<u>Column</u>	<u>Contents</u>
1 - 4	FILE
5 - 6	Number of the file (must be between 1 and 20 with a leading 0 if it is less than 10).
9 - 13	First reading to be plotted (right adjusted number).
14 - 18	Last reading to be plotted (right adjusted number).
20 - 49	Any characters; it is used as a title

The "reading" numbers referred to are the numbers printed in the column headed "READINGS" by RAWCON.

Successive cards must be increasing, i.e., either specify a higher file number or have a first reading number greater than the last reading on the previous card.

Processing is terminated by a card with *STOP* in columns 1 - 6.

5.3 Plotting QUACK Data

The output tape from QUACK is mounted on FORTRAN Unit 1. The program selects only certain stations and days for plotting. These are specified by input cards.

The stations are specified on a group of cards with the following format: 12 fields of 6 characters each. The first field contains the number of stations, succeeding fields contain the station numbers. All fields are right adjusted, blank filled. As many cards as required are read.

The days to be plotted are specified in a manner similar to the above. The first field contains the number of days and the succeeding fields contain the day numbers.

Plot Format

The plot format is controlled by various labeled commons. There are two BLOCK DATA programs included in the current deck to initialize these parameters. XBLK produces two frames for each profile, each frame being approximately 6" x 8". SMLBLK provides alternate values for some of the parameters which result in one frame for each profile with two plots on the frame, each plot approximately 3" x 4".

SECTION VI

TRACE

TRACE is a "ray tracing" program. It can be used to follow the propagation of radio waves through a changing atmosphere. The calculations are based on Snell's law and do not take into account diffraction, scattering, or interference.

Input to the program consists of atmospheric profiles, and control cards. The program will accept multiple profiles and interpolate between them. All profiles must be in the path of the ray being traced; no cross-path interpolation is performed. The on-path interpolation is not linear and the user (see *PROF card) has considerable influence over how it is done.

The control cards allow the user to trace groups of rays at various heights, ranges, and elevation angles. A ray is always reflected from the surface and the user may specify reflection from an elevated "level" as well.

Both printed output and a plotting tape (for a Stromberg-Carlson 4020) can be produced by the program. Printed output consists of a summary of control information, and (as an option) detailed descriptions of each ray's path.

6.1 Input Format

All control cards have the following format:

Column 1	*
Column 2 - 10	Function
Column 11 - 80	Seven parameters in fields of 10 columns. Parameter 1 in Column 11 - 20. Parameter 2 in Column 21 - 30, etc.

"Function" is an alphabetic code to tell what kind of card this is. The parameters are numeric. They may appear anywhere in the appropriate field but a decimal point must be present, even for integer values (e.g., RAY-COUNT). Not all functions use all parameters. There are two other types of cards, the cards which describe the atmosphere (see *PROF card) and the title card (see *TRACE card). The following conventions hold for the units on control cards: Height is always expressed in meters, range in kilometers and elevation in radians. However, when the program prints a range without indicating the units, it is in meters.

Many control cards turn "options" on or off. All options are off at the start of processing and are only changed by control cards.

6.2 Control Cards

*PRINT

Parameters: None

Function: Turns on the printing option.

When this option is on, detailed descriptions of the path of each ray are printed.

*NOPRINT

Parameters: None

Function: Turns off the printing option.

When this option is off only summaries are printed for each ray.

*STOP

Parameters: None

Function: Terminates processing.

*PATH

Parameters: None

Function: Resets the program and prepares it to accept profiles for a new path.

It must appear before any *PROF cards.

*PROF

Parameters: RANGE

Function: Marks the beginning of a profile.

The cards immediately following it describe the atmosphere at the indicated RANGE. The range must be larger than the range of any previous profiles in the same path, (i.e., between *PATH cards the ranges must increase). When the program encounters this card it reads profile description cards until a *PEND card is read. A profile description card has the following format:

Column 1 - 6	Either blank or contains "*LEVEL"
Column 11 - 20	Height
Column 21 - 30	Refractivity in N units at that height.

Within a profile the heights on successive profile cards must be increasing, except that successive cards with identical height and refractivity are allowed. (This allows two levels to appear at the same height.) The profiles of any path may have varying numbers of profile description cards but must all have the same number of *LEVEL cards.

The *LEVEL cards are used to describe the interpolation. Basically, the program divides the atmosphere into blocks bounded in range by the range of the various profiles and in height by lines connecting corresponding levels. See the section on interpolation for a more detailed description.

The program always constructs a level for the first and last height of a profile; if these heights are specified with *LEVEL cards, there will be multiple levels at these heights. This situation will be properly treated by the program.

*PEND

Function: Terminates the reading of profile description cards.

*TRACE

Parameters: START-RANGE, START-HEIGHT, START-ELEVATION,
STOP-RANGE, RAY-COUNT, BUMP-VARIABLE,
DELTA

Function: Initiate tracing of rays.

The number of rays which this card causes to be traced is given by RAY-COUNT. (If RAY-COUNT is 0, one ray is traced.) The first ray has a start height, range and elevation as given by the first three parameters. The start conditions of the other rays are determined by BUMP-VARIABLE and DELTA. For each succeeding ray, DELTA is added to the start range, height, or elevation depending on whether BUMP-VARIABLE is 1, 2, or 3, respectively. Tracing continues with reflections from the surface (if the surface is included in the input profiles). A ray is stopped when its range exceeds STOP-RANGE or its height leaves the range in which the atmosphere is specified. The card immediately following the *TRACE card is a title card. Columns 1 - 30 of the title card are used as a title in various places of the output.

***PLOT**

Parameters: START-RANGE, FRAME-RANGE, BOTTOM-HEIGHT,
TOP-HEIGHT, DENSITY, GRID

Function: Turns on the plotting option.

No plotting occurs when it is read. Rather, when a *TRACE card is processed all rays traced will be plotted together. Rays from multiple *TRACE cards may be plotted together using *HOLD and *ENDHOLD cards. The parameters establish the scale and grid for the plots. If they are omitted, reasonable values are used. More than one frame may be used to plot a set of rays if the range of the rays require it. FRAME-RANGE is the range (in Km) covered by each frame. Rays are plotted only when their height is greater than BOTTOM-HEIGHT, and less than TOP-HEIGHT, and their range is greater than START-RANGE. GRID determines how tall the plots are. It must be between 0 and 950. The plots are taller when it is larger. DENSITY indirectly determines the number of grid lines. It must be between 8 and GRID. There are fewer grid lines when it is larger.

***NOPLOT**

Parameters: None

Function: Turns off the plotting option.

***DELHT**

Parameters: HEIGHT-INCREMENT

Function: Specifies a maximum height difference between succeeding points in the trace.

(Under various circumstances the difference will be less than HEIGHT-INCREMENT, but it will never be more.) This value holds for all succeeding traces until another *DELHT card is encountered. Before a *DELHT card is encountered the maximum difference is 20 meters. For a detailed description of how the next point is chosen in the iteration see the section "Step Size."

***REFLECT**

Parameters: LEVEL, STOP-ATTENUATION, FREQ

Function: Turns on the reflection option.

Rays will be reflected from the level numbered LEVEL. (Note: Since the program automatically adds a level at the first input height the number of the first *LEVEL card is level 2.) At reflection, from the surface or the level, an attenuation is computed. When the strength of the ray falls below STOP-ATTENUATION tracing for that ray stops (STOP-ATTENUATION is given in dB. It may be given as positive or negative. Its absolute value is used.). FREQ is the frequency in MHz to be used in computing the attenuation. If either STOP-ATTENUATION or FREQ is omitted reasonable values are used.

*NOREFLECT

Parameters: None

Function: Turns off the reflection option.

*HOLD

Parameters: None

Function: Delimits start of rays to be collected.

Normally the rays of a single *TRACE card are plotted together. When a *HOLD card is encountered the plotting is suspended but all rays are accumulated.

*HOLDEND

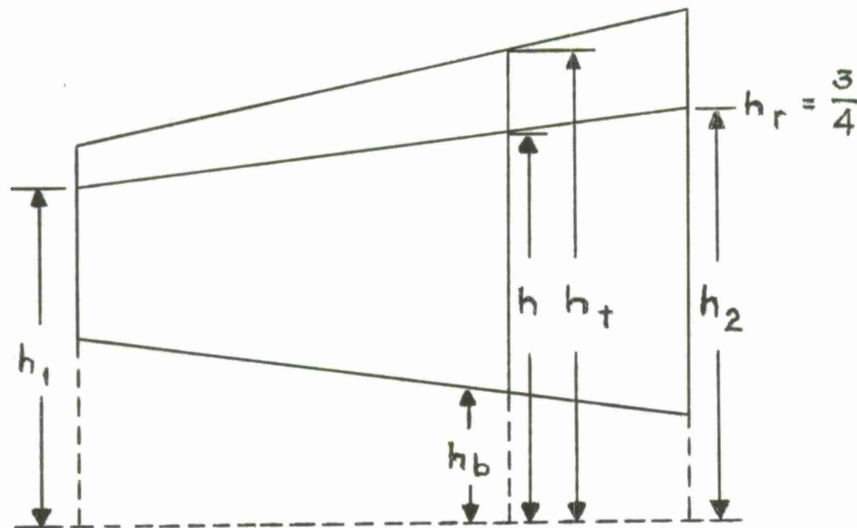
Parameters: None

Function: Delimits end of rays to be collected.

When this card is read all the rays which have been traced since the last *HOLD card are plotted together.

6.3 Interpolation

The interpolation algorithm was motivated by the following consideration: The atmosphere must be modeled in a manner which allows perturbations, such as a layer, to move up and down while the basic profile remains the same. As part of the input profile the user specifies heights which are to be used as levels. The program constructs blocks of the atmosphere bounded by the ranges of the profiles and lines connecting corresponding levels.



To determine the refractivity of an arbitrary point (H , r) the process is as follows:

1. Determine the block in which the point lies.
2. Calculate its "relative height" within the block $h_r = \frac{h - h_b}{h_t - h_b}$ where h_t , h_b are the heights of the top and bottom of the block, respectively, at range r .
3. Determine, at the ranges bounding the block, the refractivity at relative height, h_r . These heights are labeled h_1 and h_2 in the diagram. This is done by linear interpolation between points on the input profiles.
4. The refractivity is calculated by a linear interpolation along the line of relative height h_r , (i.e., the line connecting h_1 and h_2).

6.4 Step Size

The formulas used for each step of the ray are based on an integral over height. If the current height is h , the next height is determined in the following procedure.

1. Add to h either $+\text{DELHT}$ or $-\text{DELHT}$ depending on whether the ray is going up or down. Set h' to this height.
2. On the last iteration the "limits of linearity" were determined as a by-product of the calculation of refractivity. These limits are the heights, at the last range, between which the interpolation model (as described in Section 6.3) gives a linear atmosphere. If either of these heights is between h and h' , set h' to it.
3. If the difference between h and h' is less than one meter, make it one meter.
4. If the ray path has a turning point between h and h' , set h' to the height at which the elevation is 0 and change the direction of the ray. The final value of h' is the next height.

6.5 Equations

The formulas used in the iterative process are due to Colin Gardner. Their derivation, as summarized here, is contained in Pacific Missile Range, Technical Note 3280-6, "Determination of Elevation and Slant Range Errors due to Atmospheric Refraction."

The following is a summary of the derivations.

Notation.

B is the elevation angle.

θ is the earth central angle.

H is height.

N is refractive index.

We assume Snell's law for a spherical earth.

$$N_0 \left(1 + \frac{H_0}{r}\right) \cos B_0 = n \left(1 + \frac{H}{r}\right) \cos B = k$$

Then

$$d\theta = \frac{MQ}{r+H} = \frac{\cot B dH}{r+H} = \left(1 + \frac{H}{r}\right) \frac{dH}{\sqrt{n^2 \left(1 + \frac{H}{r}\right)^2 - k^2}}$$

assuming that $\frac{dN}{dH} = a$ is constant

$$ndH = \left[\frac{n}{a(r+H) + n} \right] \frac{rk \sin B}{\cos^2 B} dB$$

Thus

$$d\theta = \frac{n}{a(r+H) + n} \frac{rk \sin B}{\cos^2 B} dB$$

or

$$\Delta\theta = \int_{B_r}^{B_{r+1}} \frac{n}{a(r+H) + n} dB$$

now $\frac{n}{a(r+H) + n}$ does not vary much with height and we may take

$$\Delta\theta = \frac{\bar{n}}{a(\bar{r} + \bar{H}) + \bar{n}} \Delta B = \frac{\bar{n} \Delta B}{\Delta n (\bar{r} + \bar{H}) + \bar{n} \Delta B}$$

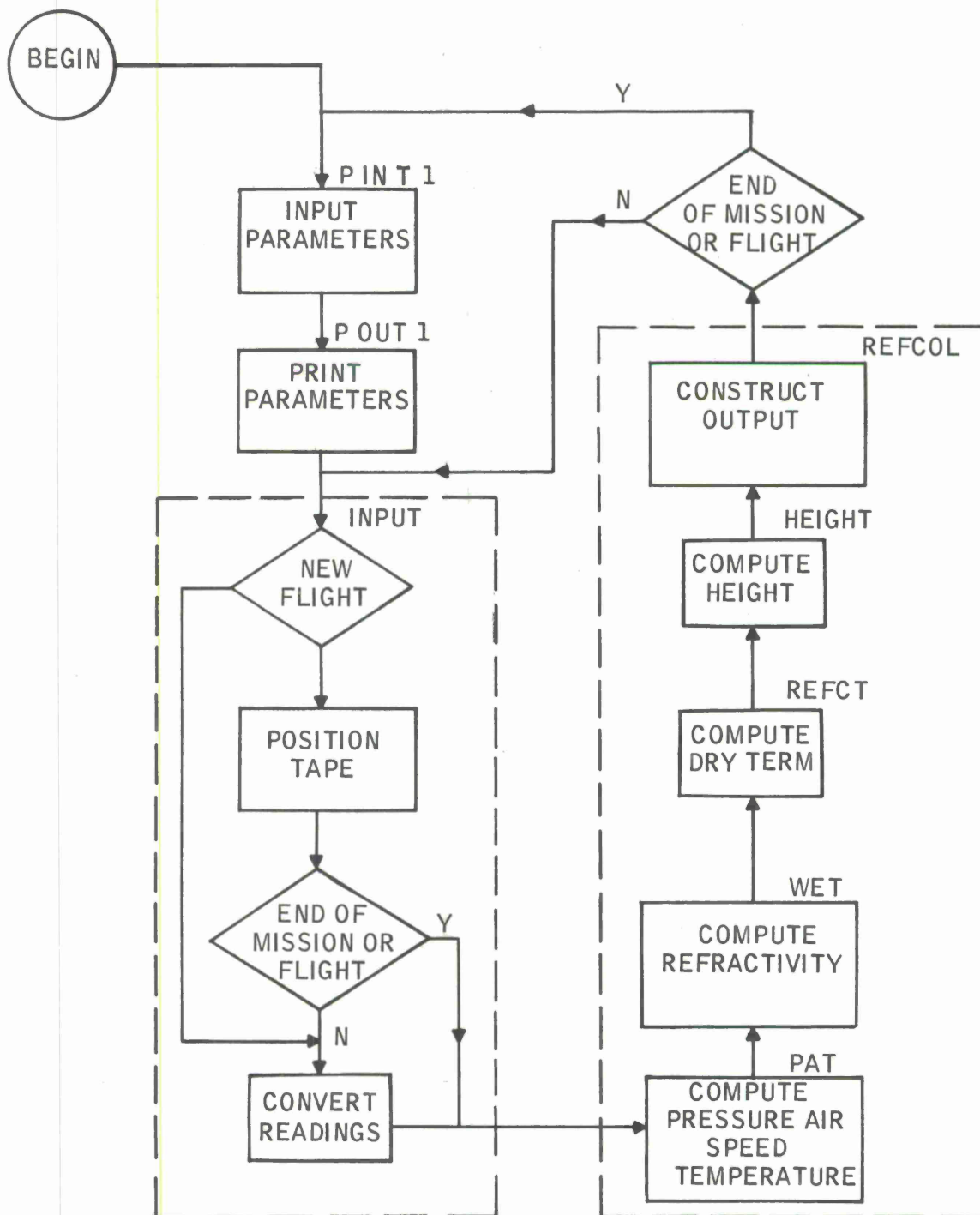
which is the formula used by TRACE.

APPENDIX I

COMPUTER PROGRAM LISTINGS

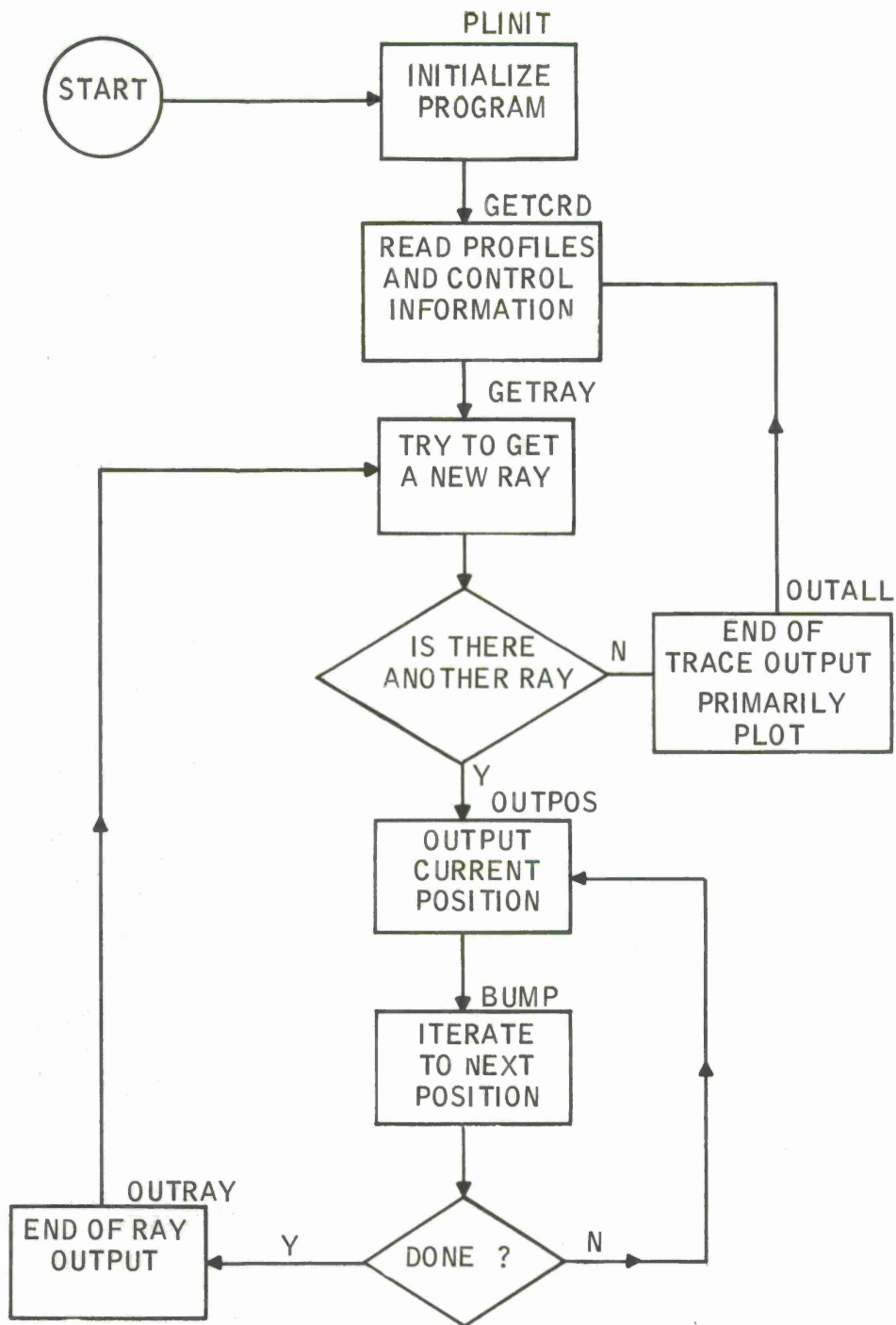
This Appendix contains listings of all the programs described in this report. They do not include the assembly language plotting routines. These are the standard routines written by North American Aviation, and available from Stromberg-Carlson.

In order to accommodate the method of reproduction some cards in these programs are split into two lines. It is always obvious when this has been done.



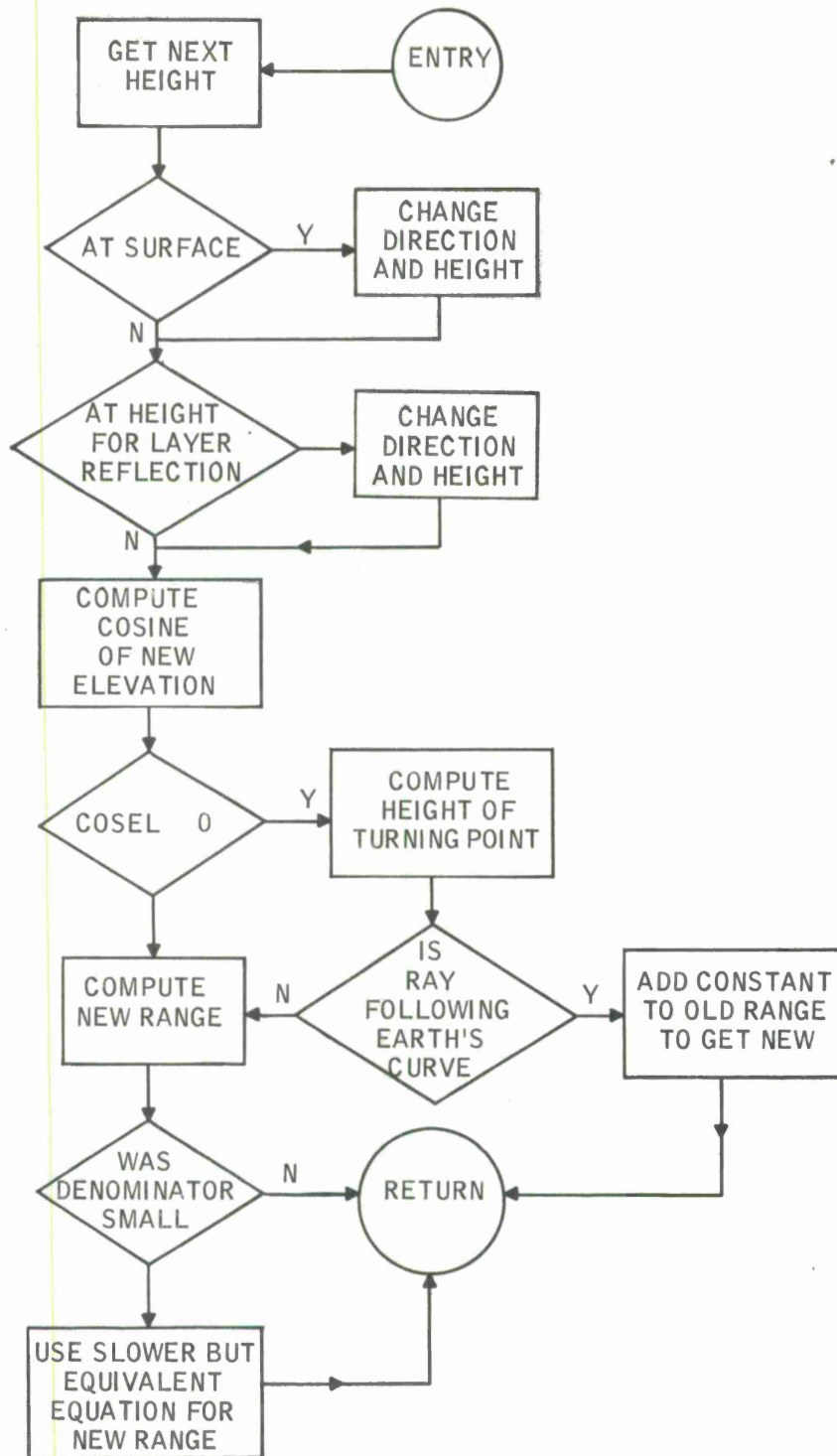
BROAD FLOW CHART OF RAWCON

A7096



OVERALL PROGRAM LOGIC OF TRACE

A7097



SUBROUTINE BUMP IN TRACE

AJ88.

AREWIND B0.

AMETAB920 SI,B0,L0,C0NC,SET.

```

      EXTEND
      RORG      0200
CHARL EQU      36*254/8
SPACELIM EQU    8
      BRU      ENDPT      FOR OPERATOR TO BRU TO.
BEGIN RES      0
      CLA
      STA      RECCNT
      BRM      QUEST
      PZE      POSMES
      BRU      STRTPT

```

*
*
*
*
*
*
*

THIS POSITIONING EFFORT IS DIRECTED.
TOWARD FINDING A TRIPLE EOF. AND
POSITIONING THE TAPE
BETWEEN THE SECOND AND THIRD EOFs.
IT IS MORE ELABORATE THAN SEEMS NECESSARY
PROBLEMS WITH THE TAPE UNITS.

```

SKIPFILE BRM      TAPEWAIT
SF1       BRM      TESTFWD
          BRU      $+2
          BRU      SF1      NOT AN EOF.
          BRM      TESTFWD
          BRU      $+2      2ND EOF.
          BRU      SF1
          BRM      TESTFWD
          BRU      $+2      3RD EOF.
          BRU      SF1
SF2       BRM      TESTBCK
          BRU      SF2
SF3       BRM      TESTFWD
          BRU      $+2
          BRU      SF3
          BRM      TESTFWD
          BRU      STRTPT
          DIR
          HLT

```

```

STRTPT RES      0      START THE PAPER TAPE
          LDA      =BUF  SET UP FOR START OF REC
          STA      PTR
          LDA      =CHARL//3-1
          STA      CHARCT
          LDX      =-3
          STB      SPACES
          CLA

```

	CAT		
	BRU	\$-1	
	EGM	06204	
WIMLDR	WIM	INPT	GET A WORD WHICH MINGT BE LEADER.
	LDA	INPT	
	ETR	=0377	GEY BYTE
	SKE	=0	
	BRU	WIM+1	GO CONTINUE.
	BRU	WIMLDR	
STRTREC	LDA	=BUF	SET UP START OF RECORD
	STA	PTR	PTR TO NEXT WORD IN OUTPUT BUFFER.
	LDA	=CHARL//3-1	
	STA	CHARCT	CHARACTER COUNTER.
	CAT		
	BRU	\$-1	
	EGM	06204	
STRTWD	LDX	=-3	3 8-BIT CHARACTERS TO A WORD.
WIM	WIM	INPT	GET NEXT CHARACTER.
	LDA	INPT	
	ETR	=0377	
	LDB	=-SPACELIM	
	SKE	=0	TEST FOR BLANK TAPE.
	STB	SPACES	
	MIN	SPACES	BUMP COUNT.
	SKN	SPACES	
	BRU	ENDPT	
ASSEM	CLB		
	EXU	LSH,2	ALIGN CHARACTER PROPERLY.
	EXU	MRGNOP,2	MERGE 2ND AND 3RD CHARS.
	STA	*PTR	
	BRX	WIM	GO GET NEX CHARACTER.
	MIN	PTR	WORD IS COMPLETE BUMP IT.
	SKR	CHARCT	CHECK CHAR COUNT.
	BRU	STRTWD	MORE TO GO.
	DSC		WRITE OUT RECORD NOW.
	WTPBIN	1,BUF,4*(CHARL//3)	
	BETP	ENDMT	
	LDA	=CHARL//3-1	
	BRM	PUTREC	
	BRU	STRTREC	
*			
*			
ENDPT			ROLL OF TAPE IS FINISHED. SEE WHAT TO DO NEXT.
	DSC		
	LDA	=CHARL//3	
	SUB	CHARCT	
	STA	CHARCT	

	WTPBIN	1,BUF,(CHARCT)	AND WRITE THAT MANY.
	BETP	ENDMT	
	LDA	CHARCT	
	SUB	=1	
	BRM	PUTREC	
	WTMARK	1	
	BRM	QUEST	FIND OUT
	PZE	EOMMES	IF THERE ARE MORE ROLLSIN
	BRU	STRTPT	THIS MISSION.
			MORE ROLLS.
			ONE EOF IS ENOUGH.
	WTMARK	1	
	BRM	QUEST	
	PZE	EOMMES	ARE THERE MORE MISSIONS.
	BRU	STRTPT	YES
	WTMARK	1	
	REWIND	1	
	BRM	QUEST	ARE THERE MORE MAG TAPES.
	PZE	TAPMES	
	BRU	BEGIN	YES.
	BRU	1	
*	ENDMT	BACKSPACE 1,(2)	
	RTPBIN	1,BUF	READ A RECORD
	BTMK	LSTMS	IF THIS
			IS END OF MSISION BRU
	BRU	ENDMT	OTHW CONTINUE
			BACK SPACEING.
LSTMS	WTMARK	1	PUT ON A 3RD EOF .
	REWIND	1	
	TYPE	REWMES	
	TYPE	REIMES	
	BRU	BEGIN	AND START OVER.
	PAGE		
PUTREC	PZE		OUTPUT A RECORD.
	MIN	RECCNT	BUMP RECORD COUNT
	BPT	2	
	BRR	PUTREC	
	STA	WORDCNT	
	BLANK	LINE,33	
	MOVE	RECMES,1,LINE,2,6	
	BINBCD	LINE,9,5,RECCNT	
	PRINT	LINE,10	PRINT HEADER.
	LDA	=BUF	
	STA	PTR	START AT
			BEGINNING OF BUFFER
	LDA	=-1	
	STA	NEWLINE	FORCE NEW LINE.
PUTL	LDX	=-3	

	LDA	*PTR	
	EXU	RSH,2	
	ETR	=0377	
	BRM	PUTCNV	
	BRX	PUTL+1	
	MIN	PTR	BUMP POINTER IN BUFFER.
	SKR	WORDCNT	BUMP COUNT.
	BRU	PUTL	KEEP GOING.
	SKN	NEWLINE	STARTING A NEW LINE.
	BRU	\$+2	NO.
	BRR	PUTREC	YES. RETURN.
	PRINT	LINE	
	BRR	PUTREC	
* PUTCNV	PZE		CONVERT A CHARACTER
	STA	TEMP1	SAVE WORD
	SKN	NEWLINE	SHOULD
	BRU	SHIFT	WE START A NEW LINE.
	LDA	=LINE	NO.
	STA	LINEPTR	YES RESET PTR.
	MIN	NEWLINE	
	STX	TEMP2	RESET INDICATOR.
	BLANK	LINE,33	
SHIFT	LDX	TEMP2	
	RES	0	
	LDA	TEMP1	
	ETR	=07	
	STA	*LINEPTR	LEAST SIGNIFICANT
			8CTAL DIGIT.
	LDA	TEMP1	
	LSH	3	
	ETR	=0700	NEXT 8CTAL DIIT
	MRG	*LINEPTR	
	STA	*LINEPTR	
	LDA	TEMP1	
	LSH	6	
	ETR	=070000	LAST DIGIT.
	MRG	*LINEPTR	
	MRG	=! 000!	
	STA	*LINEPTR	FINE FOR M OF WORD.
	MIN	LINEPTR	BUMP PTR.
	LDA	LINEPTR	
	SKG	=LINE+29	END OF LINE.
	BRR	PUTCNV	NO.
	PRINT	LINE	YES.
	SKR	NEWLINE	
	BRU	\$-1	IN CASE OF ERROR.
	BRR	PUTCNV	

QUEST	PAGE		
	PZE		ASK QUESTIONS.
	LDA	QUEST	
	ETR	=037777	EXTRACT ADDR.
	ADD	=040001	TURN ON
			INDR BIT AND BUMP.
	STA	QUEST	
	TYPE	*QUEST	
INQ	TYPEIN	BUF	GET RESPONSES.
	LDA	BUF	
	SKE	=1Y	YES RESPONSE
	BRU	\$+2	NO.
	BRR	QUEST	YES RETURN.
	SKE	=1N	NO RESPONSE.
	BRU	INGNG	NO. TRY AGAIN.
	MIN	QUEST	TAKE NEGATIVE EXIT.
	BRR	QUEST	
INGNG	TYPE	NGMES	
	BRU	INQ	
TESTFWD	PZE		
	CAT		WAIT FOR CHANNEL
	BRU	\$-1	
	SKN	DIR	IF TAPE IS GOING FORWARD
	BRU	\$+2	DONT HAVE TO
			WAIT FOR IT TO COME RDY
			OTHW. WAIT
	BRM	TAPEWAIT	
	SFB	0,1,4	
	WIM	BUF	
	DSC		
	CLA		INDICATE GOING FORWARD
	STA	DIR	
	TFT		EOF
	BRR	TESTFWD	
	MIN	TESTFWD	NO.
	BRR	TESTFWD	
TESTBCK	PZE		
	CAT		
	BRU	\$-1	
	SKN	DIR	CHECKDIRECTION
	BRM	TAPEWAIT	GOING FORWARD.
	SRB	0,1,4	
	WIM	BUF	
	DSC		
	LDA	\$-1	INDICATE GOING BACKWARD
	STA	DIR	
	TFT		TEST FOR EOF.
	BRR	T+STBCK	
	MIN	TESTBCK	NO.
	BRR	TESTBCK	

TAPEWAIT	PZE	
	TRT	0,1
	BRR	TAPEWAIT
	BRU	\$-2
	RSH	16
	RSH	8
	NOP	
RSH	EQU	\$
	LSH	16
	LSH	8
	NOP	
LSH	EQU	\$
	NOP	
	MRG	*PTR
	MRG	*PTR
MRGNOP	EQU	\$
EOMMES	TEXT	<ARE THERE ANY MORE ROLLS FOR THIS MISSION >
EOTMES	TEXT	<ARE THERE ANY MORE MISSIONS FOR THIS MAG TAPE >
RECMES	TEXT	<RECORD>
TAPMES	TEXT	<ARE THERE MORE MAGNETIC TAPES >
PBSMES	TEXT	<IS THIS A NEW MAG TAPE >
REWMES	TEXT	<END OF MAGNETIC TAPE. MOUNT NEW TAPE, AND RESTART >
RE1MES	TEXT	<CURRENT MISSION. >
NGMES	TEXT	<LEGAL RESPONSES ARE Y OR N. >
LINE	RES	33
TEMP1	PZE	
TEMP2	PZE	
DIR	PZE	
WORDCNT	PZE	
RECCNT	DATA	0,0
NEWLINE	PZE	
LINEPTR	PZE	
CHARCT	PZE	
PTR	PZE	
SPACES	PZE	
XSAV	PZE	
ASAV	PZE	
BSAV	PZE	
INPT	PZE	
BUF	RES	500
	END	BEGIN
ΔEOF.		
ΔENDJOB.		
ΔJOB.		
ΔREWIND BG.		
ΔLBAD 0,00.		


```

SIBFTC QUACK      N0DECK
C  PROGRAM TO CONTROL INPUT OUTPUT DEVICES
C  CALLS QUACK WHICH PROCESSES THE DATA
C  COMMON          INPUT - DATA READ INTO
C                  LC0DE - ARRAY IN WHICH UNPACK THE DATA
C  COMMON /X/ M0N(13) - MONTHS TO BE
C                  PROCESSED WITH OVERFLOW SLOT
C                  INUM - NUM 0F MONTHS TO BE PROCESSED, MAX 13
C                  IAREA(5,4) - AREAS TO
C                  BE PROCESSED, WITH OVERFLOW SLOT
C                  INUMX - NUM 0F AREAS TO BE PROCESSED, MAX 4
C                  ISUM(4,4,99) - 4 GROUPS
C                  0F CHARACTERISTICS FOR EACH AREA
C                  IST(200,30) - STATION AREA
C                  NUMST - NUM 0F STATIONS, MAX 200
C                  IXXI - NUMBER 0F SOUNDINGS PROCESSED,
C                  ZERO AT START 0F NEW MONTH
C  DIMENSION INPUT(316) 315 WORDS PLUS OVERFLOW, MAX 315
C                  LC0DE(6,315) 6*315 WORDS - UNPACK ARRAY
C                  IDAY(12) - NUM DAYS IN
C                  MONTH IN A FORMAT, WHERE MONTH IS
C                  GIVEN BY THE INDEX
C
C
C
C
COMMON INPUT,LC0DE
COMMON /X/ M0N(13),INUM,IAREA(5,
            4),INUMX,ISUM(4,4,99),IST(200,30),
1NUMST,IXXI
DIMENSION INPUT(316),LC0DE(6,315),IDAY(12)
DATA IDAY(1),IDAY(2),IDAY(3),IDAY(4),
      IDAY(5),IDAY(6),IDAY(7),
1IDAY(8),IDAY(9),IDAY(10),IDAY(11),IDAY(12)/
26H   31,6H   28,6H   31,6H
      30,6H   31,6H   30,6H   31,
36H   31,6H   30,6H   31,6H   30,6H   31/
      WRITE(6,300)
300 FORMAT(1H1)
C
C  INITIALIZE MONTH AND AREA COUNTER
      INUM=0
      INUMX=0
C
C  READ IN MONTHS TO BE PROCESSED IN TIME ORDER
C  SETS MONTH COUNTER, INUM MIN=1 MAX=12 IF 0THER STOP
C  BLANK CARD MARKS END 0F DATA
      DO 18 I=1,13
      INUM=INUM+1
      READ(5,4)M0N(INUM)
      4 FORMAT(I2)

```



```
18 IF(MON(INUM).EQ.0)GOTO5
888 PRINT 5013
5013 FORMAT(/,23H MONTH REQUEST IN ERROR,/)
STOP
5 INUM=INUM-1
IF(INUM.LT.1)GOTO888
PRINT 19,(MON(I),I=1,INUM)
19 FORMAT(/,24H MONTHS TO BE PROCESSED,,12(I3),/)
C
C READ IN AREAS TO BE PROCESSED
C SET ATEA COUNTER, INUMX MIN=1 MAX=4 IF OTHER STOP
C BLANK CARD MARKS END OF DATA
DO 218 I=1,5
INUMX=INUMX+1
READ(5,6)(IAREA(INUMX,J),J=1,4)
6 FORMAT(4I6)
218 IF(IAREA(INUMX,1).EQ.0)GOTO15
1888 PRINT 5113
5113 FORMAT(/,29H LAT AND LOG REQUEST IN ERROR,/)
STOP
15 INUMX=INUMX-1
IF(INUMX.LT.1)GOTO1888
PRINT 119,((IAREA(I,J),J=1,4),I=1,INUMX)
119 FORMAT(/,22H AREAS TO BE LOOKED
AT,/,3X,4HLAT1,3X,4HLAT2,3X,
14HLOG1,3X,4HLOG2,/,4(4I7,/,/),/)
C
C INITIALIZE ALL FLAGS AND COUNTERS BEFORE STARTING
ITAPE=0
IFLAG=0
IXXI=0
ICNT=1
PRINT 9123
9123 FORMAT(34H NEED BLANK TAPE ON UNIT B5 AND B6,/,
122H MOUNT TAPE ON UNIT A5,/)
GOTO24
C
C REWIND PREVIOUS INPUT TAPE AND WRITE
OPERATOR INSTRUCTIONS
C OPEN NEW FILE, SET FLAG FOR NEXT INPUT TAPE (IMT)
C INCREMENT INPUT TAPE COUNTET (ITAPE)
C START OF LOOP
10 IF(IMT.EQ.-1)GOTO22
REWIND 10
PRINT 5011
5011 FORMAT(/,22H MOUNT TAPE ON UNIT A5,
1/,24H UNMOUNT TAPE ON UNIT A6,/)
24 CALL OPEN1
IMT=-1
```

```
    ITAPE=ITAPE+1
    GOT0212
22  CONTINUE
    REWIND 9
    PRINT 5002
5002 FORMAT(/,22H MOUNT TAPE ON UNIT A6,
1/,24H UNMOUNT TAPE ON UNIT A5,/)
    CALL OPEN2
    IMT=1
    ITAPE=ITAPE+1
C
C  READY TO PROCESS DATA
C  IF IFLAG 0, BEGINNING OF RUN
212 IMON=M0N(ICNT)
    CALL QUACK(IMT,IMON,IFLAG)
C
C  IXXI, TOTAL NUMBER OF SOUNDINGS USED FOR PRESENT MONTH
C  IF IFLAG 2, MONTH REQUEST DOES NOT MATCH DATA
C  IF IFLAG 1, NEED NEW MONTH
C  IF IFLAG -1, NEED NEW TAPE
C  IF IFLAG -2, COULD NOT FIND STARTING MONTH
    PRINT 6082,IFLAG,NUMST,IXXI
6082 FORMAT(/,5H FLAG,I3,5X,19H NUMBER OF STATIONS,I5,5X,
120H NUMBER OF SOUNDINGS,I5,/)
    IF(IFLAG.EQ.2)GOT02778
    IF(IFLAG.EQ.1)GOT0602
    IF(IFLAG.EQ.-1)GOT0603
C
C  STARTING MONTH NOT ON TAPE
    PRINT 1819
1819 FORMAT(/,27H STARTING MONTH NOT
ON TAPE,/,14H FORCED FINISH,/)
    GOT0816
C
C  MONTH REQUEST DOES NOT MATCH INPUT DATA
2778 ICNT=ICNT-1
    PRINT 2887,(M0N(I),I=1,ICNT)
2887 FORMAT(/,55H NEXT MONTH REQUEST
DOES NOT MATCH WITH NEXT INPUT DAT
1A,/,19H PROCESSED MONTHS, ,12(I3),/)
    GOT0816
C
C  NEED NEW MONTH
602 CALL OUTPUT(IMON)
    IF(ICNT.EQ.INUM)GOT01777
C
C  PROCESS NEXT MONTH
    ICNT=ICNT+1
    IXXI=0
```

G0T0212

C
C FINISHED ALL MONTHS DESIRED BUT ADDITIONAL INPUT

1777 PRINT 1778,(M0N(I),I=1,ICNT)

1778 F0RMAT(/,49H ADDITIONAL INPUT

BUT FINISHED ALL MONTHS DESIRED,

1/,8H M0NTHS ,12(I3),/)

G0T0816

C
C NEED NEW TAPES

603 PRINT 729

729 F0RMAT(/,53H KEY 35 DOWN F0R ADDITIONAL
INPUT, UP T0 FINALIZE J0B,

1/)

PAUSE

CALL KEYS(W0RD)

IF(IBIT(W0RD,35).EQ.1)G0T010

C
C NO ADDITIONAL INPUT

C CHK SUFF DAYS PRES T0 CNT MONTH OR 0NLY 0NE M0 T0 PR0CESS
IF(INPUT(4).NE.IDAY(IM0N).AND.INUM.NE.1) G0 T0 705

C
C FINISHED PRESENT MONTH

CALL 0UTPUT(IM0N)

IF(ICNT.LT.INUM)G0T0704

C
C ALL MONTHS

PRINT 2771,(M0N(I),I=1,ICNT)

2771 F0RMAT(/,30H FINISHED ALL MONTHS DESIRED, ,12(I3),/)

G0T0816

C
C PR0CESSED SOME MONTHS

705 ICNT=ICNT-1

IF(ICNT.LT.1)G0T0614

704 PRINT 701,(M0N(I),I=1,ICNT)

701 F0RMAT(/,24H PR0CESSED SOME MONTHS, ,12(I3),/)

G0T0816

C
C NO 0UTPUT

614 PRINT 714

714 F0RMAT(/,30H NO 0UTPUT, INPUT INSUFFICIENT,/)

C
C TAKE CARE 0F INPUT 0UTPUT DEVICIES

816 IF(IMT.EQ.1)G0T0811

REWIND 9

END FILE 11

END FILE 12

REWIND 11

REWIND 12

```

      PRINT 2001, ITAPE
2001 FORMAT(/,22H UNMOUNT A5, B5 AND
           B6,/,27H NUMBER OF INPUT TAPES USE
1D,I3,/,9H FINISHED,/)
      STOP
811 REWIND 10
      END FILE 11
      END FILE 12
      REWIND 11
      REWIND 12
      PRINT 2002, ITAPE
2002 FORMAT(/,22H UNMOUNT A6, B5 AND
           B6,/,27H NUMBER OF INPUT TAPES USE
1D,I3,/,9H FINISHED,/)
      STOP
      END
$IBFTC WORK      N0DECK
C  PROGRAM TO CHANGE FORMAT OF SOUNDING FROM ALPHANUMERIC TO
C  FLOATING PT AND INTERGER.      COMPILIES
C          STATISTICS ON THE QUALITY
C  OF THE STATION.  CALLS DUCT WHICH
C          LOOKS FOR LAYERS AND COMPILIES
C  STATISTICS.      CHECKS DATA TO SEE
C          IF PROCESSING MONTH OF DATA
C  DESIRED, IF NOT RETURN.
C  DIMENSION INPUT(316),KPUT(21,15) INPUT ARRAY
C          LCODE(6,315),FCODE(6,315)
C          USED TO UNPACK AND EVALUATE
C          LEVEL DATA
C          IEND(6,9),FEND(6,9) USED
C          TO UNPACK AND EVALUATE
C          END DATA
C          ILT(12) USED AS A GENERAL UNPACKING ARRAY
C  DATA IRA0B ALPHANUMERIC RAOB IDENTIFIER
C          IW XV OCTAL EQUIVALENT RECORD MARK
C
      SUBROUTINE QUACK(IJLM,IM0N,IFLAG)
      COMMON INPUT,LCODE
      COMMON /X/ M0N(13),INUM,IAREA(5,
           4),INUMX,ISUM(4,4,99),IST(200,30),
1INUMST,IXXI
      DIMENSION INPUT(316),KPUT(21,15),
           LCODE(6,315),FCODE(6,315),
1IEND(6,9),FEND(6,9),ILT(12)
      EQUIVALENCE (INPUT,KPUT),(LCODE,FCODE),(IEND,FEND)
      DATA IRA0B,IW XV/6HRA0B ,0606060606072/
C
C  IF IFLAG 0, LOOKING FOR FIRST MONTH DESIRED
C  IF IFLAG -1, NEW INPUT TAPE

```

```
C IF IFLAG 1, NEW MONTH
C
C
C START
  IF(IFLAG.EQ.-1)GOTO2
C
C INITIALIZE FOR PROCESSING OF NEW MONTH
  IXFLA=0
  NUMST=0
C
C ZERO ISUM ARRAY
  DO 1101 I=1,INUMX
  DO 1101 J=1,4
  DO 1101 K=1,99
1101 ISUM(I,J,K)=0
  CALL OUTX(1,0.,0.,0.,0.)
  IF(IFLAG.EQ.1)GOTO312
C
C START OF LOOP
  2 CONTINUE
C
C REMOVE C FROM FOLLOWING CARD IF WANT
  TO STOP PROCESSING AFTER A
  A GIVEN NUMBER OF SOUNDINGS
C   IF(IXXI.GT.   )GOTO4076
C
C READ RECORDS
  CALL IN(L,L0CK,IJLM)
  IF(IFLAG.EQ.0.AND.L0CK.EQ.1)GOTO1333
  IF(L0CK.EQ.1)GOTO778
C
C TEST TO SEE IF WANT TO EVALUATE LEVEL
  DATA OF PRESENT SOUNDING
312 IF(INPUT(1).NE.IRA00)GOTO2
  CALL DECODE(INPUT(5),ILT(1),1)
  IF(ILT(5).EQ.48)ILT(5)=0
  IM0NX=10*ILT(5)+ILT(6)
  IF(IM0NX.NE.IM0N)GOTO4062
C
C EVALUATE LEVEL DATA
C
C TEST FOR LAST RECORD MARK
  IF(INPUT(L-1).EQ.IWXV.AND.INPUT(L).EQ.IWXV)L=L-1
  IF(INPUT(L).NE.IWXV)L=L+1
C
C INITIALIZE OVERFLOW FLAG
C CHECK SIZE OF SOUNDING THAT CAN READ IN
  JFLAG=0
  IF(L.LE.315)GOTO808
```

```

C
C FLAG IF NOT READ IN COMPLETE SOUNDING
C OVERFLOW, HAVE AT LEAST 48 DATA LEVELS,
      BUT DO NOT HAVE END INFO
      JFLAG=88
      L=322
C
C IREC, NUMBER OF COMPLETE LOGICAL RECORDS
C IREM, NUMBER OF DATA WORDS REMAINING LESS END DATA
808 IREC=L/21
      IREM=MOD(L,21)-11
      IF(IREM.EQ.-11)IREM=-10
      IF(IREM)28,28,3
28 IREC=IREC-1
      IREM=IREM+20
C
C DECODE THROUGH LAST LEVEL DATA POINT
3 CALL DECODE(INPUT(9),LCODE(1,1),12)
      K=13
      IF(IREC.LT.2)GOTO4
      DO 10 I=2,IREC
      CALL DECODE(KPUT(1,I),LCODE(1,K),20)
10 K=K+20
      4 CALL DECODE(KPUT(1,IREC+1),LCODE(1,K),IREM)
C
C NUMBER OF PRESSURE LEVELS WITHIN THE SOUNDING
C DO NOT EXCEPT SOUNDING IF LESS THAN 4 PRESSURE LEVELS
      LL=(L-(19+(L/21)))/6
      IF(IREM.EQ.10)LL=(L-(18+(L/21)))/6
      L=LL
      IF(L.LT.4)GOTO2
C
C EVALUATE LEVEL DATA
C N CONTROLS DATA WITHIN EACH LEVEL
C NN CONTROLS NUMBER OF LEVELS PROCESSED
C II WORD WITHIN DATA
C JJ LEVEL OF DATA
      N=0
      NN=0
      DO 99 J=1,315
      IF(J.EQ.N+7)N=N+6
      II=J-N
      JJ=1+NN
      IF(JJ.GT.L)GOTO98
      IF(II/6.EQ.1)NN=NN+1
      DO 12 I=1,6
      IF(LCODE(I,J).EQ.16)LCODE(I,J)=0
12 IF(LCODE(I,J).EQ.48)LCODE(I,J)=0
      MARK=0

```



```
      DO 11 I=1,5
      IF(LCODE(I,J).EQ.32)GOTO563
11  CONTINUE
      GOTO141
563  MARK=1
      LCODE(I,J)=0

C
C  CHECK FOR FLOATING PT
141  IF(LCODE(5,J).EQ.27)GOTO31
C
C  CHECK FOR D IN FIRST CHARACTER
      IF(LCODE(6,J).EQ.20)GOTO32
      LCODE(II,JJ)=LCODE(6,J)+10*LCODE(5,J)+100*LCODE(4,J)
      1+1000*LCODE(3,J)+10000*LCODE(2,J)
      IF(MARK.EQ.1)LCODE(II,JJ)=-LCODE(II,JJ)
      GOTO99
31  CONTINUE
      FCODE(II,JJ)=FLOAT(LCODE(6,J))*1+FLOAT(LCODE(4,J))
      1+FLOAT(LCODE(3,J))*10.0+FLOAT(LCODE(2,J))*100.0
      IF(MARK.EQ.1)FCODE(II,JJ)=-FCODE(II,JJ)
      GOTO99

C
C  REPLACE WIND DIRECTION WITH 999999 IF INCONSISTENT
32  LCODE(II,JJ)=999999
99  CONTINUE

C
C  SEE IF EXCEPTABLE VALUES FOR PRES, TEM, AND DEW PT
C  IF NOT, DO NOT CONSIDER SOUNDING
      DO 6099 J=1,LL
      IF(LCODE(1,J).GT.1100.0R.LCODE(1,J).LT.0)GOTO2
      IF(FCODE(3,J).GT.75.0R.FCODE(3,J).LT.-100.0)GOTO2
      IF(FCODE(4,J).EQ.99.0)GOTO6099
      IF(FCODE(4,J).GT.FCODE(3,J).0R.FCODE(4,
      J).LT.-100.0)GOTO2
6099 CONTINUE

C
C  CHECK PRESSURE OF FIRST FOUR LEVEL
      DATA POINTS TO SEE IF WILL
C  EXECPT SOUNDING
      98 DO 7389 I=1,4
      7389 IF(LCODE(1,I).LT.100)GOTO2
C
C  WILL EXCEPT SOUNDING, INCREMENT SOUNDING
      COUNTER, CLEAR IFLAG
      IXXI=IXXI+1
      IFLAG=99

C
C  DECODE END DATA
      IF(JFLAG.EQ.88)GOTO767
```

```

      KK=IREM+2
      IF(KK.GT.20)GOTO511
      K=21-KK
      IF(K.GT.9)K=9
      CALL DECODE(KPUT(KK,IREC+1),IEND(1,1),K)
      IF(K.EQ.9)GOTO46
      CALL DECODE(KPUT(1,IREC+2),IEND(1,K+1),9-K)
      GOTO46
511 CALL DECODE(KPUT(KK-20,IREC+2),IEND(1,1),9)
C
C  EVALUATE END DATA
46  D0 76 I=1,9
      D0 248 II=1,6
      IF(IEND(II,I).EQ.16)IEND(II,I)=0
248 IF(IEND(II,I).EQ.48)IEND(II,I)=0
      IF(I.EQ.4)GOTO76
      IF(I.EQ.6)GOTO76
      IF(I.EQ.3)GOTO74
      IEND(1,I)=IEND(6,I)+IEND(5,I)*10+IEND(4,I)*100
      1+IEND(3,I)*1000+IEND(2,I)*10000
      GOTO76
74  MARK=0
      D0 1112 II=1,6
1112 IF(IEND(II,I).EQ.32)GOTO1113
      GOTO174
1113 IEND(II,I)=0
      MARK=1
174 FEND(1,I)=FLOAT(IEND(6,I))*1+FLOAT(IEND(4,I))
      1+FLOAT(IEND(3,I))*10+0+FLOAT(IEND(2,I))*100+0
      IF(MARK.EQ.1)FEND(1,I)=-FEND(1,I)
76  CONTINUE
767 CONTINUE
C
C  IF WANT TO PRINT OUT EVALUATED DATA,
      REMOVE C FROM FOLLOWING CARDS
C      WRITE(6,6092)(INPUT(I),I=1,8)
C6092 FORMAT(/,8(1X,A6))
C      D0 6098 KJ=1,LL
C6098 WRITE(6,6038)(LCODE(K,KJ),K=1,2),(FCODE(K,KJ),K=3,4),
C      1(LCODE(K,KJ),K=5,6)
C6038 FORMAT(2I8,2F9.2,2I8)
C      IF(JFLAG.EQ.88)GOTO1489
C      WRITE(6,6048)(IEND(1,K),K=1,2),FEND(1,3),IEND(1,5),
C      1(IEND(1,K),K=7,9)
C6048 FORMAT(2I8,F9.2,4I8)
C1489 CONTINUE
C
C  COMPUTE LAT AND LOG
      CALL DECODE(INPUT(6),ILT(1),2)

```



```
MAK1=0
MAK2=0
DO 9028 I=1,12
  IF(ILT(I).EQ.16)ILT(I)=0
  IF(ILT(I).NE.32)GOTO9028
  IF(ILT(I).EQ.48)ILT(I)=0
  IF(I.LE.6)MAK1=1
  IF(I.GE.7)MAK2=1
  ILT(I)=0
9028 CONTINUE
  LAT=ILT(2)*10000+ILT(3)*1000+ILT(4)*
      100+ILT(5)*10+ILT(6)
  IF(MAK1.EQ.1)LAT=-LAT
  LBG=ILT(8)*10000+ILT(9)*1000+ILT(10)*
      100+ILT(11)*10+ILT(12)
  IF(MAK2.EQ.1)LBG=-LBG
C
C FLAG IF SIG LEVELS
DO 8109 I=1,L
8109 IF(MOD(LC9DE(1,I),50).GT.0)GOTO8110
  ISIG=0
  GOTO8111
8110 ISIG=1
8111 CONTINUE
C
C PUT DATA IN STATION ARRAY
C
C ARRAY IST(I,J) I - INDEX FOR STATIONS
      J - COUNTERS FOR A
C GIVEN STATION 1 NUM 2 LAT 3
      LBG 4 EL 5 NUM TOTAL SND
C 6 NUM SIG LEVELS 7 NUM OF SUR MEASURE
      8 AVG PRES 9 AVG TEM
C 10 AVG DEW PT
C
  IF(NUMST.EQ.0)GOTO1756
  DO 1864 KLM=1,NUMST
1864 IF(INPUT(2).EQ.IST(KLM,1))GOTO1865
C
C PREPARE ARRAY FOR NEW STATION
1756 IF(NUMST.EQ.200)GOTO8900
  NUMST=NUMST+1
  KLM=NUMST
  IST(KLM,1)=INPUT(2)
  IST(KLM,2)=INPUT(6)
  IST(KLM,3)=INPUT(7)
  IST(KLM,4)=INPUT(8)
  DO 8113 I=5,30
8113 IST(KLM,I)=0
```

```
C
C TREAT AS OLD DATA
1865 IST(KLM,5)=IST(KLM,5)+1
    IF(ISIG.EQ.1)IST(KLM,6)=IST(KLM,6)+1
    IF(IEND(1,5).EQ.0.OR.JFLAG.EQ.88)GOTO7341
    IST(KLM,7)=IST(KLM,7)+1
    IST(KLM,8)=IST(KLM,8)+LCODE(1,1)
    DEL=.5
    IF(FCODE(3,1).LT.0.0)DEL=-DEL
    IST(KLM,9)=IST(KLM,9)+IFIX(FCODE(3,1)*10.0+DEL)
    DEL=.5
    IF(FCODE(4,1).LT.0.0)DEL=-DEL
    IST(KLM,10)=IST(KLM,10)+IFIX(FCODE(4,1)*10.0+DEL)
7341 CALL LAYER(L,KLM,LAT,LOG)
    GOTO2

C
C DB NOT EXCEED MORE THAN 200 STATIONS
C IF EXCEED 200 STATIONS WRITE NOTE
8900 IF(IXFLA.EQ.1)GOTO2
    IXFLA=1
    PRINT 8901
8901 FORMAT(47H MORE THAN 200 STATIONS,
           ONLY PROCESS FIRST 200)
    GOTO2

C
C COULD NOT FIND DESIRED MONTH
4062 IF(IFLAG.EQ.0)GOTO2
    IF(IXXI.NE.0)GOTO4076

C
C MONTH REQUEST DOES NOT MATCH DATA
    IFLAG=2
    RETURN

C
C RETURN, NEW MONTH
4076 IFLAG=1
    RETURN

C
C RETURN, END OF FILE
778 IFLAG=-1
    RETURN

C
C RETURN, STARTING MONTH NOT ON TAPE
1333 IFLAG=-2
    RETURN
END
#IBFTC QT      NODECK
C PROGRAM TO CREATE A PROFILE OF TEM,
      DEW PT, HEIGHT, REFRACTIVE
C INDEX, AND GRADIENT USING DATA GIVEN
      AND EXPANDING FOR 2MB
```

```

C PRESSURE LEVELS.
C LOOKS THROUGH PROFILE TO FIND LAYERS
  OF CONSTANT GRADIENT AND
C INCREMENT APPROPRIATE COUNTERS FOR THE GIVEN STATION .
C DIMENSION DATA(5,501) THE ARRAY
  IN WHICH THE PROFILES ARE MADE
C DATA(I,J) I=1 TEMP
  I=2 DEW PT I=3 HEIGHT
C I=4 REFRACTIVE
  INDEX I=5 GRADIENT
C J=1 1100 MB J=501 100 MB
C J=6 THER
  SCALED VALUE IN 2MB LEVELS
C
  SUBROUTINE LAYER(L,IJKL,LAT,LGG)
  COMMON INPUT,LCODE
  COMMON /X/ MON(13),INUM,IAREA(5,
    4),INUMX,ISUM(4,4,99),IST(200,30),
  1 NUMST,IXXI
  DIMENSION INPUT(316),LCODE(6,315),
    FCODE(6,315),DATA(5,501),IL(6)
  EQUIVALENCE (LCODE,FCODE)
C
C CHECK FIRST PRES LEVEL
  IF(LCODE(1,1).GT.1100)LCODE(1,1)=1100
C
C FIND FIRST REF POSITION WITH RESPECT TO PRES
  I1=551-LCODE(1,1)/2
  IBASE=I1
C
C PUT IN FIRST TEM AND DEW PT
  DATA(1,I1)=FCODE(3,1)
  DATA(2,I1)=FCODE(4,1)
C
C INSERT TEM AND DEW PT WITH REF TO 2MB PRESSURE LEVELS,
C AND EXPAND BETWEEN DATA POINTS 1 TEM 2 DEW PT
  DO 88 JK=2,L
  IF(LCODE(1,JK).GT.1100)LCODE(1,JK)=1100
  IF(LCODE(1,JK).LT.100)GOTO 89
  I2=551-LCODE(1,JK)/2
  DATA(1,I2)=FCODE(3,JK)
  DATA(2,I2)=FCODE(4,JK)
  DIV=I2-I1
  DEL2=(DATA(1,I2)-DATA(1,I1))/DIV
  MARK=0
  IF(DATA(2,I1).EQ.99.0)MARK=1
  IF(DATA(2,I2).EQ.99.0)MARK=1
  DEL3=(DATA(2,I2)-DATA(2,I1))/DIV
  III=DIV-1.0

```

```

      DO 903 I=1,III
      XI=I
      JJK=I1+I
      DATA(1,JJK)=DATA(1,I1)+DEL2*XI
      DATA(2,JJK)=DATA(2,I1)+DEL3*XI
903  IF(MARK.EQ.1)DATA(2,JJK)=99.0
      88 I1=I2
      JK=L
C
C  ELIMINATE PORTIONS OF DEW PT PROFILE
      WHERE DATA DOES NOT EXIST
C  EXPAND BETWEEN GIVEN VALUES AND IF
      LAST VALUE OF DEW PT IS GONE,
C  ASSUME TO BE -100 AT LOWEST PRESSURE LEVEL GIVEN
      89 IBA=0
      IEN=0
      DO 9901 I=IBASE,I2
      IF(DATA(2,I).NE.99.00)GOTO9905
      IF(IBA.EQ.0)IBA=I-1
      IF(I.NE.I2)GOTO9901
9905 IF(IBA.EQ.0)GOTO9901
      IEN=1
      IF(IEN.EQ.I2)DATA(2,I2)=-100.0
      DIF=((DATA(2,IEN)-DATA(2,IBA))/FLOAT(IEN-IBA))
      IBA=IBA+1
      IEN=IEN-1
      DO 9902 J=IBA,IEN
      JJ=J-1
9902 DATA(2,J)=DATA(2,JJ)+DIF
      IBA=0
      IEN=0
9901 CONTINUE
C
C  SET REF HEIGHT IN METERS
      REF=LCODE(2,1)
      IF(REF.EQ.0.0)GOTO1566
      CALL DECODE(INPUT(8),IL,1)
      DO 888 IJ=1,6
888  IF(IL(IJ).EQ.48)IL(IJ)=0
      EL=IL(6)+10*IL(5)+100*IL(4)+1000*IL(3)+10000*IL(2)
      REF=REF*3.04801-EL
C
C  COMPUTE THE HEIGHT USING REF HEIGHT
      AS FIRST HEIGHT      3 HEIGHT
1566 DATA(3,IBASE)=REF
      I1=IBASE+1
      DO 906 I=I1,I2
      P1=(552-I)*2
      P2=(551-I)*2

```

```

DATA(3,1)=REF+HEIGH(P1,P2,DATA(1,
      I-1),DATA(1,1),DATA(2,I-1),
1DATA(2,1))
906 REF=DATA(3,1)
C
C COMPUTE THE REFRACTIVE INDEX      4 REFRACTIVE INDEX
D0 22 JK=IBASE,I2
22 DATA(4,JK)=XIND(FLOAT((551-JK)*
      2),DATA(1,JK),DATA(2,JK))
C
C COMPUTE THE GRADIENT      5 GRADIENT
I3=I2-1
D0 23 JK=IBASE,I3
23 DATA(5,JK)=((DATA(4,JK+1)-DATA(4,JK))*1000.0)
      1/(DATA(3,JK+1)-DATA(3,JK))+.5
C
C FIND LAYERS OF CONSTANT GRADIENTS
C IF LAYER EXISTS, FIND HEIGHT AND THICKNESS
C
      IXFLAG=0
      XMAX=-10000
      D0 666 IIJ=1,10
      XMIN=XMAX
      IF(IIJ.EQ.6)XMIN=-XMIN
      GOT0(701,702,703,704,705,706,707,708,709,710),IIJ
701 XMAX=-1000
      ISGN=1
      GOT09817
702 XMAX=-500
      GOT09817
703 XMAX=-150
      GOT09817
704 XMAX=-100
      ISGN=2
      WW=-150.0
      GOT09817
705 XMAX=-75
      GOT09817
706 XMAX=100
      ISGN=3
      WW=75.0
      GOT09817
707 XMAX=150
      GOT09817
708 XMAX=500
      ISGN=4
      GOT09817
709 XMAX=1000
      GOT09817

```

```

710 XMAX=10000
C
C N6, FLAG IF SURFACE AND ELEVATED
      LAYERS OCCUR WITHIN A GIVEN PROFILE
9817 N6SR=0
      N6EL=0
C
C
C N3=1, LAYER OCCURING
C N4, NUM
      OF INTERVALS FOR THE LAYER
      N3=0
      N4=0
      DO 666 I=IBASE,I3
C
C TEST FOR LAYER
      IF (DATA(5,I).LT.XMIN.OR.DATA(5,I).GE.XMAX) GOT0610
C
C LAYER EXISTS
      N3=1
      N4=N4+1
      IF (I.NE.I3) GOT0666
C
C LAYER MUST END
      IENDX=I2
      GOT0611
610 IF (N3.EQ.0) GOT0666
C
C LAYER ENDED
      IENDX=I
611 IBEG=IENDX-N4
C
C FIND BOTTOM
      HT=DATA(3,IBEG)
C
C FIND THICKNESS
      TH=DATA(3,IENDX)-HT
C
C FIND THE GRADIENT
      SUM=0.0
      IX=IENDX-1
      DO 612 II=IBEG,IX
612 SUM=SUM+DATA(5,II)
      GR=SUM/FL0AT(N4)
C
C INDEX FOR THE HEIGHT
      INDX1=HT/100.0+1.0
      IF (INDX1.LT.1) INDX1=1
      IF (INDX1.GT.51) INDX1=51
C
C INDEX FOR THE THICKNESS
      INDX2=TH/25.0+1.0
      IF (INDX2.LT.1) INDX2=1
      IF (INDX2.LE.20) GOT05041
      INDX2=(TH-499.0)/250.0+21.0
      IF (INDX2.GT.23) INDX2=23
C
C INDEX FOR THE GRADIENT
5041 GOT0(741,742,742,743),ISGN
741 INDX3=IFIX((GR+1000.0)/35.6)+2
      IF (GR.LT.-1000.0) INDX3=1

```



```

      G0T0414
742  INDX3=IFIX((GR-WW)/3.04)+1
      G0T0414
743  INDX3=IFIX((GR-1000.0)/35.6)+1
      IF(GR.GT.1000.0)INDX3=25
C
C   INCREMENT LAYER COUNTERS WITH RESPECT TO THE SURFACE
C
C   ARRAY IST(I,J)      I - INDEX FOR STATIONS
C                        J - COUNTERS FOR GIVEN
C   STATION      11-20 NUM SUR LAYERS      21-30 NUM EL LAYERS
C
414  IF(N6SR.EQ.1)G0T06668
      N6SR=1
      IF(HT+EL.LE.100.0)IST(IJKL,IJ+10)=IST(IJKL,IJ+10)+1
6668 IF(N6EL.EQ.1)G0T06669
      N6EL=1
      IF(HT+EL.GT.100.0)IST(IJKL,IJ+20)=IST(IJKL,IJ+20)+1
6669 N3=0
      N4=0
C
C   FLAG IF FOUND A DUCTING LAYER
      IF(GR.LT.-157.0)IXFLAG=1
C
C   INCREMENT CHARACTERISTIC COUNTERS WITH RESPECT TO AREA
      D0 5146 KX=1,INUMX
      IF(IAREA(KX,1).EQ.999999)G0T05046
      IF(LAT.LT.IAREA(KX,1).0R.LAT.GT.IAREA(KX,2))G0T05146
      IF(LOG.LT.IAREA(KX,3).0R.LOG.GT.IAREA(KX,4))G0T05146
5046 ISUM(KX,ISGN,INDX1)=ISUM(KX,ISGN,INDX1)+1
      ISUM(KX,ISGN,INDX2+51)=ISUM(KX,ISGN,INDX2+51)+1
      ISUM(KX,ISGN,INDX3+74)=ISUM(KX,ISGN,INDX3+74)+1
5146 CONTINUE
666  CONTINUE
C
C   FLAG IF NOT FOUND DUCTING LAYER
      ISTA=IJKL
      IF(IXFLAG.EQ.0)ISTA=0
      CALL FREQX(DATA,IBASE,I3,LAT,LOG,ISTA)
      RETURN
      END
$IBFTC FREQ      N0DECK
      SUBROUTINE FREQX(DATA,IB0T,IT0P,LAT,LOG,ISTA)
      COMMON INPUT,LC0DE
      COMMON /X/ M0N(13),INUM,IAREA(5,
                     4),INUMX,ISUM(4,4,99),IST(200,30),
1NUMST,IXXI
      DIMENSION INPUT(316),LC0DE(6,315)
      DIMENSION DATA(5,501)

```

```
C
C  CONVERT N PROFILE TO M PROFILE
  DO 1 I=IBOT,ITOP
    DATA(4,I)=DATA(4,I)+(DATA(3,I)/6370.0)*1.0E+3
  1 DATA(5,I)=DATA(5,I)+157.0
C
C  SET VARIABLES TO FIND MIN TRAPPING FREQ AND HEIGHT
  FRQ=999999
  HT=999999
C
C  IF NOT FOUND DUCTING LAYER DO NOT PROCESS, TEST FLAG
  IF(ISTA.EQ.0)GOTO1492
C
C  LOOK THROUGH PROFILE FROM BOTTOM
    TO TOP FOR MIN IN M PROFILE
  IND1=IBOT+1
  IND2=ITOP-1
  DO 90 INDX=IND1,IND2
    IF(DATA(5,INDX).GE.0.0)GOTO890
    IF(DATA(5,INDX+1).LE.0.0)GOTO890
C
C  HAVE TOP OF LAYER, SET CONSTANTS
  XDT=DATA(3,INDX+1)
  XMT=DATA(4,INDX+1)
  XD2=XDT
  XM2=XMT
  ICNT=INDX-IBOT+1
  INDXX=INDX+1
  PI=0.0
C
C  FIND BOTTOM AND SUM PHASE INTEGRAL
  DO 91 IXI=1,ICNT
    INDXX=INDXX-1
    XD1=DATA(3,INDXX)
    XM1=DATA(4,INDXX)
    IF(XMT.GT.XM2.OR.XMT.LT.XM1)GOTO861
    XD1=XD2-(XD2-XD1)*(XM2-XMT)/(XM2-XM1)
    XM1=XMT
  61 S=(XM2-XM1)/(XD2-XD1)
    X1=SQRT((XM1-XMT+.001*S*(XD2-XD1))**3)
    X2=SQRT((XM1-XMT)**3)
    PI=PI+.942809*(X1-X2)/S
    IF(XD1.LE.0.0)GOTO892
    IF(XM1-XMT.EQ.0.0)GOTO892
    XD2=XD1
    XM2=XM1
  91 CONTINUE
  GOTO890
C
```



```
C FOUND A LAYER
  92 CONTINUE
    XDB=XD1
    XMB=XM1
    FREQZ=(1.0/(4.0*PI))*300.0
    IF(XDB.LT.10.0)FREQZ=(3.0/(8.0*PI))*300.0
    IF(FREQZ.GT.FRQ)GET067
    FRQ=FREQZ
    HT=XDB

C
C FLAG S DUCT AS -1 IN XDB      - GRADIENT AT SURFACE
C FLAG ES DUCT AS -2 IN XDB    + GRADIENT AT SURFACE
  67 IF(XDB.NE.0.0)GET097
    XDB=-1
    IF(DATA(5,IB0T).GT.0.0)XDB=-2

C
C INCREMENT COUNTERS
  97 CALL OUTX(2,XDB,FREQZ,LAT,LOG)
  90 CONTINUE

C
C IF WANT TO PRINT HEADING, MIN FREQ
      AND HEIGHT REMOVE C FROM
C FOLLOWING CARDS
    WRITE(6,201)((IST(ISTA,I),I=1,4),
      (INPUT(I),I=3,5),FRQ,HT
  201 FORMAT(7A6,2F9.2)

C
C IF WANT TO PRINT PROFILE REMOVE C FROM FOLLOWING CARDS
C PRES TEM DEW HGT M M GRAD
C   D0 140 K=1,250
C   P1=(551-K)*2
C   P2=(251-K)*2
C 140 WRITE(6,141)P1,(DATA(I,K),I=1,
      5),P2,(DATA(I,K+250),I=1,5)
C 141 FORMAT(6F9.2,5X,6F9.2)

C
C WRITE ON UNIT 11 (BIN)
C 1ST WRITE  STAT NUM, LAT, LOG, EL,
      HR, DAY, MONTH, BOTTOM, TOP
C 2ND WRITE  PROFILE OF TEM, DEW, HGT, M
C 3RD WRITE  MIN TRAPPING FREQ AND HGT
  1492 WRITE(11) INPUT(2),(INPUT(I),I=6,
      8),(INPUT(I),I=3,5),IB0T,IT0P
    WRITE(11) ((DATA(I,K),I=1,4),K=1,500)
    WRITE(11) FRQ,HT
    RETURN
  END
$IBFTC OUTZ      N0DECK
  SUBROUTINE OUTX(JFLAG,HGT,FREQ,LAT,LOG)
```

```

COMMON /X/ MBN(13), INUM, IAREA(5,
4), INUMX, ISUM(4,4,99), IST(200,30),
1 NUMST, IXXI
DIMENSION JDATA(4,4,30)

C
C JDATA(I,J,K)      I - GEOGRAPHIC AREA
C                   J - 1 FREQ FOR S DUCT
C                   J - 2 FREQ FOR ES DUCT
C                   J - 3 FREQ FOR E DUCT
C                   J - 4 HGT FOR E DUCT
C                   K - COUNTER DISTRIBUTION
C
      IF(JFLAG.GT.2) GOTO103
      GOTO(101,102), JFLAG

C
C ZERO OUT AREA ARRAY
101 DO 201 I=1, INUMX
      DO 201 J=1, 4
      DO 201 K=1, 30
201 JDATA(I,J,K)=0
      RETURN

C
C INCREMENT COUNTERS
C           IFREQ      DEL      50          0 - 999
C           DEL 1000    1000 - 10000
C           IHGT       DEL      200         0 - 6000
102 IF(HGT.LT.0.0) GOTO301

C
C ELEVATED DUCTS
      IFREQ=FREQ/50.0+1.0
      IF(IFREQ.LE.20) GOTO502
      IFREQ=20.0+FREQ/1000.0
      IF(IFREQ.GT.30) IFREQ=30
502 IHGT=HGT/200.0+1.0
      IF(IHGT.GT.30) IHGT=30
      DO 204 I=1, INUMX
      IF(IAREA(I,1).EQ.999999) GOTO203
      IF(LAT.LT.IAREA(I,1).OR.LAT.GT.IAREA(I,2)) GOTO204
      IF(LONG.LT.IAREA(I,3).OR.LONG.GT.IAREA(I,4)) GOTO204
203 JDATA(I,3,IFREQ)=JDATA(I,3,IFREQ)+1
      JDATA(I,4,IHGT)=JDATA(I,4,IHGT)+1
204 CONTINUE
      RETURN

C
C SURFACE OR ELEVATED SURFACE DUCTS
301 ITYPE=ABS(HGT)
      IFREQ=FREQ/50.0+1.0
      IF(IFREQ.LE.20) GOTO302
      IFREQ=20.0+FREQ/1000.0

```

```
      IF(IFREQ.GT.30)IFREQ=30
302  DO 304 I=1,INUMX
      IF(IAREA(I,1).EQ.999999)GOTO303
      IF(LAT.LT.IAREA(I,1).OR.LAT.GT.IAREA(I,2))GOTO304
      IF(LOG.LT.IAREA(I,3).OR.LOG.GT.IAREA(I,4))GOTO304
303  JDATA(I,ITYPE,IFREQ)=JDATA(I,ITYPE,IFREQ)+1
304  CONTINUE
      RETURN
C
C  OUTPUT HEIGHT AND FREQUENCY DATA
103  IMON=JFLAG-900
      WRITE(6,209)
209  FORMAT(1H1,/,/,61H OUTPUT OF HEIGHT
          AND FREQUENCY DISTRIBUTIONS FOR
          1 GIVEN AREAS)
      DO 208 I=1,INUMX
C
C  NORMALIZE FREQUENCY DISTRIBUTION
      DO 706 J=1,3
      DO 706 K=21,30
      IF(JDATA(I,J,K).EQ.0)GOTO706
      JDATA(I,J,K)=JDATA(I,J,K)/20+1
706  CONTINUE
C
C  BN PRINTER
      WRITE(6,205)IMON,(IAREA(I,J),J=1,4)
205  FORMAT(///,35H MONTH LAT1
          LAT2 LOG1 LOG2,/,5I7)
      WRITE(6,206)(JDATA(I,1,K),K=1,30)
206  FORMAT(/,10H S FREQ ,30I4)
      WRITE(6,207)(JDATA(I,2,K),K=1,30)
207  FORMAT(10H ES FREQ ,30I4)
      WRITE(6,246)(JDATA(I,3,K),K=1,30)
246  FORMAT(10H EL FREQ ,30I4)
      WRITE(6,247)(JDATA(I,4,K),K=1,30)
247  FORMAT(10H EL HGT ,30I4)
C
C  BN UNIT 12 TO BE PRINTED OR PUNCHED
      ICNT=1
      WRITE(12,212)ICNT,I,IMON,(IAREA(I,J),J=1,4)
212  FORMAT(3I2,4I6)
      DO 208 J=1,4
      IX2=0
      DO 208 K=1,2
      ICNT=ICNT+1
      IX1=IX2+1
      IX2=IX1+14
208  WRITE(12,211)ICNT,I,IMON,(JDATA(I,J,L),L=IX1,IX2)
211  FORMAT(3I2,15I4)
```

```

      RETURN
      END
$IBFTC OUT      NODECK
C  SUBROUTINE TO OUTPUT DATA AFTER EACH MONTH IS PROCESSED
C  IMON, MONTH OUTPUTING
C  UNIT 6 - LISTING      UNIT 12 - PUNCH CARDS
C
      SUBROUTINE OUTPUT(IMON)
      COMMON /X/ MON(13), INUM, IAREA(5,
      4), INUMX, ISUM(4,4,99), IST(200,30),
      1NUMST, IXXI
      DIMENSION IXXK(5)
      DATA IMAK/6H 99999/

C
C  COMPUTE AVG SURFACE MEASUREMENTS FOR EACH STATION
      DO 8104 I=1, NUMST
      DO 8104 J=8, 10
      8104 IST(I,J)=IST(I,J)/IST(I,7)
C
C  OUTPUT STATISTICS FOR EACH STATION
      IN INCREASING STATION NUMBER ORDER
C
      WRITE(12,8001)
      8001 FORMAT(//////,23HMONTHLY STATION SUMMARY)
      WRITE(6,11)NUMST, IXXI
      11 FORMAT(1H1,////,24H MONTHLY STATION SUMMARY,10X,
      115H STATION COUNT ,13,5X,21H NUMBER
      OF SOUNDINGS ,14,/)
      DO 8103 JJ=1, NUMST
      IMIN=IST(1,1)
      I=1
      DO 8105 JJ=2, NUMST
      IF(IST(JJ,1).LE.IMIN)GOTO8105
      IMIN=IST(JJ,1)
      I=JJ
      8105 CONTINUE
      WRITE(6,176)IMON,(IST(I,J),J=1,30)
      176 FORMAT(13,4(1X,A6),6(1X,I6),10(1X,I3),/,73X,10(1X,I3))
      WRITE(12,178)IMON,(IST(I,J),J=1,10),IMON,IST(I,1),
      1(IST(I,J),J=11,30)
      178 FORMAT(2H 1,12,4A6,6I6,/,2H 2,12,A6,20I3)
      8103 IST(I,1)=IMAK
C
C  OUTPUT STATISTICS FOR EACH GEOGRAPHICAL AREA
C
      WRITE(12,8002)
      8002 FORMAT(//////,23HCHARACTERISTICS BY AREA)
      DO 98 IJ=1, INUMX
C

```

C CALCULATE THE TOTAL NUMBER OF LAYERS FOR EACH DIVISION

```

      DO 4030 I=1,5
4030  IXKX(I)=0
      DO 4032 I=1,4
      DO 4031 J=52,74
4031  IXKX(I)=IXKX(I)+ISUM(IJ,I,J)
4032  IXKX(5)=IXKX(5)+IXKX(I)
      IWXX=1
      WRITE(6,2)IMON,(IAREA(IJ,J),J=1,4),IXKX(5)
  2  FORMAT(1H1,///,25H CHARACTERISTICS
        BY AREA,/,7H MONTH,3X,
        14HLAT1,3X,4HLAT2,3X,4HLOG1,3X,4HLOG2,/,5(1X,I6),5X,
        22HTOTAL NUMBER OF LAYERS,I5)
      WRITE(12,12)IWXX,IJ,IMON,(IAREA(IJ,J),J=1,4)
  12  FORMAT(3I2,4I6)
      DO 98 KK=1,4
      IX2=0
      DO 198 KKK=1,6
      IWXX=IWXX+1
      IX1=IX2+1
      IX2=IX1+16
      IF(IX2.GT.99)IX2=99
  198  WRITE(12,14)IWXX,IJ,IMON,(ISUM(IJ,KK,IK),IK=IX1,IX2)
  14  FORMAT(3I2,17I4)
      GOT0(610,611,612,613),KK
  610  WRITE(6,1610)IXKX(1)
  1610  FORMAT(/,18H GRADIENTS LT -150,
        22X,16HNUMBER OF LAYERS,I5)
      GOT098
  611  WRITE(6,1611)IXKX(2)
  1611  FORMAT(/,30H GRADIENTS GE -150,
        AND LE -75,11X,16HNUMBER OF LAYERS
        1,I5)
      GOT098
  612  WRITE(6,1612)IXKX(3)
  1612  FORMAT(/,28H GRADIENTS GE 75,
        AND LE 150,13X,16HNUMBER OF LAYERS,
        1I5)
      GOT098
  613  WRITE(6,1613)IXKX(4)
  1613  FORMAT(/,17H GRADIENTS GT 150,
        23X,16HNUMBER OF LAYERS,I5)
  98  WRITE(6,702)(ISUM(IJ,KK,IK),IK=1,99)
  702  FORMAT(/,4H HGT,20(1X,I4),/,4X,
        20(1X,I4),/,4X,11(1X,I4),/,4H THK,
        120(1X,I4),/,4X,3(1X,I4),/,4H GRA,
        20(1X,I4),/,4X,5(1X,I4))

```

C

C OUTPUT FREQ DATA BY AREA

```

      WRITE(12,8003)
8003  FORMAT(//////,22HFREQUENCY DATA BY AREA)
      IXM8N=IM8N+900
      CALL BUTX(IXM8N,0.,0.,0.,0.)
      RETURN
      END

$IBFTC HGT      N8DECK
C  FUNCTION HEIGH COMPUTES THE HEIGHT BETWEEN TWO SETS OF
C  METEOROLOGICAL DATA
C
      FUNCTION HEIGH(P1,P2,DEG1,DEG2,DEW1,DEW2)
      DATA A,B,C/25.0578498,-3009.47384,-5.43916634/
      RATIO1=A+B/(DEW1+273.0)+C*ALOG10(DEW1+273.0)
      RATIO2=A+B/(DEW2+273.0)+C*ALOG10(DEW2+273.0)
      WM=(10.0**RATIO1/P1+10.0**RATIO2/P2)/2.0
      HEIGH=18400.0*ALOG10(P1/P2)
      1*(1.0+(((DEG1+DEG2)/2.0)/273.0))/(1.0-0.378*WM)
      RETURN
      END

$IBFTC INDX      N8DECK
C  FUNCTION XIND COMPUTES THE REFRACTIVE
      INDEX GIVEN PRESSURE,
C  TEMPERATURE, AND DEW POINT
C
      FUNCTION XIND(P,TEM,DEW)
      DATA A,B,C/25.0578498,-3009.47384,-5.43916634/
      RATIO=A+B/(DEW+273.0)+C*ALOG10(DEW+273.0)
      XIND=(77.6/(TEM+273.0))
      1*(P+(4810.0*(10.0**RATIO))/(TEM+273.0))
      RETURN
      END

$IBMAP  HELP      N8DECK
INPUT  FILE      ,A(1),BLK=316,BCD,DEFER
KPUT   FILE      ,A(2),BLK=316,BCD,DEFER
OPEN1  SAVE
      TSX        .CLOSE,4
      PTW        KPUT
      TSX        .OPEN,4
      PZE        INPUT
      RETURN     OPEN1
OPEN2  SAVE
      TSX        .CLOSE,4
      PTW        INPUT
      TSX        .OPEN,4
      PZE        KPUT
      RETURN     OPEN2
IN     SAVE
      CLA*       5,4
      SXA        A,4

```



```

      TPL      **7
      TSX      .READ,4
      PZE      INPUT
      PZE      EOF,,*=2
      IORT     BLOCK,,**
      LXD      *-1,4
      TRA      **6
      TSX      .READ,4
      PZE      KPUT
      PZE      EOF,,*=2
      IORT     BLOCK,,**
      LXD      *-1,4
      PXA      ,4
A     AXT      **,4
      ST0*     3,4
      STZ*     4,4
      RETURN   IN
EOF   XEC      A
      CLA      =1
      ST0*     4,4
      RETURN   IN
BLOCK CONTRL  //
      COMMON   316
      END
$IBMAP DECODE  NODECK
DECODE SAVE   2,1,1
      CAL      4,4
      ADD      =5760
      STA      STORE
      CAL*     5,4
      STA      *+3
      ADD      3,4
      STA      LMQ
      AXT      **,4
      AXT      5760,2
LOOP2 AXT      6,1
LMQ   LDQ      **,4
LOOP1 CLA      =0
      LGL      6
STORE ST0      **,2
      TXI      *+1,2,-1
      TIX      LOOP1,1,1
      TIX      LOOP2,4,1
      RETURN   DECODE
      END
$DATA

```

```

$IBJOB      MAP
$USE        BLKPAR(ZPARAM),BLKPAR(ZPARAM)
$USE        BLKTAP(ZTPDNN),BLKTAP(ZTPDNT),BLKTAP(ZTPDHL)
$USE        BLKTAP(ZNFPL),BLKTAP(ZTPDMP)
$USE        XBLKS(ZHEIGHT),XBLKS(PATOUT),
             XBLKS(INPT),XBLKS(ZTAB)
$USE        XBLKS(ZLINE),XBLKS(WETOUT),XBLKS(REFOUT)
$IBFTC XBLKS  DECK
C          BLOCK DATA TO ENSURE THAT COMMONS
             ALL HAVE PROPER SIZE.

          BLOCK DATA
          COMMON /ZHEIGHT/ HEIGHT(5)
          COMMON /PATOUT/ PAT(10)
          COMMON /INPT/ YINPT(15)
          COMMON /ZTAB/ TAB(402)
          COMMON /ZLINE/ LINE(200)
          COMMON /WETOUT/ WET(4)
          COMMON /REFOUT/ REF(6)
          END
$IBFTC BLKTAP  DECK
          BLOCK DATA
C          TAPE DESCRIPTION
          COMMON /ZTPDNN/ NMISS,NAMES(20)
          COMMON /ZTPDNT/ NTMPER(20)
          COMMON /ZTPDMP/ MAPT(13)
          COMMON /ZTPDHL/ ISPEC(2,20)
          COMMON /ZNFPL/ NFPL
          DATA NMISS,(NAMES(I),I=1,14) /14,
1 6HCAR001,6HCAR002,6HCAR003,6HCAR004,
   6HCAR005,6HCAR006,6HCAR007,
2 6HCAR008,6HCAR009,6HCAR010,6HCAR011,
   6HCAR012,6HCAR013,6HCAR014/
          DATA (NTMPER(I),I=1,14) /14*1/
          DATA ((ISPEC(I,J),I=1,2),J=1,14) /
X 6,9, 15,18, 10,15, 14,17, 12,
   16, 12,16, 13,16, 9,12, 6,7,
X 11,12, 16,18, 2,6, 8,10, 6,10 /
          DATA (MAPT(I),I=1,13) /1,2,3,4,5,6,8,7,9,10,11,2,12/
          DATA NFPL /12/
          END

```


\$IBFTC XAIDA DECK

C AIRCRAFT DATA REDUCTION DRIVER PROGRAM, REFCOL-INPUT
C

```
SUBROUTINE AIDA
COMMON /ZNFLT/NFLT
COMMON /ZPARAM/DUM(25),PRBCS,DUMP
COMMON /INPT/PAR,LST
LOGICAL PRBCS
LOGICAL DUMP
LOGICAL PAR
LOGICAL LST
1  FORMAT(1H1)
   NFLT = 0
100 NFLT = NFLT+1
   CALL PINT1
   CALL POUT1
   IF(DUMP) WRITE(6, 1)
150 CALL INPUT
   IF (PRBCS) CALL REFCOL
   IF (.NOT.LST) GO TO 150
   GO TO 100
END
```

```
$IBFTC XPINT1 DECK
      SUBROUTINE PINT1
C THIS SUBROUTINE INPUTS THE PARAMETERS CARDS
C IT ASSIGNS THE VALUE OF THE PARAMETER
      ACCORDING TO THE TYPE
C THE NAME OF THE PARAMETER ISS USED
      TO FIND THE TYPE AND OFFSET
C IN THE TABLE PARAN.
C
      COMMON /ZPARAN/ NPAR, PARAN(2,1)
      COMMON /ZPARAM/ PARAM(1)
      INTEGER PARAN,ALPH,COM(10),IPARAM(1)
      INTEGER STP,TR,FAL
      LOGICAL LPARAM(1)
      EQUIVALENCE ( PARAM(1),IPARAM(1),LPARAM(1) )
      DATA ND,TR,FAL /5H*END*,1HT,1HF/
      DATA STP/6H*STOP* /
      WRITE(6,902)
100 READ(5,900) ALPH,VAL,COM
      WRITE(6,901) ALPH,VAL,COM
      IF(ALPH .EQ. ND) RETURN
      IF( ALPH .EQ. STP) STOP
      DO 150 I=1,NPAR
      IX=I
      IF(ALPH .EQ. PARAN(1,I)) GO TO 200
150 CONTINUE
      WRITE(6,903)
      GO TO 100
200 N=PARAN(2,IX)
      GO TO(300,350,400,450,500), N
C DO ASSIGNMENT ACCORDING TO THE TYPE OF PARAMETER
C
C FLOATING POINT PARAMETER
300 PARAM(IX)=VAL
      GO TO 100
C FIXED POINT PARAMETER
350 IPARAM(IX)=VAL
      GO TO 100
C LOGICAL PARAMETER
400 IF(COM(1) .EQ. TR) LPARAM(IX)=.TRUE.
      IF(COM(1) .EQ. FAL) LPARAM(IX)=.FALSE.
      IF( COM(1).NE.TR .AND. COM(1).NE.FAL ) WRITE(6,904)
      GO TO 100
C ALPHANUMERIC PARAMETER
450 IPARAM(IX)=COM(1)
      GO TO 100
C TIME PARAMERER
500 ITIME=VAL
      IHR=ITIME/10000
```

```
IMIN= MOD (ITIME/100,100)
ISEC = MOD (ITIME,100)
PARAM(IX)=ISEC+60*(IMIN+60*IHR)
GO TO 100
900 FORMAT(A6,2X,F10.0,10A6)
901 FORMAT(1X,A6,2X,E15.5,1X,10A6)
902 FORMAT(1H1,1X,4HNAME,14X,5HVALUE,7X,8HCOMMENTS )
903 FORMAT(33H UNRECOGNIZED NAME. CARD IGNORED. )
904 FORMAT(29H ILLEGAL VALUE. CARD IGNORED.)
END
```

```

$IBFTC XHGT    DECK
C      HEIGHT    BEEBE + SULLIVAN    2.1
SUBROUTINE HEIGHT(I)
C      I GE 1 IF NOT FIRST TIME THROUGH ROUTINE FOR A SPIRAL
C      RADIUS = RADIUS OF EARTH
C      ZS = HEIGHT OF REFERENCE SURFACE ABOVE SEA LEVEL
C      PR = PRESSURE, FN = REFRACTIVE
          INDEX, FKF = TEMPERATURE, EF = VAPOR
C      Z = GEOPOTENTIAL HEIGHT, R = SEMIMINOR
          AXIS, A = SEMIMAJOR AXIS
COMMON /ZPARAM/ZS,DUM(21),RADIUS,R,A,DUM2(30),Z0FS1
COMMON /ZHEIGHT/FN,EF,Z,RH0,RMF
COMMON /PATOUT/PR,DUM1(5),FKF
IF (I.GE.1) GO TO 1
Z = Z0FS1 + ZS
FKS1 = FKF*(1.0 + 0.388 * EF/PR)
GO TO 2
1 FKS2 = FKF*(1.0 + 0.388 * EF/PR)
VAL = ALOG (0LDPR / PR)
DELPSI = 14.645 * (FKS1 + FKS2) * VAL
DELZ = DELPSI *(R/A) * (1.0 +
          2.0 * Z0LD / R + DELPSI / A )
Z = Z0LD + DELZ
FKS1 = FKS2
2 RH0 = Z / RADIUS
RMF = FN*(1.0 + RH0)*0.000001 + RH0
Z0LD = Z
0LDPR = PR
RETURN
END

```

\$IBFTC XINPUT DECK

```

C      SUBROUTINE INPUT
C      INPUT COMMONS
        COMMON /ZNFLT/NFLT
        COMMON /ZPARAM/DUM(47),MISID,TSTART,TSTOP
C      TAPE DESCRIPTION COMMONS
        COMMON /ZTPDNN/NMISS,NAMES(1)           /ZTPDNT/NTMPER(1)
        1      /ZTPDMP/MAPT(13)                 /ZTPDHL/ISPEC(2,1)
C      OUTPUT COMMON
        COMMON /INPT/PAR,LST,X(13)
C      COMMUNICATION WITH LOWER SUBROUTINES
        COMMON /ZFLTIM/IH1,IM1,IS1,IH2,IM2,IS2
        COMMON /ZNWMIS/NWMIS                     /ZMISNO/MISNO
        1      /ZMISSR/MISSR                     /ZISPTR/ISPTR
        2      /ZTAB/P,N,TAB(1)                 /ZEOM/EOM
        3      /ZRTIME/RTIME                     /ZIREL/IREL
        4      /ZLINE/LINE(1)
        COMMON /ZCERR/CERR
        LOGICAL NWMIS,EOM,P,PAR,LST,MSL
        LOGICAL CERR
        LOGICAL LOGCOM
        DATA LFLT,LMIS /0,0/
C      FORMATS
        1 FORMAT(46H NO SUCH MISSION ID
                6N TAPE - FLIGHT IGNORED. )
        2 FORMAT(48H MISSION REQUESTS OUT
                6F SORT - FLIGHT IGNORED. )
C      IS THIS A NEW FLIGHT
        IF (NFLT .EQ. LFLT) GO TO 1000
        LFLT = NFLT
C      YES, INITIALIZE IF NFLT = 1
        IF (NFLT .NE. 1) GO TO 105
        NWMIS = .TRUE.
        MISNO = 1
C      FIND NO. OF REQUESTED MISSION
        105 ISPTS = 1
        DO 110 MISS=1,NMISS
        MISSR=MISS
        IF(MISID .EQ. NAMES(MISSR)) GO TO 120
        110 ISPTS = ISPTS+NTMPER(MISSR)
C      CANNOT FIND MISSION WITH PROPER ID
        WRITE(6,1)
        GO TO 2010
C      IS THE MISSION THE SAME AS THE LAST (ERROR IF LESS)
        120 IF (MISSR .GE. LMIS) GO TO 130
        WRITE(6,2)
        GO TO 2010
        130 MSL = MISSR .EQ. LMIS

```

```

      IF (MSL) GO TO 200
      LMIS = MISSR
C    UPDATE MISSION-ASSOC. CONSTANTS
      ISPTR = ISPTS
      IREL = ISPEC(1,ISPTR)
C    IF THE TAPE HAS TO BE MOVED, MOVE
      IT AND UPDATE PHYSICAL TAPE
C    POSITION INDICATORS
      NMS = MISSR-MISNO
      EOM = .FALSE.
      IF (NMS .LE. 0) GO TO 200
      DO 150 I = 1,NMS
145  CALL SFDATP
      CALL RDATP
      IF (N .NE. 0) GO TO 145
150  CONTINUE
      MISNO = MISSR
      NWMIS = .TRUE.
C    UPDATE FLIGHT CONSTANTS
200  IH1 = TSTART/3600.
      IM1 = AMBD(TSTART,3600.)/60.
      IS1 = AMBD(TSTART,60.)
      IH2 = TSTOP/3600.
      IM2 = AMBD(TSTOP,3600.)/60.
      IS2 = AMBD(TSTOP,60.)
      IH1R = MOD(IH1+24-IREL,24)
      IH2R = MOD(IH2+24-IREL,24)
      RQRST = 3600*IH1R + 60*IM1 + IS1
      RQRET = 3600*IH2R + 60*IM2 + IS2
C    IF SAME MISSION, SKIP 1ST READ
      IF (MSL) GO TO 310
C    FIND FIRST LINE
300  CALL RDLINE
310  IF (EOM) GO TO 2010
      IF (RTIME .LT. RQRST) GO TO 300
C    OUTPUT A LINE
1000 PAR = CERR
      X(1) = AMBD(RTIME+3600.*FLBAT(IREL),86400.)
      DO 1100 I = 2,13
      K1 = 4*(MAPT(I)-1) + 1
      K4 = K1+3
      DO 1010 J = K1,K4
      IF (LINE(J) .GT. 9) GO TO 1020
1010 CONTINUE
      X(I) = 1000*LINE(K1)+100*LINE(K1+
      1)+10*LINE(K1+2)+LINE(K1+3)
      GO TO 1100
1020 PAR = .TRUE.
1100 CONTINUE

```

```

C  READ ANEW
    CALL RDLINE
    LST = EOM .OR. RTIME.GT.RQRET
    RETURN
C  FAILURE EXIT
2010 DO 2020 I = 1,13
2020 X(I) = 0.
    PAR = .TRUE.
    LST = .TRUE.
    RETURN
    END
    
```



```

C ROUTINE COMPUTES PRESSURE, AIR SPEED AND TEMPERATURE
SUBROUTINE PAT
COMMON /INPT/ P,L,XTIME,XR1,XR2,
              XR3,XALT,XEVENT,XSPEED,XPRES,XKS4T
1,XEKT,XRH,XR4,XVXT
COMMON /ZPARAM/ ZS,RFS1,RV1,RNM1,
              RKP1,ANDF1,ANWF1,ACMRVP,
1 CORMR,CORVP,CORIN,ITPR0B,IHUM,
              IRSCT,PUNCH,KPAR,BETA1,BETA2,
2 BETA3,BETA4,BETA5,ALPHA,RADIUS,R,A,PROCS,DUMP,
3 PVMIN,PVMAX,PMIN,PMAX,SVMIN,SVMAX,
              SMIN,SMAX,T4VMIN,T4VMAX,
4T4MIN,T4MAX,
5 EKVMIN,EKVMAX,EKMIN,EKMAX,VXVMIN,VXVMAX,VXMIN,VXMAX,
6MISSID,TSTART,TSTOP,CPRES,CSPEED,
              CKS4T,CEKT,CVXT,Z0FS1,CHKFC
COMMON /PATOUT/ PRES,SPEED,S,TKS4,
              TEK,TVTX,FKF,TF,PTEMP,FNDF
TLIN(X,A,B,Y,Z) = (X-A)*(Z-Y)/(B-A)+Y
T73 = 273.16
ONE = 1.0
DPDN = 1.
PMK4 = TLIN(XPRES,PVMIN,PVMAX,PMIN,PMAX) + CPRES
PRES = PMK4 + DPDN
SPEED = TLIN(XSPEED,SVMIN,SVMAX,SMIN,SMAX) + CSPEED
S = SPEED**2/PRES
TKS4 = TLIN(XKS4T,T4VMIN,T4VMAX,T4MIN,T4MAX) + CKS4T
TEK = TLIN(XEKT,EKVMIN,EKVMAX,EKMIN,EKMAX) + CEKT
TVTX = TLIN(XVXT,VXVMIN,VXVMAX,VXMIN,VXMAX) + CVXT
GO TO (1,2,3),ITPR0B
1 FKF = (TKS4 + T73 )/(ONE+ BETA1 * S)
GO TO 4
2 FKF = (TEK + T73 )/(ONE+ BETA2 * S)
GO TO 4
3 FKF = (TVTX + T73)/(ONE + BETA3 * S)
4 TF = FKF - T73
IF (PRES) 5,5,6
5 PTEMP = 0.
GO TO 7
6 PTEMP = FKF*(1000.0/PRES)**(2./7.) - T73
7 FNDF = 77.6*PRES/FKF
RETURN
END

```

\$IBFTC XP0UT1 DECK

C PARAMETER PRINT I

C

```

SUBROUTINE P0UT1
COMMON /ZPARAN/NPAR,PARAN(2,1) /ZPARAM/PARAM(1)
INTEGER PARAN
DIMENSION NAMES(5),VALUES(5),FORMS(3,
5),ITIM(15),FORM(3)
EQUIVALENCE (ITIM,VALUES)
DATA FORMS(1,1) /24H(5(1XF19.9))
DATA FORMS(1,2) /24H(5(1XI9,
10X))
DATA FORMS(1,3) /24H(5(9XL1,
10X))
DATA FORMS(1,4) /24H(5(6XA6,
8X))
DATA FORMS(1,5) /24H(5(I6,
2(1H,I2)7X))
1 FORMAT(1H1,50X,28H= CURRENT PARAMETER VALUES = )
2 FORMAT(/5(1X,1H*,5X,A6,6X,1H*))
WRITE(6,1)
DO 100 ITYPE = 1,5
NVAL = 0
DO 10 I = 1,3
10 FORM(I) = FORMS(I,ITYPE)
DO 99 I = 1,NPAR
IF (ITYPE.NE.PARAN(2,I)) GO TO 80
NVAL = NVAL + 1
NAMES(NVAL) = PARAN(1,I)
IF (ITYPE.EQ. 5) GO TO 30
VALUES(NVAL) = PARAM(I)
GO TO 80
30 V = PARAM(I)
NT = 3*NVAL-2
ITIM(NT) = V/3600.
ITIM(NT+1) = AM0D(V,3600.)/60.
ITIM(NT+2) = AM0D(V,60.)
80 IF (.NOT.(NVAL.EQ. 5 .OR. (I
.EQ. NPAR .AND. NVAL.GT. 0)))
1 GO TO 99
WRITE(6,2) (NAMES(J),J=1,NVAL)
IF (ITYPE.EQ. 5) NVAL = 3*NVAL
WRITE(6,FORM) (ITIM(J),J=1,NVAL)
NVAL = 0
99 CONTINUE
100 CONTINUE
RETURN
END

```

RAWCON--P0UT1, PRINT INPUT PARAMETERS

PAGE 2

#IBFTC XRDLIN DECK

C READ A LINE OF PAPER TAPE DATA

C

```

SUBROUTINE RDLINE
COMMON/ZTPDHL/ISPEC(2,1) /ZISPTR/ISPTR
1 /ZLINE/LINE(200) /ZTPDMP/MAPT(1)
2 /ZRTIME/RTIME
4 /ZIREL/IREL /ZEOM/EOM
5 /ZMISSR/MISSR

```

```

COMMON /ZMISN0/ MISN0
COMMON /ZPARAM/DUM1(26),DUMP,DUM2(29),CHKFC
COMMON /ZNFPL/NFPL
COMMON /ZCERR/CERR
DIMENSION CLIST(12),CHAR(200)
LOGICAL EOM,DUMP
LOGICAL CHKFC,CERR
DATA (CLIST(I),I=1,12) /1H0,1H1,
1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,

```

X 1H.,1H* /

DATA LMIS /0/

```

1 FORMAT (26(1X,4A1))/(5X,120A1))
IF (MISSR .EQ. LMIS) GO TO 200

```

C INITIALIZATION EACH MISSION

LMIS = MISSR

KK = 0

K1T = 4*(MAPT(1)-1)+1

KRH = 0

PTL = 0.

ITSPH = MOD(ISPEC(2,ISPTR)+24-IREL,24)

LEGNC = 4*NFPL + 1

200 DO 299 I = 1,200

KK = KK+1

KR = KHAR(KK)

IF(EOM) GO TO 400

LINE(I) = KR

CHAR(I) = CLIST(KR+1)

IF (KR .EQ. 10) GO TO 400

299 CONTINUE

I = 200

300 KK = KK+1

KR = KHAR(KK)

IF(EOM) GO TO 400

IF (KR .NE. 10) GO TO 300

C EOL CHARACTER REACHED

400 IF(DUMP) WRITE(6,1) (CHAR(J),J=1,I)

IF(EOM) MISN0=MISN0+1

IF (EOM) RETURN

CERR = (CHKFC .AND. I .NE. LEGNC) .OR.

1 LINE(K1T) .GT. 5 .OR. LINE(K1T+1) .GT. 9 .OR.

2 LINE(K1T+2) .GT. 5 .OR. LINE(K1T+3) .GT. 9

```
IF (.NOT. CERR) GO TO 450
PT = PTL
GO TO 500
450 PT = 60*(10*LINE(K1T)+LINE(K1T+
      1)) + 10*LINE(K1T+2) + LINE(K1T+3)
IF (PT .GE. PTL) GO TO 500
KRH = KRH+1
IF (KRH .LE. ITSPH) GO TO 500
ISPTR = ISPTR+1
KRH = MOD(ISPEC(1,ISPTR)+24-IREL,24)
ITSPH = MOD(ISPEC(2,ISPTR)+24-IREL,24)
500 RTIME = 3600.*FLBAT(KRH) + PT
PTL = PT
RETURN
END
```

```

SUBROUTINE REFCOL
COMMON /ADATA/ EH20(1000)
COMMON /INPT/ PAR, LAST, XTIME, XR1,
      XR2, XR3, XALT, XEVENT, XSPEED, XPRES,
1XKS4T, XEKT, XRH, XR4, XVXT
COMMON /ZPARAM/ ZS, RFS1, RFV1, RNM1,
      RKP1, ANDF1, ANWF1, ACMRVP,
1CORMR, CORMP, CORIN, ITPR0B, IHUM,
      IRSCT, PUNCH, KPAR, BETA1, BETA2, BETA3,
2BETA4, BETA5, ALPHA, RADIUS, R, A, PR0CS, DUMP
COMMON /PATOUT/ PRES, SPEED, S, TKS4,
      TEK, TVTX, FKF, TF, PTEMP, FNDF
COMMON /WETOUT/ ARF, AEF, ANWF, ANF
COMMON /REFOUT/ RDNM, RNM, RNWF, REF, RRF, RNF
COMMON /ZHEIGHT/ FNF, EF, Z, RH0, RMF
DIMENSION MRDNG(50), ZSPEED(50),
      ZPR(50), ZTKS4(50), ZTEK(50)
DIMENSION ZTVTX(50), ZDELN(50),
      ZN(50), ZFNDF(50), ZFNWF(50), ZTIME(50)
DIMENSION IIVENT(50), ZFNF(50),
      ZPINDX(50), ZZ(50), ZALT(50), ZPRES(50)
DIMENSION ZTF(50), ZPTEMP(50), ZMIXR(50),
      ZVAP0R(50), ZRMF(50)
DIMENSION IPAR(50)
DIMENSION ZNF(50)
LOGICAL PAR, LAST, PUNCH
LOGICAL PR0CS, DUMP
INTEGER ZTIME
DATA K0UNT/0/, II/0/
DATA ISTAR/1H*/, IBLANK/1H /
IF (KPAR) 5,5,1
5 IF (.NOT.PAR) GO TO 1
IF (LAST.AND.(II.GE.1)) GO TO 501
IF (LAST) GO TO 520
RETURN
1 CALL PAT
CALL WET
IF (RKP1) 2,2,3
2 IHUM = 1
3 IF (IHUM.EQ.1) GO TO 110
GO TO (50,50,50,55), IRSCT
50 IF (RFS1) 51,51,60
51 IHUM = 1
GO TO 110
55 IF (RFV1) 56,56,60
56 IHUM = 1
GO TO 110
60 CALL REFCT
FNF = RNF

```

```

RF2 = RRF
EF = REF
GO TO 115
110 FNF = ANF
RF2 = ARF
EF = AEF
RNM = 0.
RDNM = 0.
RNWF = 0.
115 CALL HEIGHT(KBUNT)
ZPRT = Z - ZS
IF (Z.LT.ACMRVP) GO TO 120
FNF = FNF + CORIN
RF2 = RF2 + CORMR
EF = EF + CORVP
120 ALT = 1.499 * XALT
ITIME = XTIME
ISEC = MOD(ITIME,60)
ITIME1 = ITIME/60
IMIN = MOD(ITIME1,60)
IHRS = ITIME/3600
TIMEX = 10000*IHRS + 100*IMIN + ISEC
IVENT = XEVENT/100.
IF (PRES) 200,200,201
200 PINDEX = 0.
GO TO 202
201 PINDEX = FNF*(1000./PRES)**0.714
202 KBUNT = KBUNT + 1
IF (PUNCH) WRITE(1,10) KBUNT,Z,
      FNF,RMF,TF,PTEMP,EF,PRES,RF2
II = II + 1
MRDNG(II) = KBUNT
ZSPEED(II) = SPEED
ZPR(II) = PRES - 1.
ZTKS4(II) = TKS4
ZTEK(II) = TEK
ZVTX(II) = TVTX
ZDELN(II) = RDNM
ZN(II) = RNM
ZFND(II) = FND
ZFNWF(II) = RNWF
ZTIME(II) = TIMEX
IIVENT(II) = IVENT
ZFNF(II) = FNF
ZPINDX(II) = PINDEX
ZZ(II) = ZPRT
ZALT(II) = ALT
ZPRES(II) = PRES
ZTF(II) = TF

```



```

ZPTMP(11) = PTMP
ZMIXR(11) = RF2
ZVAPBR(11) = EF
ZRMF(11) = RMF
IF (PAR) GO TO 600
GO TO 610
600 IPAR(11) = ISTAR
GO TO 500
610 IPAR(11) = IBLANK
500 IF (11.NE. 50.AND. (.NOT.LAST)) RETURN
501 WRITE(6,26)
WRITE(6,20)
WRITE(6,21)
WRITE(6,22)
502 DO 510 K = 1,11
510 WRITE(6,11) MRDNG(K),IPAR(K),ZSPEED(K),
1ZVTX(K),ZDELN(K),ZN(K),ZFND(K),ZFNWF(K)
WRITE(6,26)
WRITE(6,23)
WRITE(6,24)
WRITE(6,25)
503 DO 515 K = 1,11
515 WRITE(6,12) MRDNG(K),IPAR(K),ZTIME(K),
1ZVENT(K),ZFN(K),ZPINDX(K),
ZPTMP(K),ZMIXR(K),ZVAPBR(K),
ZRMF(K)
IF(DUMP) WRITE(6,26)
IF (.NOT.LAST) GO TO 525
520 KOUNT = 0
IF(PUNCH) ENDFILE 1
525 11 = 0
RETURN
10 FORMAT(16,2F7.1,6PF7.1,OP4F7.1,3PF7.1)
11 FORMAT(5X14,A1,7XF5.1,8XF6.1,7(8XF5.1))
20 FORMAT(4X,7HREADING,5X,8HAIRSPEED,
5X,8HPRESSURE,17X,11HTEMPERATURE
X,20X,13HREFRACTOMETER)
21 FORMAT(17X,5HKNOTS,10X,3HMB,10X,
3HKS4,7X,9HPLAT.WIRE,5X6HVORTEX,
X 7X7HDELTA N,9X1HM,10X5HN DRY,8X5HN WET)
22 FORMAT(44XSHDEG.C,8XSHDEG.C,8XSHDEG.C)
12 FORMAT(6X14,A1,3X16,4X11,4XF5.1,
4XF5.1,4XF7.1,4XF7.1,4XF6.1,4X
1F5.1,4XF5.1,4X3PF5.2,4XOPF5.2,4X6PF8.1)
23 FORMAT(4X7HREADING,4X4HTIME,3X5HEVENT,
4X1HN,8X1HK,4X10GEB.HEIGHT,
X 2X9HALTIMETER,2X8HPRESSURE,3X5HTEMP.,
2X9HPOT.TEMP.,1X6HMIXING,

```

```

X 4X5HVAPOR,8X1HM)
24 FORMAT(47X6HMETERS,5X6HMETERS,
          7X3HMB.,5X5HDEG.C,4X5HDEG.C,4X
X 5HRATIO,3X8HPRESSURE )
25 FORMAT(98X4HG/KG /)
26 FORMAT (1H1)
END

```

\$IBFTC XREFCT DECK

```

SUBROUTINE REFCT
COMMON /INPT/ PAR, LAST, XTIME, XR1,
              XR2, XR3, XALT, XEVENT, XSPEED, XPRES,
1XKS4T, XEKT, XRH, XR4, XVXT
COMMON /ZPARAM/ ZS, RFS1, RFV1, RNM1,
              RKP1, ANDF1, ANWF1, ACMRVP,
1CORMR, CORVP, CORIN, ITPROB, IHUM,
              IRSCT, PUNCH, KPAR, BETA1, BETA2, BETA3,
2BETA4, BETA5, ALPHA, RADIUS, R, A
COMMON /PATOUT/ PRES, SPEED, S, TKS4,
              TEK, TVTX, FKF, TF, PTEMP, FNDF
COMMON /WETOUT/ ARF, AEF, ANWF, ANF
COMMON /REFOUT/ RDNM, RNM, RNWF, REF, RRF, RNF
C0N1 = 1.0 + (BETA5*S*1.4/.4)
C0N2 = 1.0 + BETA5 * S
RKP = FKF * (1. + BETA4 * S)
G0 T0 (1,2,3,4), IRSCT
1 XXX = XR1
  G0 T0 5
2 XXX = XR2
  G0 T0 5
3 XXX = XR3
  G0 T0 5
4 RDELN = (XR4 - RFV1)/9.245
  G0 T0 6
5 RDELN = (XXX - RFS1)/9.245
6 RDNM = RDELN + ALPHA*(RKP-RKP1)
  RNM = RNM1 = RDNM
  RNDM = (FNDF*C0N1)/C0N2
  RNWM = RNM = RNDM
  RNWF = RNWM * (C0N2**2/C0N1)
  REF = (RNWF*FKF*FKF)/373000.
  RRF = (REF * .62197)/(PRES - REF)
  RNF = RNWF + FNDF
RETURN
END

```

```

$IBFTC XWET    DECK
SUBROUTINE WET
COMMON /ADATA/ EH20(1000)
COMMON /ZPARAM/ ZS,RFS1,RFV1,RNM1,
                RKP1,ANDF1,ANWF1,ACMRVP,
1C0RMR,C0RVP,C0RIN,ITPR0B,IHUM,
                IRSCT,PUNCH,KPAR,BETA1,BETA2,BETA3,
2BETA4,BETA5,ALPHA,RADIUS,R,A
COMMON /PATOUT/ PRES,SPEED,S,TKS4,
                TEK,TVTX,FKF,TF,PTEMP,FNDF
COMMON /WETOUT/ ARF,AEF,ANWF,ANF
F00 = 1.4/.4
UM = 1.
IEW = TKS4*10. + 501.5
IF (IEW.LE.0) GO TO 2
EW = EH20(IEW)
GO TO 3
2 EW = 0.
3 CON = 1. + BETA1*S*F00
  ARF = (UM*.62197*EW)/(PRES*CON -EW)
  AEF = PRES*ARF/ (.62197+ARF)
  ANWF = 373000.*AEF/(FKF**2)
  ANF = FNDF + ANWF
  IF (ANWF1) 6,6,4
6 ANWF1 = ANWF
  ANDF1 = FNDF
4 IF (RNM1) 7,7,5
7 CON = 1. + BETA5*S*F00
  CON2 = 1. + BETA5*S
  RNWM1 = ANWF1*CON/(CON2**2)
  RNDM1 = (ANDF1*CON)/CON2
  RNM1 = RNWM1 + RNDM1
5 RETURN
END

```

RAWCON--SFDATP, SKIP A FILE

PAGE 1

```
$IBFTC XSFDAT DECK
      SUBROUTINE SFDATP
C      SKIP TO AN END OF FILE.
      COMMON /ZTAB/ P,N,TAB(400)
100  CALL RDATP
      IF(N .NE. 0) GO TO 100
      RETURN
      END
```

\$IBFTC BLKPAR DECK

BLOCK DATA

INTEGER PARAN

COMMON /ZPARAN/NPAR,PARAN(2,60)

COMMON /ZPARAM/ ZS,RFS1,RFV1,RNM1,
RKP1,ANDF1,ANWF1,ACMRVP,

1 CORMR,CORVP,CORIN,ITPR0B,IHUM,
IRCT,PUNCH,KPAR,BETA1,BETA2,
2 BETA3,BETA4,BETA5,ALPHA,RADIUS,R,A,PR0CS,DUMP,
3 PVMIN,PVMAX,PMIN,PMAX,SVMIN,SVMAX,
SMIN,SMAX,T4VMIN,T4VMAX,
4 T4MIN,T4MAX,
5 EKVMIN,EKVMAX,EKMIN,EKMAX,VXVMIN,VXVMAX,VXMIN,VXMAX,
6 MISSID,TSTART,TSTOP,CPRES,CSPEED,
CKS4T,CEKT,CVXT,Z0FS1,CHKFC

DATA NPAR/ 57/

DATA((PARAN(I,J),I=1,2),J=1,27)/

16HZS ,1,6HRFS1 ,1,6HRFV1
 ,1,6HRNM1 ,1,6HRKP1 ,1,6HANDF1 ,1,
26HANWF1 ,1,6HACMRVP,1,6HCORMR
 ,1,6HCORVP ,1,6HCORIN ,1,6HITPR0B,2,
36HIHUM ,2,6HIRSCT ,2,6HPUNCH
 ,3,6HKPAR ,2,6HBETA1 ,1,6HBETA2 ,1,
46HBETA3 ,1,6HBETA4 ,1,6HBETA5
 ,1,6HALPHA ,1,6HRADIUS,1,6HR ,1,
56HA ,1,6HPR0CS ,3,6HDUMP ,3/

DATA((PARAN(I,J),I=1,2),J=28,47)/

16HPVMIN ,1,6HPVMAX ,1,6HPMIN
 ,1,6HPMAX ,1,6HSVMIN ,1,6HSVMAX ,1,
26HSMIN ,1,6HSMAX ,1,6HT4VMIN,1,6HT4VMAX,1,
36HT4MIN ,1,6HT4MAX ,1,6HEKVMIN,
 ,1,6HEKVMAX,1,6HEKMIN ,1,6HEKMAX ,1,
46HVXVMIN,1,6HVXVMAX,1,6HVXMIN ,1,6HVXMAX ,1/

DATA((PARAN(I,J),I=1,2),J=48,56)/

16HMISID ,4,6HTSTART,5,6HTSTOP
 ,5,6HCPRES ,1,6HCSPEED,1,6HCKS4T ,1,
26HCEKT ,1,6HCVXT ,1,6HZ0FS1 ,1/

DATA PARAN(1,57),PARAN(2,57) /6HCHKFC ,3/

LOGICAL PUNCH,PR0CS,DUMP,CHKFC

DATA RFS1 /1871./

DATA RFV1 /0./

DATA RNM1 /316./

DATA RKP1 /285.94/

DATA ANDF1 /0./

DATA ANWF1 /0./

DATA ACMRVP/0./

DATA CORMR /0./

DATA CORVP /0./

DATA CORIN /0./

```
DATA ITPRBB/1/
DATA IHUM /0/
DATA IRSCT /1/
DATA PUNCH /FALSE./
DATA KPAR /1/
DATA BETA1 /0.0002632/
DATA BETA2 /-0.0002106/
DATA BETA3 /-0.0000648/
DATA BETA4 /0.0001316/
DATA BETA5 /0.0000658/
DATA ALPHA /-0.75/
DATA RADIUS/6357000./
DATA R /6354120./
DATA A /6356363./
DATA PRBCS /TRUE./
DATA DUMP /FALSE./
DATA ZS /535.4117/
DATA PVMIN /18./
DATA PVMAX /1017./
DATA PMIN /600./
DATA PMAX /1060./
DATA SVMIN /691./
DATA SVMAX /1060./
DATA SMIN /135./
DATA SMAX /195./
DATA T4VMIN/190./
DATA T4VMAX/891./
DATA T4MIN /-40./
DATA T4MAX /35.9/
DATA EKVMIN/278./
DATA EKVMAX/769./
DATA EKMIN /-40./
DATA EKMAX /35.9/
DATA VXVMIN/241./
DATA VXVMAX/1050./
DATA VXMIN /-40./
DATA VXMAX /32./
DATA MISSID/1/
DATA TSTART/0./
DATA TSTOP /86399./
DATA CPRES /0./
DATA CSPEED/0./
DATA CKS4T /0./
DATA CEKT /0./
DATA CVXT /0./
DATA ZBFS1 /914./
DATA CHKFC /TRUE./
END
```


\$IBMAP	XKHAR	DECK	
KHAR	SAVE	1,2,3,4,5	
	ZET	NWMIS	STARTING A NEW MISSION
	TRA	STRTMS	YES.
KBEG	CLA*	3,4	GET CHARACTER COUNT
	SUB	KLAST	HOW MANY PAST LAST
	TMI	ERR	BACKWARDS IS A N ERROR
	PAX	,1	
	CLA*	3,4	THIS CHAR WILL
			BE LAST ON NEXT ENTRY
	STO	KLAST	
	TSX	NXTCHR,5	GET NEXT CHARACTER
	TIX	*-1,1,1	KEEP GOING UNTIL
			HAVE PROPER ONE.
	PAC	,1	CONVERT CHARACTER
	TXL	CKEOL,1,-33	IF IT I MORE
			THAN 33 CHECK IF IT IS EOL.
	CLA	TABLE,1	
	RETURN	KHAR	
CKEOL	LDQ	=11	11 IS A BAD CHAR
	CAS	=0200	BUT CHECK IF IT IS EOL .
	TRA	*+2	NO.
	LDQ	=10	YES. EOL IS 1/
	XCA		PUT IT IN AC.
	RETURN	KHAR	
*			
NXTCHR	EQU	*	GET NEXT CHARACTER
	LAC	CURBIT,2	BITS LEFT
	TXL	NXTWRD,2,0	
	TXH	NXTWRD,2,-8	TRA IF LESS THAN 8 BITS LEFT
	LDQ	CURWRD	GET WORD
	LGL	8	GET CHARACTER
	ANA	=0377	
	STQ	CURWRD	
	TXI	*+1,2,8	BUMB BIT COUNT.
	SCA	CURBIT,2	
	TRA	1,5	RETURN
NXTWRD	LXA	WRDCNT,3	NEED ANOTHER WORD
	TNX	RDREC,3,1	BUMP COUNT,
			AND TRA IF NO MORE.
	SXA	WRDCNT,3	
SHIFTN	LDQ	CURWRD	GET CURRENT WORD
	LGL	,2	USE WHATEVER BIS IT HAS
	LAC	CURPOS,3	CURRENT POSITION IN BUFFER.
	LDQ	,3	GET WORD.
	TXI	*+1,3,-1	BUMP.
	SCA	CURPOS,3	AND STORE.
	TXI	*+1,2,8	HOW MANY BITS OF NEW WORD.
	SXA	TEMP,2	

	LAC	TEMP,2	
	LGL	,2	SHIFT IN
	STQ	CURWRD	
	ANA	=0377	
	TXI	*+1,2,36	REMAINING IN NEW WORD
	SXA	CURBIT,2	
	TRA	1,5	
*			
RDREC	CALL	RDATP	GET NEXT RECORD
	NZT	N	IS IT EOF.
	TRA	EOF	YES.
RDREC1	AXT	TAB,4	NO. RESET POINTERS.
	SXA	CURPOS,4	
	CLA	N	
	STQ	WRDCNT	
	TRA	SHIFTN	GO SHIFT IT IN.
STRTMS	STZ	CURBIT	START OF MISSION
			NO BITS IN CURWORD.
	STZ	WRDCNT	NO WORDS SIN BUFFER.
	STZ	KLAST	AND THERE WAS NO LAST CHAR
	STZ	NWMIS	RESET NEW MISSION INDICATOR.
	TRA	KBEG	
EOF	CALL	RDATP	
	AXT	0,2	A NEW FILE ALWAYS
			STARTS A NEW CHARACTER
	ZET	N	IS IT DOUBLE END OF FILE.
	TRA	RDREC1	NO. IGNORE THE FIRST EOF.
	STL	EBM	YES. END OF MISSION.
	STL	NWMIS	
	CLA	=10	
	RETURN	KHAR	
ERR	CALL	EXIT	
TABLE	EQU	*	
	DEC	11	
	DEC	1	00000001 1
	DEC	2	00000010 2
	DEC	11	3
	DEC	4	00000100 4
	DEC	11	5
	DEC	11	6
	DEC	7	00000111 7
	DEC	8	00001000 8
	DUP	1,10	
	DEC	11	
	DEC	3	00010011 19
	DEC	11	20
	DEC	5	00010101 21
	DEC	6	00010110 22
	DEC	11	23

	DEC	11		24
	DEC	9	00011001	25
	DUP	1,6		
	DEC	11		
	DEC	0	00100000	32
CURBIT	PZE			
CURWRD	PZE			
CURPOS	PZE			
WRDCNT	PZE			
KLAST	PZE			
TEMP	PZE			
	LORG			
ZNWMIS	CONTRL	ZNWMIS		
	USE	ZNWMIS		
NWMIS	BSS	1		
ZEOM	CONTRL	ZEOM		
	USE	ZEOM		
EOM	BSS	1		
ZTAB	CONTRL	ZTAB		
	USE	ZTAB		
P	BSS	1		
N	BSS	1		
TAB	BSS	400		
	END			

\$IBMAP XLOGCO DECK

*

COMPARES TWO
ARGUMENTS AND RETURNS
LOGICAL VARIABLE.
TRUE IF THEY ARE EQUAL
FALSE IF THEY ARE NOT.

*

*

*

ENTRY LOGCOM
LOGCOM NULL

CLA* 3,4

FIRST ARG.

SUB* 4,4

TZE TRET

ZAC

SET TO RETURN FALSE.

TRA 1,4

TRET CLA =1

EQUAL. RETURN TRUE.

TRA 1,4

END

\$IBMAP	XRDATP	DECK	
INPUT	FILE	,A(1),DEFER,INPUT,BLK=254,BIN	
RDATP	SAVE		
	TSX	,OPEN,4	OPEN THE FILE EVERYTIME
	PZE	INPUT	
	STZ	N	A ZERO IN N MEANS AN EOF WAS FOUND.
	TSX	,READ,4	
	PZE	INPUT	
	PZE	RET,,*-2	
	IORT	TAB,,**	INPUT INTO TAB IN COMMON.
	LXD	*-1,4	PUT COUNT
	SXA	N,4	IN N.
RET	RETURN	RDATP	
ZTAB	CONTRL	ZTAB	
	USE	ZTAB	
P	BSS	1	
N	BSS	1	
TAB	BSS	400	
	END		
\$ENTRY		AIDA	

PLOT--BLOCK DATA FOR FILE NAMES

PAGE 1

\$IBFTC TFILES

BLOCK DATA

COMMON /ZFILE / NFILES,FILEID(20)

DATA NFILES,FILEID/20,

X 6HFILE01,6HFILE02,6HFILE03,6HFILE04,
6HFILE05,6HFILE06,6HFILE07,

X 6HFILE08,6HFILE09,6HFILE10,6HFILE11,
6HFILE12,6HFILE13,6HFILE14,

X 6HFILE15,6HFILE16,6HFILE17,6HFILE18,
6HFILE19,6HFILE20 /

END

\$IBFTC XBLK

BLOCK DATA

INTEGER FILEID

LOGICAL PLTPT

INTEGER XEMPH, HTEMPH, HTLAB, HTCHR

INTEGER SUB

LOGICAL LG0, LNUMG0

REAL LFROM, LTO, LBY

REAL LHTFR, LHTT0, LHTBY

LOGICAL LFRG0

LOGICAL LSG0, LSGRK

LOGICAL ORD

C

COMMON /ZORD/ ORD

COMMON /ZPLOTS/NFRAME, NPLOTS(10)

COMMON /ZGRIDX/ MTL(10), MTR(10),
XL(10), XR(10), DX(10), XEMPH(10),

X NXLAB(10), NXCHR(10)

COMMON /ZGRIDH/ HTL0W(10), HTMAX(10),
DHT(10), HTEMPH(10), HTLAB(10),

X HTCHR(10), MTB(10), MTT(10)

COMMON /ZSUB/ SUB(10)

COMMON /ZPLTPT/ PLTPT(25)

COMMON /ZVRTTL/ LVRSZ(2), LVRX, LVRY, LVRDUM(6), VARTTL(5)

COMMON /ZFMT/ FMT(10)

COMMON /ZNVAR/ NVAR

COMMON /ZJOB/NJOBBC, JOBBID(10)

COMMON /ZLABEL/ LG0(10), LALPH(5,
10), LSIZE(2,10), LNY(10), LNX(10)

COMMON /ZLNUM/ LNUMG0(10), LNUMY(10),
LNUMSZ(2,10), LFROM(10),

X LTO(10), LBY(10)

COMMON /ZLFR/ LHTFR, LHTT0, LHTBY,
LHTIX, LHTSIZ(2), LFRG0(10),

X LFRSIZ(2)

COMMON /ZLS/ LSG0(10), LSSZ(2), LSCHR(10), LSGRK(10)

COMMON /ZHTLBL/ LHTLX, LHTLY

C

C FMT CONTAINS THE FORMAT FOR READING THE DATA.

DATA FMT(1)/12H(I6,7F7.1) /

C

C PLTPT(I)=.TRU. MEANS PLOT THE ITH

VARIABLE AS DISCRETE POINTS

C

.FALS. MEANS PLOT IT AS A CONTINUOUS LINE.

DATA (PLTPT(I), I=1,5) /5*.FALSE./

C

C THE FOLLOWING ESTABLISHES THE VERTICAL GRID (I.E. GRID FOR

C

HORIZONTAL LINES) WHICH REFERS TO HEIGHT.

C

EACH VARIABLE COULD HAVE ITS OWN GRID IF DESIRED.

C HTLOW IS THE LOWEST HEIGHT TO BE PLOTTED
 C HTMAX IS THE LARGEST HEIGHT TO BE PLOTTED.
 C DHT IS THE HEIGHT BETWEEN GRID LINES.
 C HTEMPH INDICATES WHICH GRID LINES ARE TO BE DARKENED.
 C (HTEMPH=N MEANS DARKEN EVERY NTH GRID LINE)
 C HTLAB, AND HTCHR CONTROL THE AUTOMATIC
 LABELLING OF GRID1V.

C THEY ARE NO LONGER USED.

DATA (HTLOW(I),HTMAX(I),DHT(I),
 HTEMPH(I),HTLAB(I),HTCHR(I),I=1,5)

X / 0.,4000.,500.,10,10,0,
 X 0.,4000.,0.,0,0,0,
 X 0.,4000.,500.,10,0,0,
 X 0.,4000.,0.,0,0,0,
 X 0.,4000.,0.,0,0,0 /

C

C NFRAME=NUMBER OF SEPARATE FRAMES

C NPLOTS(I)=HOW MANY VARIABLES TO PLOT ON EACH FRAME.

DATA NFRAME,(NPLOTS(I),I=1,2) /2,2,3/
 DATA NVAR /5/

C

C MTL,MTR,MTB,MTT CONTROL THE MARGIN

SETTINGS OF LEFT RIGHT TOP A

C MTL(I),MTR(I),MTB(I),MTT(I) CONTROL

THE PLACEMENT OF THE GRID FOR

C THE ITH VARIABLE.

C THEY ARE THE MARGINS (IN RASTER UNITS)

TO BE LEFT ON THE LEFT,RIGHT,

C BOTTOM, AND TOP OF THE GRID RESPECTIVELY.

DATA (MTL(I),MTR(I),MTB(I),MTT(I),I=1,5) /

1 80,5,75,260,
 1 80,5,75,260,
 1 80,5,75,260,
 1 80,5,75,260,
 1 80,5,75,260/

C

C XL,XR,DX,XEMPH,NXLAB,NXCHR ESTABLISH

THE SCALLING FOR THE VARIABLES

C WHICH ARE PLOTTED IN THE X DIRECTION,

(I.E. VERITICAL GRID LINES)

C THEIR MEANING CORRESPONDS TO BE MEANING

FOR THE VERTICAL SCALLING.

DATA (XL(I),XR(I),DX(I),XEMPH(I),
 NXLAB(I),NXCHR(I),I=1,5) /

1 150.,400.,50.,10,0,0,
 A 300.,800.,0.,0,0,0,
 3 -5.,45.,5.,2,0,0,
 4 -5.,45.,0.,0,0,0,
 5 0.,25.,0.,0,0,0 /

```

C
C
C SUB CONTROLS THE ORDER IN WHICH THE
      VARIABLES ARE PLOTTED, S
C THE ITH VARIABLE WAS THE SUB(I)TH
      NUMBER READ FROM THE INPUT RECORD.
      DATA (SUB(I),I=1,5) /1,2,3,4,5/
C
C LVRSZ = SIZE FOR LABEL CHARACTERS
      (I.E. VARIABLE ALPHA CHARS)
C LVRX,LVRY = X AND Y COORD OF START OF VARIABLE TITLE
C
      DATA LVRSZ(1),LVRSZ(2),LVRX,LVRY /3,3,400,12/
C
C NJOBID IS PRINTED IN THE FIRST FRAME
      OF A ROLL TO IDENTIFY THE JOB.
C NJOBC=NUMBER OF CHARACTERS IN JOBD.
      DATA NJOBC,JOBD(1) /7,7H WILSON /
C
C LG0(I)=.TRUE. MEANS THAT THERE IS
      AN ALPHABETIC LABEL ASSOCIATED
C WITH THE ITH VARIABLE. THE LABEL IS CONTAINED IN
C LALPH(1,I) ... LALPH(5,I)
      DATA (LG0(I),I=1,5) /3*.TRUE,..FALSE,..TRUE. /
      DATA (LALPH(1,I),I=1,5) /
      1 30HREFRACTIVITY (N UNITS) ,
      2 30HREFRACTIVITY (M UNITS) ,
      3 30HTEMPERATURE (DEGREES C) ,
      4 0,30HVAPOR PRESSURE (MB.) /
C
C LSIZE(1,I),LSIZE(2,I) ARE THE
C LSIZE(1,I),LSIZE(2,I) INDICATE THE
      SIZE OF THE LABELING FOR THE
C ITH VARIABLE. L
      DATA ((LSIZE(I,J),I=1,2),J=1,5) /10*2/
C
C LNX(I),LNY(I) ARE THE X AND Y RASTER
      COORDINATES AT WHICH TO BEGIN
C PRINTING THE LABEL FOR THE ITH VARIABLE.
      DATA (LNX(I),LNY(I),I=1,5) /400,
      36,400, 812,400,36,2*0,400, 812 /
C
C LNUMG0(I)=.TRUE. INDICATES THAT THE
      GRID FOR THE ITH VARIABLE SHOULD
C HAVE NUMERIC LABELS ATTACHED.
      DATA(LNUMG0(I),I=1,5) /3*.TRUE,..FALSE,..TRUE./
C
C LNUMSZ(1,I),LNUMSZ(2,I) INDICATE THE
      SIZE OF THE NUMERIC LABELS.

```

```

DATA ((LNUMSZ(I,J),I=1,2),J=1,5) /10*2/
C
C LNUMY(I) IS THE Y RASTER COORDINATE
      FOR THE NUMERICA LABELS A OF THE
C ITH VARIABLE. THE X COORDINATE IS
      DETERMINED BY THE VALUSE TO BE PRINT
DATA (LNUMY(I),I=1,5) / 53,792,53,0,792 /
C
C LFROM(I),LTO(I),LBY(I) INDICATE THE
      NUMBERS TO BE USED AS NUMERIC
C LABELS. LABELS ARE PLACED UNDER THE
      GRID AT POSITIONS CORRESPONDING
C TO NUMBERS BETWEEN LFROM(I), AND LTO(I),
      SEPARATED BY LBY(I)
DATA (LFROM(I),LTO(I),LBY(I),I=1,5) /150.,400.,50.,
X 300.,800.,100., 0.,40.,10., 3*0., 0.,25.,5. /
C
C LHTFR,LHTTO,LHTBY ARE SIMILAR TO LFROM,
      LTO,LBY EXCEPT THEY ARE USED
C IN LABELING THE HEIGHT COORDINATE.
DATA LHTFR,LHTTO,LHTBY /0.,4000.,500. /
C
C LHTIX,LHTSIZ CORRESPON TO LNUMY AND
      LNUMSZ FOR THE HEIGHT CORR.
DATA LHTIX,LHTSIZ(1),LHTSIZ(2) / 30,2,2/
C
C LFRGO(I)=LABEL A FRAME WITH THE HEIGHT
      COORDINATES AND VARIABLE
C DESCRIPTION FREAD AS INPUT.
DATA (LFRGO(I),I=1,5) /,TRUE.,
      ,FALSE.,,TRUE.,2*,FALSE./
C
C ORD=,TRUE. MEANS PLOT THE POINTS IN ORDER BY HEIGHT.
C ORD=,FALSE. MEANS PLOT THE POINTS
      IN THE ORDER THE ARE READ IN.
C THIS ONLY EFFECTS VARIABLES FOR WHICH PLTPF=,FALSE.
DATA ORD/,TRUE./
C
C LSGO CONTROLS THE LABELING OF LINES FOR EACH VAR
C LSGO(I)=,TRUE. MEANS LABEL THE LINE
      WITH THE CHARACTER LSCHR(I).
C THE CHARACTER IS PLACED JUST ABOVE
      THE HIGHEST POINT WHICH WAS PLOTTE
DATA (LSGO(I),I=1,5) /5*,TRUE./
DATA (LSCHR(I),I=1,5) /6H00000N,
      6H00000M,6H00000T,7,6H00000E/
C
C LSSZ CONTROLS THE SIZE OF THE CHARACTERS
      USED TO LABEL LINES.

```

DATA LSSZ(1),LSSZ(2) /3,3/

C
C

C LSGRK INDICATES WHETHER THE CHARACTER
USED TO LABEL THE VARIABLES IS

C TO BE ROMAN OR GREEK. LSGRK(I)=.TRUE.

MEANS THE CHARACTER FOR THE

C ITH VARIABLE IS GREEK, .FALSE. MEANS ROMAN.

DATA (LSGRK(I),I=1,5) /3*.FALSE.,.TRUE.,.FALSE. /

C
C
C

C LHTLX,LHTLY = X,Y START COORD OF TITLE WHE SAYS 'HEIGHT'

DATA LHTLX,LHTLY /10,300 /
END

```

$IBFTC SMLBLK
BLOCK DATA
C ONLY SETS VARIABLES WHICH ARE DIFFERENT
FROM VALUSE REQUIRED FOR
C LARGE PLOTS
C PUTS BOTH PLOTS ON ONE FRAME IN REDUCED SIZE
C SETS UP COMMONS TO PLOT A REDUCED SIZE
INTEGER XEMPH, HTEMPH, HTLAB, HTCHR
REAL LHTFR, LHTT0, LHTBY
COMMON /ZLNUM/ LNUMG0(10), LNUMY(10),
LNUMSZ(2,10), LFR0M(10),
X LTB(10), LBY(10)
COMMON /ZVRTTL/ LVRSZ(2), LVRX, LVRY, LVRDUM(6), VARTTL(5)
COMMON /ZLABEL/ LG0(10), LALPH(5,
10), LSIZE(2,10), LNY(10), LNX(10)
COMMON /ZHTLBL/ LHTLX, LHTLY
COMMON /ZGRIDX/ MTL(10), MTR(10),
XL(10), XR(10), DX(10), XEMPH(10),
X NXLAB(10), NXCHR(10)
COMMON /ZGRIDH/ HTL0W(10), HTMAX(10),
DHT(10), HTEMPH(10), HTLAB(10),
X HTCHR(10), MTB(10), MTT(10)
COMMON /ZLFR/ LHTFR, LHTT0, LHTBY,
LHTIX, LHTSIZ(2), LFRG0(10),
X LFRSIZ(2)
COMMON /ZLS/ LSG0(10), LSSZ(2), LSCHR(10), LSGRK(10)
COMMON /ZPLOTS/NFRAME, NPLOTS(10)
DATA (HTL0W(I), HTMAX(I), DHT(I),
HTEMPH(I), HTLAB(I), HTCHR(I), I=1,5)
X / 0., 4000., 1000., 10, 0, 0,
X 0., 4000., 0., 0, 0, 0,
X 0., 4000., 1000., 10, 0, 0,
X 0., 4000., 0., 0, 0, 0,
X 0., 4000., 0., 0, 0, 0 /
DATA (MTL(I), MTR(I), MTB(I), MTT(I), I=1,5) /
1 80, 500, 75, 660,
1 80, 500, 75, 660,
1 575, 0, 75, 660,
1 575, 0, 75, 660,
1 575, 0, 75, 660 /
DATA LHTBY /1000./
DATA LNUMY /53, 392, 53, 0, 392, 5*0/
DATA (LNX(I), LNY(I), I=1,5) /150,
36, 150, 412, 670, 36, 2*0, 670, 412 /
DATA LHTLX, LHTLY / 10, 150/
DATA LSSZ /2, 2/
DATA NFRAME /1/
DATA NPLOTS(1) /5/
DATA LVRSZ, LVRX, LVRY / 3, 3, 300, 12 /

```

PLBT--BLOCK DATA FOR SMALL PLOTS

PAGE 2

END


```

$IBFTC XPLBT
  COMMON /ZVAR/ VAR(2000,5)
  COMMON /ZHT/ NPTS,HT(2000)
  COMMON /ZPLOTS/NFRAME,NPLOTS(10)
  COMMON /ZGRIDH/ HTLOW(10),HTMAX(10),
                DHT(10),HTEMPH(10),HTLAB(10),
X HTCHR(10),MTB(10),MTT(10)
  COMMON /ZGRIDX/ MTL(10),MTR(10),
                XL(10),XR(10),DX(10),XEMPH(10),
X NXLAB(10),NXCHR(10)
  COMMON /ZPLTPT/ PLTPT(25)
  COMMON /ZSUB/ SUB(10)
  COMMON /ZJOB/NJOBBC,JOBID(10)
  INTEGER SUB
  LOGICAL PLTPT
  INTEGER HTEMPH,HTLAB,HTCHR,XEMPH
  EXTERNAL TABL1V
  WRITE(6,902)
  CALL CHSIZV(15,9)
  CALL RITSTV(150,150,TABL1V)
  CALL RITE2V(50,500,1000,90,1,NJOBBC,-1,JOBID,NERR )
  CALL PRINTV(-7,7HSCHWARZ,50,50)
100 CONTINUE
  CALL INPUT
  CALL ORDER
  NVAR=0
  DO 300 NFR=1,NFRAME
  CALL FRAMEV(2)
  N2=NPLOTS(NFR)
  DO 200 I=1,N2
  NVAR=NVAR+1
  NS=SUB(NVAR)
C NS=0 INDICATES THAT THAT VARIABLE IS MISSING.
  IF(NS.EQ.0) GO TO 200
  CALL SETMIV(MTL(NVAR),MTR(NVAR),MTB(NVAR),MTT(NVAR) )
  CALL GRID1V(2,XL(NVAR),XR(NVAR),
                HTLOW(NVAR),HTMAX(NVAR),
X DX(NVAR),DHT(NVAR),XEMPH(NVAR),
                HTEMPH(NVAR),NXLAB(NVAR),
X HTLAB(NVAR),NXCHR(NVAR),HTCHR(NVAR) )
  CALL TITLE(NVAR)
  IF(.NOT.PLTPT(NFR)) GO TO 150
  CALL APLBTV(=NPTS,VAR(1,NS),HT,1,1,1,42,NERR)
  IF(NERR.GT.0) WRITE(6,900) NERR
  GO TO 200
150 NERR=0
  DO 160 J=1,NPTS
  JL=J+1
  NXL=NXV(VAR(J,NS))

```


PLOT--MAIN ROUTINE

PAGE 2

```

      NYL=NYV(HT(J))
      IF(NXL.NE.0 .AND. NYL.NE.0) GO TO 165
160  NERR=NERR+1
165  DO 180 J=JL,NPTS
      NXN=NXV(VAR(J,NS))
      NYN =NYV(HT(J))
      IF(NXN .EQ.0 .OR. NYN.EQ.0) GO TO 170
      CALL LINEV(NXL,NYL,NXN,NYN)
      NXL=NXN
      NYL=NYN
      GO TO 180
170  NERR=NERR+1
180  CONTINUE
      IF(NERR.NE.0) WRITE(6,900) NERR
200  CONTINUE
300  CONTINUE
      GO TO 100
900  FORMAT(11H THERE WERE,15,12H BAD POINTS.)
901  FORMAT(1H1,9X,6HHEIGHT/(10X,6F10.2))
902  FORMAT(1H1)
      END

```

\$IBFTC XINPUT

```

SUBROUTINE INPUT
COMMON /ZHT/ NPTS,HT(2000)
COMMON /ZVAR/ VAR(2000,5)
COMMON /ZVRTTL/ XVTTL(5),YVTTL(5),VARTTL(5)
COMMON /ZNVAR/NVAR
COMMON /ZFMT/FMT(10)
COMMON /ZFILE/ NFILES,FILEID(20)
INTEGER FILEID,FILE,FIL
INTEGER CURFIL
INTEGER XVTTL,YVTTL
LOGICAL EOF
DATA MAXPTS/2000/
DATA NSTOP/6H*STOP* /
DATA CURFIL/1/

C
C INPUT SUBROUTINE
C READ IN REQUESTS. AND APPROPRIATE DATAS.
100 NPTS = 0
READ(5,900) MISSR,KLOW,KHI,CONTEX,VARTTL
IF( MISSR .EQ. NSTOP) STOP
DO 110 FIL=1,NFILES
FILE = FIL
110 IF(FILEID(FIL).EQ.MISSR) GO TO 120
WRITE(6,901) MISSR
C THE REQUESTED MISSION IS NOT ON THE TAPE.
GO TO 100
120 IF(CURFIL .EQ. FILE) GO TO 150
IF(CURFIL .LT. FILE) GO TO 130
C THIS FILE IS BEFORE CURRENT FILE
WRITE(6,902) MISSR
GO TO 100
130 NSKIP = FILE-CURFIL
C SKIP FILES TO PROPER ONE
DO 140 N=1,NSKIP
140 CALL SKPFIL
CURFIL=FILE
150 CONTINUE
200 READ(1,FMT) KOUNT,HT(NPTS+1),(VAR(NPTS+1,I),I=1,NVAR)
IF(EOF(1)) GO TO 290
IF(KOUNT .LT. KLOW) GO TO 200
IF(KOUNT .GT. KHI) GO TO 300
NPTS = NPTS+1
IF(NPTS .GE. MAXPTS) GO TO 300
IF(KOUNT .EQ. KHI) GO TO 300
GO TO 200
290 CURFIL=CURFIL+1
300 WRITE(6,903) VARTTL,MISSR,KLOW,KHI,NPTS
IF(NPTS .EQ. 0) GO TO 100

```

PL0T--XINPUT, INPUT AIRCRAFT DATA

PAGE 2

```
RETURN
900 FORMAT(A6,2X,2I5,A1,5A6)
901 FORMAT(20HOTHER IS NO MISSION
          A6/17H REQUEST IGNORED. )
902 FORMAT(20HOREQUEST FOR MISSION
          A6,17H IS OUT OF ORDER./
          X 17H REQUEST IGNORED. )
903 FORMAT(11HOGRAPHS FOR,1X,5A6,3X,
          7HMISSION,A6,5H FROM,I5,3H TO,
          X I5,1H.,I6,13H DATA POINTS. )
904 FORMAT(A1,10I5)
905 FORMAT( 25H TOO MANY EXCLUSION CARDS)
END
```

\$IBFTC XINRAD

C INPUT ROUTINE FOR THE RADIOSONDE DATA
(WHICH HAS ALREADY BEEN

C PROCESSED ONTO A TAPE)

C

```

SUBROUTINE INPUT
INTEGER VARTTL, FIXTTL
DIMENSION KDAY(50), KSTAT(50), FIXTTL(5)
LOGICAL FIRST
LOGICAL EOF
COMMON /ZHT/ NPTS, HT(2000)
COMMON /ZVAR/ VAR(2000,5)
COMMON /ZVRTTL/ XXV(10), VARTTL(5)
COMMON /ZSUB/ NSUB(10)
DIMENSION ISUB(10)
DATA ISUB/5,4,1,3,2,5*0/
LOGICAL ALLSTA, ALLDAY
DATA FIXTTL(1) /30HSTAT, DAY, HR

```

DATA FIRST/.TRUE./

C ONLY DO INITIALIZATION ONCE
IF(.NOT.FIRST) GO TO 100

DO 10 I=1,5

10 NSUB(I)=ISUB(I)

READ(5,900) NSTAT, (KSTAT(I), I=1, NSTAT)

READ(5,900) NDAY, (KDAY(I), I=1, NDAY)

WRITE(6,901) NSTAT, (KSTAT(I), I=1, NSTAT)

WRITE(6,902) NDAY, (KDAY(I), I=1, NDAY)

ALLSTA=NSTAT.LE.0

ALLDAY = NDAY.LE.0

100 FIRST=.FALSE.

C READ HEADER RECORD

READ(1) MSTAT, NX, NX, NX, MHR, MDAY, MM0, IB0T, IT0P

IF(EOF(1)) STOP

READ(1) ((VAR(I,J), J=1,4), I=1,500)

READ(1) SKIP

IF(ALLSTA) GO TO 140

DO 130 I=1, NSTAT

130 IF(KSTAT(I).EQ.MSTAT) GO TO 140

GO TO 100

140 IF(ALLDAY) GO TO 160

DO 150 I=1, NDAY

150 IF(KDAY(I).EQ.MDAY) GO TO 160

GO TO 100

160 CONTINUE

DO 180 I=1,5

180 VARTTL(I)=FIXTTL(I)

VARTTL(3) = MSTAT

VARTTL(4)=MDAY

```

VARTTL(5) = MHR
WRITE(6,903) VARTTL
C THE DATA MUST BE MADE INTO WHAT WE WANT TO PLOT.
NPTS=ITOP-IBOT+1
DO 200 N=1,NPTS
NN=N+IBOT-1
C VAR(3) IS HT
HT(N)=VAR(NN,3)
C VAR(1) IS TEMPERATURE
VAR(N,1) = VAR(NN,1)
C VAR(2) IS DEW POINT, CHANGE IT TO VAPOR PRESSURE
C THE FORMULA USED IS AN EMPIRICAL RELATION
DETERMINED BY JOHN SKILLMAN
VAR(N,2) = 10.**((25.058-3009.5/(273.2+VAR(NN,2)))
X -5.439*ALOG10(273.2+VAR(NN,2)))
C VAR(4) IS M, IT DOES NOT HAVE TO BE CHANGED.
VAR(N,4)=VAR(NN,4)
C SETUP VAR(5) AS N.
VAR(N,5) =VAR(NN,4)-VAR(NN,3)/6371.2E3 *1.E6
C VAR 3 IS POTENTIAL TEMPERATURE
PRES=1013.**EXP(-980.*HT(N)*100./2.87E6/280.)
VAR(N,3)=(VAR(N,1)+273.)*(1000./PRES)**.286-273.0
200 CONTINUE
RETURN
900 FORMAT(I6,11A6/(12A6))
901 FORMAT(1H1,I4,10H STATIONS.11A6/(15X,12A6))
902 FORMAT(I5,6H DAYS.4X,11A6/(15X,12A6))
903 FORMAT(5X,5A6)
END

```

```

$IBFTC XORDER
SUBROUTINE ORDER
COMMON /ZORD/ ORD
COMMON /ZHT/ NPTS,HT(2000)
COMMON /ZVAR/ VAR(2000,5)
LOGICAL ORD
DATA MAXVAR/5/
LOGICAL UP,DOWN
IF(.NOT.ORD) RETURN
UP = HT(NPTS) .GT. HT(1)
C   SORT WILL BE ASCENDING OR DESCENDING
    DEPENDING ON OVERALL
C   DIRECTION.
DOWN = .NOT. UP
N = 1
100 IF (N .GE. NPTS) RETURN
    IF(UP.AND.HT(N).GT.HT(N+1) ) GO TO 150
    IF( DOWN.AND.HT(N).LT.HT(N+1) ) GO TO 150
    N = N+1
    GO TO 100
150 CONTINUE
    X = HT(N)
    HT(N)=HT(N+1)
    HT(N+1)=X
    DO 200 I=1,MAXVAR
    X=VAR( N,I)
    VAR(N,I)=VAR(N+1,I)
200 VAR(N+1,I)=X
    N=N+1
    IF(N.LE. 0) N=1
    GO TO 100
END

```

```

$IBFTC XTITLE
SUBROUTINE TITLE(NVAR)
EXTERNAL TABL1V
EXTERNAL TABL2V
INTEGER SUB
INTEGER XVTTL,YVTTL
REAL LHTFR,LHTT0,LHTBY
LOGICAL LFRG0
LOGICAL LG0,LNUMG0
LOGICAL LSG0,LSGRK
REAL LFR0M,LT0,LBY
COMMON /ZLS/ LSG0(10),LSSZ(2),LSCHR(10),LSGRK(10)
COMMON /ZHTLBL/ LHTLX,LHTLY
COMMON /ZVRTTL/ LVRSZ(2),LVRX,LVRY,LVRDUM(6),VARTTL(5)
COMMON /ZLABEL/ LG0(10),LALPH(5,
10),LSIZE(2,10),LNY(10),LNX(10)
COMMON /ZLNUM/ LNUMG0(10),LNUMY(10),
LNUMSZ(2,10),LFR0M(10),
X LT0(10),LBY(10)
COMMON /ZLFR/ LHTFR,LHTT0,LHTBY,
LHTIX,LHTSIZ(2),LFRG0(10),
X LFRSIZ(2)
COMMON /ZVAR/ VAR(2000,5)
COMMON /ZHT/NPTS,HT(2000)
COMMON /ZSUB/ SUB(10)
NUSED=0
IF(.NOT.LFRG0(NVAR)) GO TO 150
CALL CHSIZV(LVRSZ(1),LVRSZ(2))
CALL RITSTV(5*LVRSZ(1),26,TABL1V)
CALL RITE2V(LVRX,LVRY,1023,90,1,30,-1,VARTTL,NUSED)
CALL CHSIZV(LHTSIZ(1),LHTSIZ(2))
CALL RITSTV(LHTSIZ(1)*5+3,26,TABL1V)
CALL RITE2V(LHTLX,LHTLY,1023,180,
1,15,-1,15HEIGHT (METERS),NUSED)
AT=LHTFR
125 CALL BNBCDV(AT,BCD,NDS)
CALL RITE2V(LHTIX,NYV(AT),1000,90,1,NDS,-1,BCD,NUSED)
AT=AT+LHTBY
IF(AT+LE*LHTT0) GO TO 125
150 CONTINUE
IF(.NOT. LG0(NVAR)) GO TO 200
CALL CHSIZV(LSIZE(1,NVAR),LSIZE(2,NVAR) )
CALL RITSTV(5*LSIZE(1,NVAR)+3,26,TABL1V)
CALL RITE2V(LNX(NVAR),LNY(NVAR),
1000,90,1,30,-1,LALPH(1,NVAR),
X NUSED)
200 CONTINUE
IF (.NOT.LNUMG0(NVAR)) GO TO 300
CALL CHSIZV(LNUMSZ(1,NVAR),LNUMSZ(2,NVAR) )

```


PL0T=-XTITLE, LABEL PL0TS

PAGE 2

```
CALL RITSTV(5*LNUMSZ(1,NVAR)+3,26,TABL1V)
AT = LFROM(NVAR)
250 CALL BNBCDV(AT,BCD,NSD)
LEFT=NSD* (5*LNUMSZ(1,NVAR)+3) -3
CALL RITE2V(NXV(AT)-LEFT,LNUMY(NVAR),
           1023,90,1,NSD,-1,BCD,NUSED)
AT = AT+LBY(NVAR)
IF(AT.LE.LT0(NVAR)) GO TO 250
300 CONTINUE
IF(.NOT.LSG0(NVAR)) GO TO 350
NS=SUB(NVAR)
H=.0
DO 310 I=1,NPTS
  IF( H.GE. HT(I) .OR. NYV(HT(I)).EQ.0
     .OR. NXV(VAR(I,NS)) .EQ.0)
    X GO TO 310
  H=HT(I)
  X=VAR(I,NS)
310 CONTINUE
  CALL CHSIZV(LSSZ(1),LSSZ(2))
  IF(.NOT.LSGRK(NVAR)) CALL VCHARV(90,
    1,NXV(X)=5*LSSZ(1),NYV(H)+3,
X LSCHR(NVAR),TABL1V)
  IF( LSGRK(NVAR)) CALL VCHARV(90,
    1,NXV(X)=5*LSSZ(1),NYV(H)+3,
X LSCHR(NVAR),TABL2V)
350 CONTINUE
RETURN
END
$IBFTC XSKP
SUBROUTINE SKPFIL
LOGICAL EOF
10 READ(1,11) I
11 FORMAT(A1)
  IF(EOF(1)) RETURN
GO TO 10
END
```

TRACE SUBROUTINES

TRACE	Main routine controls flow of program.
XGETCR	Reads control cards and profiles. Sets up COMMONS for tracing. Returns when a *TRACE card is encountered.
XGETRA	Sets up for next ray, if any.
XBUMP	Performs iteration.
XHTINT	Function which returns refractive indexes. Its arguments are height and range and it normally performs some interpolation. Also puts in common the limits of linearity for the interpolation it performed.
XDFIND	Utility routine to locate data in a table.
XATTEN	Computes attenuation and reflection from layer or surface.
XPRINT	Produces all printed output.
SFINISH	Logical function which decides whether or not to continue tracing current ray.
TRCBLK	Block data containing certain defaults and sizes.
XPLRAY	Performs manipulation of intermediate plotting tapes at end of ray.
XOUTAL	Performs plotting of all rays traced since it was last called. This is the only routine which calls the assembly language plotting routines.
XPLTPO	Adds current position to intermediate plotting tape.

```

$IBJ08      MAP
$IBFTC TRACE
  LOGICAL SUC
  DOUBLE PRECISION NHT,NTHETA,NEL,NN,NRNG
  DOUBLE PRECISION CHT,CTHETA,CEL,CN,CRNG
  DOUBLE PRECISION NP0S(5),CP0S(5)
  LOGICAL HOLD
  COMMON /ZHOLD/HOLD
  COMMON /CURP0S/ CHT,CTHETA,CEL,CN,CRNG
  COMMON /NXTP0S/ NHT,NTHETA,NEL,NN,NRNG
  EQUIVALENCE (CP0S(1),CHT ),(NP0S(1),NHT )
C  AUTHOR: JERALD SCHWARZ
C  DATE JUNE 1969.
C
C  COMMONS AND VARIABLES
C  /CURP0S/  COMMON CONTAINING THE CURRENT
              POSITION OF THE RAY
C    CHT = CURRENT HEIGHT (IN METERS)
C    CTHETA = CURRENT THETA (EARTH
              CENTRAL ANGLE IN RADIANS)
C    CEL = CURRENT ELEVATION 0 ANGLE (IN RADIANS)
C    CN = CURRENT REFRACTIVE INDEX
C    CRNG = CURRENT RANGE (ALONG EARTH) IN METERS.
C  /NXTP0S/  COMMON CONTAINING THE NEXT POSITION OF RAY
C            I.E. POSITION BEING CALCULATED.
C            VARIABLES ARE SAME AS IN
              CURP0S EXCEPT NAMES HAVE N AS PREFIX
C  /ZTRCP/  CONTAINS TRACE CONTROL PARAMETERS
C    STRRG = START RANGE OF NEXT RAY
C    STRHT = START HEIGHT OF NEXT RAY (IN METERS)
C    STRTEL = START ELEVATION OF NEXT RAY (IN RADIANS)
C    STPRNG = RANGE AT WHICH TO STOP TRACING
C    BMPCT = NUMBER OF RAYS LEFT TO
              TRACE IN THIS SET (A SET IS DETER-
C            MINED BY VBMP AND DEL)
C    VBMP CONTROLS WHICH START PARAMETER
              SHOULD BE INCREMENTED
C            =1 INCREMENT START RANGE.
C            =2 INCREMENT START HIEGHT.
C            =3 INCREMENT START ELEVATION.
C    DEL = AMOUNT TO INCREMENT PARAMETER
              EACH TIME PROGRAM STARTS A NEW
C            RAY.
C  /ZRAD/ ERAD = EARTH'S RADIUS IN METERS.
C  /ZIXP/ IXPROF = THE NUMBER OF THE CURRENT PROFILE.
C  /ZSNELL SNELLC = THE CONSTANT OF
              SNELLS LAW FOR THIS RAY.
C            I.E.  $N*(1.+HEIGHT/RADIUS)*COS(ELEVATION)$ 
C  /ZUP/ UP = .TRUE. WHEN RAY IS PROCEEDING UPWARD

```

```

C      *FALSE WHEN RAY IN GOING DOWNWARD.
C      IT IS SET BY GETRAY AND CHANGED BY BUMP.
C /ZPROF/ CONTAINS INFORMATION ABOUT PROFILES
C      NUMP= NUMBER OF PROFILES IN CORE.
C      PHT(I,J) = HEIGHT OF ITH POINT OF JTH PROFILE.
C      PN(I,J) = REFRACTIVE INDEX OF ITH POINT OF JTH PROFILE
C      MAXP(J) = NUMBER OF POINTS IN JTH PROFILE
C      PRNG(J) = RANGE AT WHICH JTH PROFILE IS SITUATED
C /ZDIM/
C      DIMP1 = LIMIT OF NUMBER OF POINTS IN A PROFILE
C      DIMP2 = LIMIT ON NUMBER OF PROFILE
C              S IN CORE (I.E. ON A PATH)
C /ZTITLE/ TITLE=BCD ARRAY WITH LABELING
C              FOR THIS SET OF TRACES.
C /ZDELHT/ DELHT = INCREMENT IN HEIGHT BETWEEN POINTS.
C              IT IS USED BY BUMP TO GET NHT.
C /ZESC/ ESCAPE = .TRUE. INDICATES THAT
C              THE RAY IS ABOVE THE PROFILE
C              IN CURRENT USE. THUS IT HAS
C              ESCAPED AND TRACING STOPS.
C /ZPRN/ PRINT = .TRUE. INDICATE THE RAY SHOULD BE PRINTED.
C /ZLEVEL/ CONTROLS INTERPOLATION IN RAYNGE.
C      NLEV= NUMBER OF LEVELS (MUST BE
C              SAME IN EACH PROFILE OF A PATH)
C      LHT(I,J) = HEIGHT OF ITH LEVEL IN JTH PROFILE.
C              THE ATMOSPHER IS ASSUMED TO
C              BE LINEAR ALONG A PATH BETWEEN
C              THE POINTS AT HEIGHT LHT(I,J) AND RANGE (LHT
C              THE POINTS AT HEIGHT LHT(I,
C              J) AND RANGE PRNG(J) AND THE POINT
C              AT HEIGHT LHT(I,J+1) AND RANG(PRNG(J+1)
C /ZREFL/ CONTR0.S REFLECTION FROM ELEVATED LAYER.
C      REFL=.TRUE. WHEN REFLECTION IS
C              TO BE COMPUTED (SET BY *REFLECT CAR
C      LOST=.TRUE. WHEN DUE TO REFLECTIONS
C              SIGNAL HAS BECOME TOO WEAK TO
C              BE FOLLOWEDD.
C      STREN=STRENGTH OF RAY CURRENTLY
C              (AS FRACTION OF ORIGINAL )
C      STPSIG= STRENGTH AT WHICH TO STOP
C              TRACING (I AS A FRACTION OF ORIGI
C      LREFL= LEVEL NUMBER OF LAYER TO
C              BE USED FOR REFLECTION.
C      FREQ=FREQUENCY (IN HERZ) TO BE
C              USED IN CALLELATING ATTENUATION
C
C
C
C SUBROUTINES.....

```

```
C GETRAY.  
C   THIS SUBROUTINE INITIALIZES /CURPBS/  
C     FOR THE NEXT RAY TO BE  
C   TRACED. IT RETURNS THE VALUE  
C     .TRUE. IN ITS ARGUMENT IF THERE  
C   IS ANOTHER RAY TO BE DONE AS SPECIFIED  
C     ON THE TRACE CARD.  
C   OTHERWISE IT RETURNS .FALSE.  
C   HTINT(HT,RNG). HEIGHT INTERPOLATION  
C   HT= HEIGHT, RNG = RANGE.  
C   THIS FUNCTION HAS AS ITS VALUE  
C     THE INDEX OF REFRACTION AT THE  
C   GIVEN HEIGHT AND RANGE.  
C   IT HAS AN ALTERNATE ENTRY POINT  
C     HTINTN WHICH IS USED FOR THE FIRST  
C   CALL OF A RAY TO INITIALIZE THE  
C     'WINDOW' AT T WHICH THE FUNCTION  
C   IS LOOKING. AFTER THAT THE 'WINDOW'  
C     MOVES WITH THE RAY.  
C   GETCRD. GET CARDS  
C   THIS ROUTINE READS CONTROL CARDS  
C     WHICH SET VALUES IN COMMONS.  
C   IT RETURNS AFTER IT ENCOUNTERS A *TRACE CARD.  
C   BUMP.  
C   THIS ROUTINE PERFORMS THE ITERATIVE  
C     PROCESS OF DETERMINING  
C   THE NEXT POSITION OF THE RAY.  
C   IT TAKES INTO ACCOUNT TURNING  
C     POINTS, REFLECTION FROM THE SURFACE  
C   AND REFLECTION FROM A LAYER (UNDER CONTROL OF REFL) .  
C   IT CALLS HTINT .  
C   DFIND.  
C   THIS IS A UTILITY ROUTINE USED  
C     TO FIND VALUES IN ARRAYS.  
C   PRINT (ENTRY POINTS=STRTRY , OUTPOS)  
C   THIS SUBROUTINE PERFORMS THE PRINTING  
C   STRTRY IS CALLED WHENEVER AN ARRAY  
C     IS INITIALIZED. (CALLED BY GETRA  
C   OUTPOS IS CALLED (BY MAIN) FOR  
C     EVERY POINT COMPUTED. IT CALLS PLT  
C   BUTRAY  
C   THIS SUBROUTINE IS CALLED WHENEVER  
C     A RAY IS COMPLETED. ITS MAIN  
C   FUNCTION IS TO CONTROL THE TAPES  
C     USED IN THE PLOTTING.  
C   BUTALL  
C   THIS ROUTINE IS CALLED WHENEVER  
C     A GROUP OF ARRAYS HAS BEEN COMPLET  
C   IT PERFORMS THE PLOTTING. IT IS  
C     THE ONLY ROUTINE WHICH CALLS THE
```



```
C   PLOTTING ROUTINES.  THUS IF THE
C       PROGRAM IS TO BE REWRITTEN TO
C   PRODUCE PLOTS USING DIFFERENT
C       SUBROUTINES THIS IS THE ONLY
C   SUBROUTINE WHICH WOULD HAVE TO BE CHANGED.
C   PLTP0S.
C   CALL TO CONTROL THE PLOTTING FOR
C       EACH POINT (I.E. IT WRITES INTERM
C   TAPES.
C   ATTEN. FUNCTION WITH TWO ENTRY POINTS
C   ATTEN = COMPUTES THE ATTENUATION
C       COEFFICIENT FOR REFLECTION
C   FROM THE ELEVATED LAYER AND
C       THE ANGLE AND RANGE SPECIFIED.
C   SURFAT= ATTENUATION DUE TO A REFLECTION
C       FROM THE SURFACE AT THE
C   ANGLE SPECIFIED.
C
C   CALL PLINIT
C   READ CONTROL CARDS, AND PROFILES
C   100 CALL GETCRD
C   INITIALIZE FOR TRACING A RAY
C   110 CALL GETRAY(SUC)
C   IF NO MORE RAYS HAVE BEEN SPECIFIED
C       GET MORE CONTROL CARDS.
C       IF(.NOT.SUC) GO TO 160
C       CALL BUTP0S
C   130 CALL BUMP
C       CALL FINISH(SUC)
C       DO 140 I=1,5
C   140 CP0S(I)=NP0S(I)
C       CALL BUTP0S
C       IF(SUC) GO TO 150
C       GO TO 130
C   WHEN RAY IS FINISHED TAKE APPROPRIATE ACTIONS.
C   150 CALL BUTRAY
C       GO TO 110
C   160 IF(.NOT.H0LD) CALL BUTALL
C       GO TO 100
C   END
```

\$IBFTC XGETCR

SUBROUTINE GETCRD

DOUBLE PRECISION PHT,PN,PRNG

INTEGER CTL(15),TYPE

INTEGER PEND

INTEGER DIMP1,DIMP2,DIML1

INTEGER TYPE1,NAMES(3)

LOGICAL PRINT

LOGICAL HOLD

DOUBLE PRECISION LHT

REAL PLPAR(6)

REAL PLTDEF(6)

INTEGER T1,T2

LOGICAL PLOT,TEND,FIRSTR

LOGICAL REFL,L0ST

COMMON /ZREFL/ REFL,L0ST,STREN,STPSIG,LREFL,FREQ

COMMON /ZDIM/ DIMP1,DIMP2,DIML1

COMMON /ZPLOT/ PLOT,PLFRNG,PLDRNG,

PLHL0,PLHHI,PLDEN,PLHGRD,

X THT(40),TRNG(40),T1,T2,TEND,FIRSTR,NRAY

COMMON /ZLEVEL/ LHT(20,10),NLEV

COMMON /ZTITLE/ TITLE(13)

COMMON /ZPRN/PRINT

COMMON /ZPROF/ PHT(200,10),PN(200,

10),PRNG(10),MAXP(10),NMP

COMMON /ZTRCP/ STRTRG,STRHT,STRTEL,

STPRNG,BMPCT,VBMP,DEL

COMMON /ZDELHT/DELHT

COMMON /ZHOLD/ HOLD

DIMENSION TRCPAR(7)

DIMENSION PARAM(7)

DATA (CTL(I),I=1,15) /5H*PATH,

5H*STOP,5H*PROF,6H*PRINT,6H*N0PRI,

X 6H*TRACE,5H*PLOT,6H*N0PL0,6H*DELHT,6H*REFLE,6H*N0REF,

X 5H*HOLD,6H*HOLDE,2*0/

DATA MAXTP /15/

DATA (NAMES(I),I=1,3) /5HRANGE,6HHEIGHT,2HEL/

DATA PEND /5H*PEND/

DATA LEV/6H*LEVEL /

DATA PLTDEF/0.,100.,0.,4000.,100.,900./

EQUIVALENCE (PLFRNG,PLPAR(1))

EQUIVALENCE (STRTRG,TRCPAR(1))

DOUBLE PRECISION INT,X,Y,X1,X2,Y1,Y2

INT(X,X1,Y1,X2,Y2) = (X-X1)/(X2-X1)*(Y2-Y1) + Y1

WRITE(6,911)

C READ CONTROL CARDS AND PROFILES.

100 CONTINUE

READ(5,900) TYPE,TYPE1,PARAM

WRITE(6,906) TYPE,TYPE1


```

      DO 110 N=1,MAXTP
110  IF(CTL(N).EQ.TYPE) GO TO (200,
      250,300,350,360,400,500,600,370,
      X 650,660,670,680 ) , N
      WRITE(6,901) TYPE,PARAM
      GO TO 100
C *PATH
200  NUMP=0
      GO TO 100
C *STOP
250  STOP
C *PROF
300  NUMP=NUMP+1
      IF(NUMP.GT.DIMP2) GO TO 320
      PRNG(NUMP) = PARAM(1)*1000.
      WRITE(6,907) NUMP,PRNG(NUMP)
      IF(NUMP.GT.1 .AND. PRNG(NUMP).LE.PRNG(NUMP-
      1))WRITE(6,919)
      IF(NUMP.GT.1 .AND. PRNG(NUMP).LE.PRNG(NUMP-
      1))GO TO 321
      N=0
C THE GROUND IS ALWAYS A LEVEL
      NL=1
      LHT(1,NUMP)=0.
310  READ(5,902) TYPE,PHT(N+1,NUMP),PN(N+1,NUMP)
      IF(TYPE.EQ.PEND) GO TO 340
      N=N+1
      PN(N,NUMP) =1.+ PN(N,NUMP)*1.E-6
      IF(N.GE.DIMP1) GO TO 330
      IF(TYPE.EQ.LEV) GO TO 340
      CAPN= (PN(N,NUMP)-1.)*1.E6
      WRITE(6,908) TYPE,PHT(N,NUMP),CAPN
      IF(N.GT.1 .AND. PHT(N,NUMP).LT.PHT(N-
      1,NUMP) ) GO TO 325
      GO TO 310
320  WRITE(6,903)
321  NUMP=NUMP+1
      GO TO 335
325  WRITE(6,920)
      N=N+1
      GO TO 310
330  WRITE(6,904)
335  READ(5,902) TYPE
      IF(TYPE.EQ.PEND) GO TO 100
      GO TO 335
C
340  NL=NL+1
      LHT(NL,NUMP)=PHT(N,NUMP)
      CAPN=(PN(N,NUMP)-1.)*1.E6

```

```
WRITE(6,914) NL,PHT(N,NUMP),CAPN
IF(N.GT.1 .AND. PHT(N,NUMP).LT.PHT(N-
1,NUMP) ) GO TO325
IF(TYPE.NE.PEND) GO TO 310
```

```
C
  345 WRITE(6,908) TYPE
      MAXP(NUMP)=N
C ALL PROFILES MUST HAVE THE SAME NUMBER OF LEVELS
  IF(NUMP.EQ.1) NLEV=NL
  IF(NL.EQ.NLEV) GO TO 100
  NLEV=MIN1(NL,NLEV)
  WRITE(6,917) NLEV
  GO TO 100
C *PRINT
  350 PRINT= .TRUE.
      GO TO 100
C NOPRI
  360 PRINT= .FALSE.
      GO TO 100
C *DELHT
  370 DELHT = PARAM(1)
      WRITE(6,913) DELHT
      GO TO 100
C *TRACE
  400 DO 410 I=1,7
  410 TRCPAR(I)=PARAM(I)
      STRTRG=STRTRG*1000.
      STPRNG=STPRNG*1000.
      READ(5,905) TITLE
      IF(VBMP.EQ.1.) DEL=DEL*1000.
      IF(VBMP.LT.1. .OR. VBMP.GT.3.) BMPCT=1.
      WRITE(6,912) TITLE
      WRITE(6,909) STRTRG,STRHT,STRTEL,STPRNG
      NBMP=VBMP
      MBMP= BMPCT
      IF(BMPCT .GT. 1.) WRITE(6,910)
          MBMP,NAMES(NBMP),NBMP,DEL
C INITIALIZE PARAMETERS FOR PLOTTING.
  IF(HOLD.AND. NRAY.NE.0) RETURN
  NRAY=0
  T1=2
  T2=3
  TEND = .TRUE.
  REWIND T1
  REWIND T2
  RETURN
C *PLOT
  500 PLOT=.TRUE.
      DO 510 I=1,6
```

```

      PLPAR(I)=PARAM(I)
510  IF(PARAM(I).EQ.0.) PLPAR(I)=PLTDEF(I)
      PLFRNG=PLFRNG*1.E3
      PLDRNG=PLDRNG*1.E3
      WRITE(6,916) PLPAR
      GO TO 100
C *NOPL0
600  PLOT=.FALSE.
      GO TO 100
C *REFLE
650  REFL=.TRUE.
      LREFL=PARAM(1)
      IF(LREFL.EQ.0) LREFL=2
      IF(PARAM(2).LE.0.) PARAM(2)=100.
      STPSIG=-ABS(PARAM(2))
      IF(PARAM(3).EQ.0.) PARAM(3)=50.
      FREQ=PARAM(3)*1.E6
      WRITE(6,918) LREFL,STPSIG,PARAM(3)
      GO TO 100
C *NOREFL
660  REFL=.FALSE.
      GO TO 100
C *HOLD
670  HOLD=.TRUE.
      NRAY=0
      GO TO 100
C *HOLDE
680  CONTINUE
      IF(HOLD) CALL BUTALL
      HOLD=.FALSE.
      GO TO 100
900  FORMAT(A6,A4,7F10.9)
901  FORMAT(32H FOLLOWING CARD IS UNRECOGNIZED./
          1X,A6,4X,7F10.4)
902  FORMAT(A6,4X,3D10.0)
903  FORMAT(19H TOO MANY PROFILES.)
904  FORMAT(30H PROFILE HAS TOO MANY HEIGHTS.)
905  FORMAT(13A6)
906  FORMAT(5X,A6,A4)
907  FORMAT(10X,7HPROFILE,I3,3H AT,
          -3PF5.0,3H KM,8X,6HHEIGHT,6X,1HN )
908  FORMAT(5X,A6,28X,F7.0,F7.1)
909  FORMAT(10X,12HSTART RANGE=,-3PF4.0,4H KM./
          2 10X,13HSTART HEIGHT=,0PF5.0,7H METERS /
          3 10X,16HSTART ELEVATION=,0PF7.4,8H RADIAN /
          1 10X,11HSTOP RANGE=,-3PF5.0,3H KM /
          X )
910  FORMAT(10X5HTRACE,I3,6H RAYS./
          10X8HVARYING A6,10H (VARIABLE

```

X 12,4H) BY,F13.4,11H EACH TIME.)
911 FORMAT(1H1)
912 FORMAT(10X,13A6)
913 FORMAT(10X,6HDELHT=F5.0,8H METERS.)
914 FORMAT(20X,5HLEVEL,I3,11X,F7.0,F7.1)
915 FORMAT(40X,F8.0,3PF8.1)
916 FORMAT(10X,14HPLOTS START AT -3PF5.0,4H KM./
X 10X,19HEACH FRAME DISPLAYS -3PF5.0,4H KM./
X 10X,25HMINIMUM HEIGHT DISPLAYED=,0PF6.0,3H M./
X 10X,25HMAXIMUM HEIGHT DISPLAYED=,0PF6.0,3H M./
X 10X,23HGRID LINES APPROX EVERY,
OPF5.0,14H RASTER UNITS. /
X 10X,12HPLBT HEIGHT= OPF5.0,14H RASTER UNITS.)
917 FORMAT(10X, 40HWRONG NUMBER OF
LEVELS, LEVEL COUNT NOW,I3,1H.)
918 FORMAT(10X,27HRAY WILL REFLECT FROM LEVEL,I3,1H. /
X 10X,33HTRACE UNTILL SIGNAL HAS
DECREASED OPF5.0,4H DB./
X 10X,20HRAY HAS FREQUENCY OF,F6.0,6H MGHZ.)
919 FORMAT(10X,47HPROFILE RANGES MUST
INCREASE. PROFILE IGNORED.)
920 FORMAT(40X,46HHEIGHTS MUST INCREASE.
PREVIOUS POINT IGNORED.)
END

```

$IBFTC XGETRA
      SUBROUTINE GETRAY(NEWRAY)
C INITIALIZE CURPOS FOR THE PLOTTING
      OF ANOTHER RAY ACCORDING TO THE
C CONTROL INFORMATION INZTRCP
      DOUBLE PRECISION HTINTN
      DOUBLE PRECISION SNELLC
      DOUBLE PRECISION ERAD
      DOUBLE PRECISION CHT,CTHETA,CEL,CN,CRNG
      LOGICAL UP
      LOGICAL NEWRAY
      LOGICAL REFL,L0ST
      INTEGER T1,T2
      LOGICAL PLOT,TEND,FIRSTR
      COMMON /ZPLOT/ PLOT,PLFRNG,PLDRNG,
               PLHL0,PLHHI,PLDEN,PLHGRD,
X   THT(40),TRNG(40),T1,T2,TEND,FIRSTR,NRAY
      COMMON /ZREFL/ REFL,L0ST,STREN,STPSIG,LREFL,FREQ
      COMMON /ZTRCP/ STRTRG,STRTHT,STRTEL,
               STRPRG,BMPCT,VBMP,DEL
      COMMON /CURPOS/ CHT,CTHETA,CEL,CN,CRNG
      COMMON /ZRAD/ERAD
      COMMON /ZSNELL/ SNELLC
      COMMON /ZUP/ UP
      DIMENSION TRCPAR(7)
      EQUIVALENCE (STRTRG,TRCPAR(1))
      NEWRAY = .FALSE.
      IF(BMPCT.LT. .1 ) RETURN
C THERE WAS ANOTHER RAY SPECIFIED.
      NEWRAY=.TRUE.
      CRNG=STRTRG
      CHT=STRTHT
      CEL =STRTEL
      CTHETA=CRNG/ERAD
      STREN=0.
      UP = CEL .GE. 0.
      CN= HTINT (CHT,CRNG)
      SNELLC =CN*(1.+CHT/ERAD)*DCOS(CEL)
      NRAY=NRAY+1
      IF(NRAY.GT.40) WRITE(6,900)
      IF(NRAY.GT. 40) NRAY=40
      CALLSTRTRY
      BMPCT=BMPCT-1.
      N=VBMP+.1
      IF(N.EQ.0)RETURN
      TRCPAR(N)=TRCPAR(N)+DEL
      RETURN
900 FORMAT(58H0ATTEMPT TO PLOT MORE
           THAN 40 RAYS TOGETHER. ONLY 40 USE

```

TRACE--GETRAY, INITIALIZES NEXT RAY

PAGE 2

XD.)
END

\$IBFTC XBUMP

```

SUBROUTINE BUMP
DOUBLE PRECISION SNELLC
DOUBLE PRECISION ERAD
DOUBLE PRECISION NHT,NTHETA,NEL,NN,NRNG
DOUBLE PRECISION CHT,CTHETA,CEL,CN,CRNG
DOUBLE PRECISION HTINT,HTINTN
DOUBLE PRECISION CTAN,NTAN,XSQ,XD1,HALFPI
DOUBLE PRECISION DARCOS
DOUBLE PRECISION COSEL,CRN,RNN
DOUBLE PRECISION HLIN1,HLIN2
DOUBLE PRECISION RBAR,HBAR,DENOM,DTHETA
DOUBLE PRECISION PHT,PN,PRNG
DOUBLE PRECISION LHT
DOUBLE PRECISION A,B,C,D,RT1,RT2
LOGICAL UP
LOGICAL ESCAPE
LOGICAL TURN
LOGICAL REFL,LOST
COMMON /ZREFL/ REFL,LOST,STREN,STPSIG,LREFL,FREQ
COMMON /ZINTL/ HLIN1,HLIN2
COMMON /NXTPOS/ NHT,NTHETA,NEL,NN,NRNG
COMMON /CURPOS/ CHT,CTHETA,CEL,CN,CRNG
COMMON /ZESC/ ESCAPE
COMMON /ZUP/ UP
COMMON /ZRAD/ERAD
COMMON /ZSNELL/ SNELLC
COMMON /ZDELHT/ DELHT
COMMON /ZIX/ IXP1,IXP2,IXH1,IXH2,IXL,IXP,IXH
COMMON /ZPRBF/ PHT(200,10),PN(200,
10),PRNG(10),MAXP(10),NUMP
COMMON /ZLEVEL/ LHT(20,10),NLEV
C THIS SETS UP THE NEXT POINT FOR STEP.
C IT INCREMENTS HIGHT, AND ELEVATION.
LOST=.FALSE.
TURN=.FALSE.

```

100 CONTINUE

```

C HTLIN1, AND HLIN2 ARE THE BOUNDS IN
      WHICH THE ATMOSPHERIC MODEL IS
C LINEAR AROUND THE LAST HEIGHT FOR WHICH HTINT WAS CALLED.
C TO INSURE PROPER TRACING, ESPECIALLY
      NEAR LAYERS THESE HEIGHTS SHOULD
C BE EXPLICITLY USED.
      IF(UP) NHT=DMIN1(CHT+DELHT,HLIN2)
      IF(UP) NHT=DMAX1(CHT+1.,NHT)
      IF(.NOT.UP) NHT=DMAX1(CHT-DELHT,HLIN1)
      IF(.NOT.UP) NHT=DMIN1(CHT-1.,NHT)
C SPECIAL ACTION IF WE GO BELOW 0 HEIGHT.
      IF(NHT .GE. 0.) GO TO 150

```



```

C HAVE'NT QUITE REACHED THE SURFACE. GO TO IT THIS TIME.
  IF( CHT .GT. 0.) GO TO 140
C START BACK UP.
C MAY WANT A NEW PROFILE FIRST.
  UP=.TRUE.
  CEL=ABS(CEL)
  IF(REFL) STREN=STREN+ALOG10(SURFAT(CEL))
  IF(REFL) LOST=STREN.LT.STPSIG
  GO TO 100
140 NHT=0.
150 CONTINUE
C NOTE. IXL IS THE VALUE SET BY THE
      LAST CALL TO HTINT WHICH SHOULD HAVE
C BEEN THE CALL DURING BUMP FOR CHT
  IF(.NOT.REFL .OR. .NOT.UP .OR. IXL.NE.LREFL) GO TO 160
C WANT THE REFLECT FROM A LAYER
  CEL=-CEL
C CORRECT FOR LANT OF LAYER
  IF(IXP1.NE.IXP2) CEL=CEL+
  X 2.*DATAN( DBLE( (LHT(LREFL,IXP2)-LHT(LREFL,IXP1)) /
  X (PRNG(IXP2)-PRNG(IXP1)) ))
  UP=CEL.GE.0.
C RECOMPUTE SNELLS CONSTANT
  SNELLC=CN*(1.+CHT/ERAD)*DCOS(CEL)
  STREN=STREN+ALOG10(ATTEN(CEL,CRNG))
  LOST=STREN.LT.STPSIG
CH
  CHT= (LHT(LREFL,IXP2)-LHT(LREFL,
      IXP1)) * (CRNG-PRNG(IXP1)) /
  * (PRNG(IXP2) -PRNG(IXP1)) + LHT(LREFL,IXP1)
  CN==
  CN=HTINT(CHT,CRNG)
  GO TO 100
160 CONTINUE
  NN=HTINT(NHT,CRNG)
  C0SEL = SNELLC/(NN*(1.+NHT/ERAD))
C CHECK IF WE HAVE REFLECTION
  IF(C0SEL .GT. 1.) GO TO 200
  NEL =DARCOS(C0SEL)
  IF(.NOT.UP) NEL=-NEL
  GO TO 210
C
C HERE WHEN WE HAVE A TURNING POINT.
200 CONTINUE
  IF(CEL.NE.0.) GO TO 202
  NHT=(CHT+NHT)/2.
  IF(ABS(NHT-CHT).GT..4) GO TO 160
  NHT=CHT
  NEL=CEL

```

```

IF(TURN) GO TO 205
TURN=.TRUE.
UP=.NOT.UP
GO TO 100
202 CONTINUE
C NEW HEIGHT SOLUTION OF FOLLOWING QUADRATIC EQUATION
C  $SNELLC*ERAD=(CN+DNDH*(NHT-CHT))*(ERAD+NHT)$ 
C WHERE DNDH IS LOCAL DERIVATIVE OF N
 $DNDH=(NN-CN)/(NHT-CHT)$ 
A=DNDH
B=DNDH*(ERAD-CHT)+CN
C=ERAD*(CN-DNDH*CHT-SNELLC)
D=SQRT(B*B-4.*A*C)
RT1=.5*(-B+D)/A
RT2=.5*(-B-D)/A
T1=(CHT-RT1)*(NHT-RT1)
T2=(CHT-RT2)*(NHT-RT2)
NHT=RT2
IF(T1.LE.0.0.AND.(T2.GT.0.0.OR.T2.LE.0.0.AND.ABS(T1-
      CHT).LE.
1ABS(T2-CHT))) NHT=RT1
IF(T1.GE.0.0.AND.T2.GE.0.0.AND.T1.LT.T2) NHT=RT1
IF(T1.GT.0.0.AND.T2.GT.0.0) GO TO 205
NEL=0.
NN=SNELLC/(1.+NHT/ERAD)
UP=.NOT.UP
IF(TURN.AND. ABS(NHT-CHT).LT..001) GO TO 205
TURN=.TRUE.
IF(ABS(NHT-CHT).LT..001) GO TO 100
GO TO 210
C WAVE IS JUST FOLLOWING CURVATURE
205 DTHETA=.0015
GO TO 250
C NOW WE COMPLETE THE NEW POSITION OF
      THE RAY BY DETERMINING THETA AND
C RANGE. THIS ROUTINE USES FORMULA'S
      DERIVED BY GARDINER. SEE
C PACIFIC MISSILE RANGE TECHNICAL NOTE 3280-6.
C DETERMINATION OF ELEVATION AND SLANT
      RANGE ERRORS DUE TO ATMOSPHERIC
C REFRACTION.
210 CONTINUE
RBAR=.5*(CN+NN)
HBAR=(CHT+NHT)/2.
DENOM=(NN-CN)*(HBAR+ERAD)+RBAR*(NHT-CHT)
IF(DABS(DENOM).LT..01) GO TO 300
DTHETA=(NEL-CEL)*RBAR*(NHT-CHT)/DENOM
IF(DTHETA.LT.0.00.OR.DTHETA.GT..0100) GO TO 300
250 CONTINUE

```

TRACE=BUMP, 'BUMPS' RAY TO NEXT POSITION PAGE 4

NTHETA=CTHETA + DTHETA

NRNG = NTHETA*ERAD

RETURN

C DENOMINATOR IS TOO SMALL.

C THIS SHOULDN'T HAPPEN OFTEN, WHEN
IT DOES USE ALTERNATE FORM

C OF EQUATIONS.

300 CONTINUE

CTAN = DSIN(CEL)/DCOS(CEL)

NTAN= DSIN(NEL)/DCOS(NEL)

XSQ= ((NTAN-CTAN)/(1.+NTAN*CTAN))**2

XD1= (NN*(NHT=CHT)*(1./DCOS(CEL)+1./DCOS(NEL)))/

X (SNELLC*(NTAN+CTAN)*(1.+NTAN*CTAN)*ERAD)

DTHETA=XD1*(1.+XSQ/3.-XSQ*XSQ/5.+XSQ*XSQ*XSQ/7.)

GO TO 250

END

*IBFTC XHTINT

DOUBLE PRECISION FUNCTION HTINT(HT,RANGE)

C RETURNS THE INDEX OF REFRACTION AT
THE GIVEN HEIGHT AND RANGE.

DOUBLE PRECISION HT,RANGE

DOUBLE PRECISION INT,X,X1,X2,Y1,Y2

DOUBLE PRECISION LINT,REL,HT1,HT2

DOUBLE PRECISION LINTH1,LINTL0

DOUBLE PRECISION LHT

DOUBLE PRECISION PHT,PN,PRNG

DOUBLE PRECISION HLIN1,HLIN2

LOGICAL LAYER

DOUBLE PRECISION HTREL

LOGICAL ESCAPE

LOGICAL ESC

COMMON /ZINTL/ HLIN1,HLIN2

COMMON /ZESC/ ESCAPE

COMMON /ZPROF/ PHT(200,10),PN(200,
10),PRNG(10),MAXP(10),NUMP

COMMON /ZLEVEL/ LHT(20,10),NLEV

COMMON /ZIX/ IXP1,IXP2,IXH1,IXH2,IXL,IXP,IXH

C
INT(X,X1,Y1,X2,Y2) = (X-X1)/(X2-X1)*(Y2-Y1) + Y1C
C HTINTN IS CALLED AT THE BEGINNING OF A RAY.
ENTRY HTINTN(HT,RANGE)

100 CONTINUE

C
C TEST IF THERE IS TO BE INTERPOLATION IN HEIGHT ONLY

IXP=1

IF(NUMP.EQ.1) GO TO 310

C
C FIND THE PROFILES FOR RANGE INTERPLATION.

CALL DFIND(RANGE,PRNG,NUMP,IXP1,ESC)

IXP2=IXP1+1

C IF PAST LAST PROFILE USE IT ONLY

IF(ESC) GO TO 300

C IF THERE ARE LEVELS FIND OUT WHICH ONE THIS IS IN.

IF(NLEV.EQ.1) GO TO 210

NL=NLEV-1

DO 200 IXL=1,NL

LINTH1=INT(RANGE, PRNG(IXP1), LHT(IXL+1,IXP1),

X PRNG(IXP2),LHT(IXL+1,IXP2))

200 IF(LINTH1.GE.HT) GO TO 220

210 HT1=HT

HT2=HT

GO TO 230

220 LINTL0=INT(RANGE,PRNG(IXP1),LHT(IXL,IXP1),PRNG(IXP2),

X LHT(IXL,IXP2))

```

REL= (HT-LINTL0)/(LINTHI-LINTL0)
HT1=LHT(IXL,IXP1) + REL*(LHT(IXL+
      1,IXP1) -LHT(IXL,IXP1))
HT2=LHT(IXL,IXP2) + REL*(LHT(IXL+
      1,IXP2) -LHT(IXL,IXP2))
230 CALL DFIND(HT1,PHT(1,IXP1),MAXP(IXP1),IXH1,ESC)
    ESCAPE=ESC
    CALL DFIND(HT2,PHT(1,IXP2),MAXP(IXP2),IXH2,ESC)
    ESCAPE=ESC.0R.ESCAPE
    RN1=INT(HT1, PHT(IXH1,IXP1),PN(IXH1,IXP1),
X   PHT(IXH1+1,IXP1),PN(IXH1+1,IXP1) )
    RN2=INT(HT2, PHT(IXH2,IXP2),PN(IXH2,IXP2),
X   PHT(IXH2+1,IXP2),PN(IXH2+1,IXP2) )
    HTINT=INT(RANGE,PRNG(IXP1),RN1,PRNG(IXP2),RN2 )
    HLIN1=DMAX1(
X   INT(PHT(IXH1,IXP1),LHT(IXL,IXP1),
      LINTL0,LHT(IXL+1,IXP1),LINTHI),
X   INT(PHT(IXH2,IXP2),LHT(IXL,IXP2),
      LINTL0,LHT(IXL+1,IXP2),LINTHI))
    HLIN2=DMIN1(
XINT(PHT(IXH1+1,IXP1),LHT(IXL,IXP1),
      LINTL0,LHT(IXL+1,IXP1),LINTHI),
XINT(PHT(IXH2+1,IXP2),LHT(IXL,IXP2),
      LINTL0,LHT(IXL+1,IXP2),LINTHI))
    RETURN
C COME HERE WHEN THERE IS ONLY ONE PROFILE TO BE USED
300 CONTINUE
    IXP=IXP2
    IF(RANGE.LE.PRNG(1)) IXP=1
C INTERPOLATE THE INDEX IN THAT PROFILE
310 CALL DFIND(HT,PHT(1,IXP),MAXP(IXP ),IXH,ESCAPE)
    HTINT=INT(HT,PHT(IXH,IXP),PN(IXH,IXP),
X   PHT(IXH+1,IXP),PN(IXH+1,IXP))
    HLIN1=PHT(IXH,IXP)
    HLIN2=PHT(IXH+1,IXP)
C FOR USE IN CASE OTHER ROUTINES WANT TO LOCATE RAY
    IXP1=IXP
    IXP2=IXP
    IXH1=IXH
    IXH2=IXH
    CALL DFIND(HT,LHT(1,IXP),NLEV,IXL,ESC)
    RETURN
END

```



```

$IBFTC XDFIND
SUBROUTINE DFIND(X,DATA,LIM,L1,   ESC)
DOUBLE PRECISION DATA(1),X
LOGICAL ESC

C
C THIS SUBROUTINE LOCATES THE ENTRIES
C           IN A TABLE OF ASCENDING VALUES
C WHICH BRACKET X.
C DATA IS THE TABLE. IT MUST BE ARRANGED IN ASCENDING ORDER.
C   I.E. DATA(I) .LE. DATA(I+1)
C LIM IS THE NUMBER OF ENTRIES IN THE TABLE.
C L1 ON RETURN IS THE LOWER SIDE OF THE BRACKET. I.E.
C   DATA(L1) .LE. X .LE. DATA(L1+1)
C IF X FALLS OUTSIDE THE TABLE ESC IS
C           SET TRUE, AND L1 IS SET TO 1
C OR LIM-1 DEPENDING ON WHETHER X IS
C           BELOW OR ABOVE THE RANGE COVERED
C BY THE TABLE.
C
C IF L1 ON ENTRY IS WITHIN THE LIMITS
C           OF THE TABLE, THE SEARCH FOR
C L1 WILL START FROM ITS CURRENT VALUE.
C           (THIS WILL MAKE REPEATED
C CALLS MORE EFFICIENT IN MANY CASES).
C
C
      ESC=.FALSE.
      IF(L1.GT.0.AND.L1.LT.LIM) GO TO 110
      L1=1
110  CONTINUE
      IF(DATA(L1).GT.X)GO TO 150
      IF(DATA(L1+1) .GT. X) RETURN
      L1=L1+1
      IF(L1.LT.LIM)GO TO 110
      L1=LIM-1
      GO TO 200
150  IF(L1.LE.1)GO TO 200
      L1=L1-1
      GO TO 110
200  CONTINUE
210  ESC=.TRUE.
      RETURN
      END

```

```

$IBFTC XATTEN
  REAL FUNCTION ATTEN(EL,RNG)
C  COMPUTE THE REFLECTION COEFFICIENT
      FOR THE SPECIFIED LAYER AT THE
C  GIVEN RANGE FOR A RAY WITH THE GIVEN ELEVATION.
  DOUBLE PRECISION EL,RNG
  DOUBLE PRECISION PHT,PN,PRNG
  DOUBLE PRECISION LHT
  LOGICAL REFL,L0ST
  COMMON /ZREFL/ REFL,L0ST,STREN,STPSIG,LREFL,FREQ
  COMMON /ZIX/ IXP1,IXP2,IXH1,IXH2,IXL,IXP,IXH
  COMMON /ZPR0F/ PHT(200,10),PN(200,
      10),PRNG(10),MAXP(10),NUMP
  COMMON /ZLEVEL/ LHT(20,10),NLEV
  DATA PI/3.141592/
C  TEST IF THERE IS ONLY ONE KPROFILE.
  X=0.
  IF(IXP1.EQ.IXP2) GO TO 110
  X= (RNG-PRNG(IXP1))/(PRNG(IXP2)-PRNG(IXP1))
110 CONTINUE
  HT=(1.-X)*LHT(LREFL,IXP1) + X*LHT(LREFL,IXP2)
  SLOPE= (HTINT(HT+10.,RNG)-HTINT(HT,RNG)) *1.E-7
  VATTEN=SLOPE*(3.E8/FREQ)/(8.*PI*SIN(EL)**3)
  ATTEN=AMIN1(VATTEN**2,1.)
  RETURN
  ENTRY SURFAT(EL)
C  COMPUTE REFLECTION COEFFICIENT FROM SURFACE.
C  USE FORMULAS FROM ESSA TECH. REPT.
      ERL 79-ITS 67, PAGE 8-4.
C  ASSUME WAVE HEIGHT OF 3 METERS.
  DATA WHT/3./
  DATA PI/3.14159/
  VATTEN= AMAX1(EXP(-2.*PI*.39*WHT*SIN(EL)/(3.E8/FREQ)),
  X SQRT(ABS(SIN(EL)))) )
  ATTEN=VATTEN**2
  RETURN
  END

```



```

$IBFTC XPRINT
SUBROUTINE STRTRY
DOUBLE PRECISION CHT,CTHETA,CEL,CN,CRNG
LOGICAL PRINT
LOGICAL REFL,L0ST
C
COMMON /ZREFL/ REFL,L0ST,STREN,STPSIG,LREFL,FREQ
COMMON /ZPRN/PRINT
COMMON /ZTITLE/ TITLE(13)
COMMON /CURPOS/ CHT,CTHETA,CEL,CN,CRNG
C
IF(,NOT,PRINT) GO TO 300
WRITE(6,904)
WRITE(6,905) TITLE
WRITE(6,900)
WRITE(6,902)
WRITE(6,903)
NCT=0
RETURN
ENTRY OUTPOS
CALL PLTP0S
IF(,NOT,PRINT) RETURN
200 CORN=(CN-1.)*1.E6
WRITE(6,901) CRNG,CHT,CEL,CORN,CTHETA
NCT=NCT+1
IF(NCT.LT.50) RETURN
NCT=0
WRITE(6,904)
WRITE(6,900)
WRITE(6,902)
WRITE(6,903)
RETURN
300 WRITE(6,906) CHT,CRNG,CEL
RETURN
ENTRY OUTRAY
WRITE(6,907) CHT,CRNG,CEL,STREN
CALL PLTRAY
RETURN
900 FORMAT(10X,5HRANGE,9X,6HHEIGHT,
          6X,9HELEVATION,11X,4HREF.,10X,
          X 5HTHETA)
901 FORMAT(-3PF15.4,0PF15.1,F15.4,F15.1,E15.3)
902 FORMAT(10X,5(2HIN,13X)/12X3HKM.,
          9X6HMETERS,8X7HRADIANS,8X7HN UNITS
          X 8X7HRADIANS)
903 FORMAT(1H0)
904 FORMAT(1H1)
905 FORMAT(20X,8HNEW RAY,5X,13A6/1X)
906 FORMAT(1H0,9X,13HSTART HEIGHT=F6.0,
          3H M.,5X12HSTART RANGE=,

```

TRACE--PRINT, DETAILED LISTING OF RAY

PAGE 2

```
X -3PF5.0,4H KM.,5X9HSTART EL=OPF8.4,5H RAD. )  
907 F0RMAT(11X,12HSTOP HEIGHT=F6.0,  
          3H M.,6X11HSTOP RANGE=-3PF5.0,  
X 4H KM.,6X8HSTOP EL=OPF8.4,5H  
          RAD.,6X,12HATTENUATION=F6.0,4H DB.)
```

END

\$IBFTC XFIN

```
SUBROUTINE FINISH(DONE)
DOUBLE PRECISION CHT,CTHETA,CEL,CN,CRNG
DOUBLE PRECISION NHT,NTHETA,NEL,NN,NRNG
LOGICAL ESCAPE
LOGICAL DONE
LOGICAL REFL,L0ST
COMMON /ZREFL/ REFL,L0ST,STREN,STPSIG,LREFL1,LREFL2
COMMON /CURPOS/ CHT,CTHETA,CEL,CN,CRNG
COMMON /NXTPOS/ NHT,NTHETA,NEL,NN,NRNG
COMMON /ZTRCP/ STRTRG,STRTHT,STRTEL,
               STPRNG,BMPCT,VBMP,DEL
COMMON /ZESC/ ESCAPE
DONE= NRNG.GE.STPRNG .OR. ESCAPE .OR. L0ST
IF(NRNG.LE.STPRNG) RETURN
X=(STPRNG-CRNG)/(NRNG-CRNG)
NRNG=STPRNG
NHT=CHT+ X*(NHT-CHT)
NTHETA=CTHETA+ X*(NTHETA-CTHETA)
NEL=CEL+X*(NEL-CEL)
NN=CN+ X*(NN-CN)
RETURN
END
```

```
$IBFTC TRCBLK
BLOCK DATA
DOUBLE PRECISION ERAD
INTEGER DIMP1,DIMP2,DIML1
LOGICAL HOLD
COMMON /ZHOLD/ HOLD
COMMON /ZDIM/ DIMP1,DIMP2,DIML1
COMMON /ZRAD/ERAD
COMMON /ZHTLAY/ HTLAY
COMMON /ZDELHT/ DELHT
DATA ERAD/ 6371.2D3/
DATA DIMP1,DIMP2 /200,10/
DATA DIML1 /20/
DATA HOLD/.FALSE./
DATA DELHT /20./
END
```

```

$IBFTC XPLTRA
      SUBROUTINE PLTRAY
      INTEGER T1,T2
      LOGICAL PLOT,TEND,FIRSTR
      LOGICAL PRINT
      COMMON /ZPRN/PRINT
      COMMON /ZPLOT/ PLOT,PLFRNG,PLDRNG,
                   PLHL0,PLHHI,PLDEN,PLHGRD,
      X   THT(40),TRNG(40),T1,T2,TEND,FIRSTR,NRAY
C THIS ROUTINE FINDS THE END OF TAPE T1 AND MARKS IT. ON T2.
      IF(.NOT.PLOT) RETURN
C IF ALREADY AT END GO MARK T2
      IF(TEND) GO TO 120
      110 READ(T1) THT,TRNG
      IF(THT(1).LT.-5.) GO TO 120
      THT(NRAY)=-1.
      WRITE(T2) THT,TRNG
      GO TO 110
C AT END OF TAPE MARK IT.
      120 CONTINUE
      DO 125 N=1,NRAY
      125 THT(N)=-1.
      THT(1)=-10.
      WRITE(T2) THT,TRNG
      ENDFILE T2
      REWIND T1
      REWIND T2
C SWITCH ROLES TO T1 AND T2.
      I=T1
      T1=T2
      T2=I
C NO LONGER ON FIRST RAY OR AT END OF TAPE
      TEND=.FALSE.
      RETURN
      900 FORMAT(1H0,9X,12HATTENUATION=F5.0,4H DB.)
      END

```

```

$IBFTC XOUTAL
SUBROUTINE OUTALL
LOGICAL MORE
LOGICAL OVER
LOGICAL HOLD
REAL THTX(40),TRNGX(40)
INTEGER TIN
INTEGER T1,T2
LOGICAL PLOT,TEND,FIRSTR
LOGICAL PRINT
COMMON /ZPRN/ PRINT
COMMON /ZHOLD/ HOLD
COMMON /ZTITLE/ TITLE(13)
COMMON /ZPLOT/ PLOT,PLFRNG,PLDRNG,
               PLHL0,PLHHI,PLDEN,PLHGRD,
X   THT(40),TRNG(40),T1,T2,TEND,FIRSTR,NRAY
C THIS ROUTINE USES THE TAPE PRODUCED
  BY OUTPOS AND OUTRAY TO PLOT
C THE RAY PATHS.
  IF(.NOT.PLOT) RETURN
  MORE=.TRUE.
  TIN=T1
  FRMRNG=PLFRNG
190  T0RNG=FRMRNG + PLDRNG
  IF(.NOT.MORE) RETURN
  MORE=.FALSE.
  REWIND TIN
  CALL SETMIV(50,0,50,1023-50-IFIX(PLHGRD))
  CALL DXDYV(1,FRMRNG/1000.,T0RNG/
             1000.,DX,N,I,NX,PLDEN,IERR)
  CALL DXDYV(2,PLHL0,PLHHI,DY,M,J,NY,PLDEN,IERR)
  CALL GRID1V(3,FRMRNG/1000.,T0RNG/1000.,PLHL0,PLHHI,
X   DX,DY,N,M,I,J,NX,NY)
  CALL RITE2V(300,20,1000 ,90,1,30, -1,TITLE,IERR)
C READ IN THE RIST POINTS
220 READ(TIN) THT,TRNG
C READ IN THE NEXT SET OF POINTS
230 READ(TIN) THTX,TRNGX
  WRITE(6,1) (THTX(I),TRNGX(I), I=1,NRAY)
1   FORMAT(10F10.0)
C IF AT END OF TAPE, WE ARE DONE.
  IF(THTX(1) .GT. -5.) GO TO 240
  FRMRNG=T0RNG
  IF(MORE) GO TO 190
  RETURN
240 OVER=.TRUE.
  DO 300 N=1,NRAY
  MORE=MORE .OR. (TRNGX(N).GT.T0RNG .AND. THTX(N).GE.0.)
  IF( TRNGX(N).LE.T0RNG .AND. THTX(N).GE.0.)
    OVER=.FALSE.

```

```

IF(THT(N).LT.0. .OR. TRNGX(N).LT.FRMRNG
   .OR. TRNG(N).GT.TORNG)
X   GO TO 290
IF(TRNG(N).EQ.TORNG .OR. TRNGX(N).EQ.FRMRNG) GO TO 290
IF(THTX(N).GT.PLHHI) GO TO 290
IF(TRNG(N).GE.FRMRNG.AND.TRNGX(N).LE.TORNG) GO TO 270
IF(TRNG(N).LT. FRMRNG) GO TO 260
C AT END OF FRAME.
  THTX(N)=THT(N)+(TORNG-TRNG(N))/(TRNGX(N)-TRNG(N))*
  X (THTX(N)-THT(N))
  TRNGX(N)=TORNG
  GO TO 270
C AT START OF FRAME.
260 THT(N)=THT(N) + (FRMRNG-TRNG(N))/(TRNGX(N)-TRNG(N))*
  X (THTX(N)-THT(N))
  TRNG(N)=FRMRNG
270 CONTINUE
C HAVE A RAY SEGMENT TO PLOT, ALSO
  INDICATE THAT THE MAY STILL BE
C MORE POINTS TO PLOT.
  NH=NYV(THT(N))
  NR= NXV(TRNG(N)/1000.)
  NHX=NYV(THTX(N))
  NRX=NXV(TRNGX(N)/1000.)
  IF(NH.NE.0 .AND. NR.NE.0 .AND.
     NHX.NE.0 .AND. NRX.NE.0)
  X   CALL LINEV(NR,NH,NRX,NHX)
290 TRNG(N)=TRNGX(N)
  THT(N)=THTX(N)
300 CONTINUE
  IF(.NOT.OVER) GO TO 230
C HAVE FINISHED A FRAME.
  FRMRNG=TORNG
  GO TO 190
  ENTRY PLINIT
C
C GARBAGE FOR PLOTTING STARTUP
  EXTERNAL TABL1V
  CALL RITSTV(24,17,TABL1V)
  CALL RITE2V(100,500,1000,90,1,7,-1,7HWILSON ,N)
C
  HOLD=.FALSE.
  PRINT=.FALSE.
  PLOT=.FALSE.
  RETURN
  END

```



```

$IBFTC XPLTPB
SUBROUTINE PLTPBS
DOUBLE PRECISION CHT,CTHETA,CEL,CN,CRNG
LOGICAL PLOT,TEND,FIRSTR
INTEGER T1,T2
COMMON /CURPBS/ CHT,CTHETA,CEL,CN,CRNG
COMMON /ZPLOT/ PLOT,PLFRNG,PLDRNG,
               PLHLO,PLHHI,PLDEN,PLHGRD,
X   THT(40),TRNG(40),T1,T2,TEND,FIRSTR,NRAY
C THIS SUBROUTINE OUTPUTS A TAPE WHICH
      WILL LATTER BE USED TO
C PRODUCE A PLOT.
C EACH LOGICAL RECORD CONTAINS THE HEIGHT
      AND RANGE OF UP TO 40 RAYS.
C A NEGATIVE HEIGHT INDICATES THAT THERE
      IS NO DATA PRESENT FOR THAT
C RAY IN THIS RECORD. A VALUE .LT.
      -5. FOR THE FIRST HEIGHT INDICATES
C THAT THIS IS THE END OF THE TAPE. (I.E. A LOGICAL EOF).
C
      IF(.NOT.PLOT) RETURN
C EXCEPT FIRST RAY, OR WHEN END OF TAPE
      IS REACHED READ IN AN OLD
C RECORD.
      IF ( TEND) GO TO 200
      READ(T1) THT,TRNG
C TEST FOR END OF TAPE.
      IF(THT(1) .GE. -5.) GO TO 200
      TEND =.TRUE.
      THT(1)=-1.
C WILL USE THT(1) IN WRITTING SO DON'T LET IT BE -10.
200 THT(NRAY) = CHT
      TRNG(NRAY)=CRNG
      WRITE(T2) THT,TRNG
      RETURN
      END

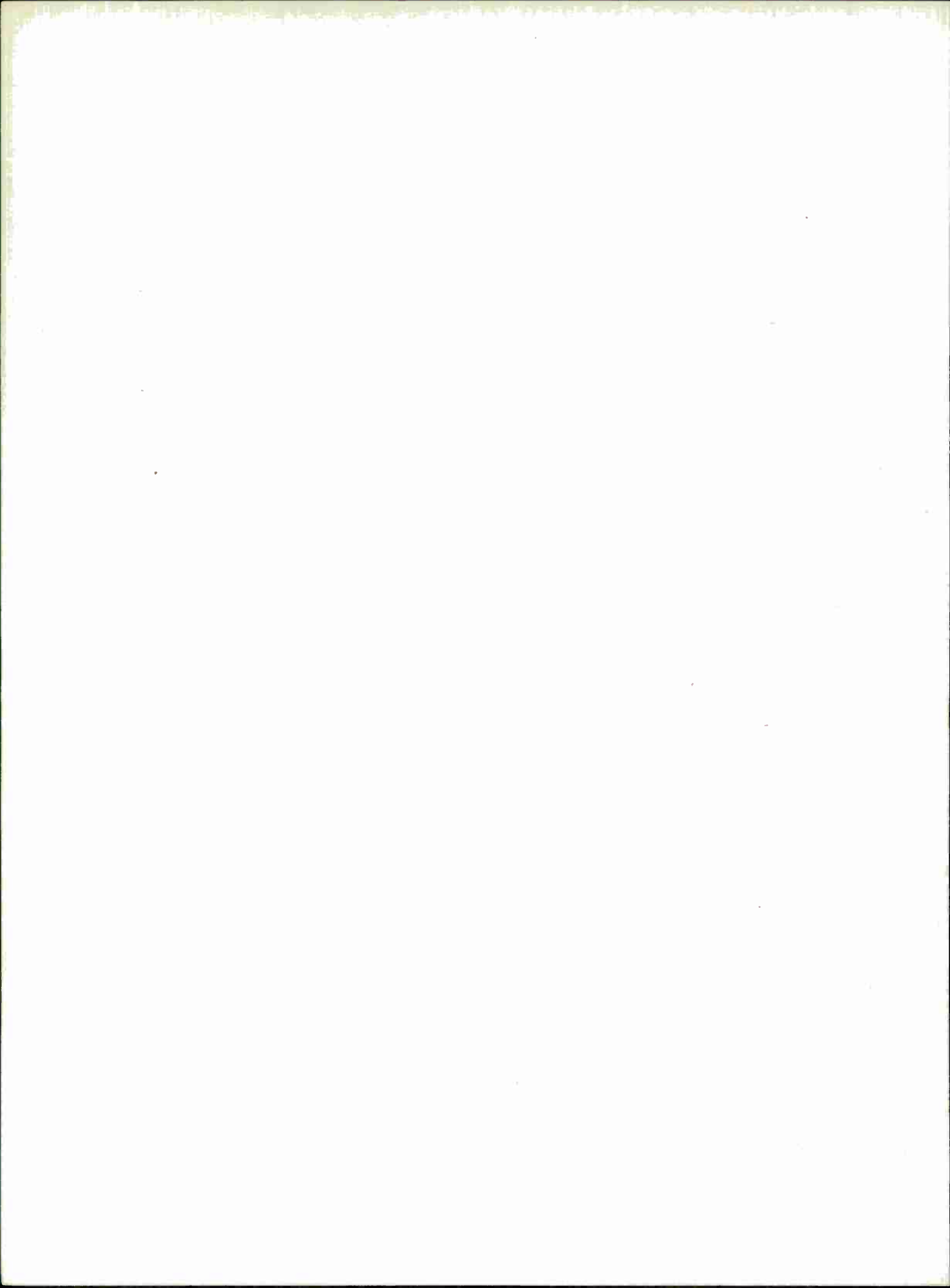
```

```
$IBFTC XARCOS
/DOUBLE PRECISION FUNCTION DARCOS(X)
DOUBLE PRECISION X
DOUBLE PRECISION PI
DATA PI /3.141592653589793D0/
DARCOS= DATAN(DSQRT(1.D0-X*X)/X)
IF(X.LT.0.D0) DARCOS=PI+DARCOS
RETURN
```

CONTROL CARD SUMMARY

B10827

FUNCTION		PARAMETERS						
COL. 1-10		COL. 11-20	COL. 21-30	COL. 31-40	COL. 41-50	COL. 51-60	COL. 61-70	COL. 71-80
* PRINT	TURNS ON PRINTING OPTION							
* NO PRINT	TURNS OFF PRINTING OPTION							
* STOP	TERMINATES PROCESSING							
* PATH	RESETS PROGRAM FOR NEW PROFILE							
* PROF	INITIATES READING OF PROFILE	RANGE						
* PEND	TERMINATES READING OF PROFILE							
* TRACE	INITIATES RAY TRACING	START- RANGE	START- HEIGHT	START- ELEVATION	STOP- RANGE	BUMP- COUNT	BUMP- VARIABLE	DEL
* PLOT	TURNS ON PLOTING OPTION	START- RANGE	FRAME- RANGE	FRAME- BOTTOM	FRAME- TOP	DENSITY	GRID	
* NO PLOT	TURNS OFF PLOTING OPTION							
* DELHT	CONTROLS STEP SIZE	DEL						
* REFLECT	TURNS ON REFLECTION OPTION	LEVEL	STOP- ATTENUATION	FREQ				
* NO REFLECT	TURNS OFF REFLECTION OPTION							
* HOLD	DELIMITS START OF RAY COLLECTION							
* HOLD END	DELIMITS END OF RAY COLLECTION							
	PROFILE DESCRIPTION	HEIGHT	N					
* LEVEL	PROFILE DE- SCRIPTION (LEVEL)	HEIGHT	N					
	TITLE (alphanumeric title in Col. 1 - 30)							



DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Syracuse University Research Corporation MerriII Lane, University Heights, Syracuse, New York 13210		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
3. REPORT TITLE DESCRIPTION OF COMPUTER PROGRAMS FOR THE ANALYSIS AND PRESENTATION OF TRADE WINDS DATA		2b. GROUP N/A	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) None			
5. AUTHOR(S) (First name, middle initial, last name) Jerald Schwarz			
6. REPORT DATE December 1969		7a. TOTAL NO. OF PAGES 147	7b. NO. OF REFS 0
8a. CONTRACT OR GRANT NO. FI9628-68-C-0209		9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TR-70-32	
b. PROJECT NO.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c.			
d.			
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Aerospace Instrumentation Program Office, Electronic Systems Division, AFSC, USAF, L G Hanscom Field, Bedford, Mass. 01730	
13. ABSTRACT An investigation of the Trade Wind Duct was carried out from March 6 through March 25, 1969 in the Northern part of the Caribbean Sea. An instrumented aircraft was used to record meteorological and radio refractivity data in digitized format for computer analysis. In addition, extensive radio-sonde data was included in the analysis to support the aircraft measurements and provide a basis for weather analysis. In order to assimilate, process and present such a large amount of data it was imperative that machine processing be used. The following report describes the various programs which were used in the analysis and presentation of the data. A ray-tracing program was also developed to analyze radio wave propagation in relation to Trade Wind Duct characteristics. This program has the advantage that horizontal changes in the Duct can be included. Most ray-tracing programs assume that the vertical variation of refractivity is spherically stratified.			

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
1. Ray Tracing Program 2. Radio-Meteorology 3. Meteorological Data Analysis 4. TradeWinds Data Analysis Program 5. Computer Analysis of Radio-Meteorological Data.						

