AIRPOL-4A ALGORITHMS

by

William A. Carpenter
Research Engineer

(The opinions, findings, and conclusions expressed in this report are those of the author and not necessarily those of the sponsoring agencies.)

Virginia Highway & Transportation Research Council
(A Cooperative Organization Sponsored Jointly by the Virginia Department of Highways and Transportation and the University of Virginia)

Charlottesville, Virginia

June 1976
VHTRC 76-R58

2780

ABSTRACT

This report provides a detailed explanation of the inner workings of the computer program AIRPOL-4A, a computer model for predicting the impact of highway generated air pollution. The report is intended to serve both as a supportive document for AIRPOL-4A and as a detailed road map of the program should modifications be required in the future.

FINAL REPORT

AIRPOL-4A ALGORITHMS

by

William A. Carpenter
Research Engineer

INTRODUCTION

AIRPOL-4A[1] is a slightly modified version of the computerized model AIRPOL-4,[2,3] a Gaussian line-source formulation of an air quality prediction model for use in estimating the impact, in terms of carbon monoxide (CO) concentration levels, of a highway on the micro-region of such a facility. Version 4 was upgraded to Version 4A in response to the release by the U. S. Environmental Protection Agency of Supplement 5 to AP-42, <u>Compilation of Air Pollutant Emission Factors</u>, and the impending requirement by the U. S. Federal Highway Administration that Supplement 5 emission factors be employed in the preparation of environmental impact statements. The modifications to AIRPOL-4 which produced AIRPOL-4A were all in the realm of input/output functions. Thus, the mathematical structure of AIRPOL-4A is unchanged from that of AIRPOL-4, and as a consequence the documents establishing AIRPOL-4 as a theoretically sound and functionally accurate, reliable and efficient model[2,3,4,5] apply equally as well to AIRPOL-4A.

The purpose of this report is to provide in-depth explanations of the algorithms making up AIRPOL-4A. This report will thus serve both to answer questions concerning the inner workings of AIRPOL-4A and to guide those who may in the future be required to modify the program.

The text of this report is directly referenced to the AIRPOL-4A program listing (and <u>vice versa</u>) contained in the Appendix by the computer card numbers shown to the left of the text.

MAIN PROGRAM

Description of Program Parameters

The program parameters are all those variables and constants used in the AIRPOL-4A program. Definitions of all parameters may be found in the program listing in the Appendix. The parameter definitions are alphabetical (with numeric entries following alphabetic entries) within each of the six categories; logical, integer, and real scalars and logical, integer, and real arrays.

Input Formats

A4A0000	Format specification for reading a Header Card as described in Reference 1.
A4A0010	Format specification for pre-reading a Data Card. Note that under this format no occurrence of "illegal data in field" can occur, thus the inclusion of comments on END and ENS Cards(1) is allowed.
A4A0020 to A4A0030	Format specification for completely reading a Data Card as described in Reference 1.

Output Formats

A4A0040 to A4A0110	Format specification for printing the heading information for the check/correct output page.
A4A0120 to A4A0130	Format specification for informing the user that a null site (two sequential ENS Cards or a Header Card followed by an ENS Card) has been encountered.
A4A0140 to A4A0150	Format specification for informing the user that a null division (a Header Card followed by zero or more ENS Cards followed by one END Card) has been encountered.
A4A0160 to A4A0170	Format specification for printing the columnar headings on the check/correct output page when METRIC = TRUE, i.e., metric inputs.
A4A0180 to A4A0190	Format specification for printing the columnar headings on the check/correct output page when METRIC = FALSE, i.e., American Engineering inputs.
A4A0200 to A4A0210	Format specification for printing underscores for columnar headings on check/correct output page.
A4A0220 to A4A0230	Format specification for echoing the Data Card inputs on the check/correct output page.
A4A0240 to A4A0250	Format specification for printing question marks on the check/correct output page below those Data Card inputs for which default values were used.
A4A0260	Format specification for leaving two blank lines on the output.
A4A0270 to A4A0350	Format specification for printing the heading information on the analysis output page.

A4A0360 Format specification for printing analysis results on the analysis output page.

A4A0370 Format specification for printing the left margin characters on the analysis output page.

A4A0380 Format specification for printing stability classes on the analysis output page.

A4A0390 Format specification for printing calendar years on the analysis output page.

A4A0400 Format specification for printing a horizontal line to separate the CLASS(·) categories on the analysis output page.

A4A0410 Format specification for leaving a single blank line on the output.

A4A0420 Format specification for printing horizontal lines to separate the YEAR(·) categories on the analysis output page.

A4A0430 Format specification for skipping to next top of page on output.

A4A0440 Format specification for informing the user that no analyses were performed for the current site since the total emissions for the site were zero.

AIRPOL-4A Specifications

A4A0450 to A4A0590 Specify the logical, integer and real scalars and vectors used in AIRPOL-4A.

A4A0600 to A4A0640 Equate (in memory) the vector ARRAY(·) with the Data Card input parameters. This equivalence speeds up the Data Card input process while keeping the code readable by people.

A4A0650 Make the memory area called MINE jointly accessible by the main program and the subroutine FIXIT(·,·) and BLOCK DATA.

A4A0660 to A4A0880 Initialize those program variables which will act as constants throughout the code. See the program listing for details.

AIRPOL-4A Operating Code

A4A0890 Position the line printer at top of the first output page.

A4A0900 Call REREAD to initialize a nonstandard FORTRAN record/buffer management system. See the program listing for details.

A4A0910 Read a Header Card and check for an end of file condition. Branches to this statement occur after the processing of each END Card encountered in the data set.

A4A0920 Set FIRST to TRUE because the next Data Card to be read will be the first Data Card for the current site.

A4A0930 Set DIDONE to FALSE because no data cards have yet been processed for the current division.

A4A0940 Reset the page counter to zero for the initialization for the current division.

A4A0950 Reset the lane group counter to zero for the initialization of the current site.

A4A0960 to A4A0970 Set the maximum indices for the output categories (48 = 8 distances x 3 years x 2 classes) from the array SUPOUT(·,·).

A4A0980 to A4A0990 Initialize receptor height (1) to 1.5 meters = (approximately) 5.0 feet.

A4A1000 to A4A1030 Reset the accumulating array SUPOUT(·,·) to zero for the initialization of the current site.

A4A1040 Default to 60 minutes if the value of TIME is non-positive.

A4A1050 to A4A1060 Find the equivalent of TIME in hours and minutes.

A4A1070 Find the ratio of TIME to PTIME.

A4A1080 to A4A1140 CLASS(·) is input as a hexadecimal number. This loop equates 1 & A, 2 & B, ··· and 6 & F, and 0 & Δ, 7 & Δ, 8 & Δ, and 9 & Δ and leaves CLASS(·) as a hexadecimal Δ, A, B, C, D, E, or F, and ICLASS(·) as a decimal 0, 1, 2, 3, 4, 5, or 6. Note: Δ = blank.

A4A1150 to A4A1190 — Ensure that at least one class is valid (default is 2 or B) and that the two classes are not redundant.

A4A1200 to A4A1210 — If either class is empty (i.e., = 0), set the appro-iate J1 or J48 to skip that class.

A4A1220 to A4A1370 — Find an upper bound on the downwind distance from source to receptor at which the maximum integration error in INTGRL will be ≤ 0.001 meter/kilometer of roadway located beyond such an upper bound. This upper bound is found by solving for all points (P, DIST) such that

$$\left(\exp - \tfrac{1}{2}\left(\frac{P}{\sigma_P}\right)^2\right) \Big/ \left(\sigma_P \times \sigma_Z\right) \le 10^6.$$

Since the maximum DIST for this condition will be at P = 0, we solve $\sigma_P \times \sigma_Z = 10^6$ for DIST. (Note σ_P and σ_Z are functions of DIST). The solution is found interactively using a Newton-Raphson technique. If the solution is found to be $\ge 2 \times 10^4$, the algorithm terminates and sets the solution equal to 2×10^4, since 20 kilometers represents an effective value of infinity.

A4A1380 to A4A1440 — Find the valid wind speed inputs, place them sequentially in the vector WSIN(·) while filling the unused portion of WSIN(·) with an effective value of infinity, and maintain a count in W6 (which is previously zeroed) of the number of valid wind speeds.

A4A1450 to A4A1500 — If W6 = 0, then set W6 = 6 and fill WSIN(·) with the appropriate metric or American Engineering default wind speeds.

A4A1510 — Set W12 = 2 × W6 for later use on the analysis output page. The reader will note that there are two columns of output for each wind speed.

A4A1520 to A4A1610 — Store the metric equivalents of the input wind speeds in MWSIN(·) and the American Engineering equivalents in WSIN(·).

A4A1620 — Pre-read a Data Card and test for an end of file. Branches to this card occur after the completion of each lane group.

A4A1630 to A4A1660	If the Data Card was an ENS or END Card then branch to either the diagnostics area of AIRPOL-4A, if this was the first Data Card in the current site, or to the output area, if this was not the first card in the current site.
A4A1670	Reread the complete Data Card (previously determined not to be an END or ENS Card) to obtain the lane group data.
A4A1680 to A4A1690	Initialize CUT and FILL to FALSE for the processing of this lane group.
A4A1700	Increment the lane group counter.
A4A1710	Set receptor height (1) for this lane group to receptor height (1) for the first lane group in this site.
A4A1720 to A4A1770	Fill CORECT(·) with blanks and fill the even numbered elements of FMT(·) with T1's. CORECT(·) will be overwritten with question marks and FMT(·) will be overwritten with appropriate format elements in those locations corresponding to defaulted inputs.
A4A1780 to A4A1860	If this is the first Data Card in the current site, fill DOYRAL(·) with FALSE's, increment the page counter, and print the complete check/correct page and columnar headings.
A4A1870	Echo the Data Card inputs.
A4A1880	Initialize LOCATR, the check/correct pointer, to 1.
A4A1890 to A4A1990	Test TFVOL(·) AND EF(·) for validity. If invalid inputs are found call FIXIT (·,·). Set DOYR(I) = TRUE if, and only if, the total emissions for year I are positive. Set DOYRAL(I) = TRUE if, and only if, at least one lane group in the current site had positive emissions for year I.
A4A2000 to A4A2020	If SOBS is invalid, use the default value and call FIXIT(·,·).
A4A2030	Set CUT = TRUE if, and only if, SOBS = 1 or SOBS = 2.

A4A2040 to A4A2200	Set H = 1.5 meters = (approximately) 5.0 feet. If HEIGHT > 0 Set FILL = TRUE. If CUT = FALSE and HEIGHT < 0 set HEIGHT = ABS(HEIGHT), set FILL = TRUE, and call FIXIT(·,·). If CUT = FALSE, set H = H + HEIGHT. If CUT = TRUE and HEIGHT = 0, set SOBS = 0, set CUT = FALSE, and call FIXIT(·,·). If CUT = TRUE and HEIGHT > 0 set HEIGHT = -HEIGHT and call FIXIT(·,·).
A4A2210 to A4A2220	Convert ULENGH and DLENGH from kilometers or kilofeet to meters or feet.
A4A2230 to A4A2330	If CUT = FALSE and CWIDTH ≠ 0, set CWIDTH = 0 and call FIXIT(·,·). If CUT = TRUE and CWIDTH < 0, set CWIDTH = - CWIDTH and call FIXIT(·,·). If CUT = TRUE and CWIDTH = 0, set CWIDTH = 100 meters = (approximately) 328 feet and call FIXIT(·,·).
A4A2340 to A4A2420	If CUT = FALSE and CLENGH ≠ 0, set CLENGH = 0 and call FIXIT(·,·). If CUT = TRUE and CLENGH < 0, set CLENGH = - CLENGH and call FIXIT(·,·). If CUT = TRUE and CLENGH = 0, set CLENGH = ULENGH and call FIXIT(·,·).
A4A2430	CUT1 = TRUE if, and only if, SOBS = 1.
A4A2440	CUT2 = TRUE, if and only if, SOBS = 2.
A4A2450	CHT = magnitude of the cut depth.
A4A2460	If CUT = TRUE, locate the effluent on the road surface. This eliminates the possibility that the effluent is emitted at a height above the top of the cut for shallow cuts.
A4A2470 to A4A2510	Equate 1 & D and 2 & U (case may be input as either numeric or alpha). If a non-valid CASE is found set CASE = 1 and call FIXIT(·,·).
A4A2520 to A4A2560	If CUT1 = TRUE, force ALPHA = 0, and call FIXIT(·,·), if ALPHA was not previously = 0.
A4A2570	If ALPHA is not between 0 and 90, set ALPHA = 90 and call FIXIT(·,·).
A4A2580 to A4A2670	If FIRST = TRUE, set Z(2) = TZ(2) = HT2 where HT2 defaults to 3 meters = (approximately) 10 feet, if it is either negative or equal to TZ(1). Call FIXIT(·,·) if HT2 defaults.

A4A2680 to A4A2710	If FIRST = FALSE, force HT2 = Z(2) = TZ(2), and call FIXIT(·,·), if HT2 was not already equal to TZ(2).
A4A2720 to A4A2740	If DIN < 0, set DIN = 0 and call FIXIT(·,·).
A4A2750 to A4A2770	If any receptor point will be < 0, set INC = 0 and call FIXIT(·,·).
A4A2780 to A4A2820	If FIRST = TRUE, fill D(·) with the eight receptor distances and set SITE = SITEID.
A4A2830 to A4A2970	If the inputs are not in metric form, convert all computational variables to their metric equivalents.
A4A2980 to A4A3010	Print the check/correct results via the two vectors CORECT(·) and CHANGE(·).
A4A3020	Set FIRST = FALSE.
A4A3030	If the total emissions for all years for the current lane group are zero, get the next lane group to be processed.
A4A3040 to A4A3080	Evaluate loop independent variables prior to entering analysis loop.
A4A3090 to A4A3110	If CUT1 = TRUE, use stability class F parameters to determine the effective upwind roadway length.[2]
A4A3120 to A4A3190	Initiate outer loop and convert vector references to scalar references for use in inner loops.
A4A3200 to A4A3220	If FILL = TRUE or CUT2 = TRUE, modify the offset parameters as described in Reference 2.
A4A3230 to A4A3250	Finish computing the outer-loop dependent variables for use in the inner loops.
A4A3260 to A4A3280	Initiate middle loop, and compute middle loop variables for use in inner loop.
A4A3290	Move all receptors 0.5 meters further from roadway to avoid question of where two coincident lines intersect. (Two parallel lines intersect at $\pm\infty$ but coincident lines intersect everywhere.)
A4A3300	Initiate inner loop.

A4A3310	If all receptors are the same distance from the road, perform inner loop only once.
A4A3320 to A4A3390	Initialize inner loop variables taking care not to generate an overflow with TAN (90°).
A4A3400 to A4A3500	Any roadway points lying outside the envelope described by $\left\lvert \frac{P}{\sigma_P} \right\rvert \leq 6$ will contribute less than 1.5×10^{-5} meters/kilometer of roadway to INTGRL, a two order of magnitude safety factor in the allowable error of 0.007 meter for INTGRL with up to 4 kilometers of such roadway. Thus, we solve $\lvert P \rvert = 6 \times \sigma_P$ for R (see Reference 2, figures 2 and 3), assuming that σ_P is a linear function of DIST (this will introduce some error but it will be a conservative error, i.e., it will increase our safety factor). There are two solutions to the above equation. If $P < 0$ the solution will be R^- and if $P > 0$ the solution will be R^+. In the downwind case, R^- is a lower bound of the envelope and R^+ is an upper bound (when it exists — an anomolous result occurs when XALPHA ≤ XALSTR, i.e. when the intersection is at $+\infty$). In the upwind case, R^- does not exist since no roadway points have both $P < 0$ and DIST > 0, and R^+ may be either an upper or lower bound depending on both the relation of XALPHA to XALSTR and the location of the point (0, DIST), which may be on either the upwind or downwind side of the line source. If XALPHA < XALSTR then R^+ will be a lower bound of the envelope, regardless of the location of 0, DIST). However, if XALPHA ≥ XALSTR and (0, DIST) is upwind of the line source, then the lower bound will be $+\infty$ (regardless of an anomolous R^+ result), while if XALPHA ≥ XALSTR and (0, DIST) is downwind of the line source, then R^+ will be an upper bound of the envelope.
A4A3510 to A4A3520	The upper and lower envelope bounds must be compared with the user specified bounds and the maximum envelope length ZUB(·) found earlier.
A4A3530	If the final effective roadway length is 4 meters or less, let INTGRL be zero.

A4A3540 to A4A3700	Compute the end points of the integration intervals. The first two intervals at the lower and of the effective roadway are made one meter each. The remaining ten intervals are split on either side of the point P = 0 (or the midpoint if P = 0 is not an element of the interval XLB to XUB), and increase in length on either side away from the point P = 0 in the ratio of 1:2:3:5:10 under the restriction that the first four on either side have maximum lengths of 10, 20, 30, and 50 meters. (The fifth one on either side is unconstrained.)
A4A3710 to A4A3870	Integration is performed numerically on each of the twelve integration intervals using the Cote's method of order six, C6. Since the last point of each integration interval is the first point of the next interval, the integration is initialized by evaluating the last point of the zeroth (nonexistent) interval so that a smooth iterative approach to C6 over each of twelve intervals may be accomplished.
A4A3890	Initialize loop for twelve integrals.
A4A3900 to A4A3940	Set the first point, R(1) = R(7), the previous last point, and R(7) = RR(M+1), the next end point. Also the integrand evaluated at R(1) is Y(1) = Y(7), the integrand evaluated at the previous last point.
A4A3950 to A4A4140	Evaluate R(·) and Y(·) over each of the twelve subintervals. This loop guards against exponential underflow and modifies the integrand in the CUT1 case.
A4A4150 to A4A4170	Evaluate INTGRL for the M^{th} subinterval using C6 and add it to the current total for intervals 1 through M-1.
A4A4180 to A4A4190	Store the integration results by class, receptor distance,and receptor height, i.e., (I,K,J) transformed to (L,J) by linearizing (I,K).
A4A4200 to A4A4230	Increment receptor distance and return to appropriate sub-loop to evaluate next integral.
A4A4240 to A4A4390	Multiply the elements of TEMP(·,·) by the appropriate emissions factors (all for the current lane group) and add the results to those for the previous lane groups in this site, while taking care not to perform unnecessary or redundant calculations.

A4A4400	Return to process the next lane group.
A4A4410 to A4A4430	If the total emissions factors for all lane groups for all years in the current site were zero, print a message stating such and branch to preparation for next site.
A4A4440 to A4A4530	Multiply the last two columns of SUPOUT(·,·) by the appropriate integration constants, conversion factors, and wind speed factors to fill out the entire SUPOUT(·,·) array.
A4A4540 to A4A4720	Arrange the receptor heights and receptor distances in their proper metric and American Engineering forms for use as headings on the analysis output page.
A4A4730 to A4A4750	Increment the page counter and output the analysis page heading information.
A4A4760 to A4A4910	Output the array SUPOUT(·,·) with appropriate left margin labels and vertical and horizontal separating lines while optimizing output by not printing uncalculated portions of SUPOUT(·,·).
A4A4920	If there are no more analyses to be performed, terminate processing.
A4A4930	If the site just output was the last in the current division, return to read the next Header Card.
A4A4940 to A4A5010	Re-initialize the site dependent variables and branch to read the first lane group Data Card in the next site.
A4A5020 to A4A5040	An end of file was encountered while reading lane group Data Cards. If the read was the first for the current site, all previously processed data were output when the last card (an ENS Card) was encountered; therefore, terminate processing. If the read was not the first for the current site, there are processed data awaiting output. Thus, assume the site is complete, set LAST = TRUE, and branch to printout the site results.
A4A5050 to A4A5080	A null site has been encountered. Increment the page counter, print a message, and return to read the first Data Card in the next site.

A4A5090 to A4A5130	If at least one site in the current division has been processed, return to read the next Header Card; otherwise, increment the page counter, print the null division message, and return to read the next Header Card.
A4A5140 to A4A5150	A normal end of file (one following an END Card) has been encountered, terminate processing.

SUBPROGRAM FIXIT

FIX0000 to FIX0040	Subroutine specification, parameter specifications, and local and common specifications.
FIX0050	Convert LOCATR to a form for indexing FMT(·).
FIX0060	Place VALUE in CHANGE (LOCATR).
FIX0070 to FIX0080	Place the tab and f formats corresponding to the invalid datum in the appropriate locations in FMT(·).
FIX0090	Increment the error pointer.
FIX0100	Place a question mark in CORECT (INDEX).
FIX0110 to FIX0120	Return to calling program.

SUBPROGRAM BLK DATA

BKD0000 to BKD0040	Subroutine specification, and common specifications.
BKD0050 to BKD0060	Data statement for F(·) the array of f format specifications for use in FIXIT.
BKD0070	Data statement to fill in the unchanging portion (the commas and parentheses) of FMT(·).
BKD0080	Data statement to place a question mark in QUEST.
BKD0090 to BKD0100	Data statement for TAB(·) the array of tab format specifications for use in FIXIT.
BKD0110	End of BLK DATA.

REFERENCES

1. Carpenter, W. A., "AIRPOL-4A, An Introduction and User's Guide",Virginia Highway & Transportation Research Council, VHTRC 76-R38, February 1976.

2. Carpenter, W. A., "The Theory and Mathematical Development of AIRPOL-4", Virginia Highway & Transportation Research Council, VHTRC 75-R49, May 1975.

3. Carpenter, W. A., "Introduction to AIRPOL-4, A User's Guide", Virginia Highway & Transportation Research Council, VHTRC 75-R56, May 1975.

4. Carpenter, W. A., "Analysis and Comparative Evaluation of AIRPOL-4," Virginia Highway & Transportation Research Council, VHTRC 75-R55, June 1975.

5. Carpenter, W. A., "Supportive Data and Methods for the Evaluation of AIRPOL-4", Virginia Highway & Transportation Research Council, VHTRC 75-R57, May 1975.

2796

APPENDIX

AIRPOL-4A PROGRAM LISTING

```
C**********************************************************************C
C**********************************************************************C
C                                                                      C
C                                                                      C
C                ****  ***  *****  *****   ****  *                     C
C               *    *  *   *   *  *    * *    * *                     C
C               *    *  *   *   *  *    * *    * *                     C
C               ******  *   *****  *****  *    * *                     C
C               *    *  *   *   *  *      *    * *                     C
C               *    * ***  *   *  *       ****  ******                C
C                                                                      C
C                                                                      C
C                      VERSION 4A -- NOVEMBER, 1975                    C
C                                                                      C
C**********************************************************************C
C**********************************************************************C
C**                                                                  **C
C**                                                                  **C
C**   AIRPOL-4A IS A GAUSSIAN LINE-SOURCE FORMULATION OF AN AIR-QUALITY **C
C**   PREDICTION MODEL FOR USE IN ESTIMATING THE IMPACT, IN TERMS OF CO **C
C**   CONCENTRATIONS, OF A HIGHWAY ON THE MICRO-REGION OF SUCH A FACILITY. **C
C**                                                                  **C
C**   AIRPOL-4A IS THE PROPERTY OF AND WAS DEVELOPED FOR THE VIRGINIA **C
C**   DEPARTMENT OF HIGHWAYS AND TRANSPORTATION BY:                  **C
C**                                                                  **C
C**                          WILLIAM A. CARPENTER                    **C
C**                           RESEARCH ENGINEER                      **C
C**                                                                  **C
C**   OF THE DATA SYSTEMS AND ANALYSIS SECTION OF THE VIRGINIA HIGHWAY AND **C
C**   TRANSPORTATION RESEARCH COUNCIL, P.O. BOX 3817, UNIVERSITY STATION, **C
C**   CHARLOTTESVILLE, VIRGINIA 22903.  TELEPHONE 804-977-0290.      **C
C**                                                                  **C
C**                                                                  **C
C**********************************************************************C
C**********************************************************************C
C                                                                      C
C                                                                      C
C                                                                      C
C                  **** DESCRIPTION OF PARAMETERS ****                 C
C                                                                      C
C                                                                      C
C                                                                      C
C                        *** LOGICAL SCALARS ***                       C
C                                                                      C
C                                                                      C
C   CUT        CALCULATED.                                             C
C              = .TRUE., IF ROADWAY IS IN A CUT.                       C
C                                                                      C
C   CUT1       CALCULATED.                                             C
C              = .TRUE., IF ROADWAY AND RECEPTORS ARE IN A CUT.        C
C                                                                      C
C   CUT2       CALCULATED.                                             C
C              = .TRUE., IF ROADWAY ONLY IS IN A CUT.                  C
C                                                                      C
C   DIDONE     CALCULATED.                                             C
C              = .TRUE., IF AT LEAST ONE SITE HAS BEEN PROCESSED FOR THE C
C              CURRENT DIVISION.                                       C
```

```
C                                                                        C
C                                                                        C
C    FILL       CALCULATED.                                              C
C               = .TRUE.. IF THE SOURCE IS AN ELEVATED ROADWAY.          C
C                                                                        C
C    FIRST      CALCULATED.                                              C
C               = .TRUE.. IF THE DATA CARD BEING PROCESSED WAS PRECEEDED BY C
C               EITHER AN 'ENS' OR A 'HEADER' CARD.                      C
C                                                                        C
C    LAST       CALCULATED.                                              C
C               = .TRUE.. IF AN END-OF-FILE IS ENCOUNTERED AND THE CURRENT C
C               SITE HAS NOT YET BEEN OUTPUT.                            C
C                                                                        C
C    METRIC     INPUT. FORMAT(L1), COLUMN 43, HEADER CARD.               C
C               = .TRUE.. IMPLIES METRIC INPUTS, AND                     C
C               = .FALSE.. IMPLIES AMERICAN ENGINEERING INPUTS.          C
C                                                                        C
C                                                                        C
C                                                                        C
C                                                                        C
C                          *** INTEGER SCALARS ***                       C
C                                                                        C
C                                                                        C
C    CASE       INPUT. FORMAT(A1), COLUMN 66, DATA CARD.                 C
C               = 'D'. IMPLIES DOWNWIND RECEPTORS, AND                   C
C               = 'U'. IMPLIES UPWIND RECEPTORS.                         C
C                                                                        C
C    CASESN     CALCULATED.                                              C
C               = +1, IF CASE = 'D', AND                                 C
C               = -1, IF CASE = 'U'.                                     C
C                                                                        C
C    CLS        CALCULATED.                                              C
C               NUMERIC INTERMEDIARY IN CHECK/CORRECT PROCESS FOR CLASS(I). C
C                                                                        C
C    HADD       CONSTANT.                                                C
C               = 'ADD'.                                                 C
C                                                                        C
C    HEND       CONSTANT.                                                C
C               = 'END'.                                                 C
C                                                                        C
C    HENS       CONSTANT.                                                C
C               = 'ENS'.                                                 C
C                                                                        C
C    I          CALCULATED.                                              C
C               AN INDEXING VARIABLE.                                    C
C                                                                        C
C    IC         CALCULATED.                                              C
C               AN INDEXING VARIABLE = ICLASS(I).                        C
C                                                                        C
C    IDTZ       CALCULATED.                                              C
C               AN INDEXING VARIABLE = INDEX(IFIX(DTZ/100)+1).           C
C                                                                        C
C    J          CALCULATED.                                              C
C               AN INDEXING VARIABLE.                                    C
C                                                                        C
C    JJ         CALCULATED.                                              C
C               AN INDEXING VARIABLE.                                    C
C                                                                        C
```

```
C                                                                          C
C   J1          CALCULATED.                                                C
C               = 1. IF CLASS(1) IS ANALYZED, AND                          C
C               = 25, OTHERWISE.                                           C
C                                                                          C
C   J48         CALCULATED.                                                C
C               = 48, IF CLASS(2) IS ANALYZED, AND                         C
C               = 24, OTHERWISE.                                           C
C                                                                          C
C   K           CALCULATED.                                                C
C               AN INDEXING VARIABLE.                                      C
C                                                                          C
C   K1          CALCULATED.                                                C
C               = 1, IF CLASS(1) IS ANALYZED, AND                          C
C               = 9, OTHERWISE.                                            C
C                                                                          C
C   K2          CALCULATED.                                                C
C               AN INDEXING VARIABLE.                                      C
C                                                                          C
C   K16         CALCULATED.                                                C
C               = 16, IF CLASS(2) IS ANALYZED, AND                         C
C               = 8, OTHERWISE.                                            C
C                                                                          C
C   L           CALCULATED.                                                C
C               AN INDEXING VARIABLE.                                      C
C                                                                          C
C   LG          CALCULATED.                                                C
C               THE COUNT OF THE NUMBER OF LANE GROUPS COMPRISING THE CURRENT C
C               SITE.                                                      C
C                                                                          C
C   LGID        INPUT, FORMAT(A1), COLUMN 9, DATA CARD.                    C
C               AN ALPHANUMERIC IDENTIFIER.                                C
C                                                                          C
C   LK          CALCULATED.                                                C
C               AN INDEXING VARIABLE.                                      C
C                                                                          C
C   LOCATR      CALCULATED.                                                C
C               AN INDEXING VARIABLE.                                      C
C                                                                          C
C   M           CALCULATED.                                                C
C               AN INDEXING VARIABLE.                                      C
C                                                                          C
C   PAGE        CALCULATED.                                                C
C               A PAGE INDEXER.                                            C
C                                                                          C
C   SITE        CALCULATED.                                                C
C               = SITEID FROM THE FIRST LANE GROUP IN THE CURRENT SITE.    C
C                                                                          C
C   SITEID      INPUT. FORMAT(A4). COLUMNS 5-7, DATA CARD.                 C
C               AN ALPHANUMERIC IDENTIFIER.                                C
C                                                                          C
C   SOBS        INPUT. FORMAT(I1). COLUMN 44, DATA CARD.                   C
C               = 1. IF SOURCE AND RECEPTORS ARE IN A CUT,                 C
C               = 2. IF SOURCE ONLY IS IN A CUT, AND                       C
C               = 0 OR BLANK, OTHERWISE.                                   C
C                                                                          C
C   TIME        INPUT. FORMAT(I3). COLUMNS 39-41. HEADER CARD.             C
C               THE DESIRED SAMPLING INTERVAL, IN MINUTES.                 C
```


2799

2800

```
C                                                                         C
C    TIME1, 2   CALCULATED.                                               C
C               THE SAMPLING TIME IN HOURS AND MINUTES, RESPECTIVELY.     C
C                                                                         C
C    WS         CALCULATED.                                               C
C               THE NUMBER OF WIND SPEEDS TO BE ANALYZED.                 C
C                                                                         C
C    W12        CALCULATED.                                               C
C               = 2*WS.                                                   C
C                                                                         C
C    XD         CONSTANT.                                                 C
C               = 'D'.                                                    C
C                                                                         C
C    XU         CONSTANT.                                                 C
C               = 'U'.                                                    C
C                                                                         C
C                                                                         C
C                                                                         C
C                         *** REAL SCALARS ***                            C
C                                                                         C
C                                                                         C
C    AHIC       CALCULATED.                                               C
C               = AH(IC).                                                 C
C                                                                         C
C    ALPHA      INPUT, FORMAT(F2.0), COLUMNS 68-69, DATA CARD.            C
C               THE ACUTE ROAD/WIND ANGLE, IN DEGREES.                    C
C                                                                         C
C    BHIC       CALCULATED.                                               C
C               = BH(IC).                                                 C
C                                                                         C
C    BLANK      CONSTANT.                                                 C
C               = ' '.                                                    C
C                                                                         C
C    CA         CALCULATED.                                               C
C               = COSINE(ALPHA).                                          C
C                                                                         C
C    CHT        CALCULATED.                                               C
C               = ABSOLUTE VALUE(CUT DEPTH), IN METERS.                   C
C                                                                         C
C    CLENGH     INPUT, FORMAT(F3.2), COLUMNS 62-64, DATA CARD.            C
C               THE UPWIND CUT LENGTH, WHEN SOHS = 1, IN FEET OR METERS.  C
C                                                                         C
C    CW         CALCULATED.                                               C
C               = CWIDTH/(2*SQRT(2*PI)), IN METERS.                       C
C                                                                         C
C    CWIDTH     INPUT, FORMAT(F3.0), COLUMNS 58-60, DATA CARD.            C
C               THE CUT WIDTH, WHEN SOHS = 1 .OR. 2, IN FEET OR METERS.   C
C                                                                         C
C    C1         CALCULATED.                                               C
C               = DRC*CA, IN METERS.                                      C
C                                                                         C
C    C2         CALCULATED.                                               C
C               = DRC*SA, IN METERS.                                      C
C                                                                         C
C    DIN        INPUT, FORMAT(F3.0), COLUMNS 74-76, DATA CARD.            C
C               THE PERPENDICULAR DISTANCE FROM THE SOURCE TO THE FIRST   C
C               RECEPTOR, IN FEET OR METERS.                              C
C                                                                         C
```

```
C                                                                          C
C                                                                          C
C   DIST      CALCULATED.                                                  C
C             THE DISTANCE FROM A RECEPTOR TO A POINT, R(I), ON THE SOURCE, C
C             IN METERS.                                                   C
C                                                                          C
C   DK        CALCULATED.                                                  C
C             THE PERPENDICULAR DISTANCE FROM THE SOURCE TO A RECEPTOR, IN C
C             METERS.                                                      C
C                                                                          C
C   DKC       CALCULATED.                                                  C
C             = DK*CASESN, IN METERS.                                      C
C                                                                          C
C   DLENGH    INPUT, FORMAT(F3.2), COLUMNS 54-56, DATA CARD.               C
C             THE DOWNWIND SOURCE LENGTH, IN KILOFEET OR KILOMETERS.       C
C                                                                          C
C                                                                          C
C   DTH       CALCULATED.                                                  C
C             = DIST+DTHOFF, IN METERS.                                    C
C                                                                          C
C   DTHOFF    CALCULATED.                                                  C
C             = HOFSET(IC) (MODIFIED, IF FILL = .TRUE.), IN METERS.        C
C                                                                          C
C   DTZ       CALCULATED.                                                  C
C             = DIST+DTZOFF, IN METERS.                                    C
C                                                                          C
C   DTZOFF    CALCULATED.                                                  C
C             = ZOFSET(IC) (MODIFIED, IF CUTZ = .TRUE.), IN METERS.        C
C                                                                          C
C   H         CALCULATED.                                                  C
C             THE TOTAL ELEVATION OF THE EFFECTIVE SOURCE ABOVE THE        C
C             SURROUNDING TERRAIN, IN METERS.                              C
C                                                                          C
C   HEIGHT    INPUT, FORMAT(F3.0), COLUMNS 46-48, DATA CARD.               C
C             THE HEIGHT OF FILL OR DEPTH OF CUT, IN FEET OR METERS.       C
C                                                                          C
C   HT2       INPUT, FORMAT(F2.0), COLUMNS 71-72, DATA CARD.               C
C             RECEPTOR ELEVATION(2) RELATIVE TO THE SURROUNDING TERRAIN,   C
C             IN METERS.                                                   C
C                                                                          C
C   INC       INPUT, FORMAT(F3.0), COLUMNS 78-80, DATA CARD.               C
C             THE INCREMENTAL DISTANCE BETWEEN RECEPTORS, IN FEET OR METERS. C
C                                                                          C
C   INTGRL    CALCULATED.                                                  C
C             THE VALUE OF THE GAUSSIAN INTEGRAL, IN 1/METERS.             C
C                                                                          C
C   JOB       INPUT, FORMAT(A4), COLUMNS 1-4, ALL CARDS.                   C
C             AN ALPHANUMERIC IDENTIFIER.                                  C
C                                                                          C
C   MWSI      CALCULATED.                                                  C
C             = MWSIN(1).                                                  C
C                                                                          C
C   ONEX      CONSTANT.                                                    C
C             = 'T1'.                                                      C
C                                                                          C
C   RTIME     CONSTANT.                                                    C
C             = 15.5 MINUTES.                                              C
C                                                                          C
```


2801

2802

```
C
C
C   QUEST      CONSTANT.
C              = '?'.
C
C   RATIO      CALCULATED.
C              = FLOAT(TIME)/RTIME, DIMENSIONLESS.
C
C   R1         CALCULATED.
C              = R(1).
C
C   R7         CALCULATED.
C              = R(7).
C
C   SA         CALCULATED.
C              = SINE(ALPHA).
C
C   SHM        CALCULATED.
C              SIGMA IN THE HORIZONTAL, IN METERS.
C
C   STEP       CALCULATED.
C              AN INCREMENTING VARIABLE.
C
C   SZM        CALCULATED.
C              SIGMA IN THE VERTICAL, IN METERS.
C
C   TFACTR     CALCULATED.
C              THE SAMPLING TIME CONVERSION FACTOR, DIMENSIONLESS.
C
C   ULENGH     INPUT, FORMAT(F3.2), COLUMNS 50-52, DATA CARD.
C              THE UPWIND SOURCE LENGTH, IN KILOFEET OR KILOMETERS.
C
C   WFACTR     CALCULATED.
C              THE WIND SPEED MULTIPLYING FACTOR, IN METERS/SECOND.
C
C   X          CALCULATED.
C              A UTILITY VARIABLE.
C
C   XALPHA     CALCULATED.
C              = ALPHA IN RADIAN MEASURE.
C
C   XALSTR     CALCULATED.
C              THE CRITICAL ANGLE, IN RADIANS, AT WHICH ONE EDGE OF THE
C              MAXIMUM-ACCEPTANCE ENVELOPE, I.E., THAT TRIANGULAR REGION
C              OUTSIDE OF WHICH ANY LINE SOURCE UP TO 4 KILOMETERS IN LENGTH
C              WILL CONTRIBUTE LESS THAN 0.02 PPM CO AT THE RECEPTOR WITH A
C              SAFETY FACTOR OF TWO ORDERS OF MAGNITUDE, BECOMES PARALLEL TO
C              THE SUBJECT LANE GROUP.
C
C   XLR        CALCULATED.
C              THE LOWER BOUND OF INTEGRATION, IN METERS.
C
C   XLR2       CALCULATED.
C              = XLR*2.
C
C   XN         CALCULATED.
C              = SLOPE OF BOUNDARY LINE OF MAXIMUM-ACCEPTANCE ENVELOPE
C              RELATIVE TO A LINE PERPENDICULAR TO THE WIND VECTOR.
```

```
C                                                                          C
C                                                                          C
C    XOB        CALCULATED.                                                C
C               THE UPPER BOUND OF INTEGRATION, IN METERS.                 C
C                                                                          C
C    X1, X2     CALCULATED.                                                C
C               THE EXPONENTS IN THE GAUSSIAN FORMULATION, DIMENSIONLESS.  C
C                                                                          C
C    XX1, XX2   CALCULATED.                                                C
C               = EXP(X1), EXP(X2), RESPECTIVELY.                          C
C                                                                          C
C    ZDIST      CALCULATED.                                                C
C               AN INTERMEDIARY USED TO SOLVE FOR ZOB(IC), IN METERS.      C
C                                                                          C
C    ZOBIC      CALCULATED.                                                C
C               = ZOB(IC).                                                 C
C                                                                          C
C    ZZ1, ZZ2   CALCULATED.                                                C
C               THE (REAL, VIRTUAL) SOURCE/RECEPTOR VERTICAL OFFSETS,      C
C               RESPECTIVELY, IN METERS.                                   C
C                                                                          C
C    3.846244E-5                                                           C
C               = (1/(2*PI))*(1 HOUR/3600 SECONDS)*(1 KILOMETER/1000 METERS)*  C
C               (870 PPM CO/(1 GRAM CO/(1 METER)**3)).                     C
C                                                                          C
C    1.745329E-2                                                           C
C               = NUMBER OF RADIANS/1 DEGREE.                              C
C                                                                          C
C    0.166667                                                              C
C               = APPROXIMATELY 1/6.                                       C
C                                                                          C
C    0.447040                                                              C
C               = NUMBER OF (METERS/SECOND)/(1 MILE/HOUR).                 C
C                                                                          C
C    1.609344                                                              C
C               = NUMBER OF KILOMETERS/1 MILE.                             C
C                                                                          C
C    3.280840                                                              C
C               = NUMBER OF FEET/1 METER.                                  C
C                                                                          C
C    5.013257                                                              C
C               = 2*SQRT(2*PI).                                            C
C                                                                          C
C    28.781749                                                             C
C               = A CONSTANT SUCH THAT THE EXPECTED ERROR OF INTEGRATION   C
C               ASSOCIATED WITH ASSUMING EXP(-0.5*(X>=28.781749)) = 0 IS LESS  C
C               THAN 0.007(METERS)**-1, FOR A SOURCE LENGTH OF 1000 METERS.    C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
```

2804

```
C
C
C                              *** LOGICAL ARRAYS ***
C
C
C   DOYR(3)    CALCULATED.
C              DOYR(I) = .TRUE.. IF YEAR(I) IS TO BE ANALYZED FOR THE
C              CURRENT LANE GROUP.
C
C   DOYRAL(3)  CALCULATED.
C              DOYRAL(I) = .TRUE., IF YEAR(I) WAS ANALYZED FOR AT LEAST ONE
C              LANE GROUP IN THE CURRENT SITE.
C
C
C
C
C                              *** INTEGER ARRAYS ***
C
C
C   A1(48)     CONSTANT.
C              CONTAINS ALPHANUMERIC OUTPUT STRING.
C
C   CLASS(2)   INPUT. FORMAT(2(Z1.X1)). COLUMNS 54-56. HEADER CARD.
C              THE DESIRED PASGUILL STABILITY CLASSES. IN EITHER ALPHA OR
C              NUMERIC FORM.
C
C   ICLASS(2)  CALCULATED.
C              ICLASS(I) = NUMERIC REPRESENTATION OF CLASS(I) AFTER THE
C              CHECK/CORRECT PROCESS.
C
C   INDEX(201) CONSTANT.
C              CONTAINS INDICES BASED ON PLUME TRAVEL DISTANCE FOR USE WITH
C              AZ(I.J) AND BZ(I.J). WHERE I = INDEX(.) AND J = ICLASS(.).
C
C   TITLE(9)   INPUT. FORMAT(8A4.A1). COLUMNS 5-37. HEADER CARD.
C              ALPHANUMERIC DIVISION IDENTIFICATION.
C
C   YEAR(3)    INPUT. FORMAT(3(I2.1X)). COLUMNS 45-52. HEADER CARD.
C              THE LAST TWO DIGITS OF THE CALENDAR YEARS FOR WHICH TRAFFIC
C              VOLUMES AND EMISSION FACTORS ARE SUPPLIED.
C
C
C
C
C                              *** REAL ARRAYS ***
C
C
C   AH(6)      CONSTANT.
C              CONTAINS SIGMA (HORIZONTAL) LOG-LOG INTERCEPTS. IN METERS.
C              INDEXED BY ICLASS(.).
C
C   ARRAY(19)  I/O DUMMY.
C              USED AS A CONTIGUOUS BLOCK OF MEMORY FOR DATA CARD I/O.
C
C   AZ(9.6)    CONSTANT.
C              CONTAINS SIGMA (VERTICAL) LOG-LOG INTERCEPTS, IN METERS.
C              INDEXED BY INDEX(.) AND ICLASS(.), RESPECTIVELY.
C
```

2805

```
C
C    BH(6)       CONSTANT.
C                CONTAINS SIGMA (HORIZONTAL) LOG-LOG SLOPES. DIMENSIONLESS.
C                INDEXED BY ICLASS(.).
C
C    BZ(9,6)     CONSTANT.
C                CONTAINS SIGMA (VERTICAL) LOG-LOG SLOPES, DIMENSIONLESS.
C                INDEXED BY INDEX(.) AND ICLASS(.), RESPECTIVELY.
C
C    CHANGE(17)CALCULATED.
C                CHANGE(I) CONTAINS THE CHANGE MADE TO ARRAY(I+2) UNDER THE
C                PROGRAM'S CHECK/CORRECT PROCESS.
C
C    CORECT(17)CALCULATED.
C                CORECT(I) CONTAINS A '2', IF ARRAY(I+2) WAS CHANGED UNDER THE
C                PROGRAM'S CHECK/CORRECT PROCESS.
C
C    D(8)        CALCULATED.
C                CONTAINS THE RECEPTOR DISTANCES RELATIVE TO THE FIRST LANE
C                GROUP IN THE CURRENT SITE.
C
C    DWS(6)      CONSTANT.
C                CONTAINS THE DEFAULT WIND SPEEDS, IN MILES/HOUR.
C
C    FF(3)       INPUT, FORMAT(3(5.2,1X)), COLUMNS 26-42, DATA CARD.
C                THE EMISSION FACTORS, IN GRAMS(CO)/VEHICLE/MILE OR
C                GRAMS(CO)/VEHICLE/KILOMETER, FOR THE THREE PREDICTION YEARS
C                FOR THE CURRENT LANE GROUP.
C
C    F(17)       CONSTANT.
C                F(I) CONTAINS THE OUTPUT FORMAT FOR ARRAY(I+2).
C
C    FMT(69)     CALCULATED.
C                CONTAINS THE RUN-TIME-GENERATED OUTPUT FORMAT FOR CHANGE(.).
C
C    HOFSET(6)   CONSTANT.
C                CONTAINS THE OFFSET DISTANCES FOR SIGMA (HORIZONTAL), IN
C                METERS.  INDEXED BY ICLASS(.).
C
C    MD(8)       CALCULATED.
C                CONTAINS THE RECEPTOR DISTANCES RELATIVE TO THE FIRST LANE
C                GROUP IN THE CURRENT SITE, IN METERS.
C
C    MDWS(6)     CONSTANT.
C                CONTAINS THE DEFAULT WIND SPEEDS, IN METERS/SECOND.
C
C    MWSIN(6)    CALCULATED.
C                CONTAINS THE WIND SPEEDS USED IN THE ANALYSIS, IN
C                METERS/SECOND.
C
C    MZ(2)       CALCULATED.
C                CONTAINS THE TWO RECEPTOR ELEVATIONS RELATIVE TO THE FIRST
C                LANE GROUP IN THE CURRENT SITE, IN METERS.
C
C    POWER(6)    CONSTANT.
C                CONTAINS THE EXPONENTS FOR TRANSLATING SHM (3 TO 10 MINUTES)
C                TO SHM (PREDICTION INTERVAL, MINUTES), DIMENSIONLESS.  INDEXED
C                BY ICLASS(.).
```

2806

```
C
C
C   R(7)        CALCULATED.
C               CONTAINS SEVEN EQUALLY SPACED ROADWAY DISTANCES, IN METERS.
C               FOR AN INTERVAL OF INTEGRATION.
C
C   RR(13)      CALCULATED.
C               CONTAINS THE END POINTS, ARRANGED IN ASCENDING ORDER, OF THE
C               TWELVE INTERVALS OF INTEGRATION, IN METERS.
C
C   SUPOUT(12,48)CALCULATED.
C               THE OUTPUT TABLE OF SUPERIMPOSED CO LEVELS, IN PPM.
C
C   TAB(17)     CONSTANT.
C               TAB(I) CONTAINS THE OUTPUT LOCATION FOR ARRAY(I+2).
C
C   TEMP(2,16)CALCULATED.
C               CONTAINS THE GAUSSIAN INTEGRALS, IN 1/METER, FOR THE CURRENT
C               LANE GROUP.  INDEXED BY RECEPTOR ELEVATION, AND RECEPTOR
C               DISTANCE-CROSS-CLASS, RESPECTIVELY.
C
C   TFVOL(3)    INPUT, FORMAT(3(F4.0,1X)), COLUMNS 11-24, DATA CARD.
C               THE TRAFFIC VOLUMES, IN VEHICLES/HOUR, FOR THE THREE
C               PREDICTION YEARS FOR THE CURRENT LANE GROUP.
C
C   TZ(2)       CALCULATED.
C               CONTAINS THE TWO RECEPTOR ELEVATIONS FOR THE CURRENT SITE
C               RELATIVE TO THE FIRST LANE GROUP IN THE SITE, IN FEET OR
C               METERS.
C
C   WIND(6)     INPUT, FORMAT(6(1X,F3.1)), COLUMNS 58-80, HEADER CARD.
C               THE DESIRED WIND SPEEDS, IN MILES/HOUR OR METERS/SECOND.
C
C   Y(7)        CALCULATED.
C               CONTAINS THE VALUES OF THE GAUSSIAN INTEGRAND AT THE SEVEN
C               R(I)'S FOR AN INTEGRATION INTERVAL, IN 1/METERS.
C
C   Z(2)        CALCULATED.
C               CONTAINS THE TWO RECEPTOR ELEVATIONS FOR THE CURRENT LANE
C               GROUP, IN FEET OR METERS.
C
C   ZOFSET(6)   CONSTANT.
C               CONTAINS THE OFFSET DISTANCES FOR SIGMA (VERTICAL), IN METERS.
C               INDEXED BY ICLASS(.).
C
C   ZOB(6)      CALCULATED.
C               CONTAINS THE MAXIMUM LENGTH OF THE MAXIMUM-ACCEPTANCE ENVELOPE
C               IN METERS.  INDEXED BY ICLASS(.).
C
C   ZZ(2)       CALCULATED.
C               CONTAINS THE TWO RECEPTOR ELEVATIONS RELATIVE TO THE FIRST
C               LANE GROUP IN THE CURRENT SITE, IN FEET.
C
C
C
C
C
C
```

```
      C                         **** FORMATS ****                                       C
      C                                                                                 C
      C                                                                                 C
      C                                                                                 C
      C                   *** FILE 5 = CARD READER ***                                  C
      C                                                                                 C
      C                                                                                 C
0001   5000 FORMAT(9A4,A1,1X,I3,1X,L1,1X,3(I2,1X),2(Z1,1X),5(F3.1,1X),F3.1)      A4A0000
0002   5010 FORMAT(4X,A3)                                                         A4A0010
0003   5011 FORMAT(4X,A3,1X,A1,1X,3(F4.0,1X),3(F5.2,1X),I1,1X,F3.0,1X,           A4A0020
          -2(F3.2,1X),F3.0,1X,F3.2,1X,A1,1X,2(F2.0,1X),F3.0,1X,F3.0)             A4A0030
      C                                                                                 C
      C                                                                                 C
      C                                                                                 C
      C                   *** FILE 6 = LINE PRINTER ***                                 C
      C                                                                                 C
      C                                                                                 C
0004   6000 FORMAT('1AIRPOL -- VERSION 4A',T50,9A4,A1,T120,'JOB NO. ',A4/,T5,   A4A0040
          -'NOVEMBER  1975',T99,'PAGE',I2,//T10,'QUESTION MARKS (?) INDICATE   A4A0050
          -A PROGRAMATIC CHANGE OF INPUTS UNDER'/T10,'THE INPUT CHECK/CORRECT A4A0060
          - PROCESS.  ITEMS BELOW QUESTION MARKS ARE'/T10,'VALUES ACTUALLY US A4A0070
          -ED BY THE PROGRAM IN PERFORMING THE ANALYSES.'///T9,'S    L   ',3('T A4A0080
          -FVOLM  '),3('  EF    '), 'L  SOURC    UW//DW     CUT   CUT1  C  A A4A0090
          -LFA',3('  OBS   ')/T13,2(4X, 'YR-1     YR-I     YR-3'),3X,'O  HIGHT   A4A0100
          -  LENGTH    WIDTH  LNTH  A',T108,'HT-2  DIST1   INC',T9,'1     G')   A4A0110
0005   6001 FORMAT(///T19,'WARNING -- NULL SITE.  NO DATA CARDS FOUND IN THIS  A4A0120
          -SITE.  CONTINUE PROCESSING WITH NEXT SITE.')                          A4A0130
0006   6002 FORMAT(///T13,'WARNING -- NULL DIVISION.  NO DATA CARDS FOUND IN T A4A0140
          -HIS DIVISION.  CONTINUE PROCESSING WITH NEXT DIVISION.')              A4A0150
0007   6010 FORMAT(T9,'T    I   ',3('  VPH    '),3('G/V/KM  '),'C      M      KM    A4A0160
          -KM      M      KM    S    DEG    M       M        M')                 A4A0170
0008   6020 FORMAT(T9,'T    I   ',3('  VPH    '),3('G/V/MI  '),'C     FT    KFT   A4A0180
          -KFT     FT    KFT    S    DEG    FT      FT       FT')                A4A0190
0009   6030 FORMAT(1H+,T6,'___  _',6('  ______'),'  _  _____',2('  ____'),'  _ A4A0200
          -____  ____  _',2('  ____'),2('  _____')/)                             A4A0210
0010   6040 FORMAT(T8,A3,2X,A1,3F8.1,3F8.2,I3,F7.1,2F6.2,F7.1,F6.2,2X,A1,       A4A0220
          -2F6.1,2F7.1)                                                          A4A0230
0011   6060 FORMAT(T19,6(A1,7X),T60,2(4X,A1),6X,A1,2(5X,A1),6X,A1,3X,A1,4X,A1, A4A0240
          -2(5X,A1),6X,A1)                                                       A4A0250
0012   6075 FORMAT(/)                                                            A4A0260
0013   6100 FORMAT('1AIRPOL -- VERSION 4A',T60,'SITE ID = ',A3,T120,'JOB NO. ' A4A0270
          -,A4/T5,'NOVEMBER  1975',T50,9A4,T99,'PAGE ',I2//' A SUPERPOSITION   A4A0280
          -OF CO LEVELS'/' FROM THE ABOVE',I2,' LANE GROUPS.',T67,'AVERAGE WI A4A0290
          -ND SPEEDS IN MPH (M/S)'/T30,6(11X,F5.2)/' CONCENTRATION AVERAGING   A4A0300
          -TIME',T31,6( 9X,'(',F5.2,')')/1H+,T36,6(14('_'),2X)/T7,I2,' HR. :   A4A0310
          -',I2,' MIN.',T31,6(8X,'OBSERVER')/  T36,6('HEIGHTS, FT(M)',2X)/     A4A0320
          -T23,'OBSERVER',T33,12(4X,F4.1)/T12,'YEAR   DISTANCE, FT(M)',12(2X, A4A0330
          -'(',F4.1,')')/1H+,T11,6('_'),2X,15('_'),12(2X,6('_'))//T35,96('_') A4A0340
          -/1H+,T75,'CO LEVELS IN PPM')                                          A4A0350
0014   6110 FORMAT(T18,'I',T19,F5.0,2X,'(',F5.1,')',2X,'I',12(1X,F5.2,2X))     A4A0360
0015   6120 FORMAT(1H+,A1,5X,'II')                                               A4A0370
0016   6125 FORMAT(1H+,3X,Z1)                                                    A4A0380
0017   6130 FORMAT(1H+,T12,'19',I2)                                              A4A0390
0018   6140 FORMAT(1H+,129('_'))                                                 A4A0400
0019   6150 FORMAT(1X)                                                           A4A0410
0020   6155 FORMAT(1H+,T6,123('_'))                                              A4A0420
0021   6160 FORMAT(1H1)                                                          A4A0430
0022   6170 FORMAT(T43,'ALL TEVOLMEF PRODUCTS = 0.  ANALYSES NOT PERFORMED.')   A4A0440
```


2807

2808

```
                   C                                                                            C
                   C                                                                            C
                   C                                                                            C
                   C****************************************************************************C
                   C****************************************************************************C
                   C                                                                            C
                   C                      **** START OF AIRPOL-4 CODE ****                      C
                   C                                                                            C
                   C                                                                            C
                   C                                                                            C
                   C                                                                            C
0023                     LOGICAL CUT,CUT1,CUT2,DIDONE,FILL,FIRST,LAST,METRIC             A4A0450
                   C                                                                            C
0024                     INTEGER CASE,CASESN,CLS,HADD,HEND,HENS,I,IC,IDT2,J,JJ,J1,J48,K,K1, A4A0460
                        -K2,K16,L,L6,LGID,LK,LOCATR,M,PAGE,SITE,SITEID,SOBS,TIME,TIME1,   A4A0470
                        -TIME2,W6,W12,X0,XU                                               A4A0480
                   C                                                                            C
0025                     REAL AHIC,ALPHA,BHIC,BLANK,CA,CHT,CLENGH,CW,CWIDTH,C1,C2,DIN,DIST, A4A0490
                        -DK,DKC,DLENGH,DTH,DTHOFF,DTZ,DTZOFF,H,HEIGHT,HT2,INC,INTGRL,JOB, A4A0500
                        -MWSI,ONEX,PTIME,QUEST,RATIO,R1,R7,SA,SHM,STEP,SZM,TFACTR,ULENGH, A4A0510
                        -WFACTR,X,XALPHA,XALSTR,XLB,XLH2,XN,XOB,X1,X2,XX1,XX2,ZDIST,ZOBIC, A4A0520
                        -ZZ1,ZZ2                                                          A4A0530
                   C                                                                            C
0026                     LOGICAL DOYR(3), DOYRAL(3)                                       A4A0540
                   C                                                                            C
0027                     INTEGER A1(48),CLASS(2),ICLASS(2),INDEX(201),TITLE(9),YEAR(3)    A4A0550
                   C                                                                            C
0028                     REAL AH(6),ARRAY(19),AZ(9,6),BH(6),BZ(9,6),CHANGE(17),CORECT(17), A4A0560
                        -D(8),DWS(6),EF(3),F(17),FMT(69),HOFSET(6),MD(6),MDWS(6),MWSIN(6), A4A0570
                        -MZ(2),POWER(6),R(7),RR(13),SUPOUT(12,48),TAB(17),TEMP(2,16),     A4A0580
                        -TFVOL(3),TZ(2),WIND(6),WSIN(6),Y(7),Z(2),ZOFSET(6),ZOB(6),ZZ(2)  A4A0590
                   C                                                                            C
0029                     EQUIVALENCE (ARRAY(1),SITEID), (ARRAY(2),LGID), (ARRAY(3),TFVOL), A4A0600
                        -(ARRAY(6),EF(1)), (ARRAY(9),SOBS), (ARRAY(10),HEIGHT),           A4A0610
                        -(ARRAY(11),ULENGH), (ARRAY(12),DLENGH), (ARRAY(13),CWIDTH),      A4A0620
                        -(ARRAY(14),CLENGH), (ARRAY(15),CASE), (ARRAY(16),ALPHA),         A4A0630
                        -(ARRAY(17),HT2), (ARRAY(18),DIN), (ARRAY(19),INC)                A4A0640
                   C                                                                            C
0030                     COMMON /MINF/ FMT,TAB,F,CHANGE,LOCATR,CORECT,QUEST               A4A0650
                   C                                                                            C
0031                     DATA AH /0.4998,0.3399,0.2001,0.1137,0.09524,0.06387/            A4A0660
                   C                                                                            C
0032                     DATA AZ /0.1181,0.04661,0.01164,6*0.0001818,3*0.1301,6*0.05566,  A4A0670
                        -9*0.1103,3*0.08671,2*0.1648,4*0.4533,2*0.07748,3*0.1375,2*0.3038, A4A0680
                        -2*0.8287,4*0.05943,0.1080,0.1859,2*0.5836,1.5252/                A4A0690
                   C                                                                            C
0033                     DATA A1 /4*' ', 'S', 'T', 'A', 'B', 'I', 'L', 'I', 'T', 'Y', ' ', A4A0700
                        -'C', 'L', 'A', 'S', 'S', 9*' ', 'S', 'T', 'A', 'B', 'I', 'L', 'I', A4A0710
                        -'T', 'Y', ' ', 'C', 'L', 'A', 'S', 'S', 5*' '/                   A4A0720
                   C                                                                            C
0034                     DATA BLANK /' '/                                                 A4A0730
                   C                                                                            C
0035                     DATA BH /0.8679,0.8760,0.8961,0.9251,0.9011,0.9000/              A4A0740
                   C                                                                            C
0036                     DATA BZ /1.0401,1.2155,1.4587,6*2.1280,3*0.9596,6*1.0962,9*0.9119, A4A0750
                        -3*0.8647,2*0.7614,4*0.6149,2*0.8299,3*0.7293,2*0.6146,2*0.4936,  A4A0760
                        -4*0.7976,0.7042,0.6256,2*0.4751,0.3708/                          A4A0770
                   C                                                                            C
0037                     DATA DWS /0.0,0.2,0.7,1.2,2.3,3.8/                               A4A0780
```

```
0038          DATA HADD,HEND,HENS /'ADD','END','ENS'/                           A440790
         C                                                                         C
0039          DATA HOFSET /7.8848,12.0129,20.5217,34.3906,45.9989,72.0415/      A440800
         C                                                                         C
0040          DATA INDEX /2*1,2,2*3,4,4*5,10*6,20*7,60*8,101*9/                 A440810
         C                                                                         C
0041          DATA LAST /.FALSE./                                               A440820
         C                                                                         C
0042          DATA HOWS /0.0,0.1,0.3,0.5,1.0,1.7/                               A440830
         C                                                                         C
0043          DATA ONEX /'T1'/                                                  A440840
         C                                                                         C
0044          DATA POWER /0.65,0.56,0.46,0.35,0.23,0.10/                        A440850
         C                                                                         C
0045          DATA PTIME /15.5/                                                 A440860
         C                                                                         C
0046          DATA XD,XU /'D','U'/                                              A440870
         C                                                                         C
0047          DATA ZOFSET /11.5155,12.7736,17.4995,27.0231,35.5362,57.2639/     A440880
         C                                                                         C
0048          WRITE(6, 6160)                                                    A440890
0049          CALL REREAD                                                       A440900
         C                                                                         C
         C          ACTIVATION OF REREAD BY A SINGLE CALL CAUSES ALL SUBSEQUENT    C
         C     READS (FROM FILES OTHER THAN 99) TO FILL THE FILE 99 BUFFER IN      C
         C     ADDITION TO THEIR OWN BUFFERS.  THUS, FROM HERE ON, IN THIS PROGRAM, C
         C     A READ FROM FILE 99 WILL EFFECTIVELY CONSTITUTE A REREAD OF THE     C
         C     PREVIOUS RECORD FROM FILE 5 (5 IS THE ONLY INPUT FILE IN AIRPOL-4).  C
         C     THE SUBPROGRAM REREAD IS AVAILABLE FROM THE SHARE FORTRAN LIBRARIAN, C
         C     MS. SANDRA WARD, UNIVERSITY OF WATERLOO, WATERLOO, ONTARIO, CANADA.  C
         C                                                                         C
         C                                                                         C
         C     READ A HEADER CARD AND CHECK FOR END OF FILE.                       C
         C                                                                         C
0050        1 READ(5, 5000, END = 999) JOB,TITLE,TIME,METRIC,YEAR,CLASS,WIND    A440910
         C                                                                         C
         C     INITIALIZE CONTROL VARIABLES AND CLEAR COMPUTATIONAL PORTION OF     C
         C     OUTPUT ARRAY.                                                       C
         C                                                                         C
0051          FIRST = .TRUE.                                                    A440920
0052          DIDONE = .FALSE.                                                  A440930
0053          PAGE = 0                                                          A440940
0054          LG = 0                                                            A440950
0055          J1 = 1                                                            A440960
0056          J48 = 48                                                          A440970
0057          TZ(1) = 1.5                                                       A440980
0058          IF(.NOT. METRIC) TZ(1) = 5.0                                      A440990
0059          DO 2 J = 11, 12                                                   A441000
0060          DO 2 I = 1, 48                                                    A441010
0061          SUPOUT(J, I) = 0.0                                                A441020
0062        2 CONTINUE                                                          A441030
         C                                                                         C
         C     DO VALIDITY CHECK/CORRECT ON HEADER CARD INPUTS.                    C
         C                                                                         C
0063          IF(TIME .LE. 0) TIME = 60                                         A441040
0064          TIME1 = TIME/60                                                   A441050
0065          TIME2 = TIME-TIME1*60                                             A441060
0066          RATIO = FLOAT(TIME)/PTIME                                         A441070
```

```
C
      DO 20 J = 1, 2
      CLS = CLASS(J)
      CLS = CLS-(CLS/9)*9
      CLS = CLS-(CLS/7)*CLS
      ICLASS(J) = CLS
      CLASS(J) = CLS+9
   20 CONTINUE
      IF(ICLASS(1) .NE. ICLASS(2)) GO TO 21
      ICLASS(2) = 0
      IF(ICLASS(1) .NE. 0) GO TO 22
      CLASS(1) = 11
      ICLASS(1) = 2
C
   21 IF(ICLASS(1) .EQ. 0) J1 = 25
   22 IF(ICLASS(2) .EQ. 0) J46 = 24
C
      DO 29 I = 1, 2
      IC = ICLASS(I)
      IF(IC .EQ. 0) GO TO 29
      X = AH(IC)*RATIO**POWER(IC)
      ZDIST = 1.0E4
      ZUR(IC) = 2.0E4
      BHIC = BH(IC)
      DO 27 J = 1, 5
      K = INDEX(IFIX(ZDIST/100)+1)
      DIST = (1.0E6/(X*A/(K,IC)))**(1.0/(BHIC+B/(K,IC)))
      IF(DIST .GT. 2.0E4) GO TO 29
      IF(ABS(DIST-ZDIST) .LE. 10.0) GO TO 28
      ZDIST = DIST
   27 CONTINUE
   28 ZUR(IC) = DIST
   29 CONTINUE
C
      W6 = 0
      DO 30 J = 1, 6
      WSIN(J) = 1.0E70
      IF(WIND(J) .LE. 0.0) GO TO 30
      W6 = W6+1
      WSIN(W6) = WIND(J)
   30 CONTINUE
      IF(W6 .NE. 0) GO TO 32
      W6 = 6
      DO 31 J = 1, 6
      WSIN(J) = HDWS(J)
      IF(.NOT. METRIC) WSIN(J) = DWS(J)
   31 CONTINUE
   32 W12 = W6*2
      IF(.NOT. METRIC) GO TO 35
      DO 34 J = 1, 6
      X = WSIN(J)
      MWSIN(J) = X
      WSIN(J) = X/0.447040
   34 CONTINUE
      GO TO 37
   35 DO 36 J = 1, 6
      MWSIN(J) = WSIN(J)*0.447040
   36 CONTINUE
```

```
C
C     READ A DATA CARD AND CHECK FOR END OF FILE
C
   37 READ(5, 5010, END = 996) SITEID
C
C     DETERMINE IF THIS DATA CARD WAS AN ENS OR END CONTROL CARD.
C
      IF(FIRST .AND. (SITEID .EQ. HADD .OR. SITEID .EQ. HENS)) GO TO 997
      IF(SITEID .EQ. HADD .OR. SITEID .EQ. HENS) GO TO 270
      IF(FIRST .AND. SITEID .EQ. HEND) GO TO 998
      IF(SITEID .EQ. HEND) GO TO 270
C
C     THE CARD IS A LANE-GROUP DATA CARD.  REREAD TO OBTAIN THE DATA.
C
      READ(99, 5011) ARRAY
C
C     INITIALIZE LANE-GROUP CONTROL VARIABLES.
C
      FILL = .FALSE.
      CUT = .FALSE.
      LG = LG+1
      Z(1) = TZ(1)
      DO 40 J = 1, 17
      CORECT(J) = BLANK
   40 CONTINUE
      DO 41 J = 1, 34
      FMT(2*J) = ONEX
   41 CONTINUE
      IF(.NOT. FIRST) GO TO 44
C
C     THIS IS THE FIRST CARD IN THE SITE.  INITIALIZE SITE VARIABLES,
C     AND PRINT HEADINGS.
C
      DO 43 J = 1, 3
      DOYRAL(J) = .FALSE.
   43 CONTINUE
      PAGE = PAGE+1
      WRITE(6, 6000) TITLE,JOB,PAGE
      IF(METRIC) WRITE(6, 6010)
      IF(.NOT. METRIC) WRITE(6, 6020)
      WRITE(6, 6030)
C
C     ECHO THE INPUT.
C
   44 WRITE(6, 6040) ARRAY
C
C     DO VALIDITY CHECK/CORRECT ON DATA CARD INPUTS.
C
      LOCATR = 1
      DO 70 I = 1, 3
      DOYR(I) = .TRUE.
      IF(TFVOL(I) .GE. 0.0) GO TO 50
      TFVOL(I) = 0.0
      CALL FIXIT(0.0, I)
   50 IF(EF(I) .GE. 0.0) GO TO 60
      EF(I) = 0.0
      CALL FIXIT(0.0, I+3)
   60 DOYR(I) = .NOT.((TFVOL(I)*EF(I)) .EQ. 0.0)
```

2812

```
0157          DOYRAL(I) = DOYRAL(I) .OR. DOYR(I)                          A4-1980
0158       70 CONTINUE                                                    A4-1990
        C                                                                       C
0159          IF(SOBS .GE. 0 .AND. SOBS .LE. 2) GO TO 90                  A4-2000
0160          SOBS = 0                                                    A4-2010
0161          CALL FIXIT(SOBS, 7)                                         A4-2020
0162       90 CUT = SOBS .NE. 0                                           A4-2030
        C                                                                       C
0163          H = 1.5                                                     A4-2040
0164          IF(.NOT. METRIC) H = 5.0                                    A4-2050
0165          IF(CUT) GO TO 92                                            A4-2060
0166          IF(HEIGHT .GT. 0.0) FILL = .TRUE.                           A4-2070
0167          IF(HEIGHT .GE. 0.0) GO TO 91                                A4-2080
        C                                                                       C
        C     TO GET HERE, THE PROGRAM FOUND A NEGATIVE SOURCE-HEIGHT FOR A     C
        C     NON-CUT SECTION OF HIGHWAY.                                       C
        C                                                                       C
0168          HEIGHT = ABS(HEIGHT)                                        A4-2090
0169          FILL = .TRUE.                                               A4-2095
0170          CALL FIXIT(HEIGHT, 8)                                       A4-2100
0171       91 H = H+HEIGHT                                                A4-2110
0172          GO TO 100                                                   A4-2120
        C                                                                       C
0173       92 IF(HEIGHT .LT. 0.0) GO TO 100                               A4-2130
0174          IF(HEIGHT .NE. 0.0) GO TO 93                                A4-2140
        C                                                                       C
        C     TO GET HERE, THE PROGRAM FOUND A CUT OF ZERO DEPTH.               C
        C                                                                       C
0175          CUT = .FALSE.                                               A4-2150
0176          SOBS = 0                                                    A4-2160
0177          CALL FIXIT(SOBS, 7)                                         A4-2170
0178          GO TO 100                                                   A4-2180
        C                                                                       C
        C     TO GET HERE, THE PROGRAM FOUND A POSITIVE SOURCE-HEIGHT FOR A CUT. C
        C                                                                       C
0179       93 HEIGHT = -HEIGHT                                            A4-2190
0180          CALL FIXIT(HEIGHT, 8)                                       A4-2200
        C                                                                       C
0181      100 ULENGH = ULENGH*1000.0                                      A4-2210
0182          DLENGH = DLENGH*1000.0                                      A4-2220
        C                                                                       C
0183      110 IF(CUT) GO TO 111                                           A4-2230
0184          IF(CWIDTH .EQ. 0.0) GO TO 120                               A4-2240
        C                                                                       C
        C     TO GET HERE, THE PROGRAM FOUND A NON-ZERO CUT WIDTH FOR A NON-CUT C
        C     SECTION OF HIGHWAY.                                               C
        C                                                                       C
0185          CWIDTH = 0.0                                                A4-2250
0186          CALL FIXIT(0.0, 11)                                         A4-2260
0187          GO TO 120                                                   A4-2270
        C                                                                       C
0188      111 IF(CWIDTH .GT. 0.0) GO TO 120                               A4-2280
        C                                                                       C
        C     TO GET HERE, THE PROGRAM FOUND A NON-POSITIVE CUT WIDTH FOR A CUT C
        C     SECTION OF HIGHWAY.                                               C
        C                                                                       C
0189          CWIDTH = -CWIDTH                                            A4-2290
0190          IF(CWIDTH .NE. 0.0) GO TO 112                               A4-2300
```

```
      CWIDTH = 100.0
      IF(.NOT. METRIC) CWIDTH = 328.
  112 CALL FIXIT(CWIDTH, 11)
C
  120 IF(SOBS .EQ. 1) GO TO 121
      IF(CLENGH .EQ. 0.0) GO TO 130
C
C     TO GET HERE, THE PROGRAM FOUND A NON-ZERO CUT LENGTH FOR A NON-CUT1
C     SECTION OF HIGHWAY.
C
      CLENGH = 0.0
      CALL FIXIT(0.0, 12)
      GO TO 130
C
  121 IF(CLENGH .GT. 0.0) GO TO 130
C
C     TO GET HERE. THE PROGRAM FOUND A NON-POSITIVE CUT LENGTH FOR A CUT1
C     SECTION OF HIGHWAY.
C
      CLENGH = -CLENGH
      IF(CLENGH .EQ. 0.0) CLENGH = ULENGH
      CALL FIXIT(CLENGH, 12)
C
  130 CUT1 = SOBS .EQ. 1
      CUT2 = SOBS .EQ. 2
      CHT = ABS(HEIGHT)
      IF(CUT) H = 0.0
C
  140 IF(CASE .EQ. XD) CASE = 1
      IF(CASE .EQ. XU) CASE = 2
      IF(CASE .EQ. 1 .OR. CASE .EQ. 2) GO TO 150
      CASE = 1
      CALL FIXIT(XD, 13)
C
  150 IF(.NOT. CUT1) GO TO 151
C
C     FOR A CUT1 SECTION OF HIGHWAY, FORCE ALPHA = 0.
C
      IF(ALPHA .EQ. 0.0) GO TO 160
      ALPHA = 0.0
      CALL FIXIT(ALPHA, 14)
      GO TO 160
C
  151 IF(ALPHA .GE. 0.0 .AND. ALPHA .LE. 90.0) GO TO 160
      ALPHA = 90.0
      CALL FIXIT(ALPHA, 14)
C
  160 IF(.NOT. FIRST) GO TO 162
C
C     THIS IS THE FIRST DATA CARD IN THE SITE.  SET UP HT1 AND HT2.
C
      IF(HT2 .NE. TZ(1) .AND. HT2 .GE. 0.0) GO TO 161
      HT2 = 3.0
      IF(.NOT. METRIC) HT2 = 10.0
      CALL FIXIT(HT2, 15)
  161 Z(2) = HT2
      TZ(2) = HT2
      GO TO 170
```

2814

```
C                                                                        C
C     THIS IS NOT THE FIRST DATA CARD IN THE SITE.  CHECK HT2.           C
C                                                                        C
  162 IF(HT2 .EQ. IZ(2)) GO TO 170                                A4-2680
      HT2 = IZ(2)                                                 A4-2690
      Z(2) = HT2                                                  A4-2700
      CALL FIXIT(HT2, 15)                                         A4-2710
C                                                                        C
  170 IF(DIN .GE. 0.0) GO TO 180                                  A4-2720
      DIN = 0.0                                                   A4-2730
      CALL FIXIT(DIN, 16)                                         A4-2740
C                                                                        C
  180 IF((7.0*INC+DIN) .GE. 0.0) GO TO 190                        A4-2750
      INC = 0.0                                                   A4-2760
      CALL FIXIT(INC, 17)                                         A4-2770
C                                                                        C
  190 IF(.NOT. FIRST) GO TO 200                                   A4-2780
C                                                                        C
C     THIS IS THE FIRST DATA CARD IN THE SITE.  SET UP DISTANCE          C
C     REFERENCES RELATIVE TO THIS LANE GROUP AND DETERMINE THE SITE.     C
C                                                                        C
      DO 191 I = 1, 8                                             A4-2790
      D(I) = DIN+(I-1)*INC                                        A4-2800
  191 CONTINUE                                                    A4-2810
      SITE = SITEID                                               A4-2820
C                                                                        C
  200 IF(METRIC) GO TO 220                                        A4-2830
C                                                                        C
C     IF NOT METRIC INPUTS, CONVERT VARIABLES TO METRIC.                 C
C                                                                        C
      DO 211 I = 1, 3                                             A4-2840
      EF(I) = EF(I)/1.609344                                      A4-2850
  211 CONTINUE                                                    A4-2860
      H = H/3.280840                                              A4-2870
      HEIGHT = HEIGHT/3.280840                                    A4-2880
      ULENGH = ULENGH/3.280840                                    A4-2890
      DLENGH = DLENGH/3.280840                                    A4-2900
      CWIDTH = CWIDTH/3.280840                                    A4-2910
      CLENGH = CLENGH/3.280840                                    A4-2920
      CHT = CHT/3.280840                                          A4-2930
      Z(1) = Z(1)/3.280840                                        A4-2940
      Z(2) = Z(2)/3.280840                                        A4-2950
      DIN = DIN/3.280840                                          A4-2960
      INC = INC/3.280840                                          A4-2970
C                                                                        C
C     OUTPUT CHECK/CORRECT RESULTS.
C                                                                        C
  220 WRITE(6, 6060) CORECT                                       A4-2980
      LOCATR = LOCATR-1                                           A4-2990
      IF(LOCATR .GE. 1) WRITE(6, FMT) (CHANGE(I), I = 1, LOCATR)  A4-3000
      WRITE(6, 6075)                                              A4-3010
      FIRST = .FALSE.                                             A4-3020
C                                                                        C
C     IF NO ANALYSES ARE TO BE DONE, GET NEXT DATA CARD.                 C
C                                                                        C
      IF(.NOT. (DOYR(1) .OR. DOYR(2) .OR. DOYR(3))) GO TO 37      A4-3030
C                                                                        C
C                                                                        C
```

```
C                                                                           C
C     BEGIN THE ANALYSES                                                    C
C                                                                           C
      CW = CWIDTH/5.013257                                           A4A3040
      CASESN = -((-1)**CASE)                                         A4A3050
      XALPHA = 1.745329E-2*ALPHA                                     A4A3060
      CA = COS(XALPHA)                                               A4A3070
      SA = SIN(XALPHA)                                               A4A3080
      IF(.NOT. CUT1) GO TO 230                                       A4A3090
C                                                                           C
C     DETERMINE LENGTH OF CONTRIBUTING HIGHWAY FOR A CUT1 HIGHWAY.          C
C                                                                           C
      X = CLENGH*((CWIDTH/0.06387)**1.1111-72.0415)                  A4A3100
      OLENGH = AMIN1(OLENGH,X)                                       A4A3110
C                                                                           C
  230 DO 252 I = 1,2                                                 A4A3120
      IC = ICLASS(I)                                                 A4A3130
      IF(IC .EQ. 0) GO TO 252                                        A4A3140
      AHIC = AH(IC)                                                  A4A3150
      BHIC = BH(IC)                                                  A4A3160
      ZOHIC = ZOH(IC)                                                A4A3170
      DTZOFF = ZOFSET(IC)                                            A4A3180
      DTHOFF = HOFSET(IC)                                            A4A3190
C                                                                           C
C     FIND NEW OFFSET VALUES FOR FILL AND CUT2 SECTIONS OF HIGHWAY.          C
C                                                                           C
      IF(FILL) DTZOFF = ((1.5 +0.25*HEIGHT)/AZ(1,IC))**(1.0/BZ(1,IC))  A4A3200
      IF(CUT2) DTHOFF = ((3.0+AMIN1(CWIDTH/6.0,0.500000*CHT))/AHIC)**  A4A3210
     -(1.0/BHIC)                                                     A4A3220
C                                                                           C
      TFACTR = RATIO**POWER(IC)                                      A4A3230
      XN = 6.0*TFACTR*AHIC                                           A4A3240
      XALSTR = ATAN(XN)                                              A4A3250
C                                                                           C
      DO 251 J = 1,2                                                 A4A3260
      ZZ1 = (Z(J)-H)                                                 A4A3270
      ZZ2 = (Z(J)+H)                                                 A4A3280
      DK = DIN+0.5                                                   A4A3290
C                                                                           C
      DO 250 K = 1,8                                                 A4A3300
C                                                                           C
C     DO NOT PROCESS ALL EIGHT RECEPTORS IF THEY ARE IN THE SAME LOCATION.  C
C                                                                           C
      IF(K .GT. 1 .AND. INC .EQ. 0.0) GO TO 242                      A4A3310
C                                                                           C
      INTGRL = 0.0                                                   A4A3320
      XUP = 1.0E70                                                   A4A3330
      XLP = -1.0E70                                                  A4A3340
      X = -1.0E70                                                    A4A3350
      DKC = DK*CASESN                                                A4A3360
C                                                                           C
C     AVOID MULTIPLICATION BY INFINITY.                                     C
C                                                                           C
      IF(ALPHA .NE. 90.0) X =-DKC*TAN(XALPHA)                        A4A3370
C                                                                           C
      C1 = DKC*CA                                                    A4A3380
      C2 = DKC*SA                                                    A4A3390
C                                                                           C
```


2815

```
C
C     DETERMINE UPPER AND LOWER BOUNDS OF INTEGRATION BASED ON INPUT LIMIT
C     AND MAXIMUM ACCEPTANCE ENVELOPE.
C
      IF(CASE .EQ. 2) GO TO 231
      XLB = -(XN*(DTHOFF+C2)-C1)/(SA+XN*CA)
      IF(XALPHA .GT. XALSTR) XUB = (XN*(DTHOFF+C2)+C1)/(SA-XN*CA)
      GO TO 234
  231 IF(XALPHA .LT. XALSTR) GO TO 233
      IF((DK/SA) .LE. DTHOFF) GO TO 232
      XLB = 1.0E70
      GO TO 234
  232 XUB = (XN*(DTHOFF+C2)+C1)/(SA-XN*CA)
      GO TO 234
  233 XLB = (XN*(DTHOFF+C2)+C1)/(SA-XN*CA)
  234 XLB = AMAX1(-ULENGH,X,XLB)
      XUB = AMIN1(ULENGH,XUB,ZUBIC)
C
C     IF EFFECTIVE ROAD LENGTH IS .LE. 4 METERS, OMIT INTEGRATION.
C
      IF((XUB-XLB) .LE. 4.0) GO TO 242
C
C     DETERMINE END POINTS OF INTEGRATION INTERVALS.
C
      RR(1) = XLB
      RR(2) = XLB+1
      XLB2 = XLB+2
      RR(3) = XLB2
      RR(8) = (XLB2+XUB)/2.0
      X = 1E70
      IF(CASE .NE. 2 .AND. SA .NE. 0.0) X = C1/SA
      IF(X .LT. XUB) RR(8) = X
      RR(7) = RR(8)-AMIN1(10.0,(RR(8)-XLB2)/21.0)
      RR(6) = RR(7)-AMIN1(20.0,(RR(7)-XLB2)/10.0)
      RR(5) = RR(6)-AMIN1(30.0,(RR(6)-XLB2)/6.0)
      RR(4) = RR(5)-AMIN1(50.0,(RR(5)-XLB2)/3.0)
      RR(9) = RR(8)+AMIN1(10.0,(XUB-RR(8))/21.0)
      RR(10) = RR(9)+AMIN1(20.0,(XUB-RR(9))/10.0)
      RR(11) = RR(10)+AMIN1(30.0,(XUB-RR(10))/6.0)
      RR(12) = RR(11)+AMIN1(50.0,(XUB-RR(11))/3.0)
      RR(13) = XUB
C
C     EVALUATE INTEGRAND AT LOWER BOUND OF FIRST INTEGRATION INTERVAL.
C
      DIST = C2+XLB*CA
      DTZ = DIST+DTZOFF
      IDTZ = INDEX(IFIX(DTZ/100)+1)
      SZM = AZ(IDTZ,IC)*DTZ**BZ(IDTZ,IC)
      DTH = DIST+DTHOFF
      SHM = TFACTR*AHIC*DTH**BHIC
      X = ((-C1+XLB*SA)/SHM)**2
      X1 = X+(ZZ1/SZM)**2
      X2 = X+(ZZ2/SZM)**2
      XX1 = 0.0
      XX2 = 0.0
      IF(X1 .GT. 28.781749) GO TO 238
      XX1 = EXP(-0.5*X1)
      IF(X2 .LE. 28.781749) XX2 = EXP(-0.5*X2)
```

```
      IF(CUTI .AND. SHM .GT. (CW+0.1)) SHM = CW*(1.0+EXP(-1.0/(SHM-CW)))  A4A3850
  238 Y(7) = (XX1+XX2)/(SHM*SZM)                                          A4A3860
      R(7) = XLR                                                          A4A3870
C                                                                               C
C     EVALUATE INTEGRAL OVER EACH OF THE TWELVE INTERVALS.                      C
C                                                                               C
      DO 241 M = 1,12                                                     A4A3880
      R(1) = R(7)                                                         A4A3890
      R1 = R(1)                                                           A4A3900
      R(7) = RR(M+1)                                                      A4A3910
      R7 = R(7)                                                           A4A3920
C                                                                               C
C     THE VALUE OF THE INTEGRAND AT THE LOWER BOUND OF THE M'TH INTERVAL        C
C     IS THE SAME AS ITS VALUE AT THE UPPER BOUND OF THE M-1'ST INTERVAL.       C
C                                                                               C
      Y(1) = Y(7)                                                         A4A3930
      STEP = 0.0                                                          A4A3940
C                                                                               C
C     EVALUATE THE INTEGRAND AT SEVEN EQUALLY SPACED POINTS WITHIN THE          C
C     M'TH INTERVAL.                                                            C
C                                                                               C
      DO 240 L = 2, 7                                                     A4A3950
      STEP = STEP+0.166667                                                A4A3960
      R(L) = (1.0-STEP)*R1+STEP*R7                                        A4A3970
      DIST = C2+R(L)*CA                                                   A4A3980
      DTZ = DIST+DTZOFF                                                   A4A3990
      IDTZ = INDEX(IFIX(DTZ/100)+1)                                       A4A4000
      SZM = AZ(IDTZ,IC)*DTZ**BZ(IDTZ,IC)                                  A4A4010
      DTH = DIST+DTHOFF                                                   A4A4020
      SHM = TFACTR*AHIC*DTH**BHIC                                         A4A4030
      X = ((-C1+R(L)*SA)/SHM)**2                                          A4A4040
      X1 = X+(ZZ1/SZM)**2                                                 A4A4050
      X2 = X+(ZZ2/SZM)**2                                                 A4A4060
      XX1 = 0.0                                                           A4A4070
      XX2 = 0.0                                                           A4A4080
      IF(X1 .GT. 28.781749) GO TO 239                                     A4A4090
      XX1 = EXP(-0.5*X1)                                                  A4A4100
      IF(X2 .LE. 28.781749) XX2 = EXP(-0.5*X2)                            A4A4110
      IF(CUTI .AND. SHM .GT. (CW+0.1)) SHM = CW*(1.0+EXP(-1.0/(SHM-CW)))  A4A4120
  239 Y(L) = (XX1+XX2)/(SHM*SZM)                                          A4A4130
  240 CONTINUE                                                            A4A4140
C                                                                               C
C     EVALUATE THE INTEGRAL OVER THE M'TH INTERVAL, AND ADD THE RESULT          C
C     TO THE SUM OF THE INTEGRALS OF THE PRECEEDING M-1 INTERVALS.              C
C                                                                               C
      INTGRL = ((R(7)-R(1))/840.0)*(41.0*(Y(1)+Y(7))+216.0*(Y(2)+Y(6))+   A4A4150
     -27.0*(Y(3)+Y(5))+272.0*Y(4))+INTGRL                                 A4A4160
  241 CONTINUE                                                            A4A4170
C                                                                               C
C     STORE INTEGRATION RESULTS BY RECEPTOR LOCATION AND STABILITY CLASS        C
C     IN TEMPORARY AREA.                                                        C
C                                                                               C
  242 L = (I-1)*8 + K                                                     A4A4180
      TEMP(J, L) = INTGRL                                                 A4A4190
      DK = DK+INC                                                         A4A4200
  250 CONTINUE                                                            A4A4210
  251 CONTINUE                                                            A4A4220
  252 CONTINUE                                                            A4A4230
```


2817

```
C
C     MULTIPLY INTEGRATION RESULTS BY LANE GROUP EMISSIONS, AND ADD TO THE
C     SUM OF THESE PRODUCTS FOR THE PREVIOUS LANE GROUPS IN THIS SITE.
C
      K1 = 1
      K16 = 16
      IF(ICLASS(1) .EQ. 0) K1 = 9
      IF(ICLASS(2) .EQ. 0) K16 = 8
      DO 262 I = 1, 3
      IF(.NOT. DOYR(I)) GO TO 262
      X = EF(I)*TFVOL(I)
      L = (I-1)*8
      DO 261 J = 1, 2
      JJ = J+10
      DO 260 K = K1, K16
      LK = L+K+((K-1)/8)*16
      SUPOUT(JJ,LK) = SUPOUT(JJ,LK)+ TEMP(J,K)*X
  260 CONTINUE
  261 CONTINUE
  262 CONTINUE
C
C     RETURN TO PROCESS THE NEXT LANE GROUP IN THIS SITE.
C
      GO TO 37
C
C     CHECK TO SEE IF ANY ANALYSES WERE PERFORMED.
C
  270 IF (DOYRAL(1) .OR. DOYRAL(2) .OR. DOYRAL(3)) GO TO 271
      WRITE(6,6170)
      GO TO 301
C
C     CONVERT THE SUMS OF THE INTEGRATION*EMISSIONS PRODUCTS FOR THIS SITE
C     TO PPM(CO) USING THE WIND SPEED FACTORS, INTEGRATION CONSTANTS, AND
C     CONVERSION FACTORS.
C
  271 DO 280 I = 1, W6
      K2 = 2*I
      K = K2-1
      MWSI = MWSIN(I)
      WFACTR = MWSI+1.92*EXP(-0.19*MWSI)
      X = 3.846244E-5/WFACTR
C
      DO 280 J = J1, J48
      SUPOUT(K,J) = X*SUPOUT(11,J)
      SUPOUT(K2,J) = X*SUPOUT(12,J)
  280 CONTINUE
C
C     ARRANGE METRIC AND AMERICAN ENGINEERING VALUES PROPERLY FOR OUTPUT.
C
      IF(.NOT. METRIC) GO TO 283
C
      DO 281 I = 1, 2
      X = TZ(I)
      MZ(I) = X
      ZZ(I) = X*3.280840
  281 CONTINUE
C
C
```

```
          DO 282 I = 1, 8                                                 A4A4600
          X = D(I)                                                        A4A4610
          MD(I) = X                                                       A4A4620
          D(I) = X*3.280840                                               A4A4630
      282 CONTINUE                                                        A4A4640
          GO TO 286                                                       A4A4650
C                                                                               C
      283 DO 284 I = 1, 2                                                 A4A4660
          ZZ(I) = TZ(I)                                                   A4A4670
          MZ(I) = TZ(I)/3.280840                                          A4A4680
      284 CONTINUE                                                        A4A4690
C                                                                               C
          DO 285 I = 1, 8                                                 A4A4700
          MD(I) = D(I)/3.280840                                           A4A4710
      285 CONTINUE                                                        A4A4720
C                                                                               C
C         OUTPUT RESULTS OF SITE ANALYSES.                                      C
C                                                                               C
      286 PAGE=PAGE+1                                                     A4A4730
          WRITE(6,6100) SITE,JOB,TITLE,PAGE,LG,WSIN,MWSIN,TIME1,TIME2,ZZ,ZZ, A4A4740
         -ZZ,ZZ,ZZ,ZZ,MZ,MZ,MZ,MZ,MZ,MZ                                   A4A4750
C                                                                               C
          DO 300 I = J1, J48                                              A4A4760
          J = (I-1)/8+1                                                   A4A4770
          J = J-((J-1)/3)*3                                               A4A4780
          IF(.NOT. DOYRAL(J) ) GO TO 288                                  A4A4790
          L = I - (I - 1) / 8 * 8                                         A4A4800
          WRITE(6,6110) D(L),   MD(L), (SUPOUT(M, I), M = 1, W12)         A4A4810
          IF(L .EQ. 4) WRITE(6, 6130) YEAR(J)                             A4A4820
          GO TO 289                                                       A4A4830
      288 WRITE(6, 6150)                                                  A4A4840
      289 WRITE(6, 6120) A1(I)                                            A4A4850
          IF(I .EQ. 12) WRITE(6, 6125) CLASS(1)                           A4A4860
          IF(I .EQ. 36) WRITE(6, 6125) CLASS(2)                           A4A4870
          IF(.NOT. DOYRAL(J) .OR. I .EQ. 48) GO TO 300                    A4A4880
          IF(L .EQ. 8 .AND. I .NE. 24) WRITE(6,6155)                      A4A4890
          IF(I .EQ. 24 .AND. J48 .EQ. 48) WRITE(6,6140)                   A4A4900
      300 CONTINUE                                                        A4A4910
      301 IF(LAST) STOP                                                   A4A4920
C                                                                               C
C         RETURN FOR NEXT DIVISION OR SITE AS APPROPRIATE.                      C
C                                                                               C
          IF(SITEID .EQ. HEND) GO TO 1                                    A4A4930
          FIRST = .TRUE.                                                  A4A4940
          DIDONE = .TRUE.                                                 A4A4950
          LG = 0                                                          A4A4960
          DO 310 J = 11, 12                                               A4A4970
          DO 310 I = 1, 48                                                A4A4980
          SUPOUT(J, I) = 0.0                                              A4A4990
      310 CONTINUE                                                        A4A5000
          GO TO 37                                                        A4A5010
C                                                                               C
C         HAVE REACHED AN EOF ON DATA CARD READ.  OUTPUT LAST DATA SET, IF      C
C         NOT ALREADY DONE.  OTHERWISE TERMINATE PROCESSING.                    C
C                                                                               C
      996 IF(FIRST) STOP                                                  A4A5020
          LAST = .TRUE.                                                   A4A5030
          GO TO 270                                                       A4A5040
```

```
C
C     A NULL SITE HAS BEEN ENCOUNTERED.  PRINT MESSAGE AND CONTINUE
C     PROCESSING.
C
  997 PAGE = PAGE+1
      WRITE(6, 6000) TITLE,JOB,PAGE
      WRITE(6, 6001)
      GO TO 37
C
C     IF A NULL DIVISION HAS NOT BEEN DETECTED, CONTINUE PROCESSING.
C     OTHERWISE, PRINT MESSAGE, AND CONTINUE PROCESSING.
C
  998 IF(DIDONE) GO TO 1
      PAGE = PAGE+1
      WRITE(6, 6000) TITLE,JOB,PAGE
      WRITE(6, 6002)
      GO TO 1
C
C     HAVE REACHED A NORMAL EOF.
C
  999 STOP
      END
```