EPA-600/8-85/007

MARCH 1985

SAI AIRSHED MODEL OPERATIONS MANUALS

Volume II -- Systems Manual

ATMOSPHERIC SCIENCES RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27711

SAI AIRSHED MODEL OPERATIONS MANUALS

Volume II -- Systems Manual

by

J. Ames, S.R. Hayes, T.C. Myers, and D.C. Whitney

Systems Applications, Incorporated
101 Lucas Valley Road
San Rafael, California 94903

Contract No. 68-02-2429

Project Officers

Kenneth L. Demerjian
Kenneth L. Schere
Meteorology and Assessment Division
Atmospheric Sciences Research Laboratory
Research Triangle Park, North Carolina 27711

ATMOSPHERIC SCIENCES RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27711

NOTICE

The information in this document has been funded wholly or in part by the United States Environmental Protection Agency under Contract No. 68-02-2429 to Systems Applications, Inc. It has been subject to the Agency's peer and administrative review, and it has been approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

ABSTRACT

This report describes the Systems Applications, Inc. (SAI) Airshed Model System from a programmer's point of view. Included are discussions of all subroutines and how they fit together, run-time core allocation techniques, internal methods of segment handling using secondary storage, and detailed definitions of the structure of each file in the system. This manual also includes suggestions for implementing the programs on different computers.

The SAI Airshed Model System consists of 17 programs that communicate with each other through 17 data files; both the programs and files are described in detail in this manual. Chapter II presents an overview of the system, briefly describing the programs used to perform each of the five system functions. Chapter III contains guidelines for implementing the programs, and Chapters IV through VIII contain detailed descriptions of the programs by function, including hierarchical subroutine tables and short module descriptions. Chapter IX contains detailed file structure definitions for each file in the system.

The User's Manual, the companion to this report, contains a general view of the SAI Airshed Model and detailed operating instructions for the user. Included are discussions of all the files needed to run the model, the data preparation programs that produce each file, all the input formats and data preparation methods, samples of input and output, and information flow diagrams that illustrate the job stream control.

This report was submitted in fulfillment of Contract No. 68-02-2429 by Systems Applications, Inc. under the sponsorship of the U.S. Environmental Protection Agency. This report covers the period 23 August 1976 to 15 September 1979 and work was completed as of 15 September 1979.

CONTENTS

ABSTRACT iii

LIST OF FIGURES vii

LIST OF TABLES ix

LIST OF EXHIBITS xi

I INTRODUCTION 1

II OVERVIEW 2

A. M.E.B. File Preparation 2
B. M.E.B. File Segmentation. 5
C. Control Data Preparation. 5
D. Airshed Simulation. 6
E. Display and Analysis. 6

III IMPLEMENTING THE SAI AIRSHED MODEL SYSTEM. 7

A. Coding Conventions. 7
B. Implementation Strategy 8

IV M.E.B. File Preparation Programs 10

A. Common Program Features 10
B. Utility Subroutines 12
C. Individual Program Structures 15
D. Implementing the Programs 61

V M.E.B. FILE SEGMENTATION PROGRAM 65

VI CONTROL DATA PREPARATION PROGRAMS. 80

VII AIRSHED SIMULATION PROGRAM 81

A. Core Allocation 81
B. Temporary Secondary Storage Files 92
C. Boundary Handling 96

D. Computing Average Concentrations 103
E. Flow of Control. 104
F. COMMON Blocks. 117
G. Overlays 121
H. Implementing the Program 125
I. Locations of Specific Algorithms 129
VIII THE DISPLAY PROGRAM 132
A. Overall Logic Flow 132
B. Input/Output 132
C. Subroutine Definition and Hierarchy. 134
D. Implementation Considerations. 137
IX FILE STRUCTURES 138
A. Input and Output File Structure Definitions. 138
B. Internal File Structure Definitions. 233

FIGURES

Number		Page
II-1	Information Flow in the SAI Airshed Model System	3
IV-1	Typical Flow of Control for File Preparation Programs	13
IV-2	Flow of Control for AIRQUALITY File Preparation Program.	25
IV-3	Flow of Control for BOUNDARY File Preparation Program.	28
IV-4	Flow of Control for DIFFBREAK File Preparation Program.	31
IV-5	Flow of Control for EMISSIONS File Preparation Program.	34
IV-6	Flow of Control for METSCALARS File Preparation Program.	37
IV-7	Flow of Control for PTSOURCE File Preparation Program.	40
IV-8	Flow of Control for REGIONTOP File Preparation Program.	43
IV-9	Flow of Control for ROADWAY File Preparation Program.	46
IV-10	Flow of Control for TEMPERATUR File Preparation Program.	49
IV-11	Flow of Control for TERRAIN File Preparation Program.	52
IV-12	Flow of Control for TOPCONC File Preparation Program.	55
IV-13	Flow of Control for WIND File Preparation Program.	58
IV-14	Overlay Structure for Data Preparation Programs.	62

Number		Page

V-1	Flow of Control for File Segmentation	66
V-2	Flow of Control--AIRQUALITY File Segmentation Subroutine	68
V-3	Flow of Control--BOUNDARY File Segmentation Subroutine	69
V-4	Flow of Control--DIFFBREAK File Segmentation Subroutine	70
V-5	Flow of Control--EMISSIONS File Segmentation Subroutine	71
V-6	Flow of Control--PTSOURCE File Segmentation Subroutine	72
V-7	Flow of Control--ROADWAY File Segmentation Subroutine	73
V-8	Flow of Control--REGIONTOP File Segmentation Subroutine	74
V-9	Flow of Control--TEMPERATUR File Segmentation Subroutine	75
V-10	Flow of Control--TOPCONC File Segmentation Subroutine	76
V-11	Flow of Control--TERRAIN File Segmentation Subroutine	77
V-12	Flow of Control--WIND File Segmentation Subroutine	78
VII-1	Pointers to Variably Dimensioned Arrays	83
VII-2	The Four Tables Assigned to the Vector of Available Space	86
VII-3	Segment Table Structure	87
VII-4	Local Table Structure	89
VII-5	Boundary Table Structure	91
VII-6	Segment Scratch Table Structure	93
VII-7	Core Allocation and File Storage	97
VII-8	Example of a Segmented Region	98
VII-9	Hierarchy of Top-Level Subroutines	105
VII-10	Airshed Simulation Program Overlay Structure	122
VIII-1	Schematic of the Display Program Structure	133
VIII-2	Subroutine Hierarchy Diagram	136
IX-1	Files in the SAI Airshed Model System	139

TABLES

Number		Page
II-1	SAI Airshed Model Program	4
IV-1	UTILITY Modules Grouped by Usage Category	16
IV-2	Subroutines in UTILITY	18
IV-3	Subroutines in FILUTIL	23
IV-4	Routines Specific to AIRQUALITY File Preparation	27
IV-5	Routines Specific to BOUNDARY File Preparation	30
IV-6	Routines Specific to DIFFBREAK File Preparation	33
IV-7	Routines Specific to EMISSIONS File Preparation	36
IV-8	Routines Specific to METSCALARS File Preparation	39
IV-9	Routines Specific to PTSOURCE File Preparation	42
IV-10	Routines Specific to REGIONTOP File Preparation	45
IV-11	Routines Specific to ROADWAY File Preparation	48
IV-12	Routines Specific to TEMPERATUR File Preparation	51
IV-13	Routines Specific to TERRAIN File Preparation	54
IV-14	Routines Specific to TOPCONC File Preparation	57
IV-15	Routines Specific to WIND File Preparation	60
V-1	File Segmentation Routines	79
VII-1	Local Variable Overlays	90
VII-2	Arrays in the Segment Scratch Table	94
VII-3	Boundary Definition Indices	101

Number		Page
VII-4	Subroutine Hierarchy in the Airshed Simulation Program	108
VII-5	Descriptions of Airshed Simulation Program Subroutines	112
VII-6	COMMON Block/Subroutine Cross-Reference List	119
VII-7	LBL Library Functions	126
VII-8	File Unit Assignments	130
VII-9	Location of Specific Numerical Functions	131
VIII-1	Subroutine Descriptions	135
IX-1	SAI Airshed Model System Files and Their Descriptions	140
IX-2	AIRQUALITY File Stucture	146
IX-3	AVERAGE File Structure	151
IX-4	BOUNDARY File Structure	157
IX-5	CHEMPARAM File Structure	163
IX-6	DIFFBREAK File Structure	169
IX-7	EMISSIONS File Structure	174
IX-8	INSTANT File Structure	180
IX-9	METSCALARS File Structure	185
IX-10	PTSOURCE File Structure	191
IX-11	REGIONTOP File Structure	197
IX-12	ROADWAY File Structure	202
IX-13	SIMCONTROL File Structure	207
IX-14	TEMPERATUR File Structure	211
IX-15	TERRAIN File Structure	216
IX-16	TOPCONC File Structure	221
IX-17	VARIATION File Structure	226
IX-18	WIND File Structure	231

EXHIBITS

Number		Page
VII-1	Code Fragments Implementing the Core Allocation Scheme	84

I INTRODUCTION

This report describes the Systems Applications, Inc. (SAI) Airshed Model System from a programmer's point of view. Included are discussions of all subroutines and how they fit together, run-time core allocation techniques, internal methods of segment handling using secondary storage, and detailed definitions of the structure of each file in the system. This manual also includes suggestions for implementing the programs on different computers.

The SAI Airshed Model System consists of 17 programs that communicate with each other through 17 data files; both the programs and files are described in detail in this manual. Chapter II presents an overview of the system, briefly describing the programs used to perform each of the five system functions. Chapter III contains guidelines for implementing the programs, and Chapters IV through VIII contain detailed descriptions of the programs by function, including hierarchical subroutine tables and short module descriptions. Chapter IX contains detailed file structure definitions for each file in the system.

The User's Manual, the companion to this report, contains a general view of the SAI Airshed Model and detailed operating instructions for the user. Included are discussions of all the files needed to run the model, the data preparation programs that produce each file, all the input formats and data preparation methods, samples of input and output, and information flow diagrams that illustrate the job stream control.

II OVERVIEW

Figure II-1 shows the flow of information through the SAI Airshed Model System. Each rectangular box in the diagram represents one of the five major functions of the system:

> Meteorology, emissions, and initial and boundary conditions (M.E.B.) file preparation.
> M.E.B. file segmentation.
> Control data preparation.
> Airshed simulation.
> Display and analysis.

These five functions were described from the user's point of view in the User's Manual. This report describes the characteristics of the computer programs that perform each function.

There are 17 programs in the SAI Airshed Model System. Table II-1 lists the programs, grouped by function. The remainder of this chapter presents general descriptions of the programs that constitute each of the five functions and discusses additional programs. For a particular application, additional programs can be developed to convert raw data into the input format required by the data preparation programs, or to create a data file directly, bypassing the existing data preparation program.

A. M.E.B. FILE PREPARATION

The 12 programs that prepare the 12 meteorology, emissions, and initial and boundary conditions (M.E.B.) files each perform the following functions:

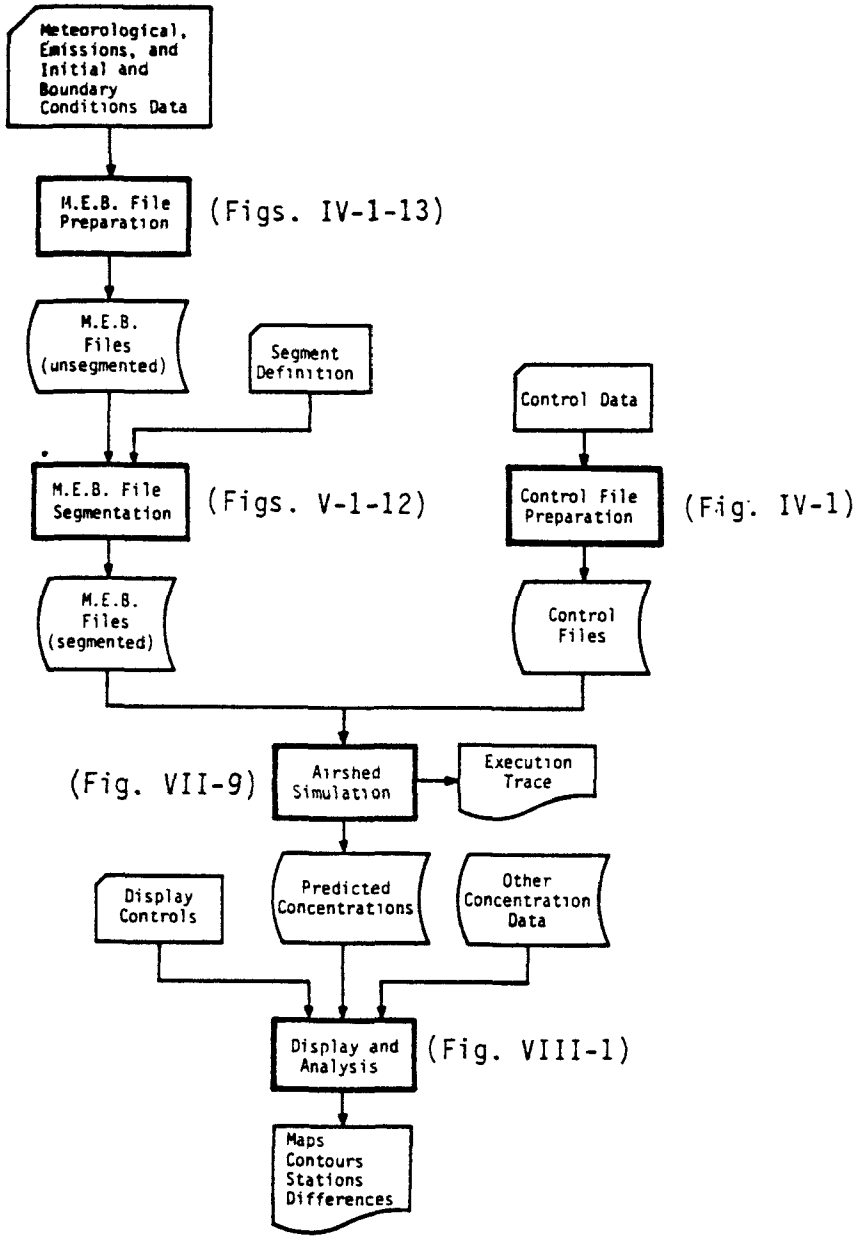


Figure II-1. Information Flow in the SAI Airshed Model System

Table II-1. SAI Airshed Model Programs

Function	Program	Purpose
M.E.B. File Preparation		
	AIRQUL	Creates the AIRQUALITY file
	BNDARY	Creates the BOUNDARY file
	DFSNBK	Creates the DIFFBREAK file
	EMISSN	Creates the EMISSIONS file
	METSCL	Creates the METSCALARS file
	PTSRCE	Creates the PTSOURCE file
	REGNTP	Creates the REGIONTOP file
	ROADWY	Creates the ROADWAY file
	TEMPTR	Creates the TEMPERATUR file
	TERAIN	Creates the TERRAIN file
	TPCONC	Creates the TOPCONC file
	WINDSET	Creates the WIND file
M.E.B. File Segmentation		
	SEGDEF	Segments any spatially varying M.E.B. file
Control Data Preparation		
	CPREP	Creates the CHEMPARAM file
	SPREP	Creates the SIMCONTROL file
Airshed Simulation		
	AIRSIM	Performs the airshed simulation and writes the INSTANT, AVERAGE, and VARIATIONS files
Display and Analysis		
	DISPLAY	Generates concentration maps and vertical profiles

> Read and verify input card packets that control the data preparation algorithms to be used, and supply observed data points.
> Allocate space for each necessary array in a "scratch vector."
> Assign "core extension" files, if necessary, for temporary storage of large arrays.
> Perform required computations to generate gridded data.
> Write output data files.

The programs were designed together as a unified package with standardized input, processing, and output modules. Thus, they all have the same basic structure and share many general purpose utility and file handling subroutines. These shared subroutines are kept in two separate libraries where they need be compiled and stored only once, but can be picked up by the loader (collector, linkage editor) when each program is prepared for execution by the system.

B. M.E.B. FILE SEGMENTATION

M.E.B. files containing data that vary spatially may need to be segmented before they are passed to the Airshed Simulation Program. The file segmentation program--SEGDEF--can be used to segment all the spatially varying M.E.B. files. This program reads control cards that define the segment layout and constructs a segment grid, verifying the proper overlapping of segment boundaries. It then reads the file to be segmented and uses the segment grid to direct output of data arrays by segment to a new file. The file segmentation program uses a scratch vector for allocation of arrays and, when necessary, core extension files for large arrays. It also uses many of the same utility subroutines used by the M.E.B. file preparation programs.

C. CONTROL DATA PREPARATION

Two programs, CPREP and SPREP, prepare the two control data files, CHEMPARAM and SIMCONTROL, respectively. They are short, simple programs, without subroutines, that read cards, perform simple validation tests, and write files in a straightforward manner.

D. AIRSHED SIMULATION

The Airshed Simulation Program, AIRSIM, reads the files created in the first three steps. It performs the simulation and writes its results, as instantaneous and time-averaged concentrations over the entire grid, to the simulation results files. It also produces a trace of its execution steps. This program contains the following features:

> Run-time allocation of arrays of variable-lengths within a vector of available space.
> Overlay structure to maximize the length of that vector.
> Use of temporary (scratch) files for intermediate storage of data arrays by segment.

E. DISPLAY AND ANALYSIS

The display and analysis function includes all programs that process the results of the airshed simulation. Currently, there is one such program--DISPLAY. It reads the simulation results files and a control deck, including station locations; it produces concentration maps at any selected level as well as vertical concentration profiles at the stations.

III IMPLEMENTING THE SAI AIRSHED MODEL SYSTEM

Since the SAI Airshed Model System contains so many separate programs and data files, its initial implementation on a different computer may seem a formidable task. To simplify the conversion task, this chapter describes some of the coding conventions used and presents an eight-step approach to the implementation of the system as a whole.

A. CODING CONVENTIONS

All SAI Airshed Model programs are written in FORTRAN IV and were originally implemented on the Lawrence Berkeley Laboratory (LBL) CDC 7600 computer system. The programs were subsequently modified to run on the Environmental Protection Agency's UNIVAC 1100 computer. This report discusses the version of the codes implemented on the EPA's UNIVAC computer. The following are some features of the code for this system:

> There are no multiple entry points or returns.

> No arrays are greater than three-dimensional.

> There are no complex or double-precision variables.

> There is no rereading of input data cards.

> All alphanumeric variables are represented by vectors containing one character per word, left justified.

> Output format statements use Hollerith counts rather than special delimiters for alphanumeric text fields.

> Routines accessing secondary storage files or large core memory are isolated modules that can be easily changed.

> All main programs begin with comments naming all files that the program uses and, if necessary, providing equivalences between local file names and FORTRAN unit numbers.

B. IMPLEMENTATION STRATEGY

This section briefly sketches the tasks necessary to make the SAI Airshed Model System operational on a different computer.

1. First, determine the locations of the programs and test decks. Read in each source file (program or data) and store it by name.
2. Locate @ ELT cards. These cards precede each predefined main program, COMMON block, and subroutine. Change or remove these cards as necessary, and attempt to compile each program module.*
3. Locate and change all language incompatibilities. Change large core and secondary storage access routines as required. Special implementation considerations for each program are described as necessary in Chapters IV through VIII.
4. Load (or link/edit or collect) each program, using subroutine libraries and overlay structures where necessary. Adjust the scratch vector in each program to make optimal use of available space.
5. Run each data preparation program with the test data decks provided. Refer to the job control diagrams in Chapters VII and IX of the User's Manual. Save the files produced.
6. Determine whether the segment size defined in the test data deck will fit in the Airshed Simulation Program

*Some compilers using the FORTRAN-77 standard will cause fatal errors to occur at program execution time if real arrays are passed through integer arrays from a main program to its subroutines. Because the Airshed Model uses this technique, the variable-type checking should be turned off during the compile step (usually by invoking the FORTRAN-66 compiler option).

(see Section VIII.A of the User's Manual). If not, make a new segment definition deck. Segment all spatially varying data files and save the segmented versions.

7. Run the Airshed Simulation Program, saving the results.
8. Run the display program, using the sample control deck and the simulation results files.

Chapter V of the User's Manual contains a worksheet that can be used to keep track of the locations and special characteristics of each program. It also contains a summary of file characteristics that can be used to aid in the preparation of job control specifications.

IV M.E.B. FILE PREPARATION PROGRAMS

All of the 12 programs used for preparing the M.E.B. files have the same basic structure and use many routines in common. This chapter first describes the features common to all the programs, including general purpose routines, scratch storage, core extension, and user subroutines. Next, a list of the utility modules is presented, followed by a detailed discussion of the structure of the programs with charts and tables showing the utility and special purpose routines called by each one. Finally, some of the considerations involved in implementing the programs are discussed, including source deck structure, scratch vector length, overlaying, I/O error handling, and core extension.

A. COMMON PROGRAM FEATURES

The M.E.B. file preparation programs were designed and written as a unified system of programs that incorporate standardized techniques of data transfer, naming conventions, subroutine interfaces, and program structure in general. The following features are common to all the M.E.B. file preparation programs.

1. General Purpose Routines

There are two libraries of utility modules that can be used by more than one program. The first library--UTILITY--contains routines that read the standard data packets, perform the method and vertical method calculations, convert input units to internal units, allocate and initialize scratch storage, perform data transfers involving core extension, if any, and control reads to other data files. Each of the routines in the second library--FILUTIL--performs either a single read or write to one of the 12 data files. Included are reads and writes of header records, other time-invariant records, time interval records, and other time-varying records.

2. Scratch Allocation

To provide maximum flexibility for defining region size and item counts, array sizes have not been preassigned in the FORTRAN programs. Instead, all available storage is contained in a single "scratch" vector. Once the system parameters are read from the CONTROL and REGION packets, the scratch storage vector is partitioned by a subroutine--SCRPRT-- that constructs a pointer to successively higher locations in the scratch vector for each array required. The value of each pointer is calculated on the basis of the size of the preceding array, with the final pointer indicating the beginning of "available scratch" storage, which can be used by any module that needs more space for its calculations.

When the processing routine is called, the array variables are passed as arguments by specifying their locations in the scratch vector. In the processing routine itself, the actual array names appear in the argument list, and the appropriate set of variable dimensions is assigned. For a more detailed description of run-time scratch storage allocation, see Section VII.A.

3. Core Extension

Some of the programs require three-or four-dimensional arrays. In these programs, a calculation is made to determine whether the scratch vector is large enough to contain all of the arrays. If it is not, a special "core extension" routine--MATSET--is called that creates a random access file for storing the entire array in two-dimensional slices and allocates space in scratch storage for a single two-dimensional slice. (Note: This feature has been disabled for use on the UNIVAC; it is a CDC computer feature.) Access to such arrays is provided through modules that transfer information between the two-dimensional slice and the entire array, whether that array is in core or in a special file. See Section D of this chapter for suggestions on implementing this feature on different computers.

4. Typical Program Structure

All of the programs have the same basic structure. The main routine reads the CONTROL and REGION data packets, reads the header records from any

input files, and writes the header records to the output file. The main program also partitions scratch storage for use by the rest of the program, sets up core extension if necessary, and finally calls the processing routine. The processing routine reads the remainder of the user's time invariant and time-varying data packets, performs all calculations on the input data, and writes the remainder of the output file. It uses a variety of standard utility modules and, for most programs, a few special-purpose modules. Figures IV-1(a) and (b) illustrate the general flow of control for all of the M.E.B. file preparation programs. These figures are used again in Section C of this chapter to show the specific subroutines used by each program.

5. User Subroutines

Each of the data processing programs except that for METSCALARS has the capability of calling one user-written subroutine for computations at ground level and another for vertical computations, if any. The interfaces for these routines are included in the code, and dummy user routines have been supplied. There is no restriction on the functions of the user routines, and all variables from the processing routine are included in each subroutine call. Access to such routines is accomplished by the specification of USER in the METHOD packet or VERTUSER in the VERTICAL METHOD packet.

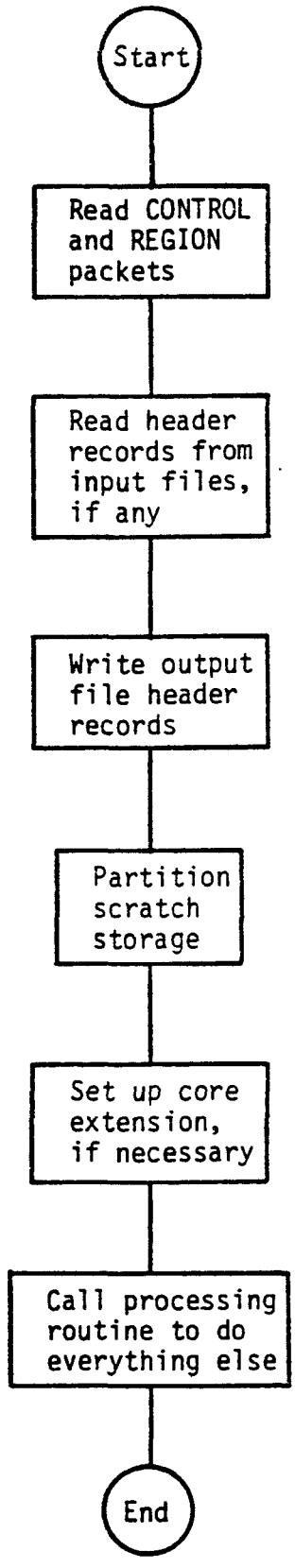
B. UTILITY SUBROUTINES

The two libraries of utility subroutines--UTILITY AND FILUTIL--are described in this section.

1. UTILITY

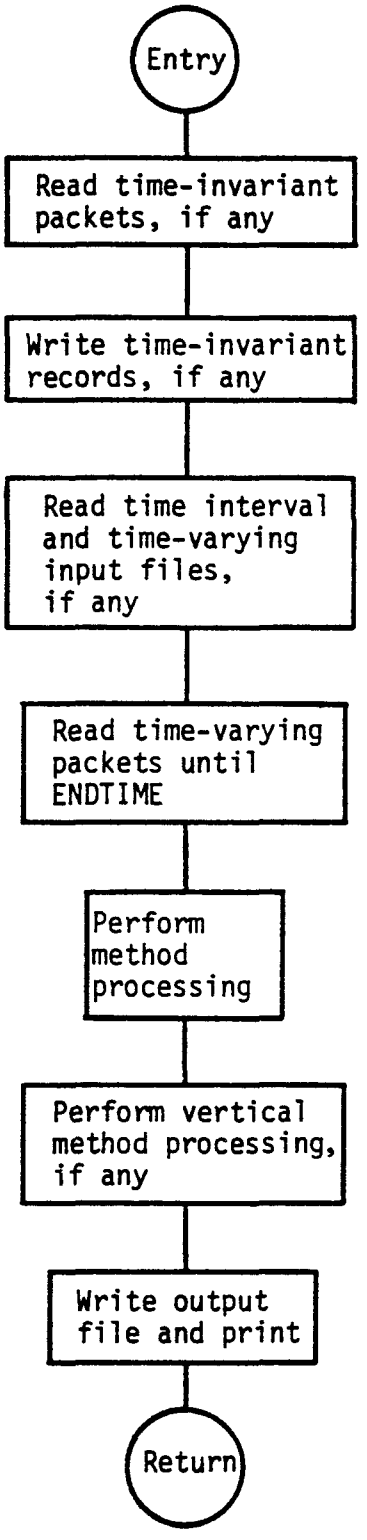
The UTILITY library subroutines can be grouped into the following categories of uses:

> Text manipulation
> Input file read
> Scratch storage manipulation
> Core extension
> Packet read



(a) Main Program

Figure IV-1. Typical Flow of Control for File Preparation Programs



(b) Processing Routine

Figure IV-1 (Concluded)

> Unit conversion
> Method computation
> Vertical method computation
> Grid printing.

Table IV-1 lists the names of the subroutines in each group. Table IV-2 lists alphabetically and describes all subroutines in UTILITY and shows the other subroutines that each calls.

2. FILUTIL

The subroutines in the FILUTIL library perform reads or writes directly to the data files. All subroutine names are six characters long, beginning with "FL" and ending with "RD" for read or "WR" for write. The middle two letters indicate the type of record being written. The subroutines in FILUTIL can be called by main programs, processing routines, and subroutines in UTILITY. Table IV-3 is an alphabetical list of all subroutines in FILUTIL with a short description of each one.

C. INDIVIDUAL PROGRAM STRUCTURES

This section contains the following information for each program:

> A structure chart showing all subroutines directly called by the main program and its processing routine (Figures IV-2 through IV-13).
> A table listing all routines not contained in either of the libraries (Tables IV-4 through IV-15).

Each structure chart follows the same general pattern of processing steps illustrated and described in Section A of this chapter. Next to each box (indicating a processing step) in the diagram is a list of subroutines called to perform that function. Other subroutines called by these subroutines are not shown in the figure, but are listed in the corresponding subroutine table or in the UTILITY subroutine table (IV-2). If any processing step is not performed by a particular program, the word "none" appears next to the corresponding box. The structure charts and tables appear in alphabetical order according to the names of the files they create.

Table IV-1. UTILITY Modules Grouped by Usage Category

Usage Category	Module
Text manipulation	COMPAR
	TXTPAK
Input file read	DFBKRD
	MTSCRD
	RGTPRD
	TMPSRD
	TPCNRD
	WINDRD
Scratch storage manipulation	SCRPRT
	SCRSET
	SINVAL
Core extension	COREXT
	MATGET
	MATLOD
	MATPUT
	MATSET
	MATSTR
Packet read	CNSTRD
	CNTLRD
	GDVLRD
	MTHDRD
	REGNRD
	SBRGRD
	STATRD
	STVLRD
	TIMERD
	UNITRD
	VPRFRD

Table IV-1 (Concluded)

Usage Category	Module
Unit conversion	UNTCNV
Method computation	EWINTR
	NSINTR
	POISSP
	POISVR
	SPLITC
	STASGN
Vertical method computation	ABSPRF
	ABPRFR
	CELLHT
	EWINTR
	HTZCLL
	NSINTR
	RELPRF
	RLPRFR
Grid printing	GRIDPR

Table IV-2. Subroutines in UTILITY

Name	Description	Subroutines Called
ABSPRF	Absolute profile method--calculates the value at level aloft from user profile	CELLHT
ABPRFR	Absolute profile ratio method--calculates the value at level aloft from user profile, relative to top and bottom values	CELLHT
CELLHT	Calculates the height of the center of a vertical cell above ground level	
CNSTRD	Reads the CONSTANTS packet, performs unit conversions if necessary, and stores values in the output matrix	TXTPAK COMPAR UNTCNV MATGET MATPUT
CNTLRD	Reads the CONTROL packet	TXTPAK COMPAR TIMERD
COMPAR	Compares two strings character by character for equality	
COREXT	Sets up core extension for matrix if it will not fit in core	SCRSET MATSET MATPUT
DFBKRD	Reads the time-variant Diffusion Break Record from DIFFBREAK	COMPAR FLDBRD FLTMRD
FILVER	Reads and verifies file header records	COMPAR FLHDRD FLRGRD FLSGRD FLSPRD

Table IV-2 (Continued)

Name	Description	Subroutines Called
EWINTR	Interpolates a value on an east-west line from values in adjacent subregions	
GDVLRD	Reads the GRID VALUES packet	TXTPAK COMPAR UNTCNV MATGET MATPUT
GRIDPR	Prints a grid of output values in accordance with a selectable format	
HTZCLL	Calculates the vertical cell number for a given height above ground level	
MATGET	Gets a matrix from scratch vector or from core extension	MATLOD
MATLOD	Recovers a record from random access mass storage	
MATPUT	Puts a matrix to scratch vector or to core extension	MATSTR
MATSET	Sets up a random access mass storage file	
MATSTR	Stores a record on random access mass storage	
MTHDRD	Reads the METHOD packet	TXTPAK COMPAR UNTCNV
MTSCRD	Reads time-variant records on METSCALARS	COMPAR FLMSRD FLTMRD
NSINTR	Interpolates a value on a north-south line from values in adjacent subregions	

Table IV-2 (Continued)

Name	Description	Subroutines Called
POISSP	Calculates full grid distribution of station values (concentrations) in one subregion by means of the POISSON method for each variable	SINVAL COMPAR
POISVR	Calculates full grid distribution of station values (meterological and other variables) in one subregion by means of the POISSON method for each variable	SINVAL
REGNRD	Reads the REGION packet	TXTPAK
RELPRF	Calculates the value at level aloft from user profile, adjusted for variation of diffusion break	CELLHT
RFTPRD	Reads time-variant records on REGIONTOP	COMPAR FLRTRD FLTMRD
RLPRFR	Calculates the value at level aloft from user profile, relative to top and bottom values, adjusted for variation of diffusion break	CELLHT
SBRGRD	Reads the SUBREGION packet	TXTPAK COMPAR SINVAL GRIDPR
SCRPRT	Allocates scratch vector to the arrays required	SCRSET
SCRSET	Allocates space to a single array and initializes space	SINVAL
SINVAL	Places a single value into every location of a vector	
SPLITC	Calculates the value of a variable as a function of the values of a set of other variables	COMPAR MATGET MATPUT

Table IV-2 (Continued)

Name	Description	Subroutines Called
STASGN	Calculates station-based interpolation values for a single grid square	
STATRD	Reads the STATIONS packet	TXTPAK COMPAR UNTCNV
STVLRD	Reads the STATION VALUES packet	TXTPAK COMPAR UNTCNV
TIMERD	Reads a time interval card (beginning and ending dates and times)	
TMPSRD	Reads time-varying records on TEMPERATUR	COMPAR FLTPRD FLTMRD
TPCNRD	Reads time-varying records on TOPCONC	COMPAR FLTCRD FLTMRD MATPUT
TXTPAK	Left-justifies text string, removing blanks, with blank fill at right	
UNITRD	Reads the UNITS packet	TXTPAK COMPAR
UNTCNV	Uses unit conversion tables to convert from input units to internal units	
VPRFRD	Reads the VERTICAL PROFILES packet	TXTPAK COMPAR UNTCNV

Table IV-2 (Concluded)

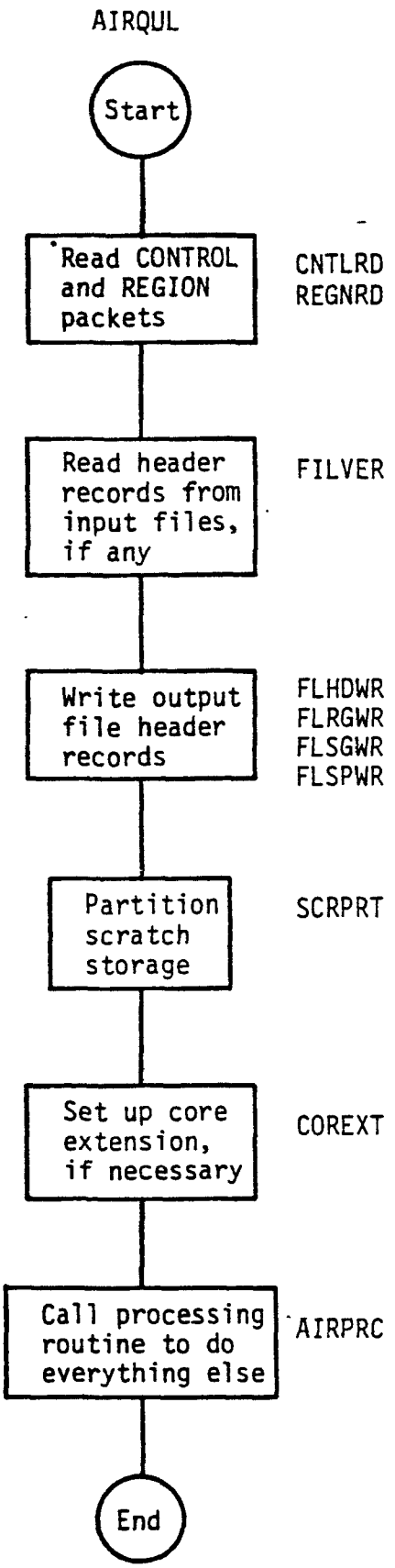
Name	Description	Subroutines Called
WINDRD	Reads time-varying records on WIND	COMPAR FLWDRD FLTMRD
WNDSPD	Calculates maximum absolute wind speeds in x- and y-directions and average wind speeds at each boundary	MATGET

Table IV-3. Subroutines in FILUTIL

Name	Description
FLAQRD	Reads Air Quality Records for a segment
FLAQWR	Writes Air Quality Records for a segment
FLBIRD	Reads Boundary Definition Record for one edge
FLBIWR	Writes Boundary Definition Record for one edge
FLBVRD	Reads Boundary Concentration Record for one edge
FLBVWR	Writes Boundary Concentration Record for one edge
FLDBRD	Reads Diffusion Break Record for a segment
FLDBWR	Writes Diffusion Break Record for a segment
FLEMRD	Reads Emissions Records for a segment
FLEMWR	Writes Emissions Records for a segment
FLHDRD	Reads File Description Header Record
FLHDWR	Writes File Description Header Record
FLMSRD	Reads Meteorological Scalars Record
FLMSWR	Writes Meteorological Scalars Record
FLPIRD	Reads time-invariant Counter Record and Point Source Definition Record
FLPIWR	Writes time-invariant Counter Record and Point Source Definition Record
FLPVRD	Reads time-varying Counter Record, Point Source Definition Record, and Point Source Emissions Record
FLPVWR	Writes time-varying Counter Record, Point Source Definition Record, and Point Source Emissions Record

Table IV-3 (Concluded)

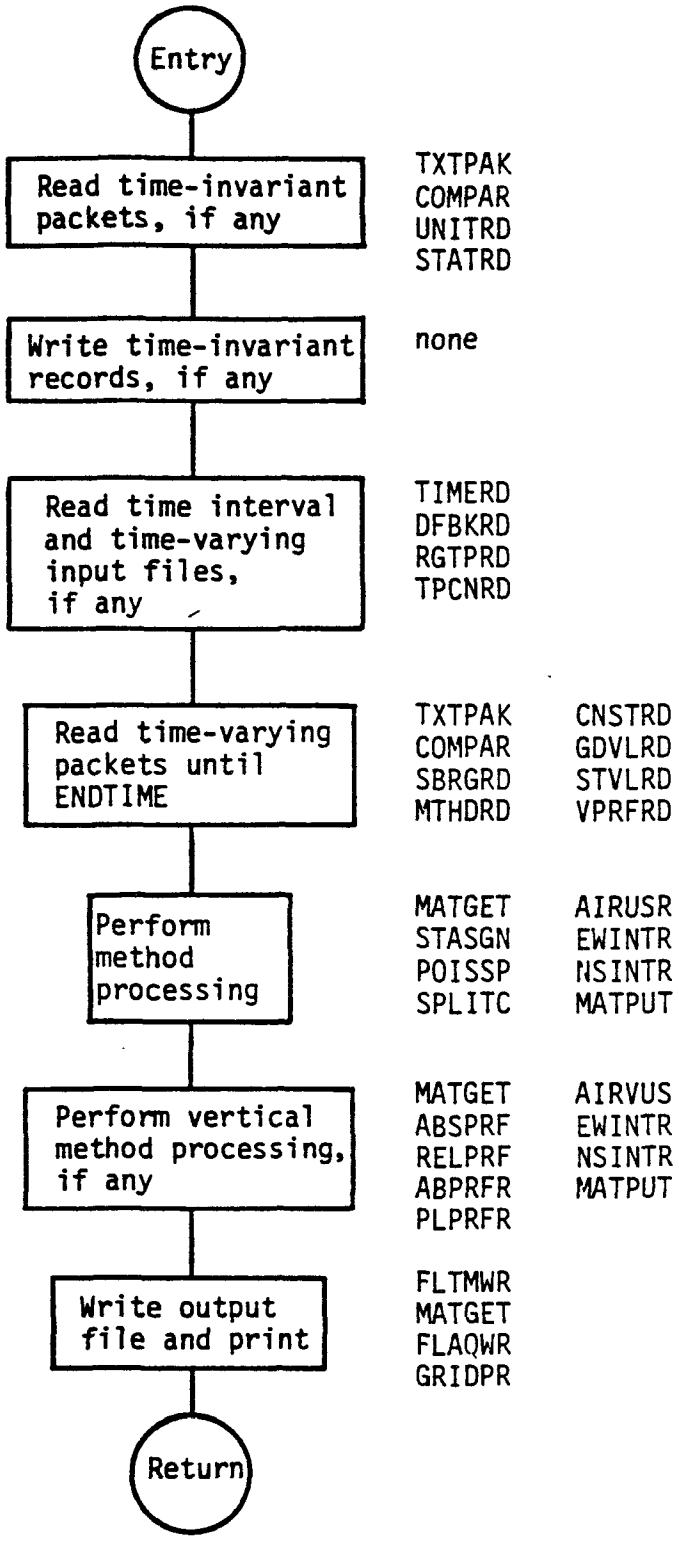
Name	Description
FLRGRD	Reads the Region Definition Header Record
FLRGWR	Writes the Region Definition Header Record
FLRTRD	Reads the Region Top Record
FLRTWR	Writes the Region Top Record
FLRWRD	Reads Roadway Emissions Records
FLRWWR	Writes Roadway Emissions Records
FLSGRD	Reads the Segment Definition Header Record
FLSGWR	Writes the Segment Definition Header Record
FLSPRD	Reads the Species Definition Header Record
FLSPWR	Writes the Species Definition Header Record
FLTCRD	Reads Top Concentration Records
FLTCWR	Writes Top Concentration Records
FLTMRD	Reads Time Interval Record
FLTMWR	Writes Time Interval Record
FLTPRD	Reads Temperature Record
FLTPWR	Writes Temperature Record
FLTRRD	Reads Terrain Records
FLTRWR	Writes Terrain Records
FLWDRD	Reads time-varying Wind Records
FLWDWR	Writes time-varying Wind Records



(a) Main Program

Figure IV-2. Flow of Control for AIRQUALITY File Preparation Program

AIRPRC

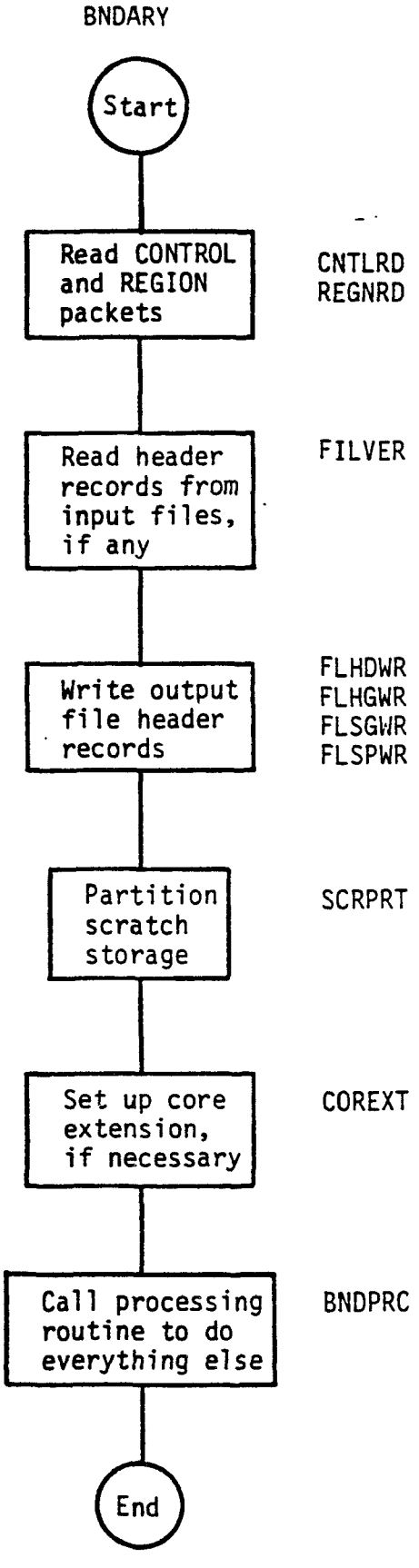


(b) Processing Routine

Figure IV-2 (Concluded)

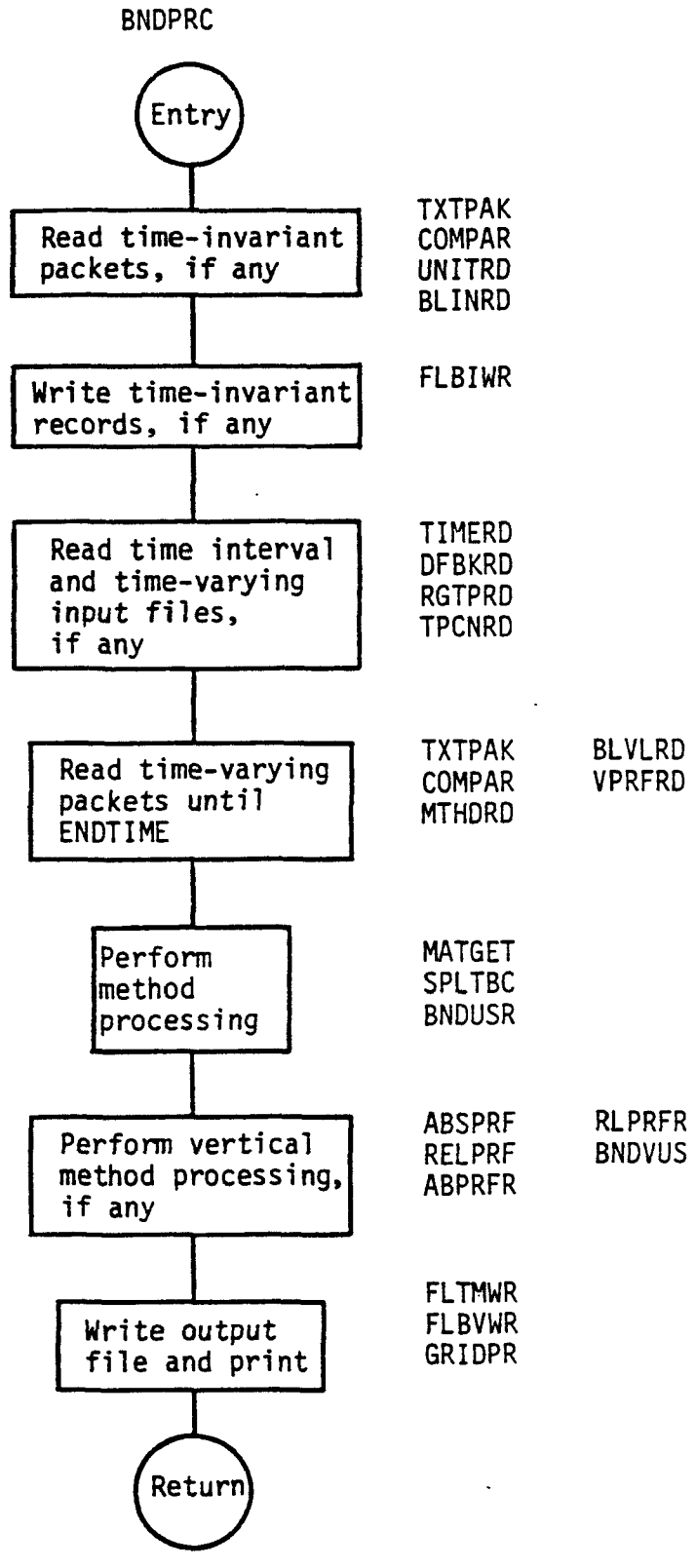
Table IV-4. Routines Specific to AIRQUALITY File Preparation

Name	Description	Routines Called
AIRPRC	Processing routine	See Figure IV-2(b)
AIRQUL	Main program	See Figure IV-2(a)
AIRUSR	USER method dummy subroutine	
AIRVUS	VERTUSER vertical method dummy subroutine	



(a) Main Program

Figure IV-3. Flow of Control for BOUNDARY File Preparation Program



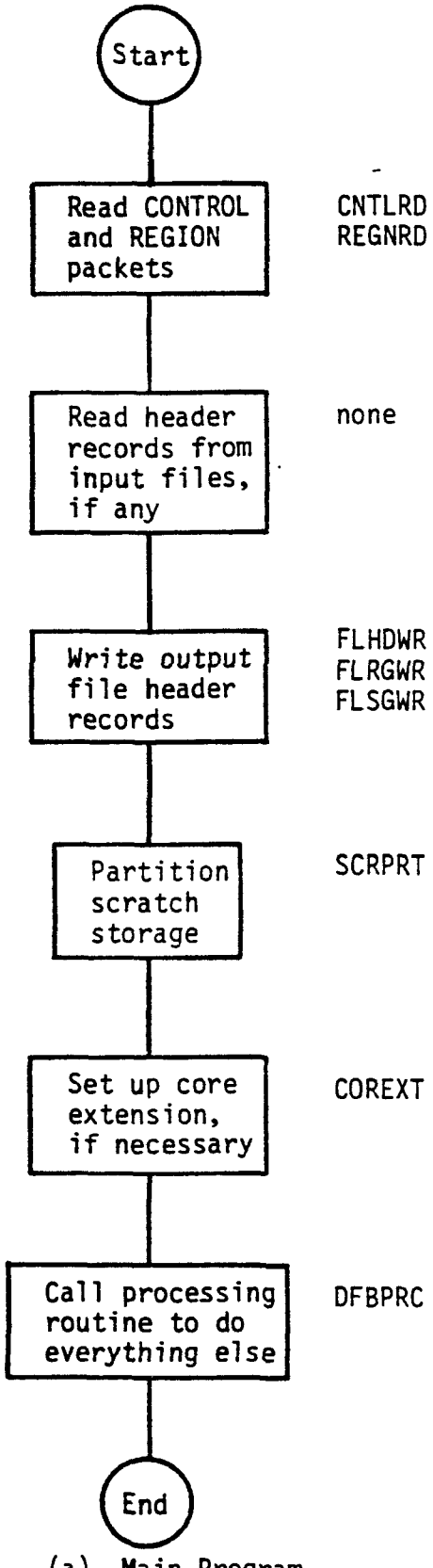
(b) Processing Routine

Figure IV-3 (Concluded)

Table IV-5. Routines Specific to BOUNDARY File Preparation

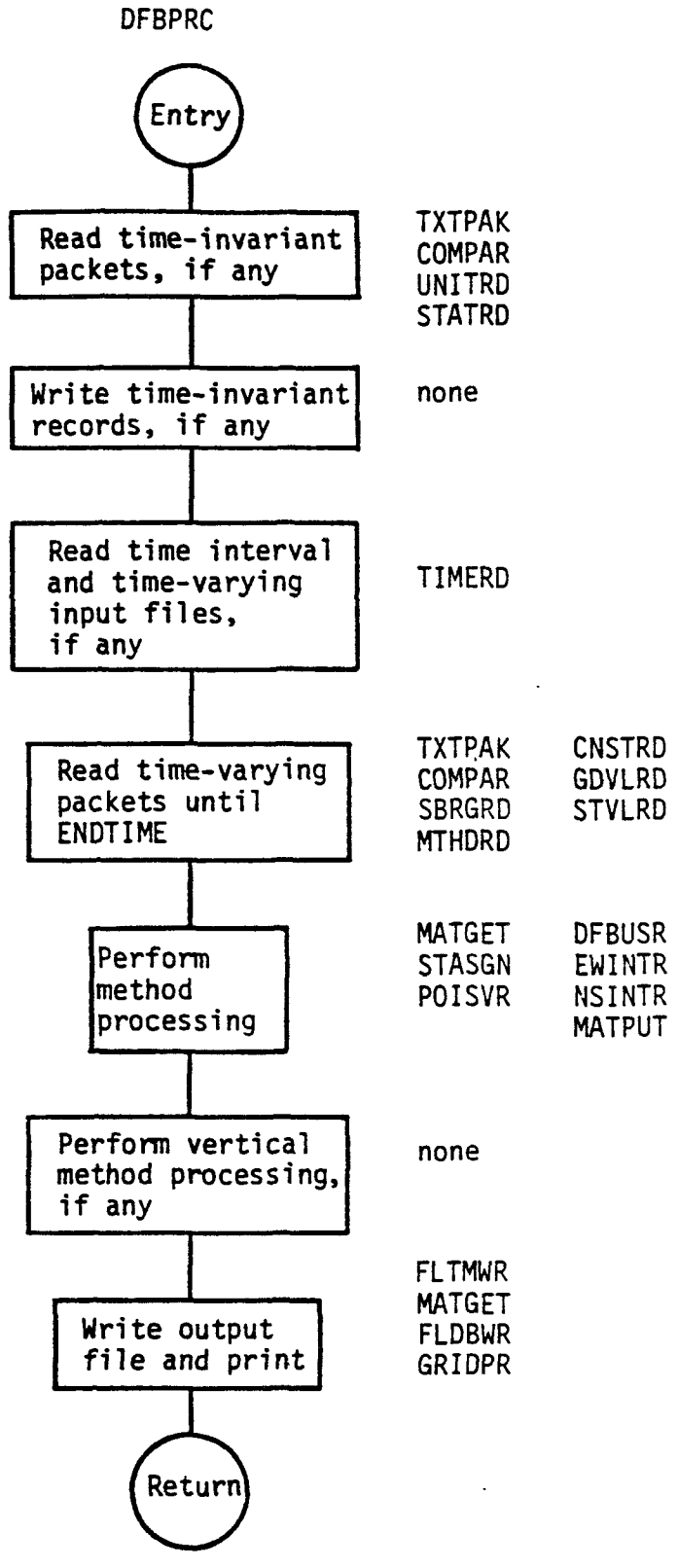
Name	Description	Routines Called
BLINRD	Reads BOUNDARIES packet and sets boundary cells	COMPAR TXTPAK UNTCVN SINVAL GRIDPR
BLVLRD	Reads BOUNDARY READINGS or CONSTANTS packet	TXTPAK COMPAR UNTCNV
BNDARY	Main program	See Figure IV-3(a)
BNDPRC	Processing routine	See Figure IV-3(b)
BNDUSR	USER method dummy subroutine	
BNDVUS	VERTUSER vertical method dummy subroutine	
SPLTBC	Performs split/combination method on boundary concentrations	

DFSNBK



(a) Main Program

Figure IV-4. Flow of Control for DIFFBREAK File Preparation Program



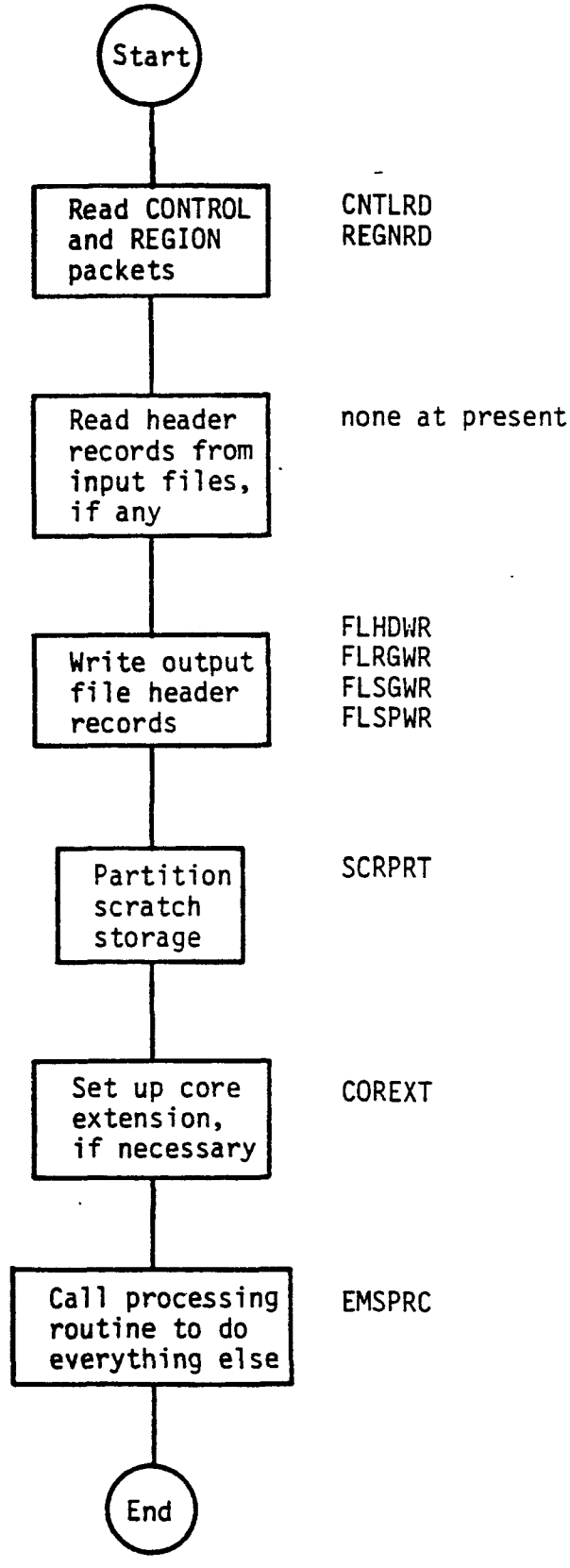
(b) Processing Routine

Figure IV-4 (Concluded)

Table VI-6. Routines Specific to DIFFBREAK File Preparation

Name	Description	Routines Called
DFBPRC	Processing routine	See Figure IV-4(b)
DFBUSR	USER method dummy subroutine	
DFSNBK	Main program	See Figure IV-4(a)

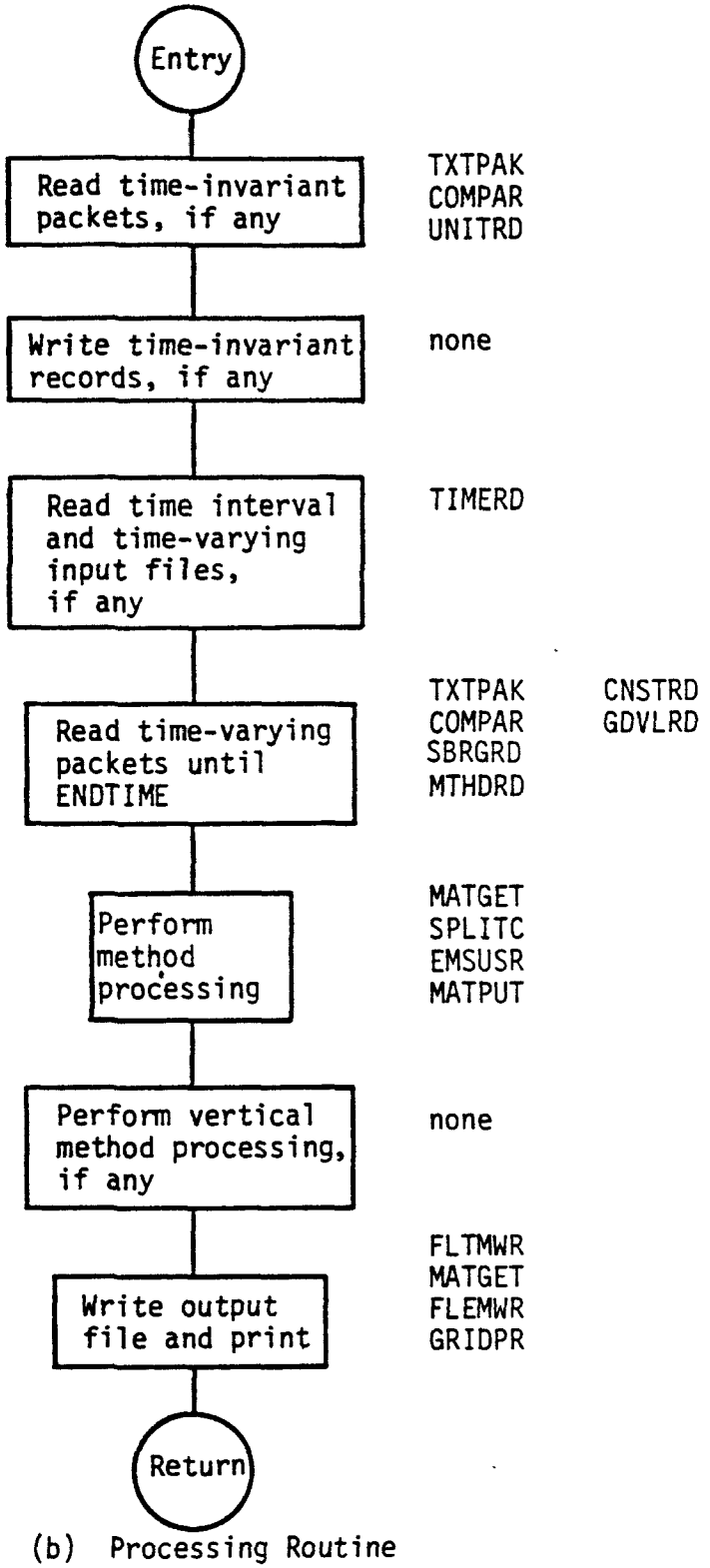
EMSPRC



(a) Main Program

Figure IV-5. Flow of Control for EMISSIONS File Preparation Program

EMSPRC



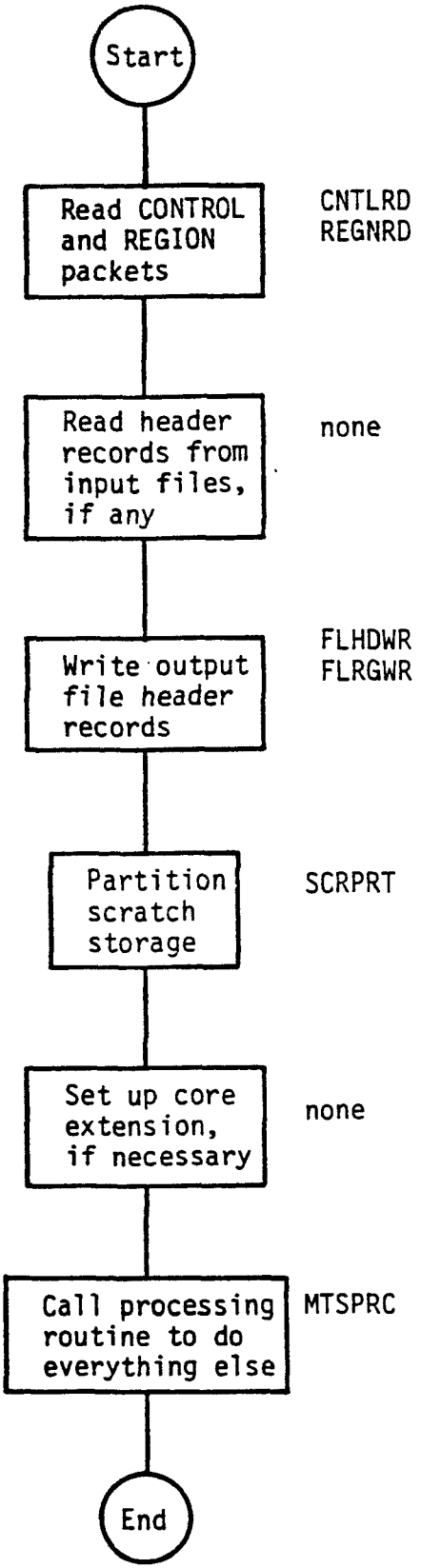
(b) Processing Routine

Figure IV-5 (Concluded)

Table IV-7. Routines Specific to EMISSIONS File Preparation

Name	Description	Routines Called
EMISSN	Main program	See Figure IV-5(a)
EMSPRC	Processing routine	See Figure IV-5(b)
EMSUSR	USER method dummy subroutine	

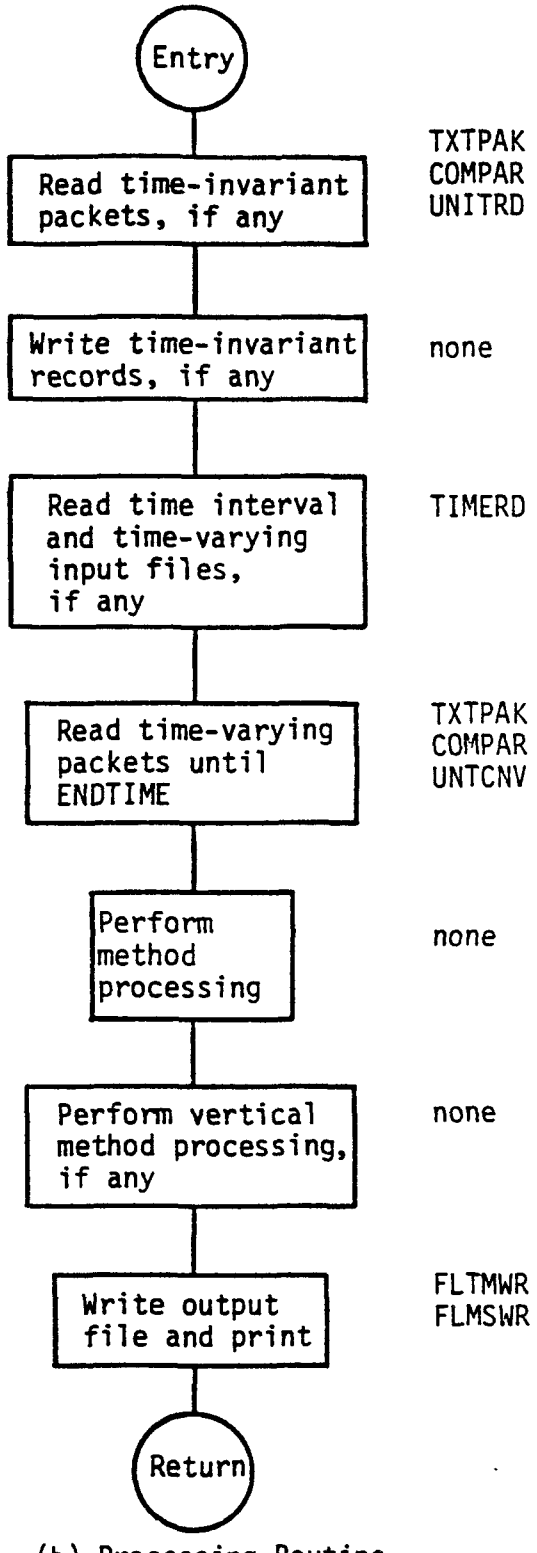
METSCL



(a) Main Program

Figure IV-6. Flow of Control for METSCALARS File Preparation Program

MTSPRC

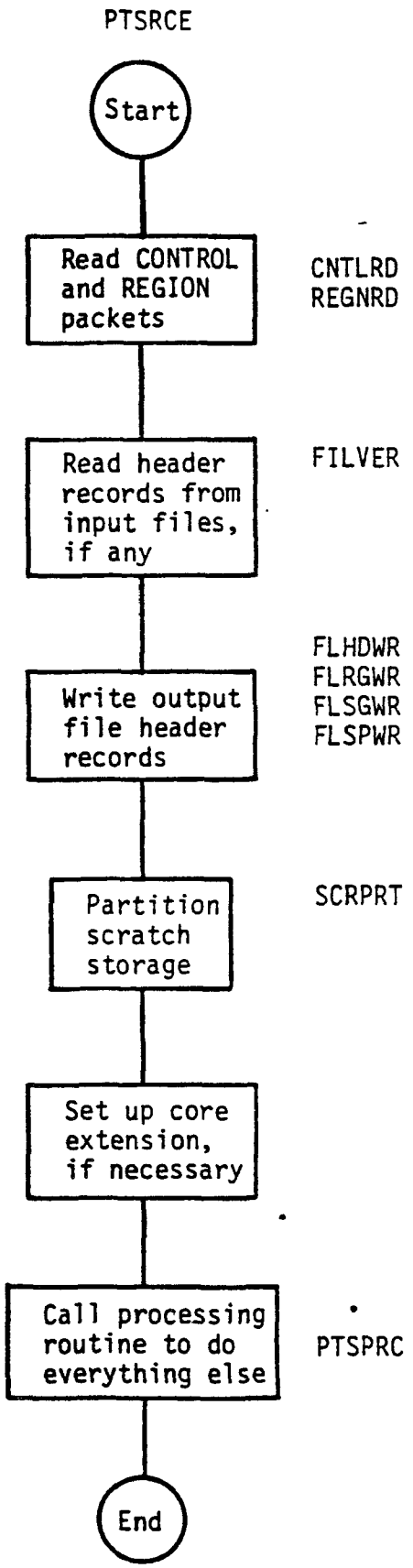


(b) Processing Routine

Figure IV-6 (Concluded)

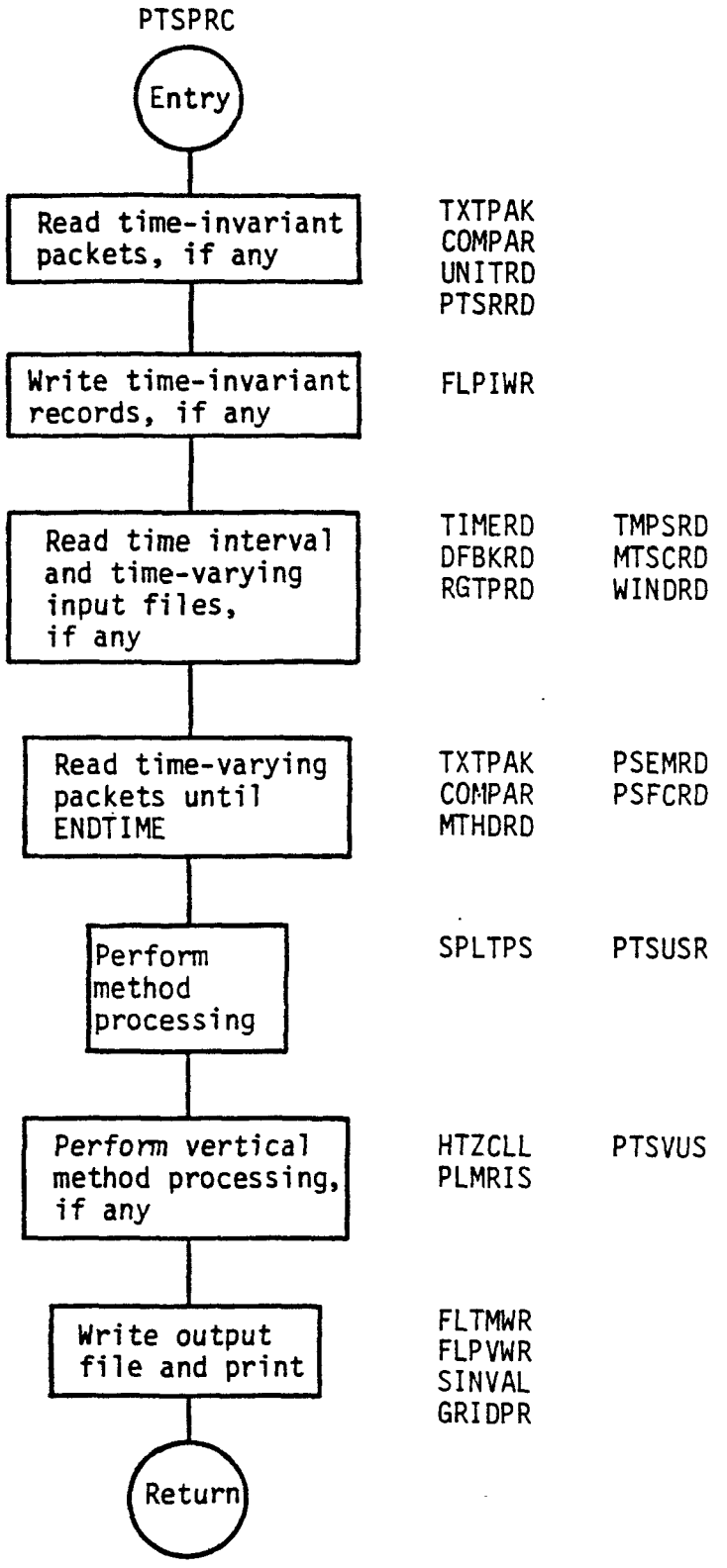
Table IV-8. Routines Specific to METSCALARS File Preparation

Name	Description	Routines Called
METSCL	Main program	See Figure IV-6(a)
MTSPRC	Processing routine	See Figure IV-6(b)



(a) Main Program

Figure IV-7. Flow of Control for PTSOURCE File Preparation Program

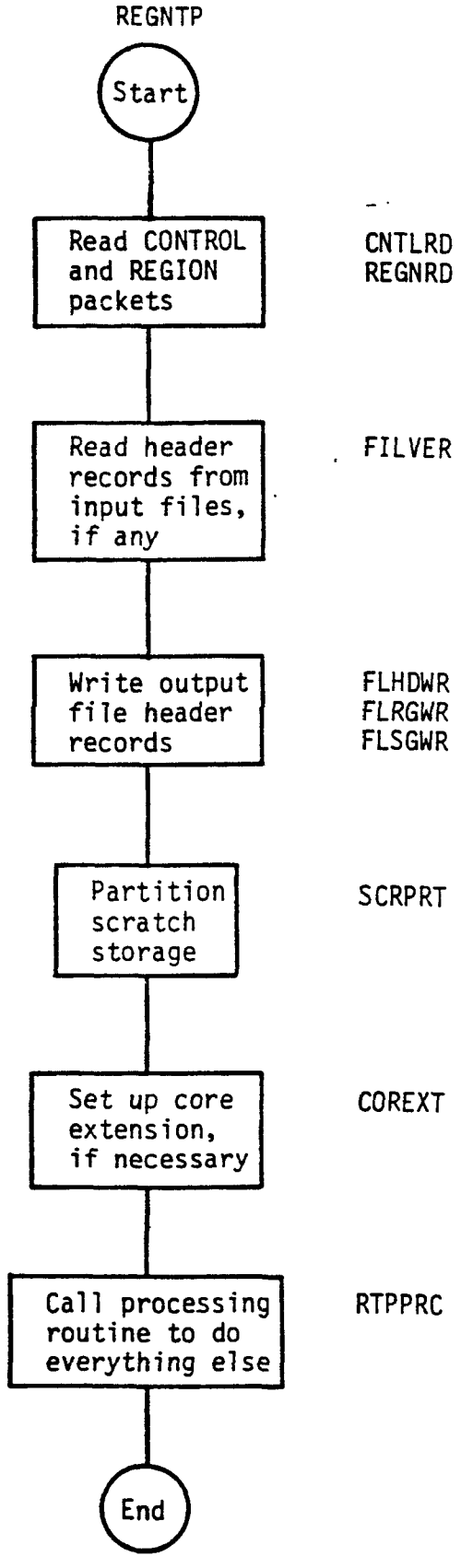


(b) Processing Routine

Figure IV-7 (Concluded)

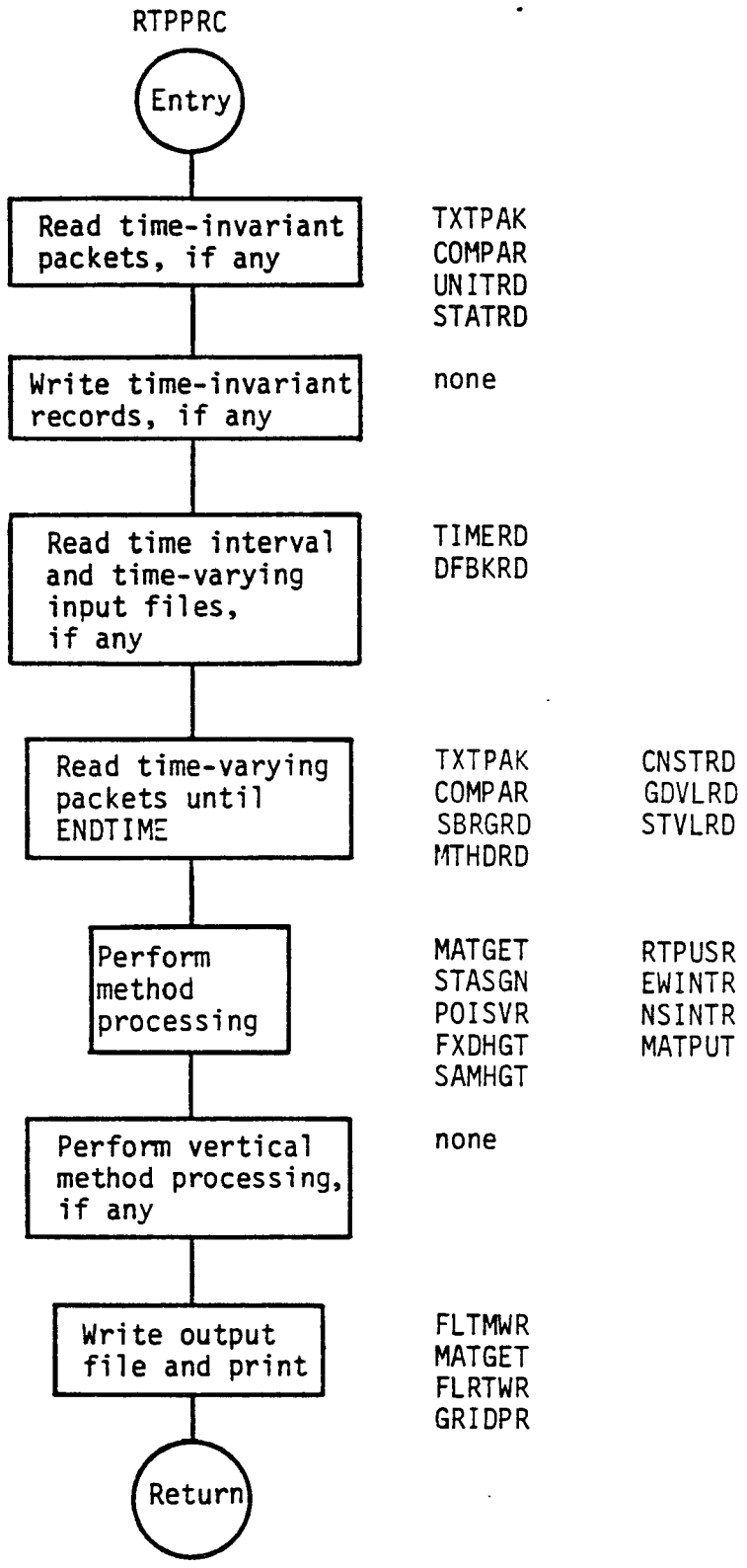
Table VI-9. Routines Specific to PTSOURCE File Preparation

Name	Description	Routines Called
PLMRIS	Calculates plume rise using Briggs formulae	
PSEMRD	Reads EMISSIONS VALUES packet	TXTPAK COMPAR UNTCNV
PSFCRD	Reads EMISSIONS FACTORS packet	TXTPAK COMPAR
PTSPRC	Processing routine	See Figure IV-7(b)
PTSRCE	Main program	See Figure IV-7(a)
PTSRRD	Reads POINT SOURCES packet	TXTPAK COMPAR UNTCNV
PTSVUS	VERTUSER vertical method dummy subroutine	
SPLTPS	Performs split/combination method on point source emissions	COMPAR



(a) Main Program

Figure IV-8. Flow of Control for REGIONTOP File Preparation Program

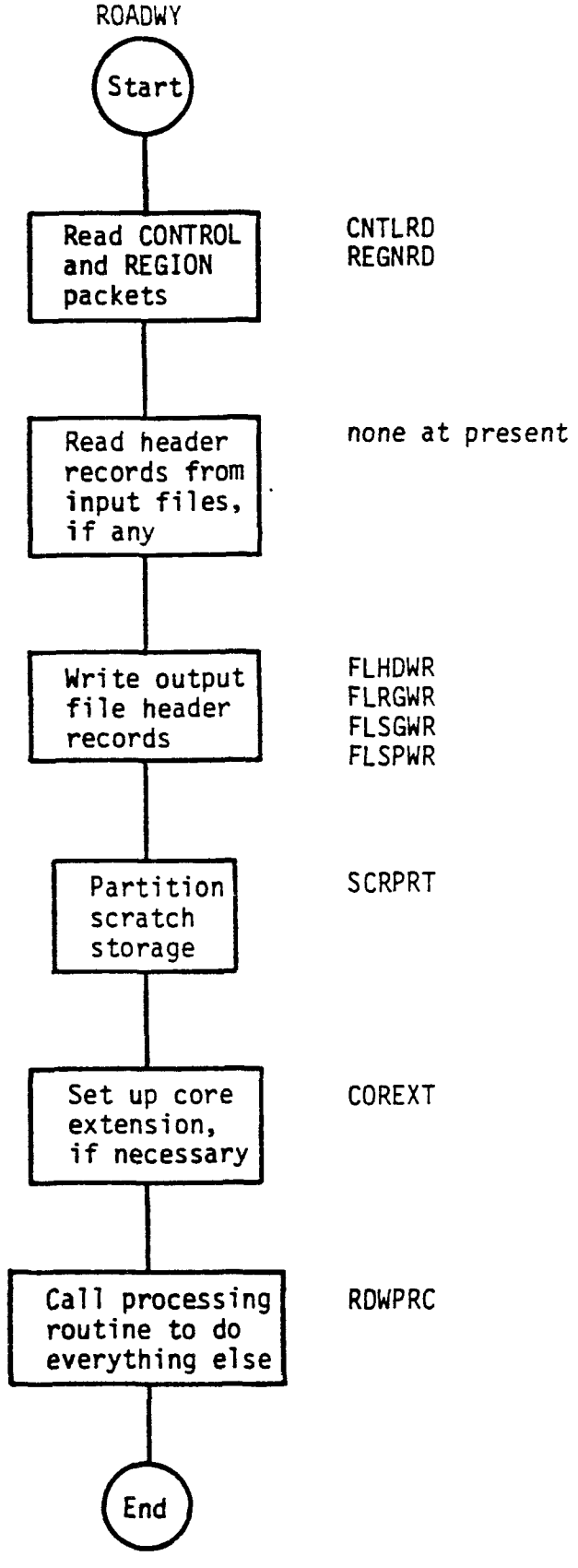


(b) Processing Routine

Figure IV-8 (Concluded)

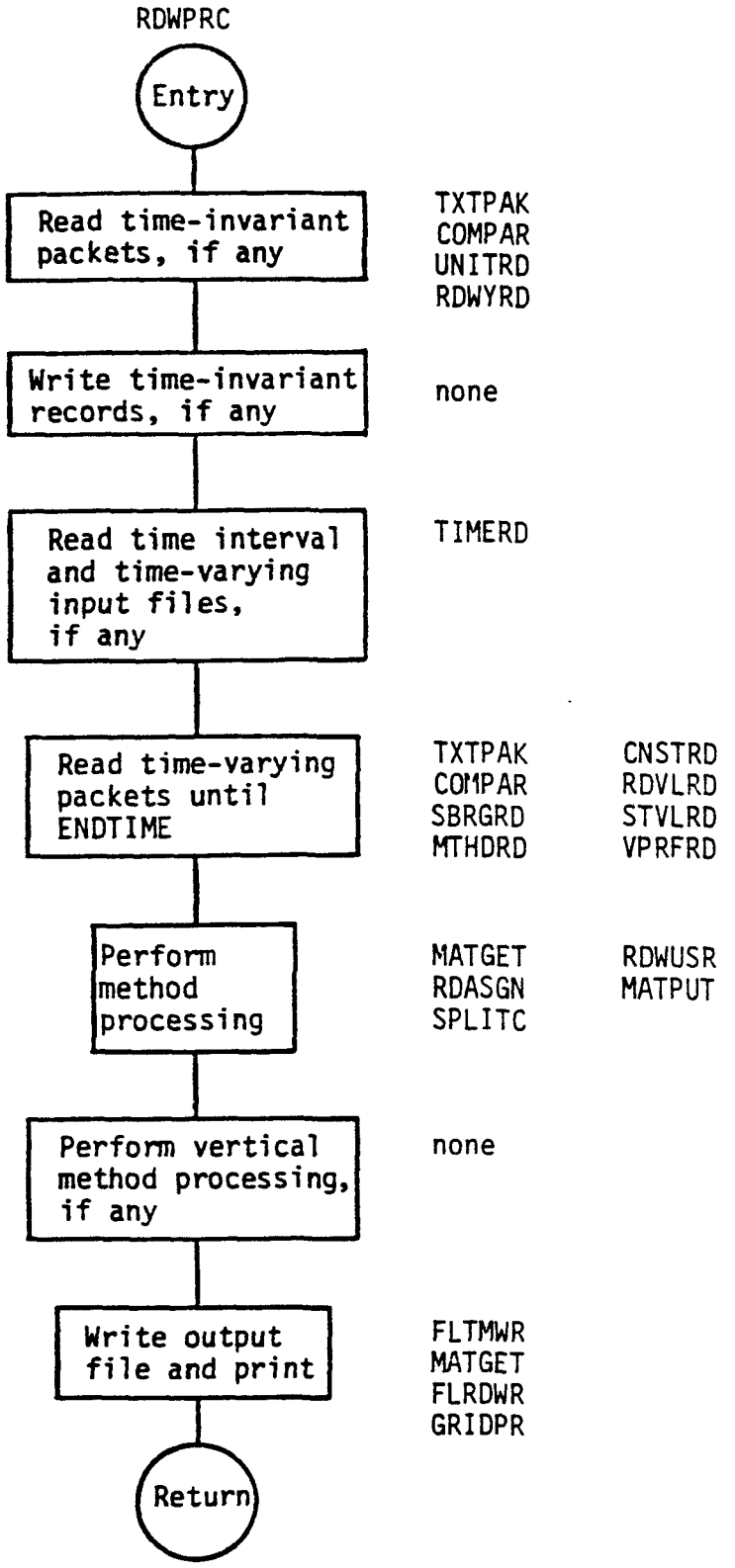
Table IV-10. Routines Specific to REGIONTOP File Preparation

Name	Description	Routines Called
FXDHGT	Performs FIXDHEIGHT method	
REGNTP	Main program	See Figure IV-8(a)
RTPPRC	Processing routine	See Figure IV-8(b)
RTPUSR	USER method dummy subroutine	
SAMHGT	Performs SAMEHEIGHT method	



(a) Main Program

Figure IV-9. Flow of Control for ROADWAY File Preparation Program

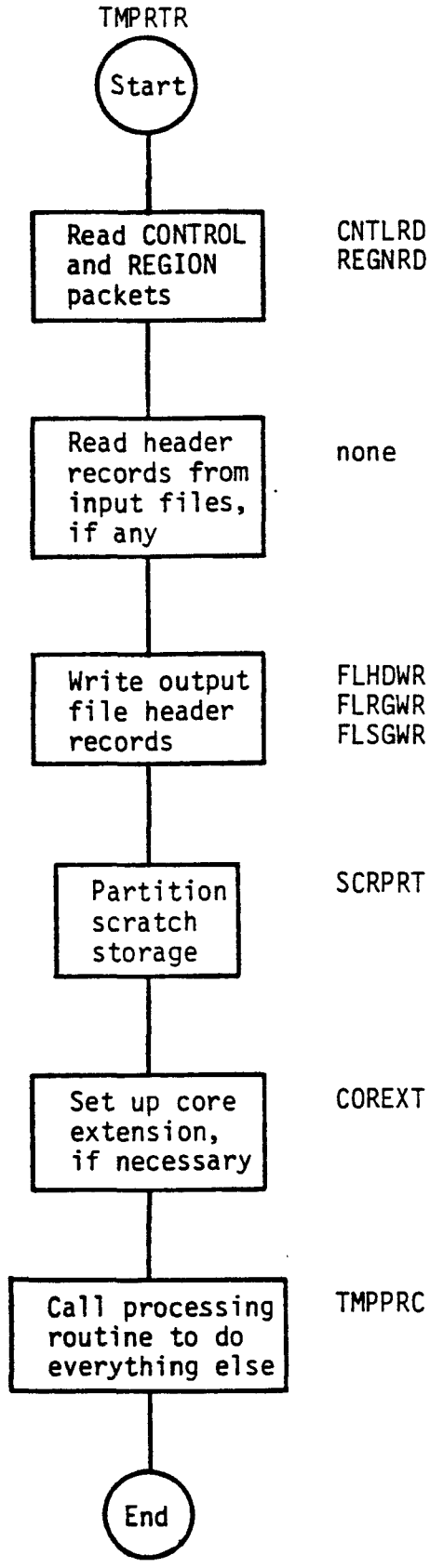


(b) Processing Routine

Figure IV-9 (Concluded)

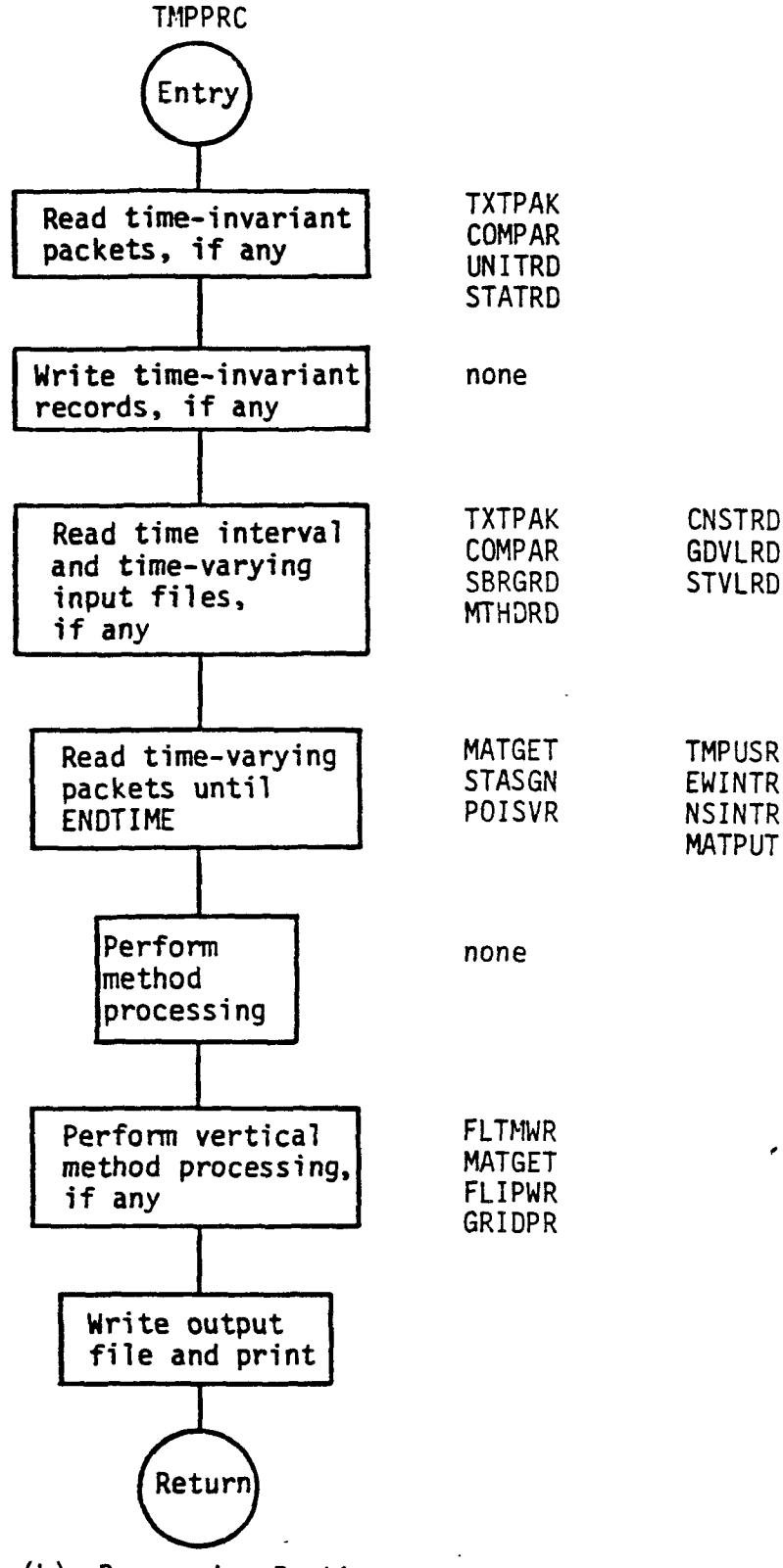
Table IV-11. Routines Specific to ROADWAY File Preparation

Name	Description	Routines Called
RDASGN	Assigns roadway values (dummy)	
RDVLRD	Reads ROADWAY VALUES packet (dummy)	
RDWPRC	Processing routine	See Figure IV-9(b)
RDWUSR	USER method dummy subroutine	
RDWYRD	Reads ROADWAY DEFINITION packet (dummy)	
ROADWY	Main program	See Figure IV-9(a)



(a) Main Program

Figure IV-10. Flow of Control for TEMPERATUR File Preparation Program

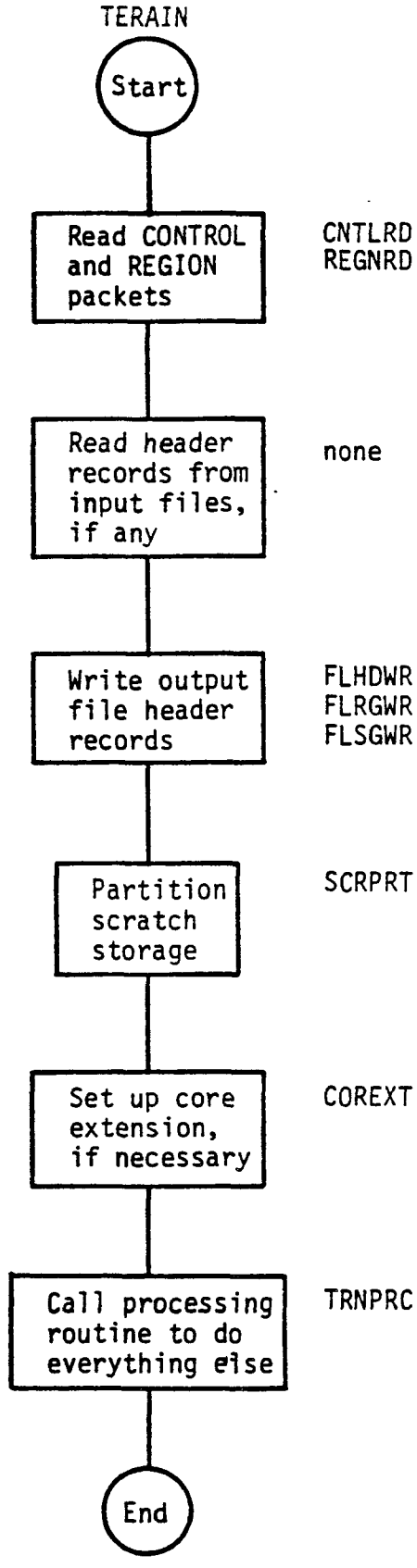


(b) Processing Routine

Figure IV-10 (Concluded)

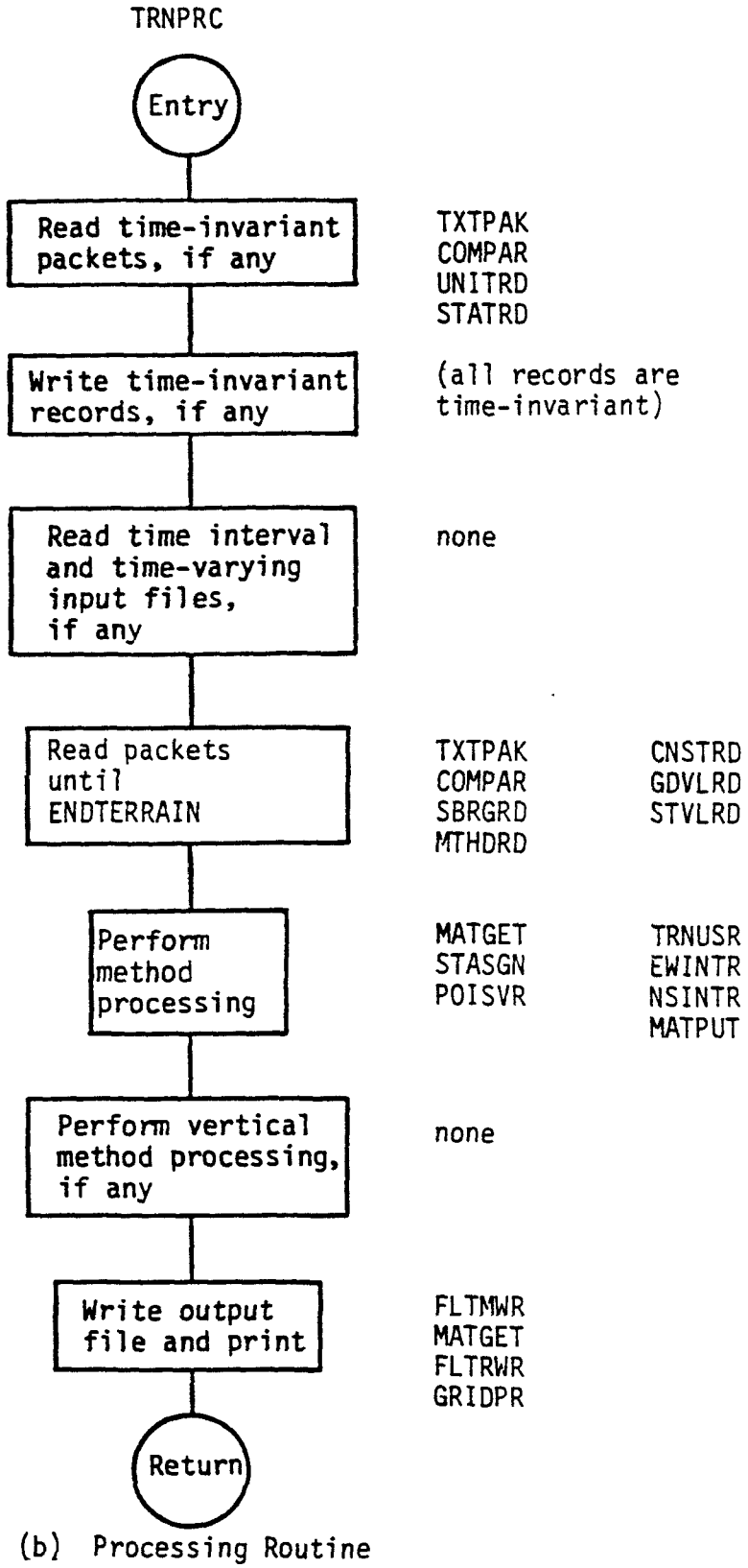
Table IV-12. Routines Specific to TEMPERATUR File Preparation

Name	Description	Routines Called
TMPPRC	Processing routine	See Figure IV-10(b)
TMPRTR	Main program	See Figure IV-10(a)
TMPUSR	USER method dummy subroutine	



(a) Main Program

Figure IV-11. Flow of Control For TERRAIN File Preparation Program

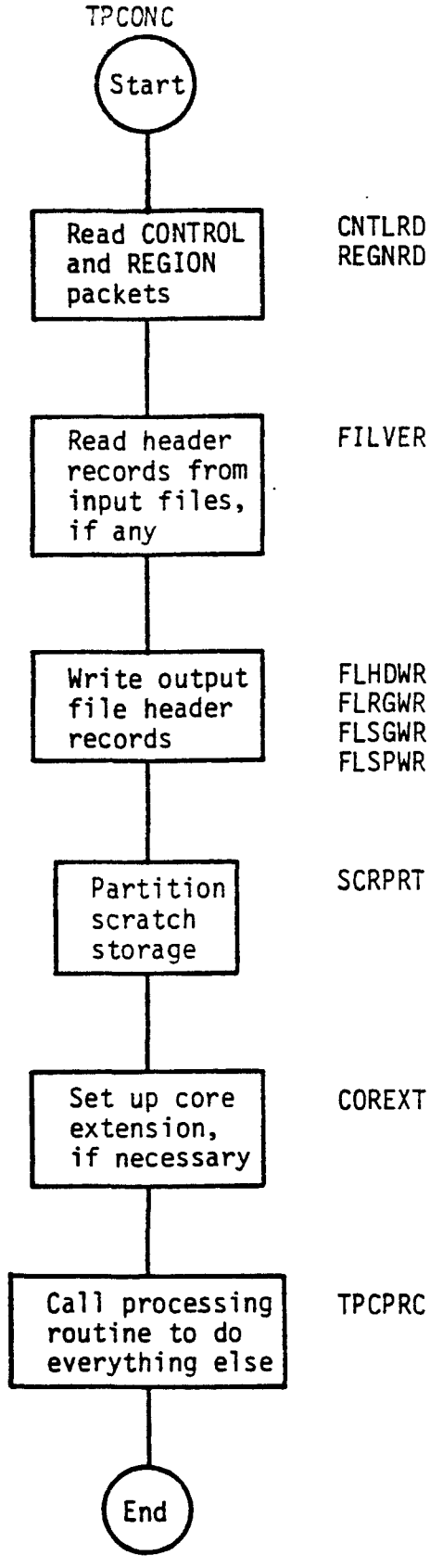


(b) Processing Routine

Figure IV-11 (Concluded)

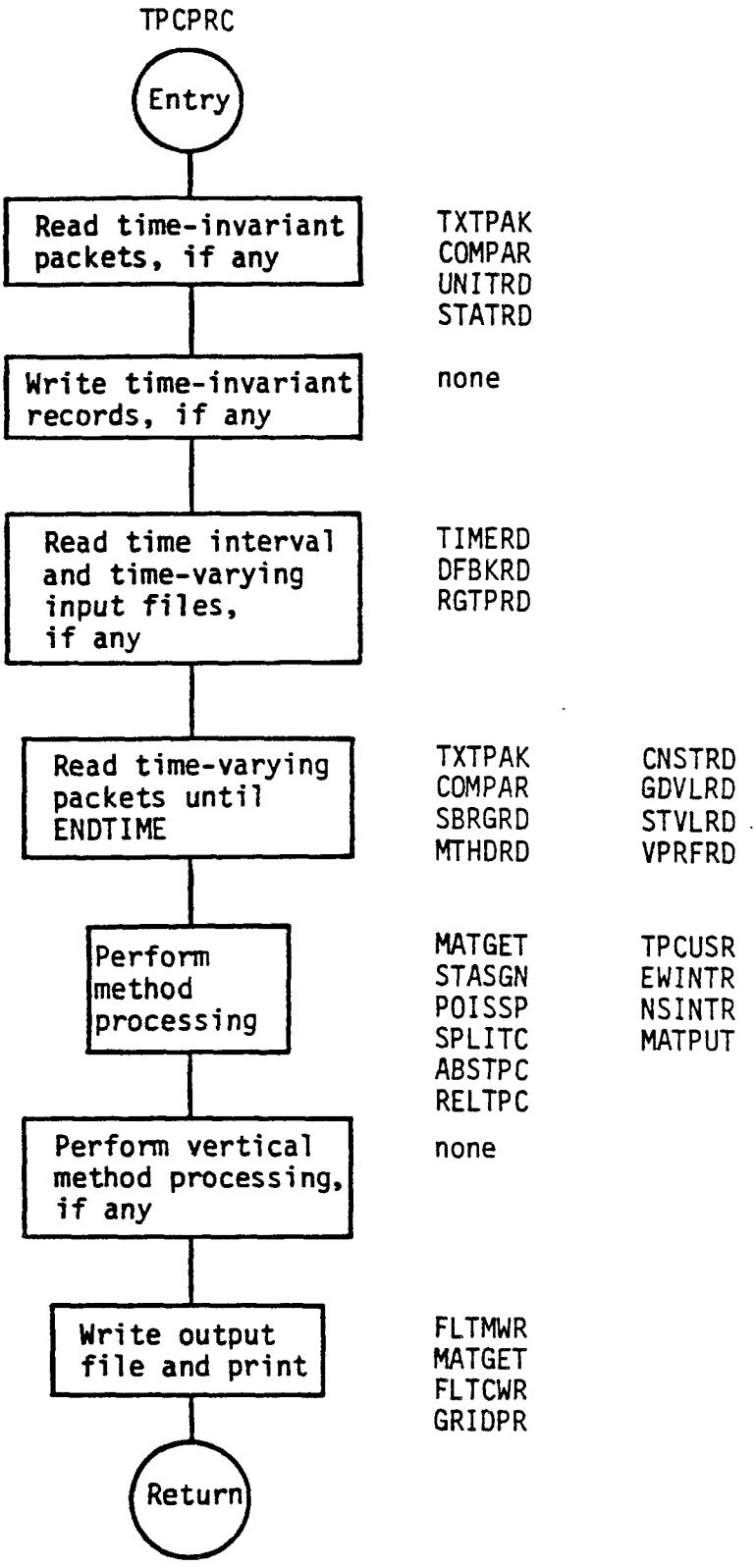
Table IV-13. Routines Specific to TERRAIN File Preparation

Name	Description	Routines Called
TERAIN	Main program	See Figure IV-11(a)
TRNPRC	Processing routine	See Figure IV-11(b)
TRNUSR	USER method dummy subroutine	



(a) Main Program

Figure IV-12. Flow of Control for TOPCONC File Preparation Program

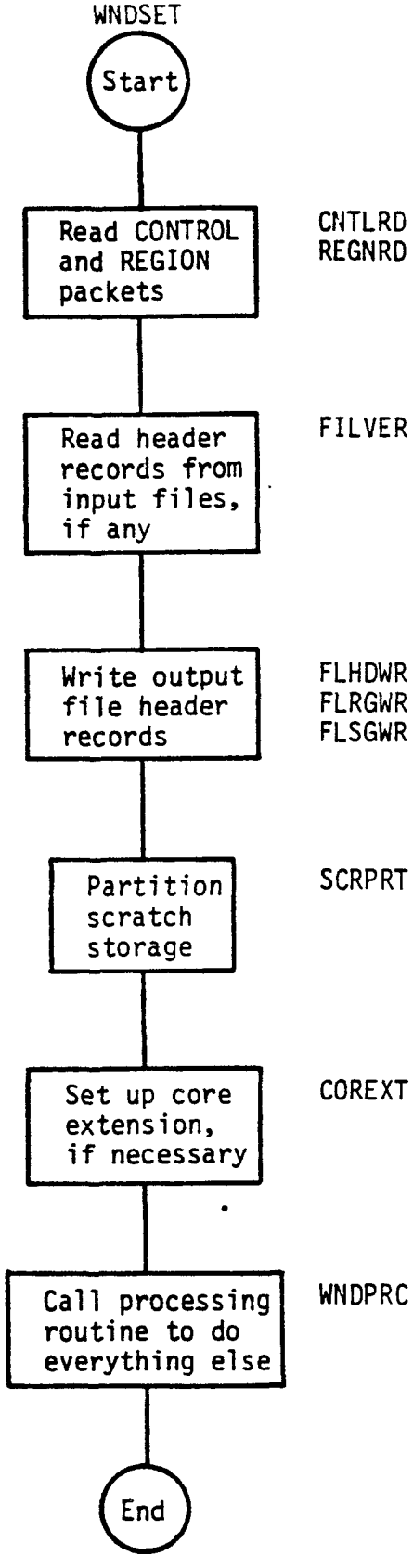


(b) Processing Routine

Figure IV-12 (Concluded)

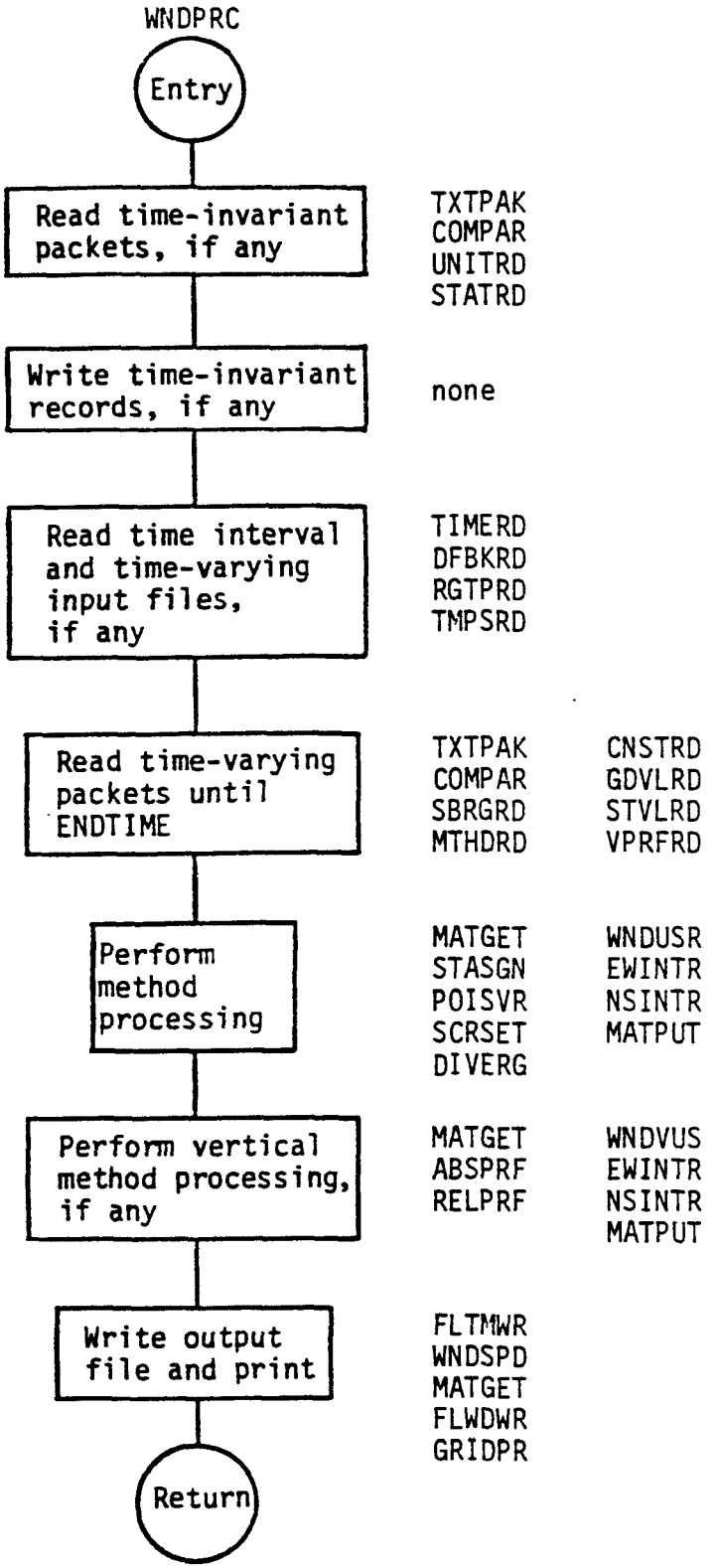
Table IV-14. Routines Specific to TOPCONC File Preparation

Name	Description	Routines Called
ABSTPC	Performs ABSTOPCONC method	
RELTPC	Performs RELTOPCONC method	
TPCONC	Main program	See Figure IV-12(a)
TPCPRC	Processing routine	See Figure IV-12(b)
TPCUSR	USER method dummy subroutine	



(a) Main Program

Figure IV-13. Flow of Control for WIND File Preparation Program



(b) Processing Routine

Figure IV-13 (Concluded)

Table IV-15. Routines Specific to WIND File Preparation

Name	Description	Routines Called
DIVERG	Performs DIVERGENCE method	CELLHT
		POISVR
		MATPUT
		MATGET
WNDPRC	Processing routine	See Figure IV-13(b)
WNDSET	Main program	See Figure IV-13(a)
WNDUSR	USER method dummy subroutine	
WNDVUS	VERTUSER vertical method dummy subroutine	

D. IMPLEMENTING THE PROGRAMS

1. Source Deck Structure

The source version of each program and each of the libraries consists of a number of "decks", each of which is a main program or subroutine and is preceded by an @ELT, I card.

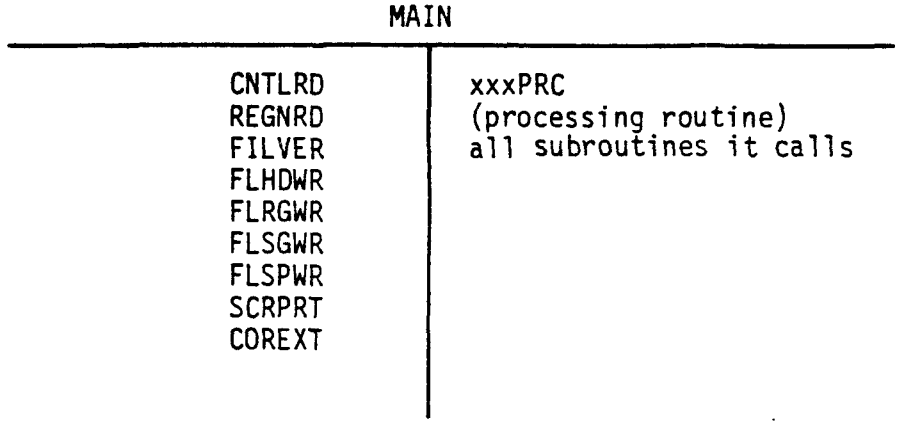
2. Scratch Vector Length

Each program contains a scratch vector JSCRTC; its length is fixed in a DIMENSION statement in the main program and is also assigned to the variable NSCRTC in a DATA statement. For efficient execution of the program, the scratch vector should be as long as possible. The length currently compiled into each program is a function of the EPA UNIVAC computer system. After the program has been recompiled on a new computer and loaded along with the library subroutines it requires, the load map should be examined to determine how much space is actually available for the scratch vector. The DIMENSION and DATA statements should then be changed and the main program recompiled.

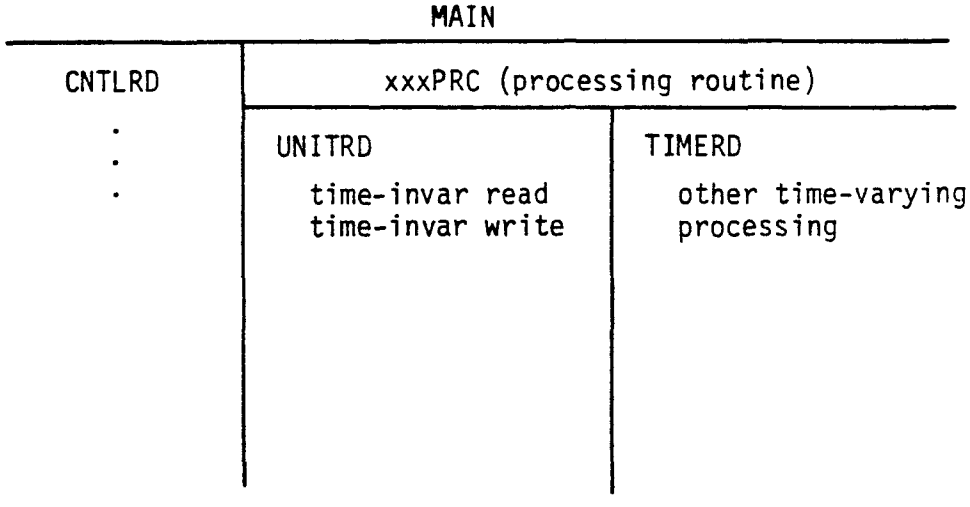
3. Overlaying

Currently, no overlay structure is built into the data preparation programs. Since there is no provision for COMMON, and all information is passed through argument lists, it is impossible to construct independent overlays as required by the LBL system (see Section VII-G). On the UNIVAC system, however, an overlay consists merely of subroutines called from a higher level overlay segment; with this less stringent requirement overlay schemes become possible.

The simplest overlay scheme is illustrated in Figure IV-14(a). In this figure, the main segment contains the main program; the first overlay segment contains routines called by the main program before the processing routine, and the second overlay segment contains the processing routine itself.



(a) For Simple Overlaying



(b) For Further Overlaying

Figure IV-14. Overlay Structure for Data Preparation Programs

A greater savings in core space can be accomplished by further subdividing the second overlay segment (processing routine) as shown in Figure IV-14(b). Here, the processing of time-invariant data is in one lower-level overlay segment and the processing of time-varying data is in another.

In all cases, routines used by more than one parallel overlay segment should be assigned to higher-level segments. For example, TXTPAK and COMPAR, used by all packet-reading subroutines, should be placed in the main overlay segment.

4. I/O Error Handling

The subroutines in the library FILUTIL contain provisions for setting flags and retaining control if errors are detectable. These routines can be modified to include end-of-file and error checks if these are available.

5. Core Extension

The most machine-dependent feature of the data preparation programs is core extension, which at present involves the execution-time assignment of random-access files. This section discusses each of the core extension routines in some detail, and suggests alternatives for other implementations.

a. COREXT

COREXT controls the assignment of three-and four-dimensional arrays. It assigns two two-dimensional arrays for working storage and then checks as to whether the entire array will fit in the remainder of the available scratch vector space. If the array will fit, it is assigned to that space. If it will not fit, COREXT allocates space for an index table and calls MATSET to set up the random-access file. It then calls MATPUT to initialize the data in the file and returns. COREXT itself contains no machine-dependent features other than the passing of buffers and an index table through arguments to lower level subroutines. (Note: MATSET has been bypassed for UNIVAC implementation. It may be useful for installation of the code on a CDC machine.)

b. MATSET

MATSET creates a FET (File Environment Table), causes large core buffers to be allocated for a new random-access file, and initializes an index table. Use of MATSET to perform these functions avoids the need to allocate files and buffers if they are not going to be used.

c. MATPUT AND MATGET

These routines transfer two-dimensional slices of arrays between the three- or four-dimensional arrays and the two-dimensional working storage. If the large array is in the scratch vector, a simple transfer of values is performed. If the large array is on a random-access file, MATPUT and MATGET call MATSTR or MATLOD, respectively, to write or read the data on the file. There is no machine-dependent code in MATPUT or MATGET.

d. MATSTR and MATLOD

These routines are called to store data to or to load data from a random-access file. Each is passed the appropriate index table entry and the length of the record, and each calls a system routine that performs the store or load. (These are dummy routines on the UNIVAC version of the code.)

Implementing the core extension feature on another machine necessitates the establishment of a medium for storing the oversized arrays and the means for accessing them. However, if a large amount of core storage is available, it may not be necessary to implement this feature at all. Furthermore, if extended core memory is available, it, rather than a secondary storage file, can be used to reduce I/O accesses.

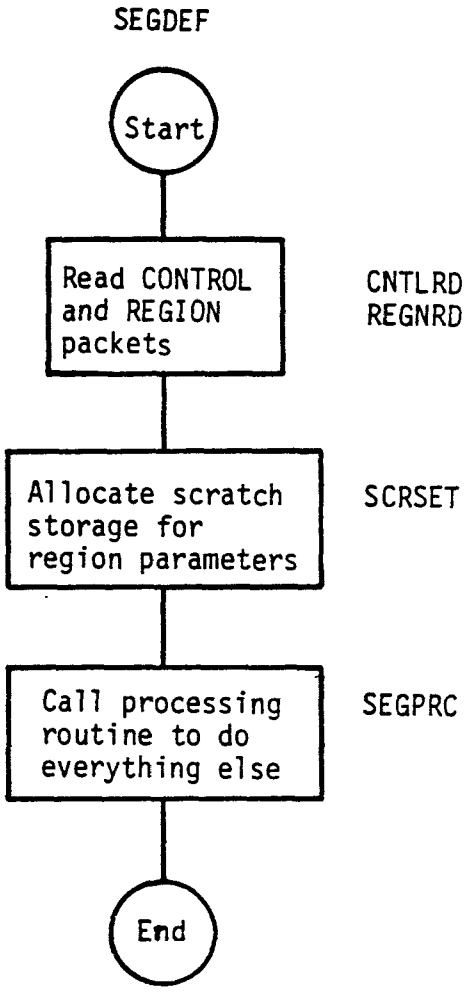
V M.E.B. FILE SEGMENTATION PROGRAM

One program--SEGDEF--is used to segment the 11 spatially varying M.E.B. data files. It is designed to segment any number of files in a single run, but because of certain computer requirements for small core file buffers, it has been necessary to restrict program usage to one file at a time.

The SEGDEF program contains many of the features described in Chapter IV: it uses routines from the same two libraries, UTILITY and FILUTIL; it uses the same scratch vector allocation scheme for variable data storage; it uses core extension if necessary for storage of three- and four-dimensional arrays.

This section describes the unique features of the file segmentation program. For information on utility routines, scratch allocation, core extension, and implementation suggestions, see the appropriate sections of Chapter IV.

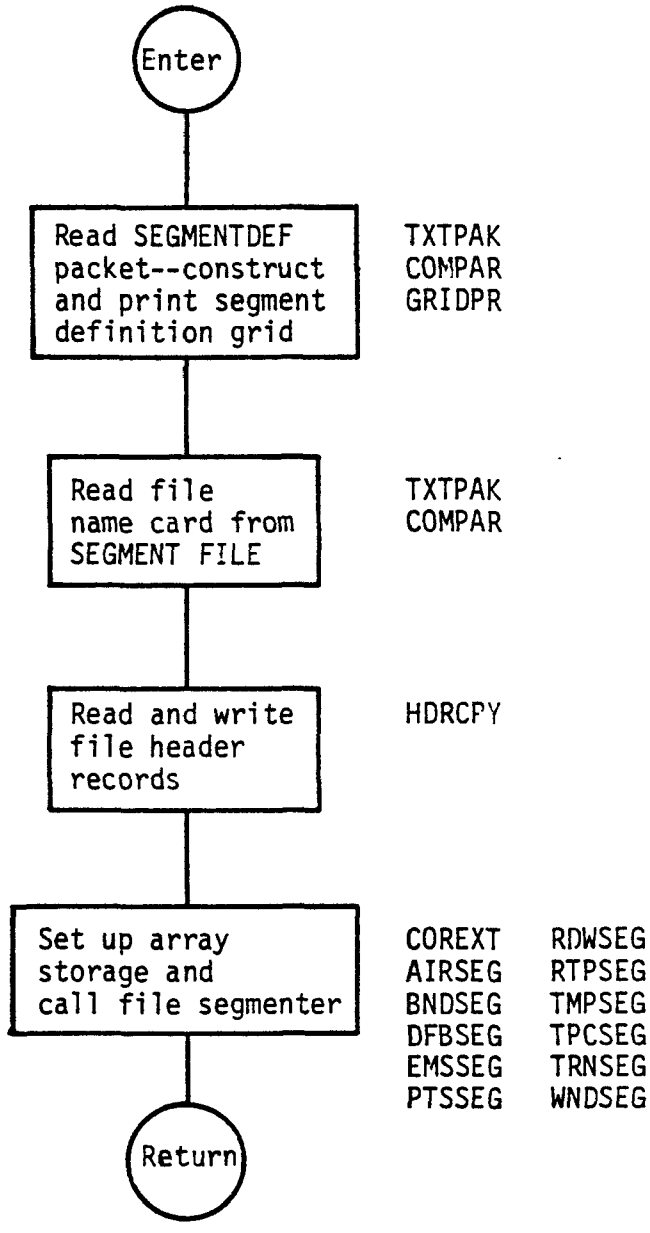
The flow of control in the program is shown in Figures V-1(a) and V-1(b). The main program reads the CONTROL and REGION packets, allocates scratch storage for the regional definition parameters, and then calls the processing routine, SEGPRC, to complete the file segmentation. SEGPRC reads the SEGMENT DEFINITION packet, assigns a segment number to each grid cell, and constructs the internal segment definition parameters -- adding one or two boundary cells as necessary on each edge. SEGPRC then reads the SEGMENT FILE packet and, for each file, calls COREXT to set up the array storage required and calls a file-specific segmentation subroutine. Figures V-2 through V-12 show the flow of control in each subroutine. Table V-1 lists all nonutility routines in the file segmentation program.



(a) Main Program

Figure V-1. Flow of Control for File Segmentation

SEGPRC



(b) Segmentation Processing Routine

Figure V-1 (Concluded)

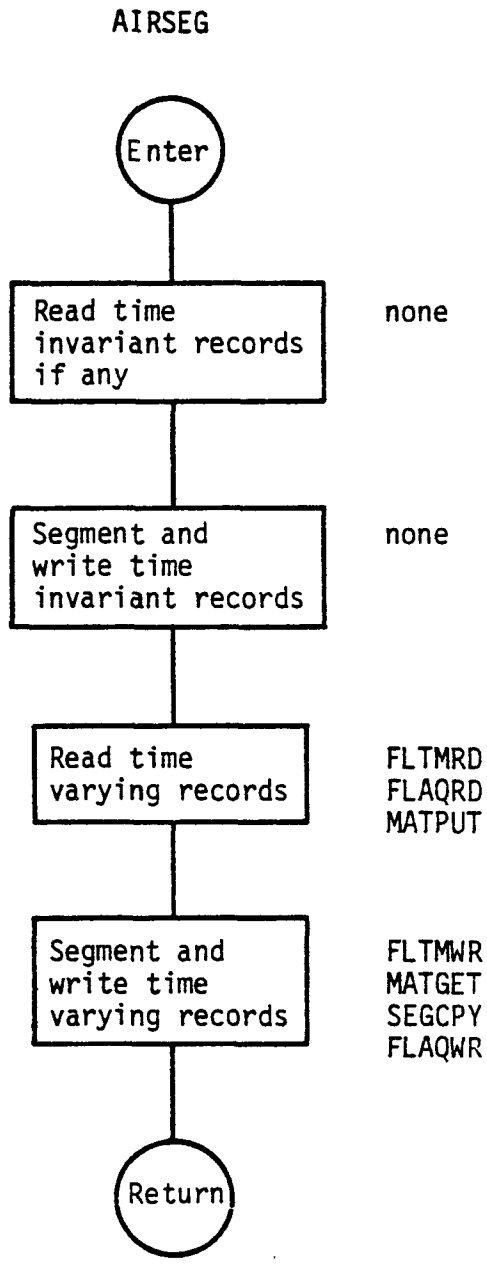


Figure V-2. Flow of Control--AIRQUALITY File Segmentation Subroutine

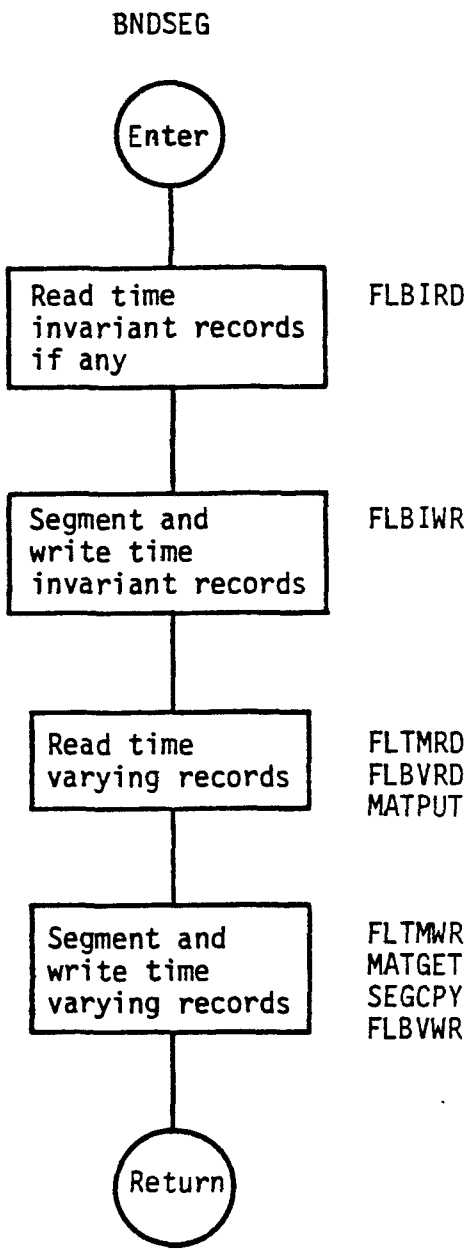


Figure V-3. Flow of Control--BOUNDARY File Segmentation Subroutine

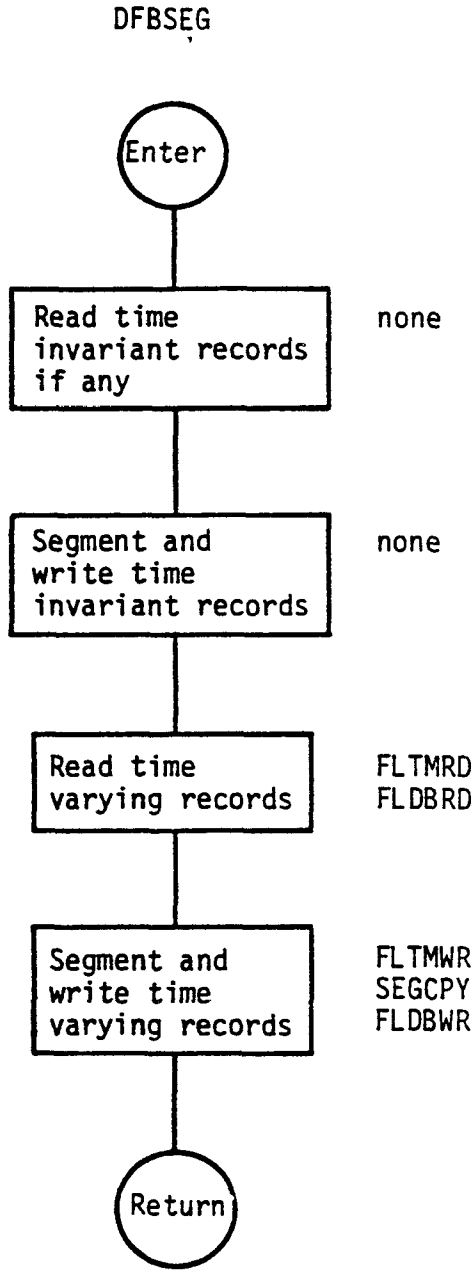


Figure V-4. Flow of Control--DIFFBREAK File Segmentation Subroutine

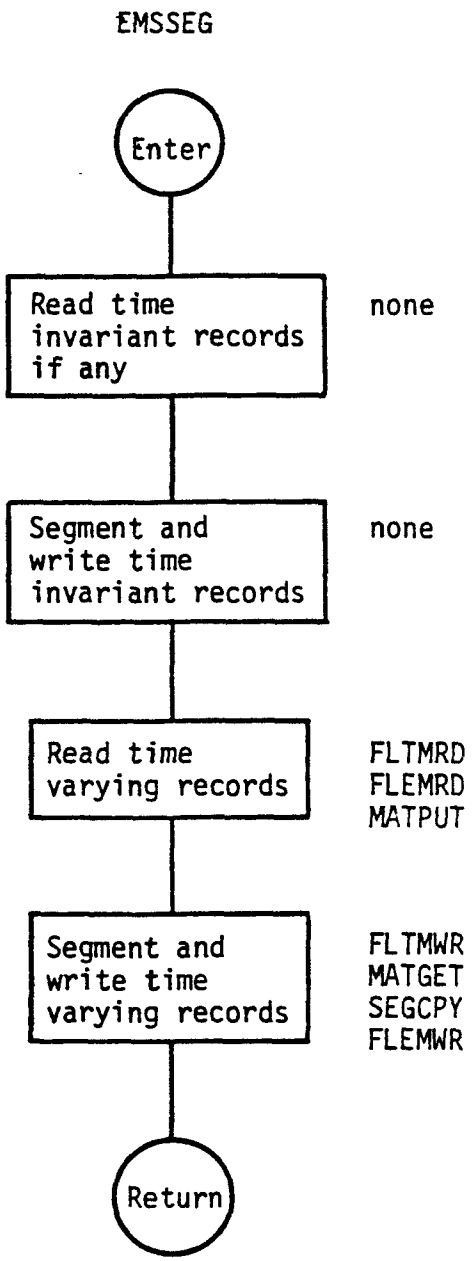


Figure V-5. Flow of Control--EMISSIONS File Segmentation Subroutine

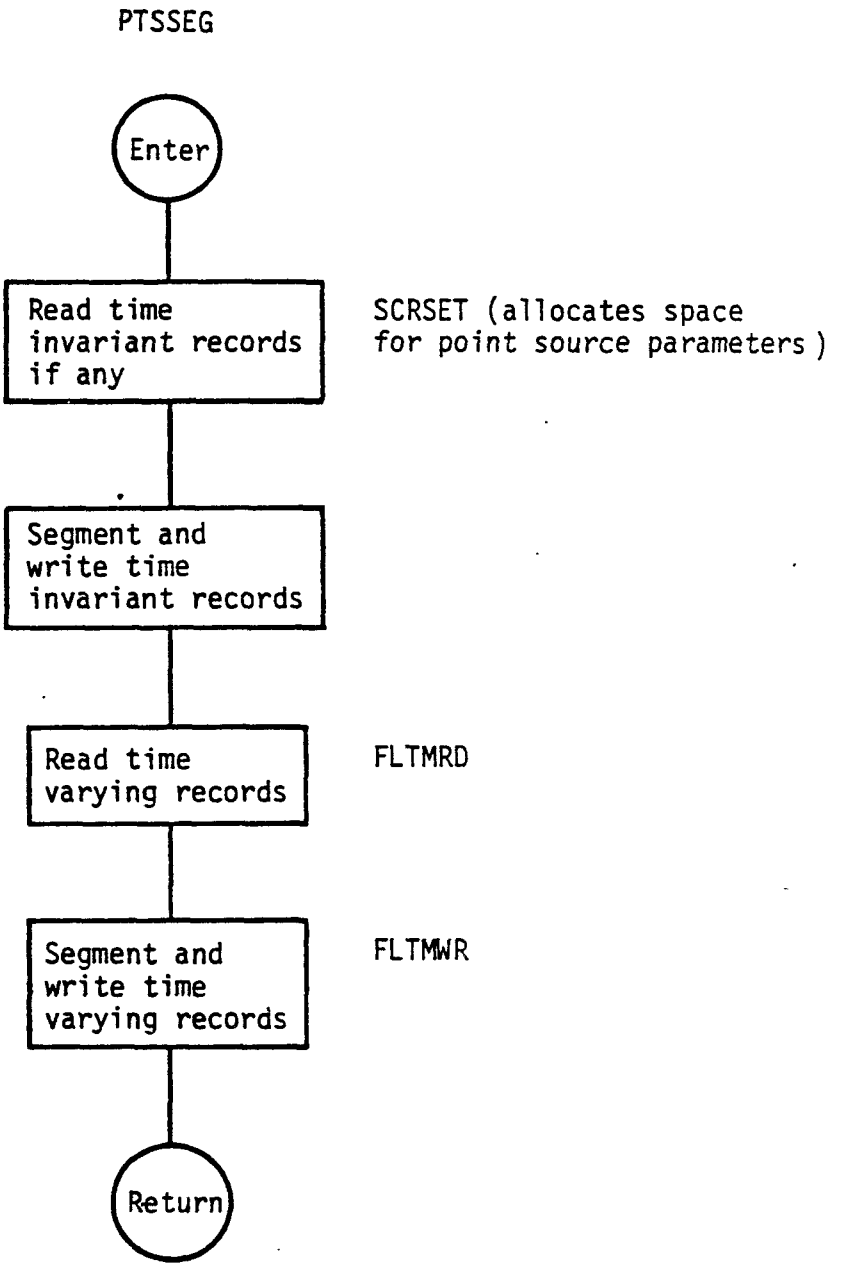


Figure V-6. Flow of Control--PTSOURCE File Segmentation Subroutine

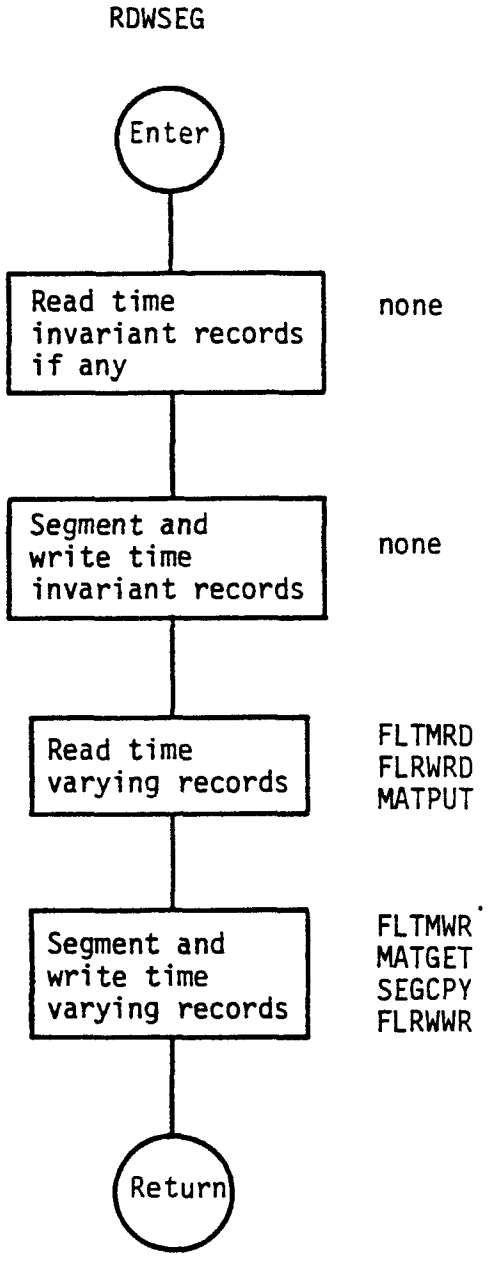


Figure V-7. Flow of Control--ROADWAY File Segmentation Subroutine

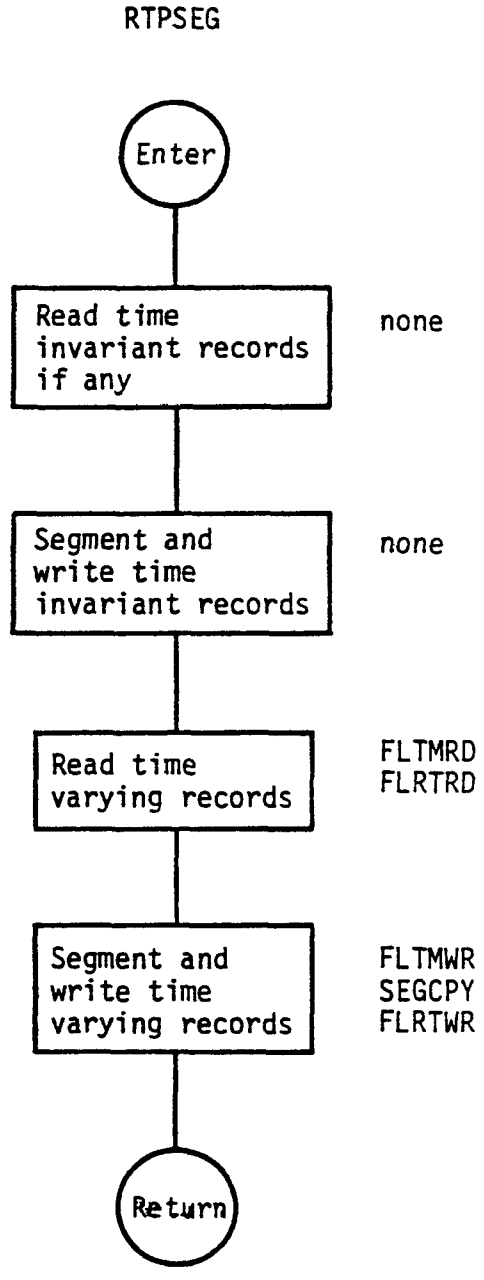


Figure V-8. Flow of Control--REGIONTOP File Segmentation Routine

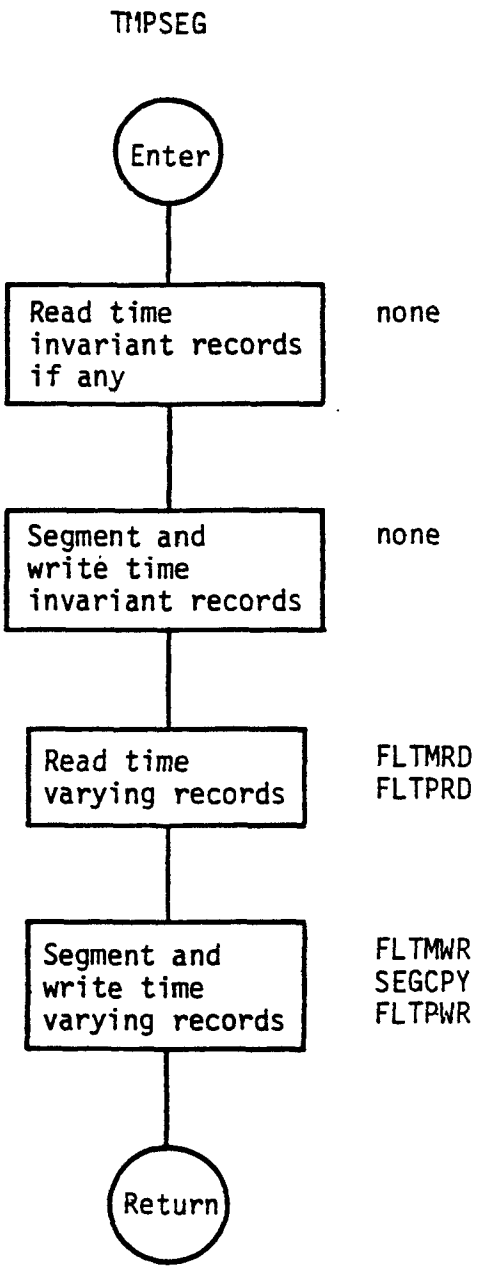


Figure V-9. Flow of Control--TEMPERATUR File Segmentation Subroutine

TPCSEG

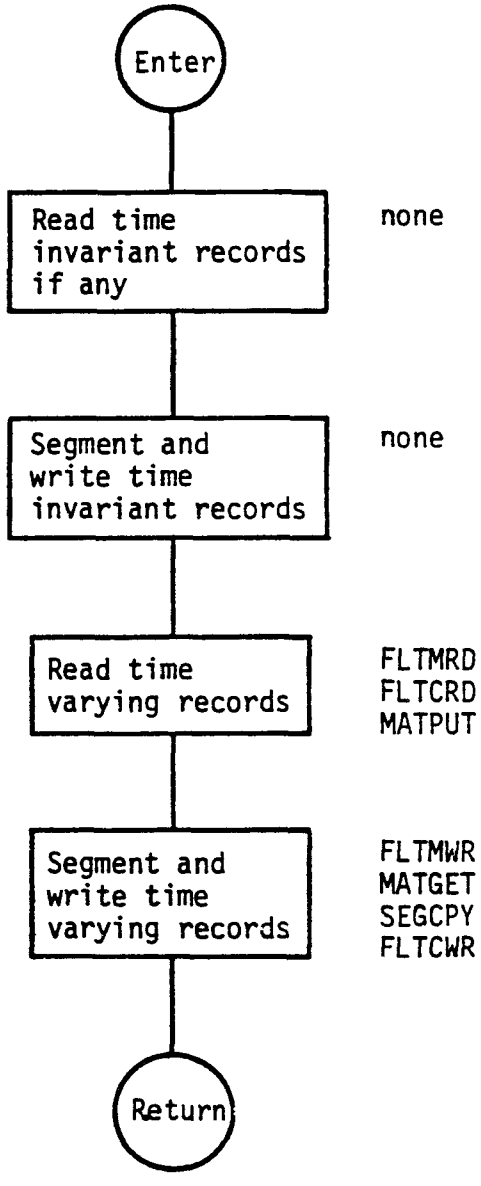


Figure V-10. Flow of Control--TOPCONC File Segmentation Subroutine

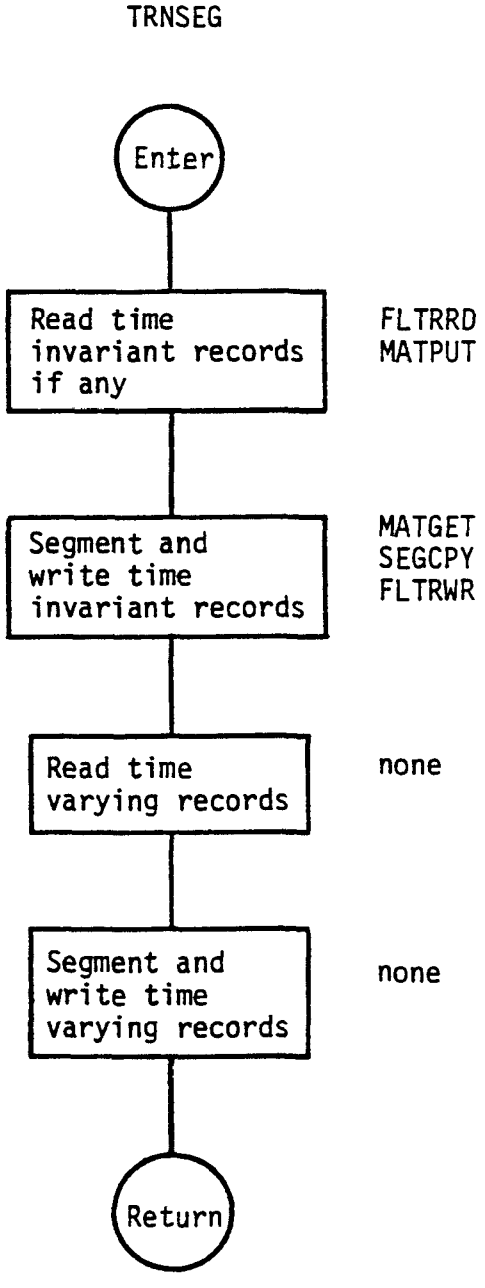


FIGURE V-11. Flow of Control--TERRAIN File Segmentation Subroutine

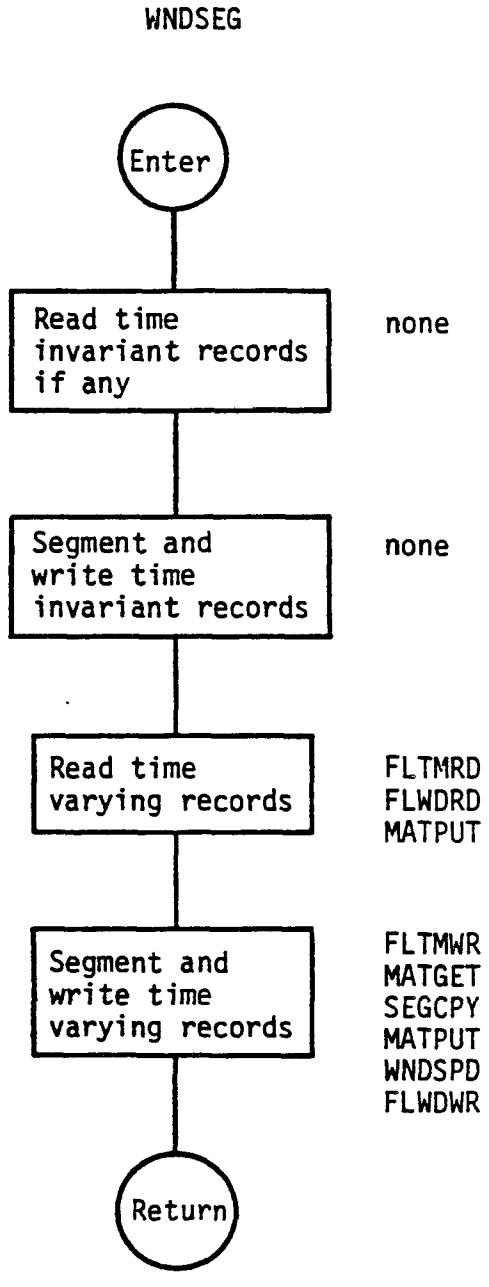


Figure V-12. Flow of Control--WIND File Segmentation Subroutine

Table V-1. File Segmentation Routines

Name	Description	Routines Called
AIRSEG	Segments AIRQUALITY file	See Figure V-2
BNDSEG	Segments BOUNDARY file	See Figure V-3
DFBSEG	Segments DIFFBREAK file	See Figure V-4
EMSSEG	Segments EMISSIONS file	See Figure V-5
HDRCPY	Reads and checks headers from unsegmented file and writes to segmented file	FLHDRD COMPAR FLRGRD FLSGRD SCRSET FLSPRD FLHDWR FLRGWR FLSGWR FLSPWR
PTSSEG	Segments PTSOURCE file	See Figure V-6
RDWSEG	Segments ROADWAY file	See Figure V-7
RTPSEG	Segments REGIONTOP file	See Figure V-8
SEGCPY	Copies one segment of data from full array	
SEGDEF	Main program for file segmentation	See Figure V-1(a)
SEGPRC	Processing routine for file segmentation	See Figure V-1(b)
TMPSEG	Segments TEMPERATUR file	See Figure V-9
TPCSEG	Segments TOPCONC file	See Figure V-10
TRNSEG	Segments TERRAIN file	See Figure V-11
WNDSEG	Segments WIND file	See Figure V-12

VI CONTROL DATA PREPARATION PROGRAMS

Two control data preparation programs--CPREP and SPREP--create the control files CHEMPARAM and SIMCONTROL, respectively. Both programs are quite simple and, therefore, should present no problems in implementation on any machine. The flow of control is as follows:

> Each card is read and its values are printed.
> Packet header and terminator fields are tested to ensure the presence of all cards.
> Numeric fields are validated where possible.
> After all cards have been read and validated, the output control file is written and its contents are printed.

VII AIRSHED SIMULATION PROGRAM

The Airshed Simulation Program is the most complex program in the SAI Airshed Model System. It reads 10-14 input data files, predicts concentrations forward in time, and writes up to three output data files as well as a printed trace of its own progress. It has been designed to handle regions of arbitrary sizes through the method of segmentation. By means of this technique, all information for a given segment is resident in core at once, and the numerical integration proceeds for the segment for one time slice as if it were an independent region.

The following sections of this chapter discuss the techniques of core allocation, the use of secondary storage, and the maintenance of boundary conditions, all of which enable the processing of multiple segments of variable lengths. Next, the method used to compute concentration averages is briefly discussed. Then, a step-by-step description of the flow of control in the program is presented with references to hierarchical diagrams. Finally, some of the considerations in implementing the program are discussed, including FORTRAN anomalies, file assignments, overlays, and the locations of the specific numerical algorithms.

A. CORE ALLOCATION

As discussed in Chapter VIII of the User's Manual, segments (i.e., rectangular subregions) of the region to be modeled are selected such that all the data arrays for one segment will fit in core at once. The maximum dimensions for a segment in the horizontal plane depend on the number of vertical levels and chemical species, the simulation options used, and the amount of core space available. The amount of space available is fixed for an implementation on a given computer, but the other

parameters determining segment size vary from application to application. Since the model will run most efficiently if the segments are as large as possible, the program is designed to make optimal usage of core space. To avoid the need for recompiling the program for each application (a task that would require detailed knowledge of the program and the special considerations for each case), a run-time core allocation technique is employed that allows the user to set the dimensions of a segment to obtain maximum core usage.

The essence of a run-time core allocation scheme lies in the establishment of pointers to the beginning of each variably dimensioned array in a "vector of available space" (VOAS). This vector is assigned when the program is implemented on a particular computer; it is given a dimension as large as possible within the core requirements of the computer system. The pointer to an array is the index of the first word of the array within the VOAS. The array can be passed to a subroutine by providing in the CALL statement the address of the beginning of the array. Figure VII-1 is a simple illustration of the technique, which is discussed below.

Three arrays--AR1, AR2, and AR3--are allocated to the vector of available space called AV. AR1 is dimensioned (NOX,NOY), AR2 is dimensioned (NOX,NOY,NOZ), and AR3 is dimensioned (NOZ,NOX), where NOX, NOY, and NOZ are read in at run time. The pointers JAR1, JAR2, and JAR3 are the indices of the beginning of AR1, AR2, and AR3, respectively, within AV.

Exhibit VII-1 shows code fragments implementing this scheme. JNEXT is a pointer to the next available word. It begins at 1. JAR1, the pointer to AR1, is first set to JNEXT, i.e., to the beginning of AV. Then, the number of words required by AR1 is calculated and JNEXT is set to point to the next available word. JAR2 is set to JNEXT. The next available word after AR2 is calculated and AR3 starts at that point. The subroutine SUB1 uses the arrays AR1, AR2, and AR3 with their dimensions passed as arguments. In the call to SUB1, the beginning of each array is passed by naming the properly indexed word in AV.

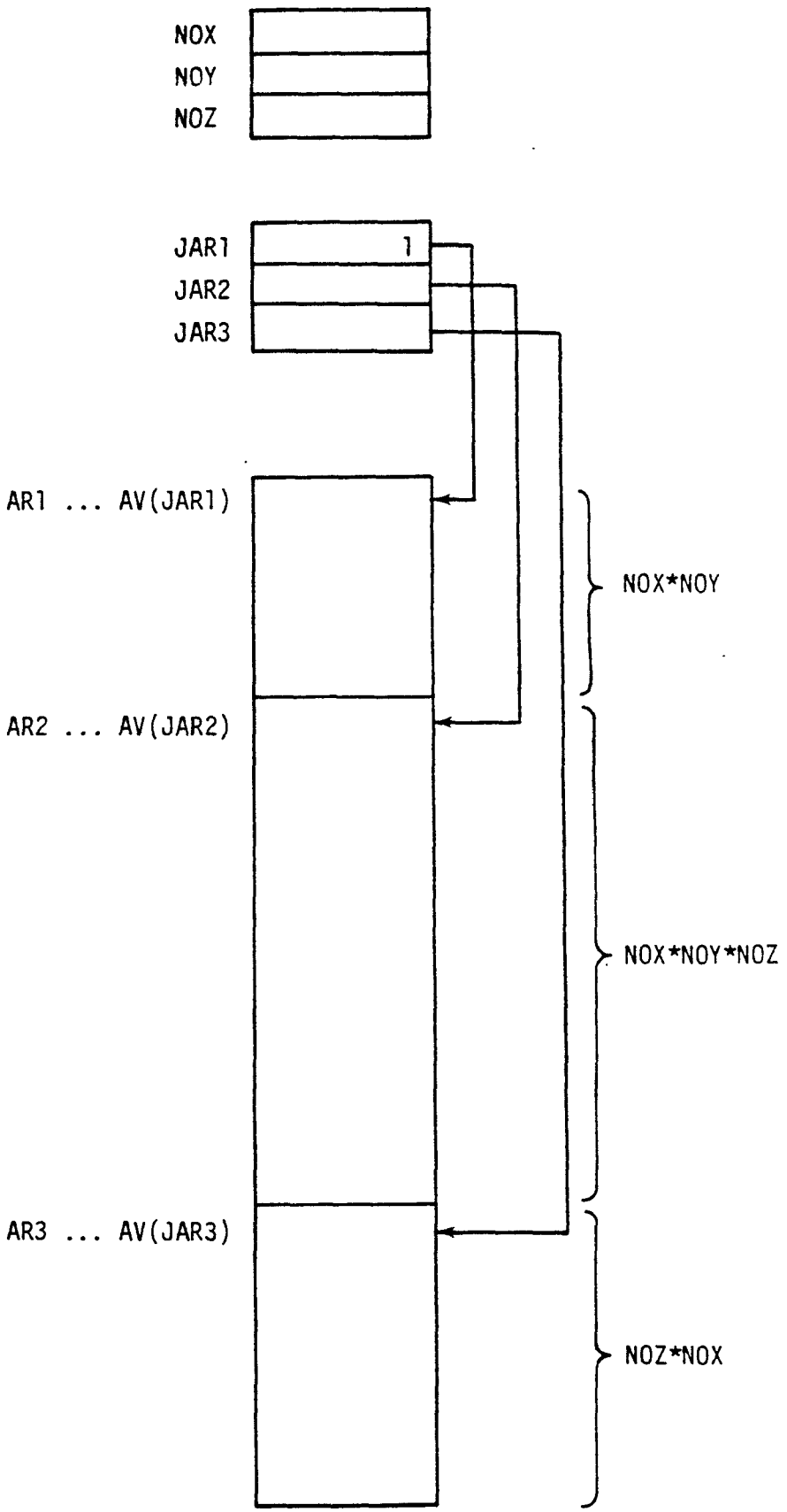


Figure VII-1. Pointers to Variably Dimensioned Arrays

```
      DIMENSION AV(10000)
C
C     *** READ VALUES FOR DIMENSIONS NOX, NOY, NOZ
C
      READ (...) NOX, NOY, NOZ
      .
      .
      .
C
C     *** SET POINTERS TO AR1, AR2, AND AR3
C
      JNEXT = 1
      JAR1 = JNEXT
      JNEXT = JNEXT + NOX*NOY
C
      JAR2 = JNEXT
      JNEXT = JNEXT + NOX*NOY*NOZ
C
      JAR3 = JNEXT
      JNEXT = JNEXT + NOX*NOZ
      .
      .
      .
C
C     *** CALL SUB1, PASSING DIMENSIONS AND
C                            FIRST WORD OF EACH ARRAY
C
      CALL SUB1 (NOX, NOY, NOZ, AV(JAR1), AV(JAR2), AV(JAR3))
      .
      .
      .
      END


      SUBROUTINE SUB1 (NOX, NOY, NOZ, AR1, AR2, AR3)
      DIMENSION AR1(NOX,NOY), AR2(NOX,NOY,NOZ), AR3(NOX,NOZ)
C
      DO 200 J=1,NOY
         DO 100 I=1,NOX
            AR1(I,J) = ...
  100       CONTINUE
  200    CONTINUE
      .
      .
      .
      RETURN
      END
```

Exhibit VII-1. Code Fragments Implementing the Core Allocation Scheme

The Airshed Simulation Program makes extensive use of array pointers; in some cases, there are pointers to other pointers. Each pointer to an array is named by the array name prefixed by a J, as in the previous example.

The program contains four tables of variable lengths that are assigned to AV:

> Segment
> Local
> Boundary
> Segment Scratch.

Figure VII-2 illustrates the assignment of the four tables in AV. Each table, which contains a number of arrays of variable lengths, is described in more detail below.

1. Segment Table

The Segment Table contains 65 data items for each segment, including the origin and dimensions of the segment, scalars and control values for the segment, and pointers to the segment's data arrays in the Boundary and Segment Scratch Tables. Since the Segment Table is of variable length depending on the number of segments in the run, there is a table of pointers to the beginning of each type of item in the Segment Table. These pointers are in the COMMON block /SEGPTR/; they are set in subroutine STABAL. During the course of a segment loop, it is desirable to have the values for the 65 data items for a given segment easily accessible without extensive indexing. Therefore, another COMMON block /SEGTAB/ contains the 65 values from the Segment Table for a given segment. These values are picked out of the Segment Table by the subroutine SEGSET. Figure VII-3 illustrates the relationship between /SEGPTR/, the Segment Table, and /SEGTAB/.

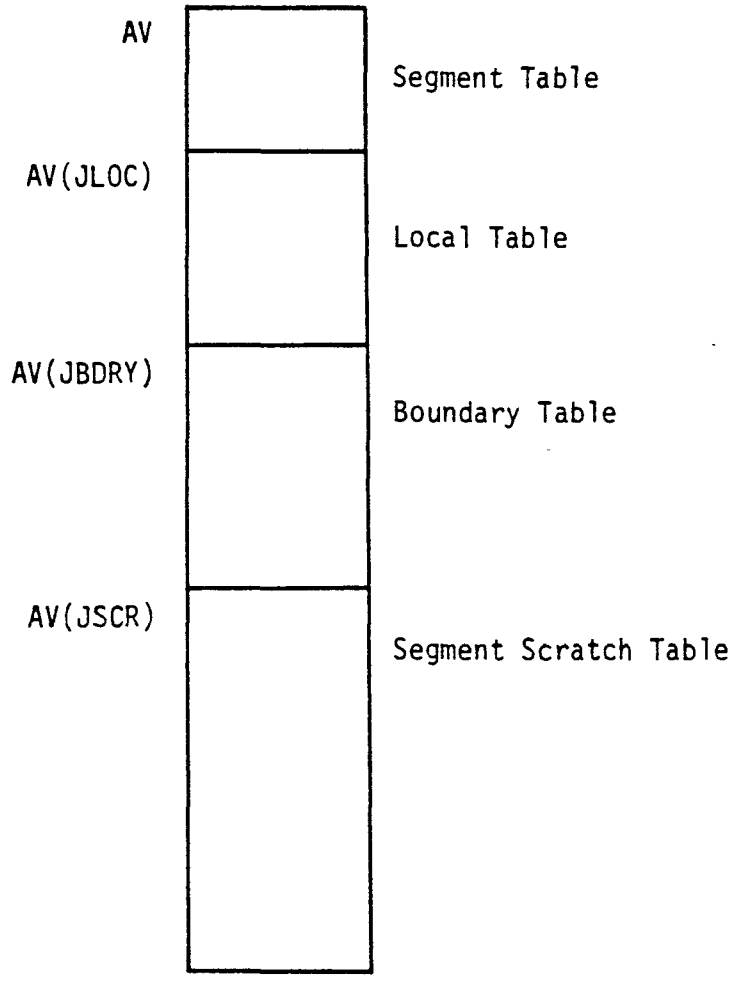


Figure VII-2. The Four Tables Assigned to the Vector of Available Space (AV)

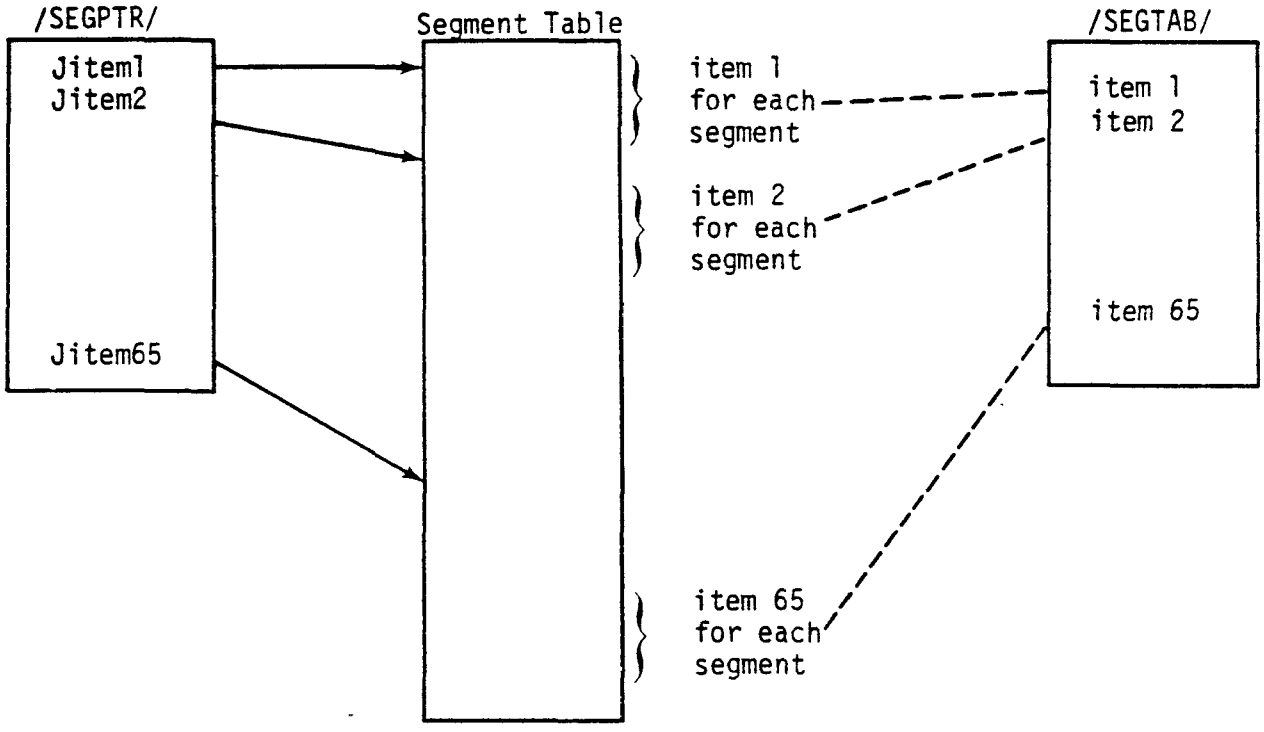


Figure VII-3. Segment Table Structure

2. Local Table

The Local Table contains vectors that are used by the integration routines. With the exception of the six STEP1 and STEP2 vectors, which depend on the length of the longest x- or y-dimension of any segment, all vectors in this table depend on the number of vertical cells and the number of species, dimensions that do not change from segment to segment. The pointers to the vectors in the Local Table are in COMMON block /LCOPTR/; their values are set in subroutine LOCAL. Some overlaying of local vectors is accomplished in LOCAL by resetting JNEXT for groups of variables. Figure VII-4 illustrates the relationship between /LOCPTR/ and the Local Table, and shows how vector overlays are done. Table VII-1 names the vectors in the Local Table and the subroutines that use them, and shows how the vectors are assigned.

3. Boundary Table

The Boundary Table contains the boundary cell definitions and the boundary concentrations in the interface cells for a given segment. In multisegment runs, the Boundary Table for all segments is saved on a secondary storage file. The table comprises eight variable length arrays. Pointers to these arrays are found in the Segment Table; they are set in subroutine BTABAL. The first four arrays define the location of the boundary cells on the west, east, south, and north edges, respectively. The remaining four contain the vertically varying concentrations for each species at each interface cell on the four edges. If any boundary edge is completely external--i.e., there are no other segments adjacent to it--the boundary concentration array for that edge will have zero length. Figure VII-5 shows the Boundary Table structure; see Section C of this chapter for a description of how the Boundary Table is used.

4. Segment Scratch Table

The Segment Scratch Table contains all the other spatially varying data arrays for a given segment; it is saved on a secondary file in <u>all</u>

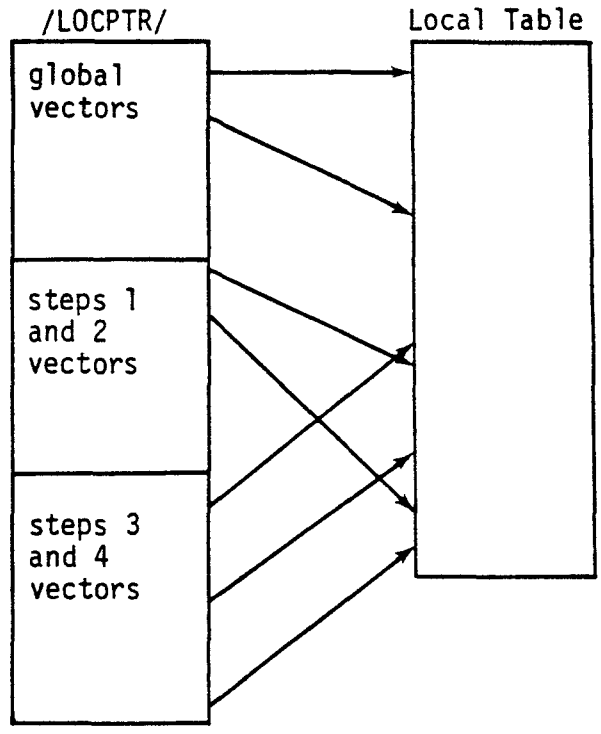


Figure VII-4. Local Table Structure

Table VII-1. Local Variable Overlays

STEP 1 and STEP 2 (General Vectors)	STEP 3 WINDZ	STEP 3 SIMDIG	STEP 4
TH(NOZ,3)			
HT(NOZ,3)			
VD(NOSPEC)			
SRC(NOSPEC)			
R(NORS,NORS)			
HS1(MAXDIM)	AER(NOZ)		
HCS1(MAXDIM)	FACT(NOZP1)		
WS1(MAXDIM)	CON(NOSPEC)		
CS1(MAXDIM)	CONT(NOSPEC,NOZ)		
AS1(MAXDIM)	XKZ(NOZP1)		
DELS1(MAXDIM)	FLUX(NOSPEC)		
	WZ(NOZP1)		
	THEDG(NOZ,4)	A(NOZ)	HOLD(NOSPEC)
	WAV(4,NOZ)	B(NOZ)	RATE(NORS)
	DHDT(NOZ)	C(NOZ)	YGF(NORS)
	JDUM(NOZ)	Y(NOZ)	GA(NORS,NORS)
		X(NOZ)	GF(NORS)
		CTEST(NOZP1)	DC(NORS)
		Z(NOZ)	
		AOMEGA(NOZ)	

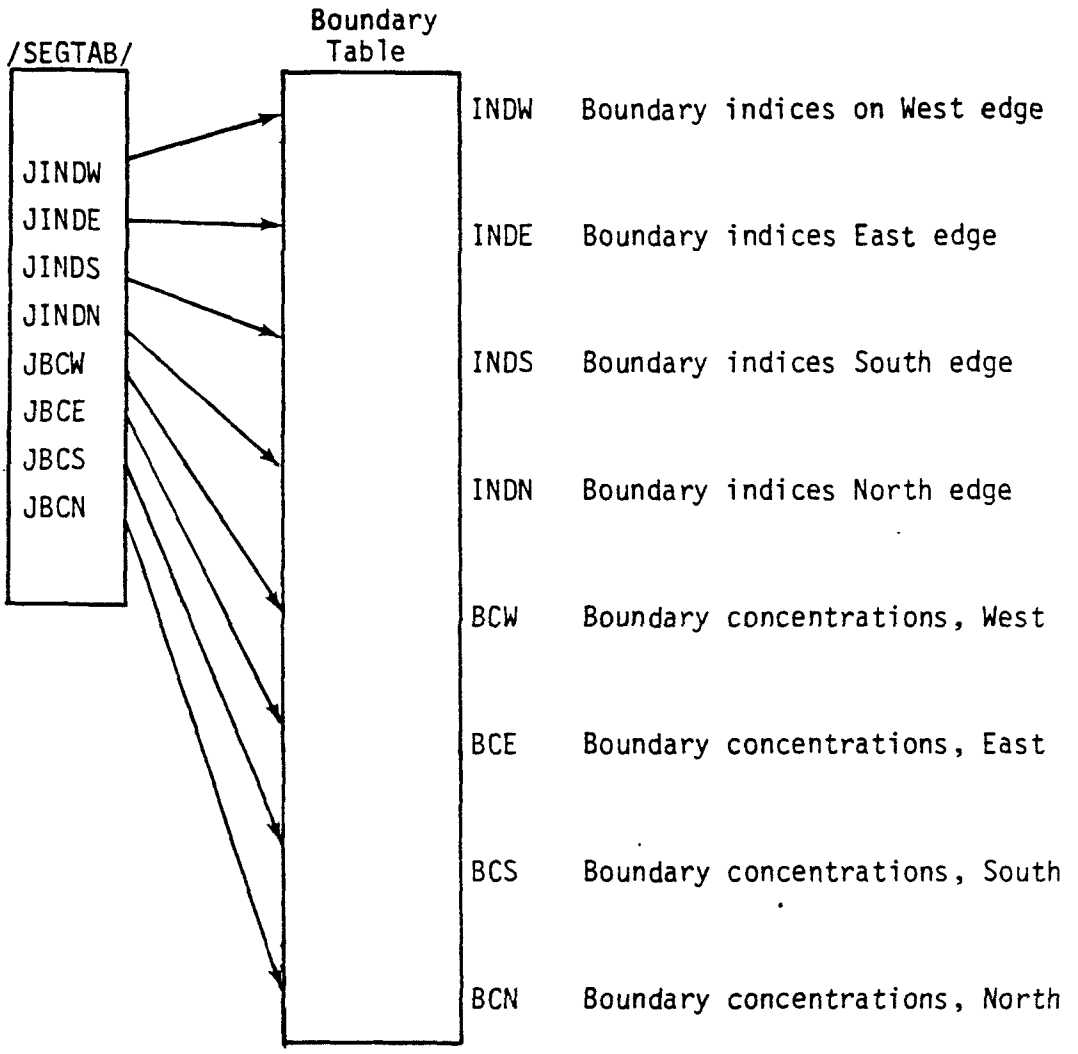


Figure VII-5. Boundary Table Structure

airshed simulation runs, even those for only one segment, because portions of the table are overlaid by other arrays when concentration averages are calculated. (See Section H.4 of this chapter for details.)

The number and lengths of the arrays in the Segment Scratch Table depend on the simulation options used and the dimensions of each segment. The pointers to the arrays in this table are found in the Segment Table; they are set in subroutine SCRAL. Figure VII-6 shows the structure of the Segment Scratch Table and Table VII-2 lists the arrays it contains.

B. TEMPORARY SECONDARY STORAGE FILES

The Segment Table and the Local Table, which are of variable length and thus not allocated until run time, remain in core throughout a simulation run. The Boundary Table and the Scratch Table, however, contain information for only one segment at a time and may differ in length from segment to segment if the segment dimensions differ. To handle this situation, the contents of each table are saved for all segments on temporary secondary storage files. Since the tables may be of different lengths for different segments, and because the length of even the longest record is not determined until run time, the DEFINE FILE feature could not be used for random access of the tables.

Instead, a procedure was devised for using a pair of sequential files to hold all data for each table. One of the pair is designated as a "write" file and the other as a "read" file. During the course of a loop over segments, data for a segment are read from the read file at the beginning of the loop and revised data for the segment are written to the write file at the end of the loop. When the loop has been terminated, the read file, with old data, has been read completely and the write file has been written with all new data. At that point, both files are rewound and their designations are switched, i.e., the write file with the new data becomes the read file for the next pass, and the read file with obsolete data becomes the write file for the next pass.

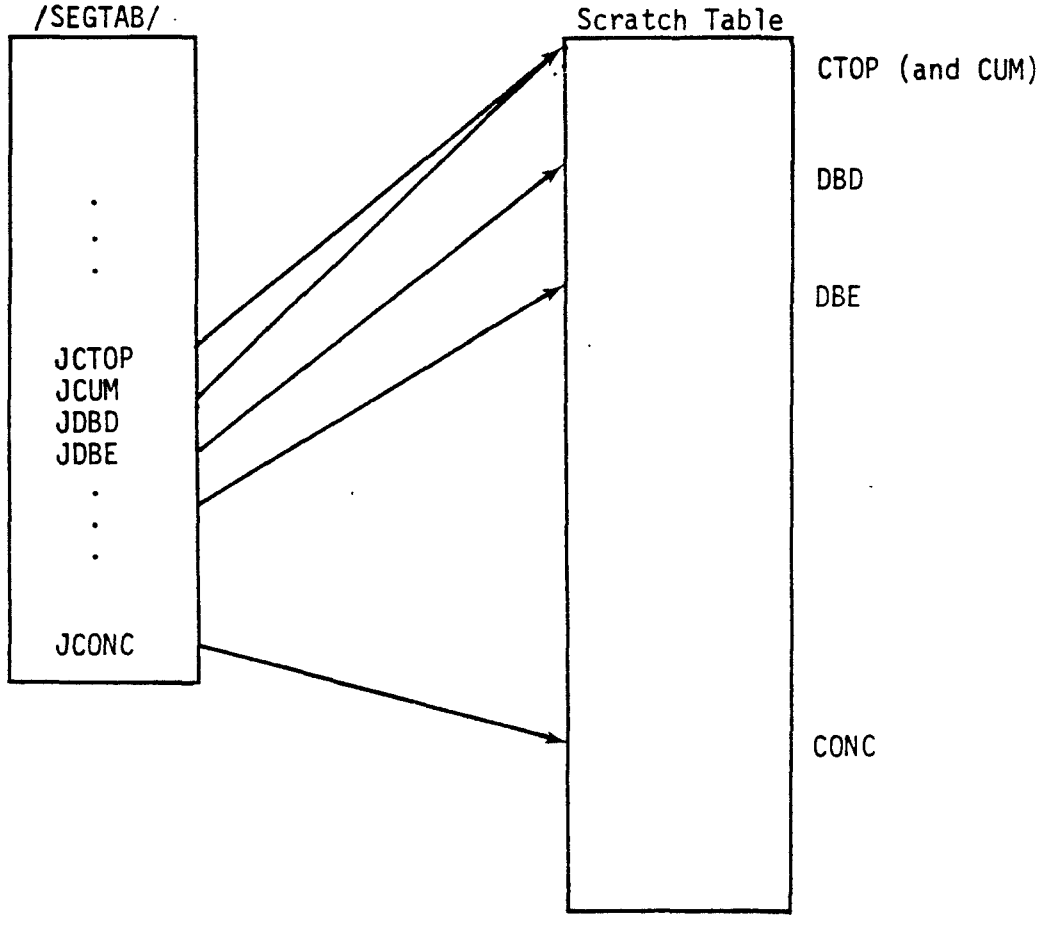


Figure VII-6. Segment Scratch Table Structure

Table VII-2. Arrays in the Segment Scratch Table

	Name	Dimensions	Meaning
	CTOP	(sp,x,y)	Concentrations at the top of the region
	DBD	(x,y)	Diffusion break slope during time interval
	DBE	(x,y)	Diffusion break at end of interval
	DB1	(x,y)	Diffusion break at beginning of time step
	DB2	(x,y)	Diffusion break at end of time step
	TOPD	(x,y)	Top of the region slope during time interval
	TOPE	(x,y)	Top of the region at end of interval
	TOP1	(x,y)	Top of the region at beginning of time step
	TOP2	(x,y)	Top of the region at end of time step
	WX	(x,y,z)	u-component of wind
	WY	(x,y,z)	v-component of wind
*	TEMP	(x,y)	Ground-level temperature
*	ROUGH	(x,y)	Surface roughness
*	VEG	(x,y)	Vegetation factor
	QT	(sp,x,y)	Total ground-level emissions
*	QR	(sp,x,y)	Emissions from roadways
*	CARM	(x,y)	Roadway microscale parameter
*	IJPS	(pts)	Point source location index
*	KPTS	(pts)	Vertical cell to receive point source emissions
*	QPTS	(sp,pts)	Point source emissions
	CONC	(x,y,z,sp)	Pollutant concentrations
*	CVAR	(sp,x,y)	Subgrid-scale concentration variations
	CUM	(x,y,z)	Temporary instantaneous concentrations
*	CUMV	(x,y)	Temporary instantaneous concentration variations

* Optional arrays.

There are four such pairs of files in the Airshed Simulation Program:

> BON1 and BON2 contain all the data from the Boundary Table for each segment. If there is only one segment, these files need not be used, because the Boundary Table will not be overlaid by anything else.

> CON1 and CON2 contain the four-dimensional array of concentration predictions and the three-dimensional array of concentration variations, if it exists.

> CUM1 and CUM2 contain the four-dimensional array of cumulative concentrations and the three-dimensional array of cumulative concentration variations, if it exists. These arrays and those in CON1 and CON2 share space in small core and thus must always be saved on secondary storage files. If there is only one segment, however, only one file is necessary for each pair of arrays.

> SEG1 and SEG2 contain the data in the Segment Scratch Table appearing before the above mentioned concentration arrays. In other words, they contain all the rest of the spatially varying data for each segment.

The Segment Table contains, for each segment, two entries to control the usage of each of the four pairs of files:

> The number of words in the record for the segment (i.e., the length of the table to be saved).
> The location of the record on the file.

For the general case of all sequential files, the second entry is not used. However, it may be possible to use an access method other than sequential for a particular host computer or application. In that case, the location of the record might be a record number or the relative location of the first word. The file control entries in the Segment Table are set in subroutine SCRAL.

One pair of subroutines is used for accessing each of the four pairs of files within a segment loop: a GET routine to obtain data from a "read" file, and a PUT routine to put data to a "write" file. The subroutine names (not surprisingly) are:

> PUTBON	> GETBON
> PUTCON	> GETCON
> PUTCUM	> GETCUM
> PUTSEG	> GETSEG.

The segment number, the read or write file unit number, the two control entries mentioned above, and the core location to which or from which data are to be transferred are all passed to each subroutine. The subroutines can easily be changed to reflect different access techniques without affecting the rest of the program. The unit numbers for the four pairs of files are set in the BLOCK DATA subprogram. These numbers can also be easily changed, if necessary.

Figure VII-7 is a summary of the core allocation and file storage procedures we have discussed so far. /SEGPTR/ contains pointers to the Segment Table and /LOCPTR/ contains pointers to the Local Table. These two tables always remain in core. Pointers to arrays in the Boundary Table are found in the Segment Table, and the Boundary Table can be stored on secondary storage files BON1 and BON2 (if there is more than one segment). The pointers to arrays in the Segment Scratch Table are also found in the Segment Table. The Segment Scratch Table, not including the concentration array, is saved on secondary storage files SEG1 and SEG2; the instantaneous concentrations and variations on CON1 and CON2; and the cumulative concentrations and variations on CUM1 and CUM2.

C. BOUNDARY HANDLING

Figure VII-8 shows a 7 X 12 cell region divided into two 7 X 8 cell segments to illustrate the following discussion of segment interface handling.

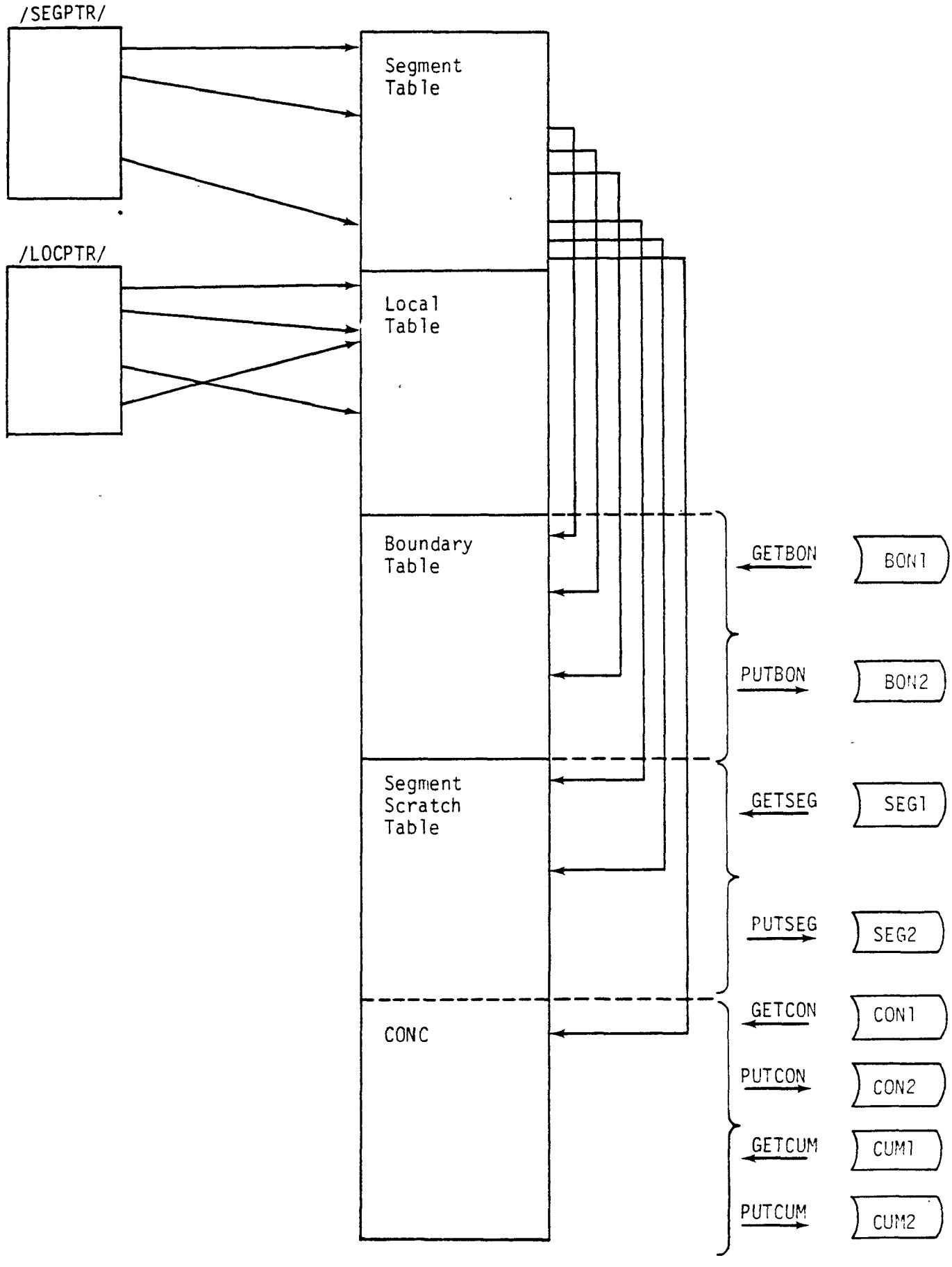


Figure VII-7. Core Allocation and File Storage

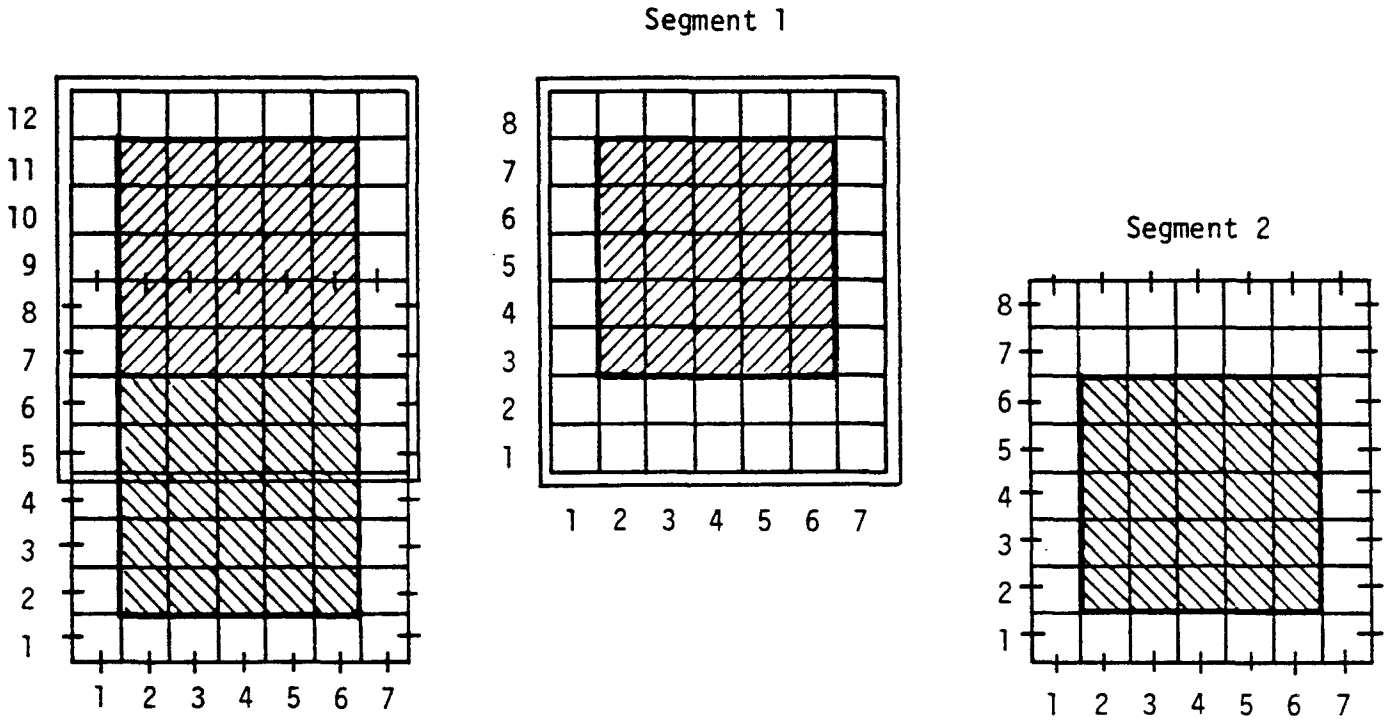


Figure VII-8. Example of a Segmented Region

First, it is necessary to define the boundary indices. The rectangular horizontal plane for each segment has four edges: West, East, South and North. Each row of the region has a West and an East boundary and each column has a South and a North boundary. The boundary indices are defined as the indices of the first and last cells within each row or column to be simulated, not including the extra cells at each edge. Thus, for each row of both segments in Figure VII-8, the West index is 2 and the East index is 6. In the North-South directions, however, the boundary indices in the two segments differ because of the two-row overlap at the segment interface. In Segment 1, the South index is 3 and the North index is 7; in Segment 2, the South and North indices are 2 and 6, respectively. These boundary indices are used to control the numerical integration for each segment as if it were an independent region.

To answer the question of where the concentrations in the outer cells come from, it is necessary to define external and internal boundaries:

> External boundary cells in a segment are those that correspond to the edges of the unsegmented region.

> Internal boundary cells in a segment are those that correspond to simulated cells in another segment.

Concentrations in the external boundary cells are read from the BOUNDARY file, and steady-state calculations may be performed on them at the beginning of each time slice. Concentrations in the internal boundary cells must be filled in from the concentration arrays for other segments.

Four numbers are read from the BOUNDARY file for each cell on each edge for each segment:

> Boundary index (index of the first or last cell to be simulated in the row or column). A boundary index of zero indicates that the row or column is an external edge.

> Adjacent segment number. If the adjacent segment number is zero, the last simulated cell is at an external boundary; if the number is not zero, the last simulated cell is at an internal boundary and the concentration values in the interface cells must come from simulated cells in the adjacent segment.

> I and J indices, within the adjacent segment, of the cell corresponding to the nearer interface cell at the edge.

Table VII-3 shows the complete set of boundary definition indices that would be read from the BOUNDARY file for the sample region. After they are read, the I and J indices for each cell are combined into one number by the following formula:

$$INDEX = (J - 1)*NOX + JI \quad ,$$

where NOX = number of cells in the x-direction in the adjacent segment. This combined index is the location of word (I,J) in an array dimensioned (NOX,NOY).

The Boundary Table contains for one segment the four sets of boundary definition indices (INDW, INDE, INDS, INDN) and four sets of boundary concentrations in the interface cells, if any (BCW, BCE, BCS, BCN). During the subroutine SETUP, which is called at the beginning of each time slice, the concentrations at the external boundaries in the CONC array are updated from the BOUNDARY file if necessary. For internal boundaries, concentrations in the Boundary Table for interface cells are filled in from the appropriate adjacent segment cells by the subroutines SEGINT and BCFILL. During the subroutine ADVANC, which controls the integration steps, the interface cells along the internal edges in CONC are filled by the concentrations from the corresponding BC arrays (subroutine CFILL) and steady-state computations are performed in external boundary cells if required (subroutine SSBC).

Table VII-3. Boundary Definition Indices

	Segment 1				Segment 2			
Edge of Row	Boundary Index	Adjacent Segment Number	Within Adjacent Segments I	Within Adjacent Segments J	Boundary Index	Adjacent Segment Number	Within Adjacent Segments I	Within Adjacent Segments J
West								
1	0	0	0	0	2	0	0	0
2	2	0	0	0	2	0	0	0
3	2	0	0	0	2	0	0	0
4	2	0	0	0	2	0	0	0
5	2	0	0	0	2	0	0	0
6	2	0	0	0	2	0	0	0
7	2	0	0	0	2	0	0	0
8	2	0	0	0	0	0	0	0
East								
1	0	0	0	0	6	0	0	0
2	6	0	0	0	6	0	0	0
3	6	0	0	0	6	0	0	0
4	6	0	0	0	6	0	0	0
5	6	0	0	0	6	0	0	0
6	6	0	0	0	6	0	0	0
7	6	0	0	0	6	0	0	0
8	6	0	0	0	0	0	0	0

Table VII-3 (Concluded)

Edge of Row	Segment 1 Boundary Index	Segment 1 Adjacent Segment Number	Segment 1 Within Adjacent Segments I	Segment 1 Within Adjacent Segments J	Segment 2 Boundary Index	Segment 2 Adjacent Segment Number	Segment 2 Within Adjacent Segments I	Segment 2 Within Adjacent Segments J
South								
1	0	0	0	0	0	0	0	0
2	3	2	2	6	2	0	0	0
3	3	2	3	6	2	0	0	0
4	3	2	4	6	2	0	0	0
5	3	2	5	6	2	0	0	0
6	3	2	6	6	2	0	0	0
7	0	0	0	0	0	0	0	0
North								
1	0	0	0	0	0	0	0	0
2	7	0	0	0	6	1	2	3
3	7	0	0	0	6	1	3	3
4	7	0	0	0	6	1	4	3
5	7	0	0	0	6	1	5	3
6	7	0	0	0	6	1	6	3
7	0	0	0	0	0	0	0	0

D. COMPUTING AVERAGE CONCENTRATIONS

Predicted pollutant concentrations are averaged over a given time interval that is input by the user by means of the SIMCONTROL file. To calculate the time averages, the cumulative concentration array must be created; it contains the time-weighted sum of the concentrations calculated during each time step from the beginning of the averaging interval.

The Interim Instantaneous Concentrations (INTERINST) file keeps a complete history of instantaneous concentrations calculated within each time slice. See Chapter IX on File Structures for a more detailed description of this file. For a given time slice, each segment is integrated in turn by one or more time steps of equal size. The instantaneous concentrations array for the segment is written to the INTERINST file at the beginning of the time slice and at the end of each time step in the slice. Thus, for n time steps there will be n + 1 sets of concentration records for the segment. When the time slice has been completed for all segments (subroutine ADVANC), the INTERINST file is rewound.

In the accumulation phase, controlled by subroutine ACCUM, the following procedures are completed for each segment:

> The current cumulative concentration arrays are brought in from CUMR (the secondary storage file to be read). These arrays overlay the instantaneous concentration arrays in the Scratch Table.

> Instantaneous concentrations are read in from INTERINST for each time step, weighted by the time step size (as described below) and added into the cumulative concentrations.

> The new cumulative concentrations are written to CUMW (the secondary storage file to write).

The concentrations at the beginning and end of the time slice are multiplied by the time step used for the slice. If there was more than one step in a slice, there will be one or more sets of concentration records between the first and the last records; these are each weighted by the value of two times the time step.

At the end of the averaging period (in subroutine FINISH), the cumulative concentration arrays are again read in and the time-weighted values are divided by (2 x total elapsed averaging interval). The resulting average concentrations are then written to the AVERAGE file, and the cumulative concentration arrays are zeroed for the next averaging period.

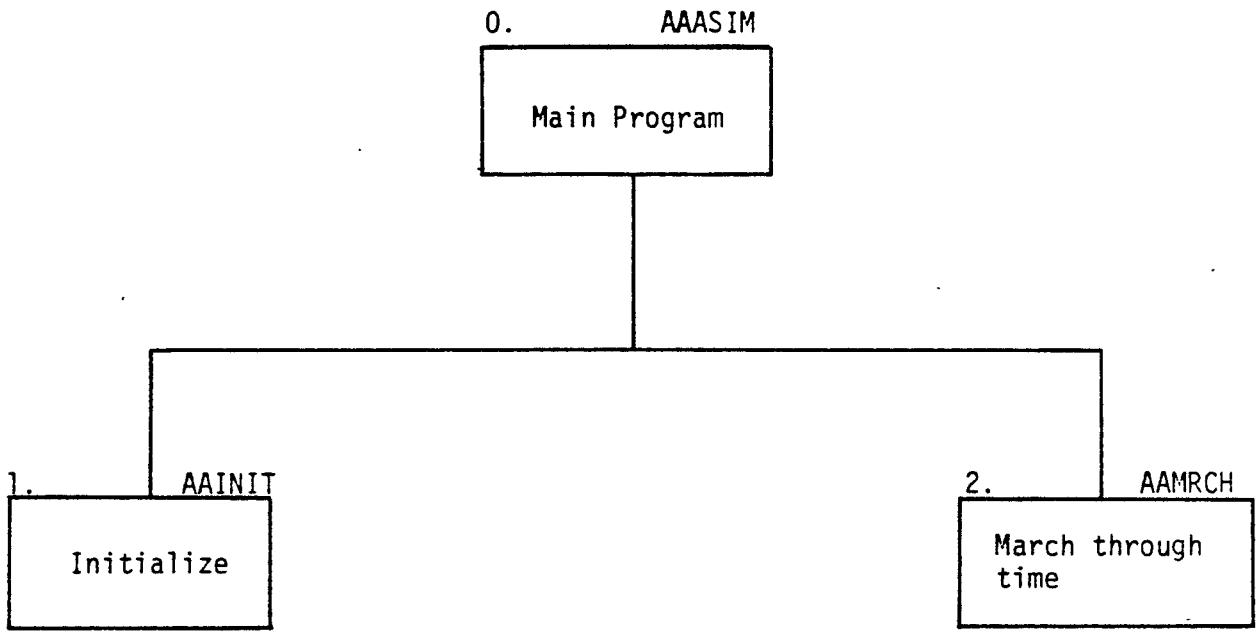
If concentration variations are being calculated, the array containing the variations follows the concentration array in the Segment Scratch Table. To calculate the average concentration variation output to the VARIATION file, the same procedure is performed. The instantaneous variations are also written to the INTERINST file, and the cumulative variations are maintained in the same way.

E. FLOW OF CONTROL

This section describes the operation of the Airshed Simulation Program and includes figures and tables that supplement the text to aid in the comprehension of the program as a whole. Figures VII-9(a) through VII-9(c) are top-down structure diagrams that illustrate the hierarchy of the highest level subroutines. Table VII-4 is a hierarchical list of subroutine names, indicating the complete calling structure of the program. Table VII-5 lists the subroutines alphabetically and includes a one-line description of the function of each.

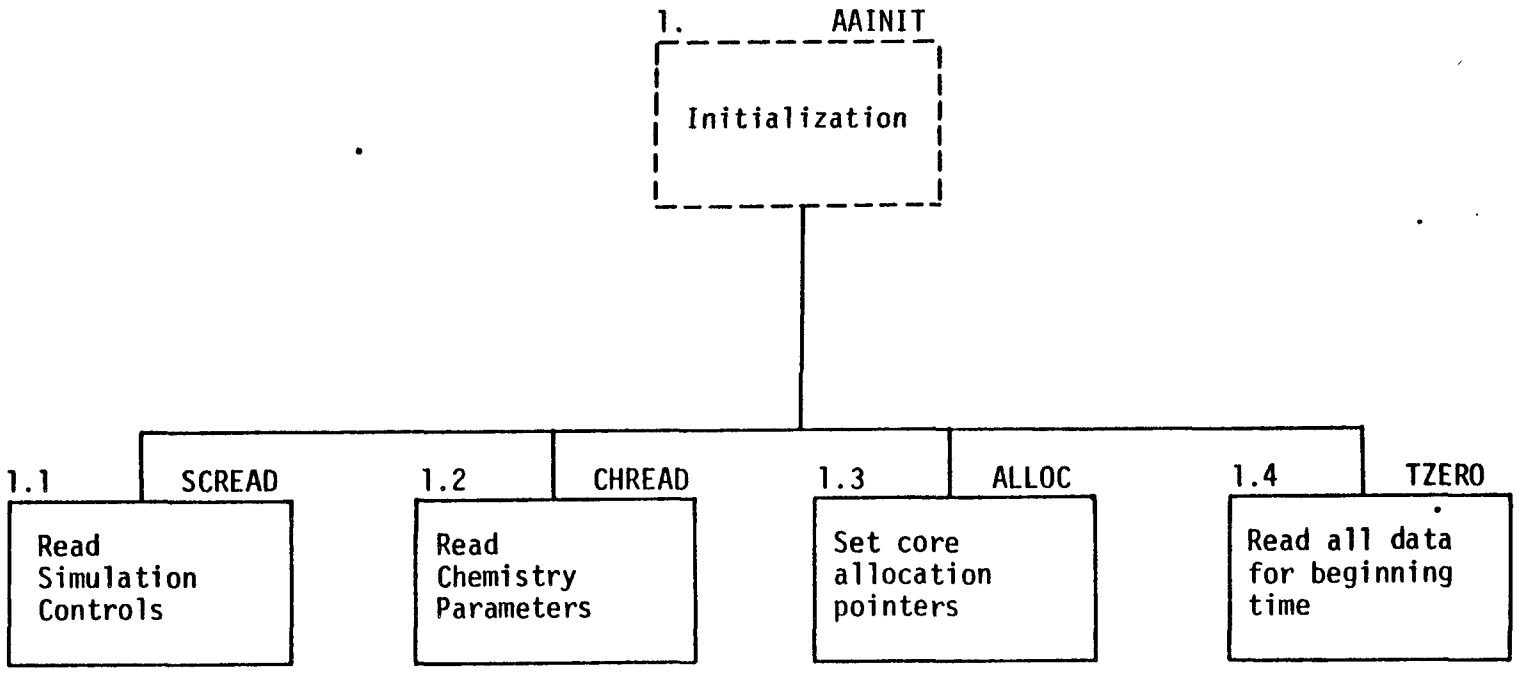
1. AAASIM

The main program AAASIM controls the calls to the two major controlling routines [Figure VII-9(a)]. The first, AAINIT, performs all one-time initialization functions and the second, AAMRCH, performs the simulation,



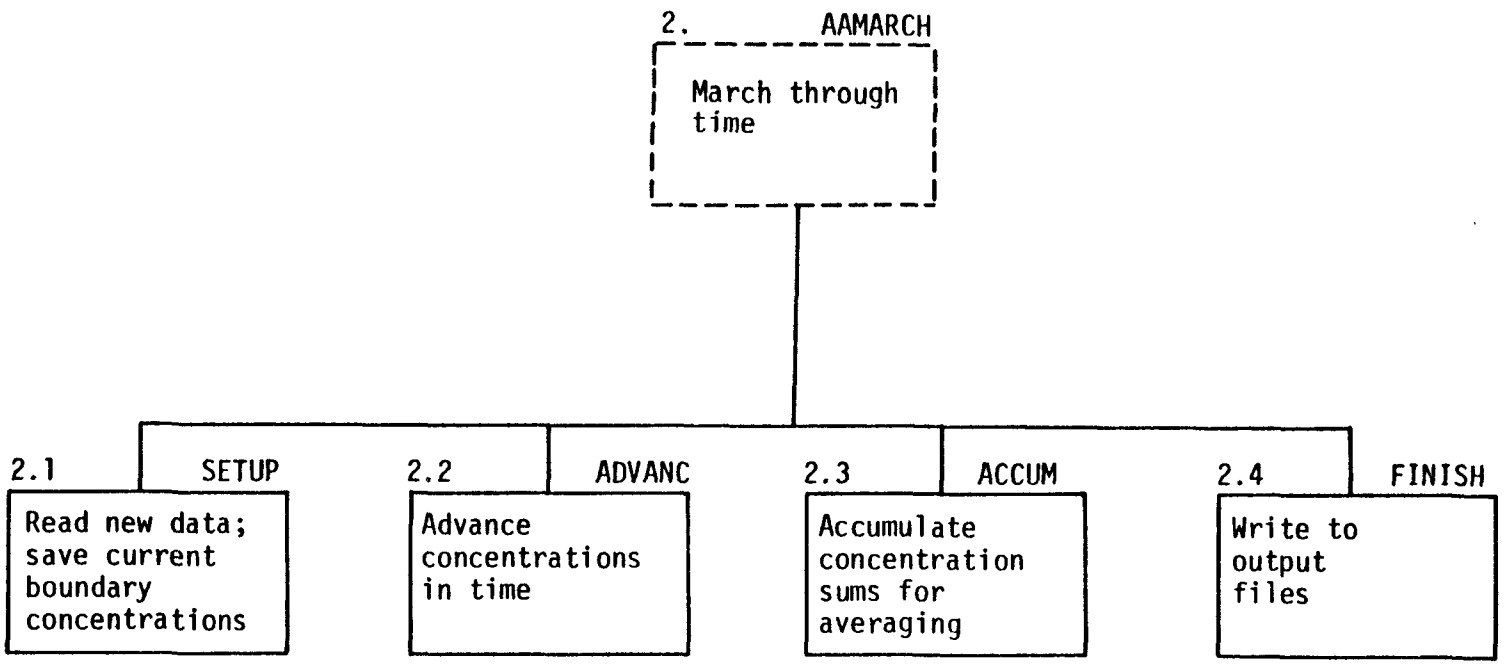
(a) AAASIM

Figure VII-9. Hierarchy of Top-Level Subroutines



(b) AAINIT

Figure VII-9 (Continued)



(c) AAMRCH

Figure VII-9 (Concluded)

Table VII-4. Subroutine Hierarchy in the Airshed Simulation Program

```
0.   AAASIM
1.      AAINIT
1.1         SCREAD
                CLFILE
                    RETURNS
1.2         CHREAD
                SPECID
                COEFID
                CLFILE
1.3         ALLOC
                STABAL
                BTABAL
                BDREAD
                PUTBON
                LOCAL
                SCRAL
1.4         TZERO
                HDRCK
                SPETAB
                FILSKP
                    MSREAD
                MSREAD
                SEGSET
                RESDB
                RESTMP
                RESEM
                RESPTS
                RESTOP
                RESRWY
                RESTER
                RESCT
                RESWND
                PUTSEG
                PUTCUM
```

Table VII-4 (Continued)

```
                CLFILE
                HDRRIT
                GETSEG
                GETBON
                ARTIME
                RESAQ
                RESBDY
                SSIC
                    PHOTOL
                    VERTCL
                    PHFACT
                    TDRATE
                    STEADY
                        CHEM
                CONPRT
                PUTCON
                SECOND
2.0     AAMRCH
2.1         SETUP
                FILCK
                FILCKE
                SEGSET
                GETBON
                GETCON
                RESBDY
                PUTCON
                MSREAD
                GETSEG
                RESDB
                RESTMP
                RESEM
                RESPTS
                RESTOP
                RESRWY
                RESCT
```

Table VII-4 (Continued)

```
            RESWND
            PUTSEG
            GETCON
            SEGINT
                GETBON
                BCFILL
                PUTBON
            SLICE
2.2     ADVANC
            SEGSET
            GETCON
            GETSEG
            GETBON
            CFILL
            SSBC
                PHOTOL
                VERTCL
                PHFACT
                TDRATE
                STEADY
                    CHEM
            WINTER
            ARTIME
            STEP1
                VERTCL
                SHASTA
            STEP2
                VERTCL
                SHASTA
            STEP34
                PHOTOL
                VERTCL
                DIFCOF
                SNKSRC
```

Table VII-4 (Concluded)

```
                MICRO
                WINDZ
                    VERTCL
                PHFACT
                TDRATE
                STEADY
                    CHEM
                STEP3
                    SIMDIG
                STEP4
                    CHEM
                    CHEM2
            DEBUG
                CRASH
            PUTCON
            PUTSEG
2.3     ACCUM
            SEGSET
            GETCUM
            ADDEM
            PUTCUM
2.4     FINISH
            SECOND
            DISPLA
            WARN
            SEGSET
            GETCUM
            PUTCUM
            GETCON
```

Table VII-5. Descriptions of Airshed Simulation Program Subroutines

Subroutine	Description of Function
AAASIM	Main program--controls calls to two major phases
AABDAT	BLOCK DATA (not executable)--sets values for labeled COMMON variables
AAINIT	Controls calls in initialization phase
AAMRCH	Controls calls in march through time
ACCUM	Controls accumulation of concentration sum at the end of each time slice
ADDEM	For a segment, adds concentrations at each time step into cumulative concentrations
ADVANC	Controls advance of concentrations one time slice
ALLOC	Controls allocation of all variable core space
ARTIME	Calculates values in an array (DB or TOP) for a given time; stores previous values
BCFILL	Stores CONC in BC for a given segment and edge
BDREAD	Reads boundary definition records for a segment
BTABAL	Allocates Boundary Table
CFILL	Stores BC in CONC at all interfaces for a segment
CHEM	Computes steady-state values for radicals, reaction rates, and partial derivatives for each species
CHEM2	Computes reaction rates and partial derivatives for night chemistry
CHREAD	Reads CHEMPARAM file
CLFILE	Closes a file
COEFID	Establishes coefficient values for chemistry
CONPRT	Prints concentrations for a segment
CRASH	Prints all relevant information at the time of a crash
DEBUG	Controls call to crash print
DIFCOF	Calculates vertical diffusivity
FILCK	Checks current time against file time interval; if at the end of the interval, reads next interval and sets flag; adjusts time slice so that it won't cross time interval

Table VII-5 (Continued)

Subroutine	Description of Function
FILCKE	Like FILCK, but tests current time against beginning of the file time interval
FILSKP	Reads time-varying data until first good time interval
FINISH	Updates time and writes files at end of a time slice
GETBON	Gets Boundary Table for a segment from secondary storage file
GETCON	Gets concentrations for a segment from secondary storage file
GETCUM	Gets cumulative concentrations for a segment from secondary storage file
GETSEG	Gets scratch arrays for a segment from secondary storage file
HDRCK	Reads and checks header records
HDRRIT	Writes header records to INSTANT and AVERAGE
LOCAL	Allocates space for local variables
MICRO	Performs surface layer microscale computations
MSREAD	Reads data record from METSCALARS file
PHFACT	Calculates vertically varying photolysis factors
PHOTOL	Multiplies photolysis rate constants by factors
PUTBON	Puts Boundary Table for a segment to secondary storage file
PUTCON	Puts concentrations for a segment to secondary storage file
PUTCUM	Puts cumulative concentrations for a segment to secondary storage file
PUTSEG	Puts scratch arrays for a segment to secondary storage file
RESAQ	Reads data for one segment from AIRQUALITY file
RESBDY	Reads data for one segment from BOUNDARY file
RESCT	Reads data for one segment from TOPCONC file

Table VII-5 (Concluded)

Subroutine	Description of Function
RESDB	Reads data for one segment from DIFFBREAK file
RESEM	Reads data for one segment from EMISSIONS file
RESPTS	Reads data for one segment from PTSOURCE file
RESRWY	Reads data for one segment from ROADWAY file
RESTER	Reads data for one segment from TERRAIN file
RESTMP	Reads data for one segment from TEMPERATUR file
RESTOP	Reads data for one segment from REGIONTOP file
RESWND	Reads data for one segment from WIND file
SCRAL	Allocates scratch storage space
SCREAD	Reads SIMCONTROL file
SEGINT	Updates segment interfaces
SEGSET	Sets segment table values for a given segment
SETUP	Sets up data for a new time slice
SHASTA	Performs SHASTA integration
SIMDIG	Solves tridiagonal system of linear equations
SLICE	Calculates time-slice size
SNKSRC	Calculates deposition velocity and reemission
SPECID	Sets species indices for chemistry
SPETAB	Sets species index table for a file
SSBC	Computes steady-state concentrations at external boundaries
SSIC	Computes steady-state values for initial concentrations
STABAL	Allocates segment table space
STEADY	Calculates steady-state values for NO, NO_2, and O_3
STEP1	Controls integration in x-direction
STEP2	Controls integration in y-direction
STEP3	Performs advection integration in z-direction
STEP4	Performs chemistry integration for a single cell
STEP34	Controls vertical and chemistry integration
TDRATE	Calculates temperature-dependent rate constants
TZERO	Sets up all data for time = t_0 (beginning of integration)
VERTCL	Calculates vertical distribution of grid cells
WINDZ	Calculates vertical winds
WINTER	Writes INTERINST file

marching through time by time slices. Each routine (both are described below) calls four major subroutines. The subroutines used in AAINIT that are not also used in AAMRCH can be overlaid when the initialization phase is completed (see Section G of this chapter for more details on overlays).

2. AAINIT

AAINIT calls four subroutines: SCREAD, CHREAD, ALLOC, AND TZERO [see Figure VII-9(b)].

SCREAD reads the SIMCONTROL file, prints its values, and closes it.

CHREAD reads the CHEMPARAM file, prints its values and closes it. Subroutines SPECID and COEFID are called, if required, to verify that the species and coefficient names, respectively, correspond to the names required by the built-in chemical mechanism. CHREAD also sets internal flags controlling optional program functions depending on the properties of the species and reactions and on the simulation options specified on SIMCONTROL.

ALLOC reads the BOUNDARY file for the overall dimensions of the region and the number and size of its segments. It then calls four subroutines to set up all core allocation pointers:

> STABAL sets up the pointers to the Segment Table in /SEGPTR/.
> LOCAL sets up the pointers to the Local Table in /LOCPTR/.
> BTABAL sets up the pointers to the Boundary Table in the Segment Table.
> SCRAL sets up the pointers to the Scratch Table in the Segment Table. SCRAL reads the PTSOURCE file to determine the maximum number of point sources in each segment.

TZERO reads the remainder of the M.E.B. data files and establishes all array values for the beginning of the simulation. For each file, it calls HDRCK to read and check the header records, SPETAB to establish species index tables based on species names (which allows species to appear in a

different order on each file), and FILSKP to skip records, if necessary, up to the first time interval applicable to the run. TZERO then reads all M.E.B. data for each segment (the subroutines beginning with RES ___) and zeros arrays where appropriate. It calls SSIC, if required, to perform steady-state calculations on the initial concentrations. It writes all data onto the appropriate secondary storage files, calls HDRRIT to write headers to the output files, and calls CONPRT, if required, to print the initial concentrations.

3. AAMRCH

AAMRCH controls the march through time. It calls SETUP, ADVANC, ACCUM, and FINISH for each time slice [see Figure VII-9(c)].

SETUP checks all file time intervals (FILCK and FILCKE) and reads new data, if necessary. It also saves all boundary interface concentrations (SEGINT) and establishes the size of the time slice that will be used (SLICE).

ADVANC calls the subroutines that perform the numerical integration. These are listed in Section I of this chapter. ADVANCE processes each segment completely for one time slice. For each segment, it first fills in the concentrations at the segment interfaces (CFILL), performs steady-state calculations, if necessary, on the external boundary concentrations (SSBC), and writes the entire concentration array to the INTERINST file (WINTER). Then, for each time step, it calculates the diffusion break and top of the region (ARTIME), performs the horizontal transport steps (STEP1 and STEP2), performs the vertical transport and chemistry steps (STEP34), and writes the concentrations at the end of the time step to INTERINST (WINTER).

ACCUM adds the instantaneous concentrations calculated for the time slice into the cumulative concentration arrays. For each segment, it calls ADDEM, which reads INTERINST for each time step and does the actual time-weighting and adding. (Details were previously given in Section D of this chapter.)

FINISH is called at the end of each time slice. It calculates and prints the elapsed CPU time for the slice and then checks to see whether an instantaneous or average output time has been reached. If so, concentrations are read from the appropriate secondary storage file and written to the appropriate output file. For averages, the time-weighting is divided out of the cumulative concentrations. FINISH also calls CONPRT if the concentrations are to be printed. If the simulation end time has been reached or if it has been detected that computer time is running out, FINISH will write end-of-file on the output files and terminate the run.

F. COMMON BLOCKS

Eleven labeled COMMON blocks are used to share data among the many Airshed Simulation Program subroutines; a short description of each one is given in this section. Table VII-6 is a cross-reference list indicating the appearance of each of the following COMMON blocks in each subroutine.

> /AVAIL/--contains the vector of available space, its length, and pointers to the beginning of its tables.

> /CHDATA/--contains species indices, names, and coefficients required by the Carbon-Bond Mechanism.

> /CHPARM/-- contains chemistry parameters input by means of the CHEMPARAM file.

> /CNTROL/--contains many control parameters of global interest:

- Dimensions and region descriptors.
- Logical control flags.
- Time controls.
- Integration controls and calculated scalar values.
- Trace and internal file unit numbers.
- Trace options and current location indicators.
- Descriptive information.

> /FILCON/--contains all file control variables.

> /LCM/--contains the large core memory vector used to store boundary tables for all segments (LBL only).

> /LOCPTR/--contains pointers to vectors in the Local Table.

> /MSCAL/--contains meteorological scalars read from METSCALARS or calculated in MSREAD.

> /SEGPTR/--contains pointers to the Segment Table entries.

> /SEGTAB/--contains the values of the Segment Table entries for a particular segment.

> /VPARAM/--contains the vertical parameters used by subroutine VERTCL.

Table VII-6. COMMON Block/Subroutine Cross-Reference List

	COMMON Block										
Subroutine	AVAIL	CHDATA	CHPARM	CNTROL	FILCON	LCM	LOCPTR	MSCAL	SEGPTR	SEGTAB	VPARAM
AAASIM	X	X	X	X	X	X	X	X	X	X	X
AABDAT		X	X	X	X		X	X	X		
AAINIT	X										
AAMRCH	X										
ACCUM				X	X					X	
ADDEM											
ADVANC				X	X		X	X	X	X	
ALLOC				X	X		X		X		X
ARTIME											
BCFILL											
BDREAD											
BTABAL											
CFILL											
CHEM		X	X								
CHEM2		X	X								
CHREAD		X	X	X	X			X			
CLFILE											
COEFID		X									
CONPRT			X	X						X	
CRASH			X	X				X		X	
DEBUG				X			X			X	
DIFCOF				X				X		X	
FILCK											
FILCKE											
FILSKP				X							
FINISH			X	X	X					X	
GETBON						X					
GETCON											
GETCUM											
GETSEG											
HDRCK				X					X		
HDRRIT			X	X	X						
LOCAL							X				
MICRO		X	X	X							
MSREAD					X			X			
PHFACT											
PHOTOL			X								
PUTBON						X					
PUTCON											
PUTCUM											
PUTSEG											
RESAQ			X		X						
RESBDY					X						
RESCT			X		X						
RESDB					X						
RESEM					X			X			
RESPTS					X			X	X	X	
RESRWY					X			X			
RESTER					X						
RESTMP					X						

Table VII-6 (Concluded)

	COMMON Block										
Subroutine	AVAIL	CHDATA	CHPARM	CNTROL	FILCON	LCM	LOCPTR	MSCAL	SEGPTR	SEGTAB	VPARAM
RESTOP					X						
RESWND					X				X	X	
SCRAL									X		
SCREAD				X	X						
SEGINT									X	X	
SEGSET				X					X	X	
SETUP				X	X			X	X	X	
SHASTA											
SIMDIG											
SLICE				X							
SMKSRC		X	X								
SPECID		X									
SPETAB											
SSBC		X	X	X				X			
SSIC		X	X	X				X			
STABAL									X		
STEADY		X	X					X			
STEP1			X	X			X			X	
STEP2			X	X			X			X	
STEP3			X	X			X				
STEP4			X	X				X			
STEP34		X	X	X			X	X		X	
TDRATE			X					X			
TZERO			X	X	X		X	X	X	X	
VERTCL											X
WINDZ				X						X	
WINTER											

G. OVERLAYS

Core space requirements can be reduced by overlaying program or data storage areas. Both of these techniques are used in the Airshed Simulation Program and are described in the following sections.

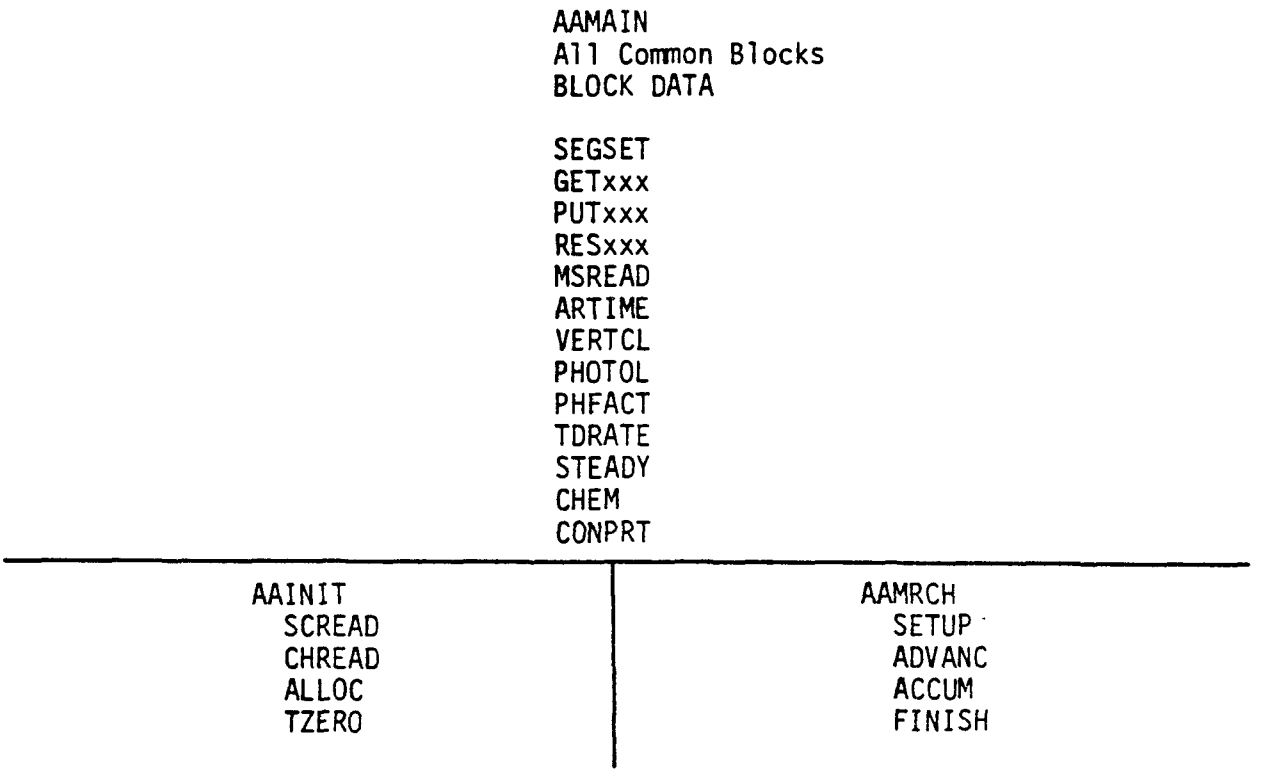
1. Program Overlays

A number of subroutines are used only in the initialization phase of the Airshed Simulation Program and are not required again. These subroutines may be overlaid by subroutines used only in the "march through time" loop.

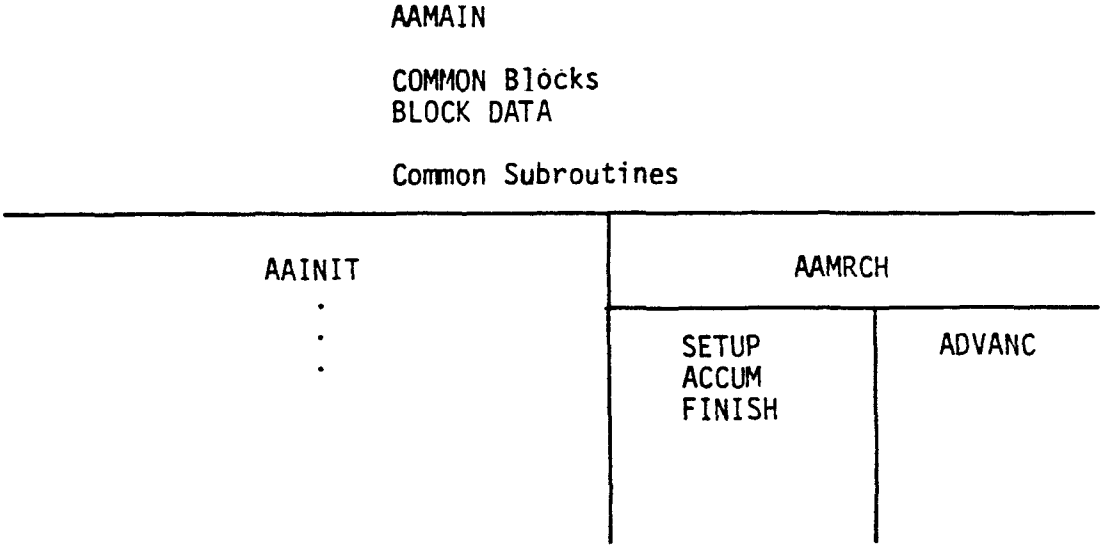
Figure VII-10(a) is a list of subroutines in the simulation program arranged by overlay possibility. At the top is the main program, that calls each major function. Next are the subroutines that are used both in the initialization and simulation phases. These routines should not be overlaid, or they must appear in both overlays. Finally, in two columns, are the initialization and simulation routines that are completely independent of each other and that may be overlaid. None of the lower-level subroutines are named here; see Table VII-4 for the complete list of subroutines that belong in each overlay. This type of program overlay should produce little additional system overhead, since the second program overlay is brought in only once, at the end of the initialization phase, and no further program shuffling is necessary.

If necessary, the "march through time" overlay could be further divided, with the ADVANC portion in one segment and the SETUP, ACCUM, and FINISH routines in another, as shown in Figure VII-10(b). This would necessitate two overlay swaps in each time slice.

Another overlay technique, which could be used with either of the two described above, consists of separating the subroutines DEBUG and CRASH from the other routines called by ADVANC. [See Figures VII-10(c) and VII-10(d).] These routines are called only when negative concentrations are predicted or when the chemistry calculations do not converge. In normal situations, these

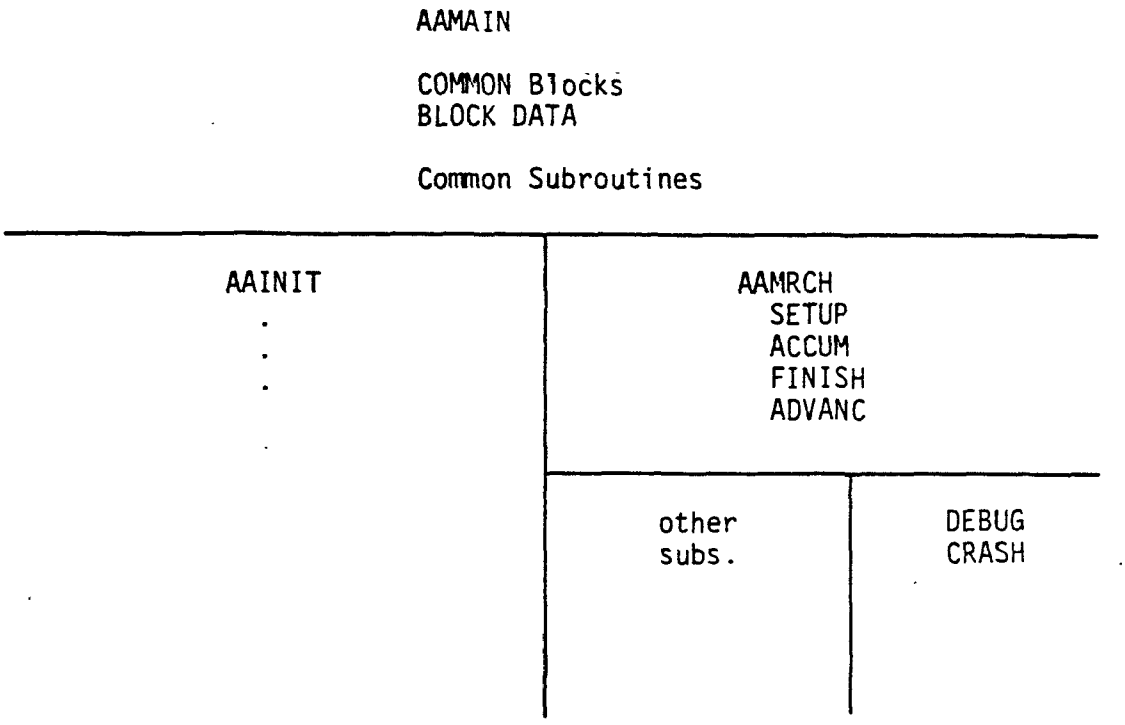


(a) Simple Overlay

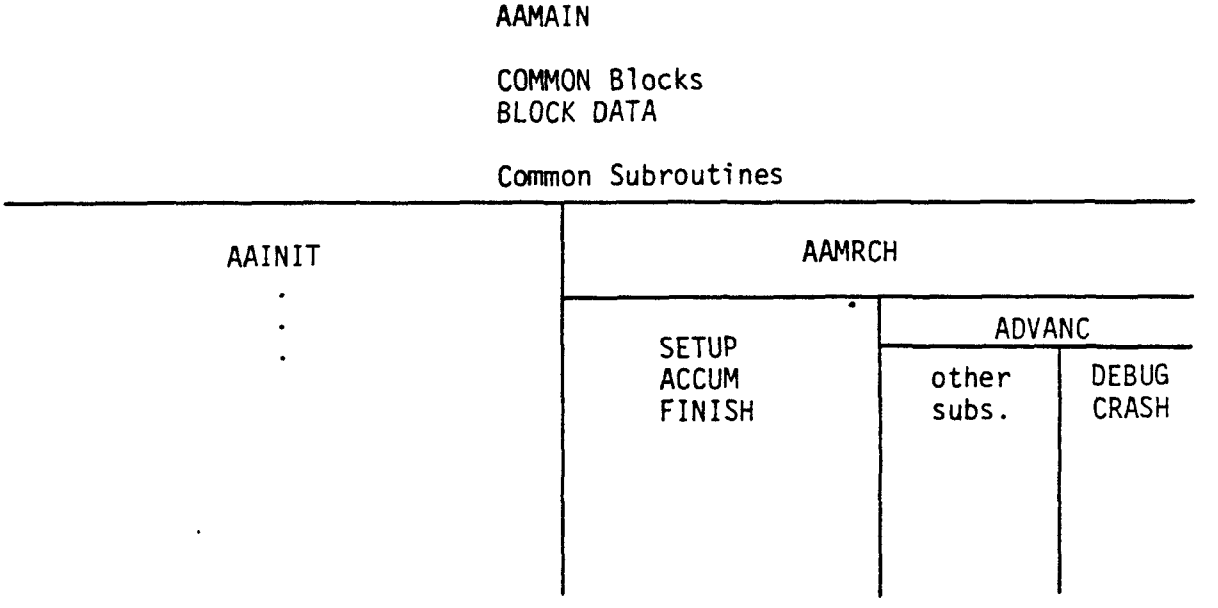


(b) Overlay Scheme with Two Swaps in Each Time Slice

Figure VII-10. Airshed Simulation Program Overlay Structure



(c) A Possible Overlay Scheme with No Swaps



(d) Another Possible Overlay Scheme with Two Swaps

Figure VII-10 (Concluded)

routines would never be called and, therefore, no additional overlay swapping would be required.

Some computer systems have more stringent requirements than others for overlays. On the LBL system, for example, each overlay must be a free-standing main program, called explicitly by the higher level overlays, sharing only COMMON blocks and subroutines within overlays above it, and lacking the ability to receive parameters passed as arguments. On the UNIVAC system, however, overlay segments contain only subroutines, and these will be swapped in as necessary when the subroutines are called. The Airshed Simulation Program has been designed so that the simplest overlay [described in Figure VII-10(a)] is possible on the LBL system. On other systems, to accommodate the largest possible vector of available space, it might be necessary to design overlay structures that are considerably more complex.

2. Data Overlays

The overlaying of data arrays is already handled within the program in the following three ways:

> All data arrays for a single segment are in core at once, but they share core space with corresponding arrays for other segments.

> Arrays containing cumulative concentrations (used to compute time-averages) share space with the instantaneous concentration arrays. Interim instantaneous concentrations read from INTERINST are stored in space ordinarily used by other data arrays.

> The local vectors allocated in subroutine LOCAL are implicitly overlaid by the manner in which their pointers are set. See Section A.2 of this chapter for a further explanation of this procedure.

H. IMPLEMENTING THE PROGRAM

1. Source Deck Structure

The source version of the Airshed Simulation Program consists of labelled COMMON block definitions as well as the main program and subroutines.

For implementation on the UNIVAC, the main program and subroutines are preceded by @ELT,I cards. The elements formed from labelled COMMON block definitions should be processed by @PDP to become PROCs. The insertion of predefined COMMON blocks within subprogram decks is accomplished with INCLUDE statements.

2. LBL Library Functions

There are five LBL functions used in the program (Table VII-7 shows where they appear):

> The EOF function is used to test for end-of-file on file reads. It can be replaced by END=nnn parameters on READ statements.

> The subroutine SECOND returns the elapsed CPU time, in seconds, from the beginning of the job. If there is no comparable function on the host computer, this function can be replaced by calls to other clocking routines or can be removed.

> The subroutine DISPLA prints a message on the CDC Dayfile, a job control stream printout. This function can be replaced by a comparable routine or can be eliminated.

> The function WARN returns a nonzero value when the computing time limit is approaching. This function, too, can be replaced by a comparable routine or can be eliminated.

Table VII-7. LBL Library Functions

Function	Subroutines That Call the Functions
EOF	ADDEM, FILCK, FILCKE, FILSKP, MSREAD, RESxxx
SECOND	TZERO, FINISH
WARN	FINISH
DISPLA	FINISH
RETURNS	CLFILE

> The subroutine RETURNS returns to the system an input file no longer needed. On the UNIVAC, this function can be replaced by a call to CLOSE (see Fortran V Library PRM, pp. 3 to 8).

Note that these functions have been bypassed for implementation on the EPA's UNIVAC computer. They may be employed for CDC computer operation of the Airshed Model.

3. Changing File Assignments

Table VII-8 shows the current assignment of unit numbers to the Airshed Simulation Program's data files. These numbers are assigned in the BLOCK DATA subprogram. If they conflict with standard unit assignments for other purposes, they can be changed simply by changing the BLOCK DATA subprogram.

4. Overlaying

Section G.1 of this chapter describes a number of overlay structures that can be used. It is suggested that the program first be compiled and loaded without overlays to determine the total core requirements of the program and data. Then, the effects of the various program overlay structures on core storage availability can be evaluated.

5. Vector of Available Space

The vector of available space is currently in Blank COMMON in order to reduce the core requirements for loading on the LBL system. If necessary, it can easily be changed by changing the COMMON declaration in the labeled COMMON AVAIL. The dimension of the vector of available space (AV) is currently 26,000 words. After the program has been compiled and loaded, the load map should be examined to determine how much space is actually available for variable length arrays. The dimension of AV can then be changed and the three modules that use it, AAASIM, AAINIT, and AAMRCH, can be recompiled.

6. Use of Large Core Memory

On the LBL system, large core memory rather than a pair of secondary storage files is used to store the Boundary Tables for each segment. Boundary

Table storage can be changed to secondary storage file accesses by using the following steps:

> Remove the COMMON/LCM/ block from AAASIM, GETBON, and PUTBON.

> Change GETBON and PUTBON to read and write files, as GETCON and PUTCON do.

> Restore statements that REWIND NUBONR and NUBONW, which appear after segment loops in TZERO, SETUP, SEGINT, and ADVANC, and have been disabled by C in Column 1.

On the UNIVAC, it may be desirable to use extended memory to store the Boundary Table. In that case, the following steps should be taken:

> Calculate the desired length of the table storage vector by the formula:

$$L = \sum_{seg} \left[8(x_{seg} + y_{seg}) + z*sp*(\text{no. interface cells})_{seg} \right]$$

> Edit the COMMON/LCM/ block to change the length of the table storage vector and to remove the LEVEL statement.

> Change @MAP directives so that the COMMON/LCM/ block is loaded last.

7. Flexibility Versus Efficiency

The Airshed Simulation Program has been designed to maximize flexibility in choosing regional dimensions while optimizing the usage of available core storage. This capability has been accomplished by assigning all spatially varying arrays at run time, when spatial dimensions are specified. This feature provides that the user can specify the largest

region possible given the selection of program options, species and vertical levels, and that the program need not be recompiled for each new use. This provision of flexibility is not, however, without some cost in efficiency. Whenever variably dimensioned arrays are passed as subroutine arguments rather than being declared with fixed dimensions within COMMON blocks, extra instructions are required for each reference. These instructions add to program size and to operating time. It may be desirable, at some time, to tailor the program for a specific application, where the program options, numbers of species and vertical levels, and maximum segment size are all well-known. Then, to achieve maximum efficiency in program operation, the arrays now assigned to available space could be fixed in dimensions, put in COMMON blocks, and removed from argument lists. This task is not conceptually difficult but, practically, it could be quite time consuming, since the use of variably dimensioned arrays in arguments is widespread within the program. The increased efficiency that would result from these changes is not known. Therefore, it is not recommended unless the number and length of computer runs is great enough to justify the changes.

I. Locations of Specific Algorithms

In addition to all the data handling processes that have occupied the bulk of descriptive material in this chapter, the primary function of the Airshed Simulation Program is to perform the integration of the governing equations described in Chapter II of the User's Manual. The routines that perform the numerical integration are called by subroutine ADVANC; routines that calculate steady-state values of initial and boundary conditions are called from subroutine SSIC (called by TZERO) and SSBC (called by ADVANC). Table VII-9 is a list of numerical functions used within the program and the subroutines in which they are performed. See also the hierarchical list of subroutines (Table VII-4) for indications of how the modules fit together.

Table VII-8. File Unit Assignments

File	Number Unit
AIRQUALITY	3
BOUNDARY	11
DIFFBREAK	12
EMISSIONS	13
METSCALARS	7
PTSOURCE	14
REGIONTOP	15
ROADWAY	16
TEMPERATUR	17
TERRAIN	4
TOPCONC	18
WIND	19
CHEMPARAM	2
SIMCONTROL	1
EXECUTION TRACE	6
AVERAGE	21
INSTANT	22
VARIATION	23
INTERINST	30
BON1	37
BON2	38
CON1	33
CON2	34
CUM1	35
CUM2	36
SEG1	31
SEG2	32

Table VII-9. Location of Specific Numerical Functions

Numerical Function	Subroutine
Vertical distribution of grid cells	VERTCL
Horizontal advection and diffusion	STEP1, STEP2, SHASTA
Emissions unit conversion and scaling	MSREAD, RESEM, RESPTS
Surface deposition	SNKSRC
Surface-layer microscale	MICRO
Vertical diffusivity	DIFCOF
Vertical winds	WINDZ
Vertical advection integration	STEP3, SIMDIG
Photolysis rates	PHFACT, PHOTOL
Temperature-dependent rates	TDRATE
Steady-state calculations	STEADY
Chemical mechanism	CHEM
Night chemistry	CHEM2
Chemistry integration	STEP4

VIII THE DISPLAY PROGRAM

Output from the Airshed Simulation Program is written in binary form onto two library storage files that contain instantaneous and time-averaged concentration predictions throughout the day in each grid cell for each modeled chemical species. The display program, in turn, processes either simulation output file; it can be run either back to back with the simulation program or separately later. In the following discussion, important aspects of display program organization and structure are considered: overall logic flow, input/output, subroutine definition and hierarchy, and implementation considerations.

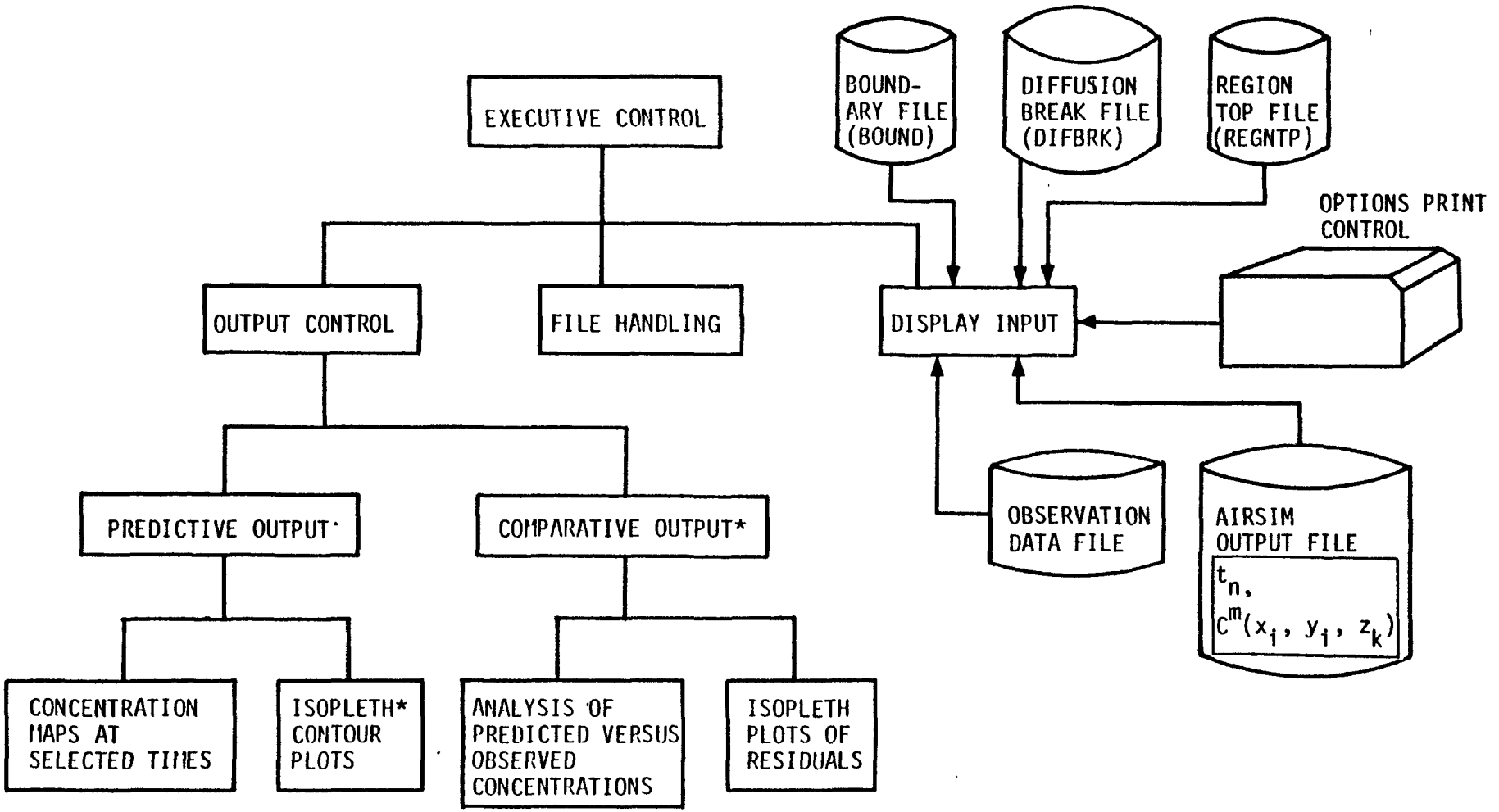
A. OVERALL LOGIC FLOW

The executive control logic of the display program performs its tasks in the following order. Card input data are processed; input files are attached and positioned at the proper simulation hour; the full concentration field for the user-specified vertical level is reconstructed from the segmented simulation output file; and concentration maps as well as station predictions are printed for all desired species and simulation hours. A schematic of this logic flow is presented in Figure VIII-1.

As noted in that figure, the comparative output/analysis portion of the display program is included only in skeletal form in the present version. Although the executive control currently can interface an observation file that is formatted in the same way as the simulation output file, no analysis is performed. Extension of the display program to do so, however, would be straightforward.

B. INPUT/OUTPUT

Input to the display program is of two types: cards and a binary data file. The card input provides print information, desired output options,



* Not available in the current program version.

Figure VIII-1. Schematic of the Display Program Structure

units conversion for printing, and the names and location of stations and landmarks. Details about input parameters and their card formats can be found in Chapter XI of the User's Manual. The data files required by the program include: an airshed simulation output file, which contains segmented concentration predictions from the simulation program [these may be either time-averaged (AVERAGE) or instantaneous (INSTANT) values]; a diffusion break file (DIFFBREAK) which details the behavior of the diffusion break profile during the modeling period; a region top file (REGIONTOP), which specifies the spatial and temporal variation of the top of the modeling region during the day; and a boundary file (BOUNDARY), which contains information necessary to reconstruct from segmentation specifications the overall characteristics of the modeling region.

Output from the display program is designed to assume two forms: prediction and comparison. The first of these includes output that is based solely on concentration values from one file. Such output may consist of concentration maps and station prediction vertical profiles. Output of the second form, though not available in the current program version, would be based on direct comparison of two concentration files. The capability to produce output of this latter type would require only a straightforward extension of the program, since the file handling logic is already included in the executive control.

C. SUBROUTINE DEFINITION AND HIERARCHY

Executive control as well as various subordinate functions (input, file handling, and output control) are handled in the main routine (DISPLY) of the display program. The subroutines used by DISPLY to produce the final program output are listed in Table VIII-1.

As an illustration of subroutine precedence (which routines are called by another routine), Figure VIII-2 presents a subroutine hierarchy diagram. As each subroutine is called by another, its name is shown beneath and indented from the calling routine.

Table VIII-1. Subroutine Descriptions

Routine	Description
CALNDR	Adds one to calendar date
DISPLY	Main program; handles executive control, input
FMT4A1	Converts integer to alpha 4A1 for printing
MAPPER	Cleans up line for printing of concentration map
PRINT1	Produces ground-level concentration maps and vertical concentration profiles above each station
PRMAP	Prints concentration map
XTIME	Performs arithmetic operations on times in minutes and 2400-hour time

```
DISPLY
    EOF*
    INT†
    AMIN1†
    MINO†
    FLOAT†
    ABS†
    PRINT1
        IFIX†
        FMT4A1
        PRMAP
            INT†
            MAPPER
                FMT4A1
```

* CDC-specific function.

† Inline function.

Figure VIII-2. Subroutine Hierarchy Diagram

D. IMPLEMENTATION CONSIDERATIONS

The source deck form of this program is like that of the Airshed Simulation Program; that is, it consists of labeled COMMON blocks, a main program, and subprograms as described in Section VII.H. Unlike the Airshed Simulation Program, the dimensions of its arrays are fixed and are passed to the subroutines in COMMON blocks. Maximum dimensions can be changed if necessary to fit the available core space.

The CDC function, EOF, can be replaced by the appropriate "END =" clause in the READ statement. The subroutine FMT4A1 can be replaced by an ENCODE statement if desired.

IX FILE STRUCTURES

The 26 files in the SAI Airshed Model are divided into three categories on the basis of their relationship to the Airshed Simulation Program: input files, output files, and internal (scratch) files. The files input to the Airshed Simulation Program are created by external data preparation programs as described in Chapter IV. The output files are generated by the Airshed Simulation Program for the storage of data by segment. The usage of these files is discussed in Chapter VII. Figure IX-1 shows the position of each file in the logical structure of the system. This figure is discussed in more detail in Chapter III of the User's Manual; Table IX-1 lists, in alphabetical order, first the input and output files and then the internal files. (The files will also be discussed in this order in the following sections). The table contains the ten-character code name for each file, its longer descriptive name, and the dimensions over which file entries vary. Dimensions are not indicated for the internal files, which are basically core images for each segment.

A. INPUT AND OUTPUT FILE STRUCTURE DEFINITIONS

This section contains, in alphabetical order, the formal descriptions of the input and output files for the Airshed Simulation Program. Each description includes a table (IX-2 through IX-18) illustrating the structure and listing the contents of each file. Also, the descriptions discuss separately the three types of data that may appear: standard header records, time-invariant data, and time-varying data. Furthermore, the contents of each record are completely defined within the text by means of word definition lists. Each data item is described by word number (order), item type (I = integer, R = real, L = logical, A = alphanumeric), and by means of a definition that includes units and dimensionality, if any.

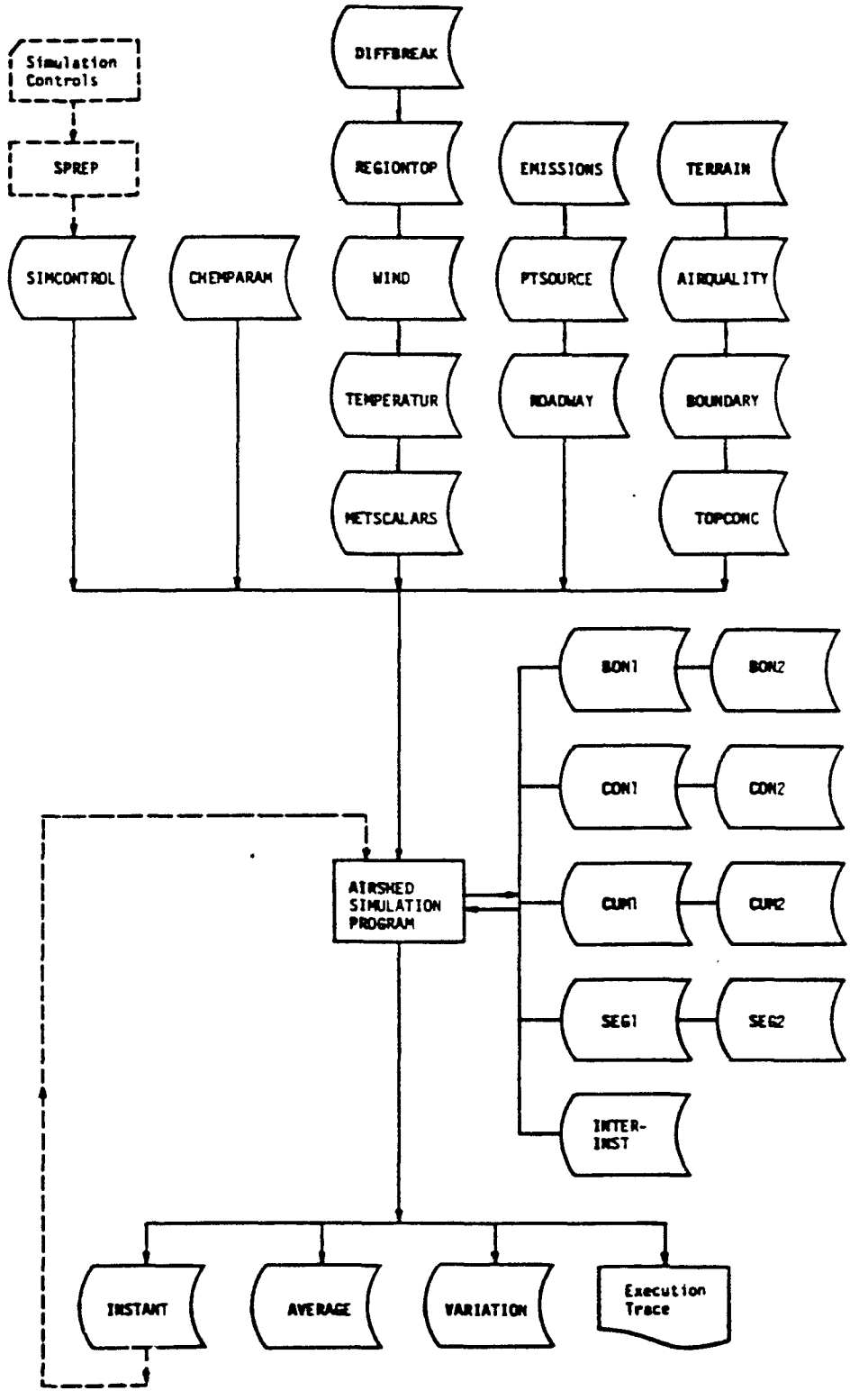


Figure IX-1. Files in the SAI Airshed Model System

Table IX-1. SAI Airshed Model System Files and Their Dimensions

(a) Input and Output Files

Code Name	Descriptive Name	Dimensions*				
AIRQUALITY	Air Quality	x	y	z	s	t
AVERAGE	Average Concentrations	x	y	z	s	t
BOUNDARY	Boundary Definition and Concentrations	(x	y)	z	s	t
CHEMPARAM	Chemistry Parameters				s	
DIFFBREAK	Diffusion Break	x	y			t
EMISSIONS	Ground-Level Emissions	x	y		s	t
INSTANT	Instantaneous Concentrations	x	y	z	s	t
METSCALARS	Meteorological Scalars					t
PTSOURCE	Point Source Emissions	(x	y	z)	s	t
REGIONTOP	Top of the Region	x	y			t
ROADWAY	Roadway Microscale Parameters	x	y		s	t
SIMCONTROL	Simulation Controls					
TEMPERATUR	Temperature	x	y			t
TERRAIN	Terrain	x	y			
TOPCONC	Concentrations at Top of Region	x	y		s	t
VARIATION	Subgrid-Scale Concentration Variation	x	y		s	t
WIND	Wind	x	y	z		t

*x = east-west axis of grid.

y = north-south axis of grid.

z = vertical axis of grid.

s = species.

t = time.

() = data that vary in space but are not present over the entire grid.

Table IX-1 (Concluded)

(b) Internal Files

Code Name	Descriptive Name
BON1	Boundary Table
BON2	Boundary Table
CON1	Concentrations
CON2	Concentrations
CUM1	Cumulative Concentrations
CUM2	Cumulative Concentrations
INTERINST	Interim Instantaneous Concentrations
SEG1	Scratch Table
SEG2	Scratch Table

1. Air Quality

The Air Quality (AIRQUALITY) File contains observed concentration values as a function of time for each species over the region (x,y,z).

a. Header Records

The AIRQUALITY File begins with four standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'AIRQUALITY', ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size --x-direction (meters)
7	R	Grid cell size --y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)

2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word

b. Time-Invariant Data

The AIRQUALITY File contains no time-invariant data.

c. Time-Varying Data

The AIRQUALITY File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region the AIRQUALITY File contains a set of Concentration Records, grouped by species, and ordered within each species by vertical level. The first eleven words of the record identify segment and species:

1	I	Segment number
2-11	A	Species name, ten characters, one character per word

The next group of words is the concentration array itself:

12+	R	Concentrations (ppm, or $\mu g/m^3$ for AEROSOLS) for each cell in one vertical level, varying by x-, then y-directions.

Table IX-2. AIRQUALITY File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span
2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution
3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size
4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time
2) For each segment:
   - For each species:
     - For each level:
       - Concentration Record
         - Segment Number
         - Species name
         - Concentrations (x,y)

2. Average Concentrations

The Average Concentrations (AVERAGE) File contains time-averaged concentrations for each species over the region.

a. Header Records

The AVERAGE File begins with four standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'AVERAGE ', ten characters, one character per word.
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file. For the AVERAGE File, the ending date and time may not correspond to the last date and time on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone.

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)

2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word

b. Time Invariant Data

The AVERAGE File contains no time-invariant data.

c. Time-Varying Data

The AVERAGE File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region the AVERAGE File contains a set of Average Concentration Records, grouped by species, and ordered within each species by vertical level. The first 11 words of the record identify segment and species:

1	I	Segment number
2-11	A	Species name, ten characters, one character per word

The next group of words is the concentration array itself:

12+	R	Concentrations (ppm, or $\mu g/m^3$ for AEROSOLS) averaged over the time interval for each cell in one vertical level, varying by x-, then y-directions.

Table IX-3. AVERAGE File Structure

a. <u>Header Records</u>

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record
   - For each species:
     - Species name

b. <u>Time-Invariant Data</u>--none

c. <u>Time-Varying Data</u>-one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - For each species:
     - For each level:
       - Average Concentration Record
         - Segment number
         - Species name
         - Concentrations (x,y)

3. Boundary Definition and Concentrations

The Boundary Definition and Concentrations (BOUNDARY) File contains the definition of external boundaries and segment interfaces and the concentrations for each species at each external boundary cell over the region (x,y,z).

a. Header Records

The BOUNDARY File begins with four standard header records.

1) The File Description Header Record contains 76 words.

1-10	A	File name = 'BOUNDARY ', ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells:

11	I	Number of cells between surface layer and and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appear in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)

2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word.

b. Time-Invariant Data

The BOUNDARY File contains the definition of the boundaries of the region, both the external boundaries defined in the data preparation and the internal segment interfaces created when the region is segmented.

For each segment there are four Boundary Definition Records, one record for each edge. The edges are defined as follows:

1	Left	West	lower limit for each row
2	Right	East	upper limit for each row
3	Bottom	South	lower limit for each column
4	Top	North	upper limit for each column

Each Boundary Definition Record defines the location of the boundary cells at an edge of a segment. The first three words identify the edge and its dimensions:

1	I	Segment number
2	I	Edge number
3	I	Number of cells on edge (i.e., number of rows or columns)

Then, for each grid index (row or column) along the edge, there is a group of four words that define the boundary location:

4	I	Index, within the segment, of the cell at the edge of the region modeled (i.e., the first or last cell simulated within the row or column). If this number = 0, this row or column is to be omitted from the simulation and the next three numbers are ignored.
5	I	Segment number in which adjacent cell is located. If this number = 0, the boundary is an external one, and the next two numbers are ignored.
6	I	x-index of adjacent cell within segment defined in Word 5.
7	I	y-index of adjacent cell within segment defined in Word 5.

c. Time-Varying Data

The BOUNDARY File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending Time (hours)

2) For each segment of the region, there is a set of Boundary Concentration Records, grouped by species, with four records (one for each edge) for each species. The first twelve words of the record identify segment, species, and edge:

1	I	Segment number.
2-11	A	Species name, ten characters, one character per word.
12	I	Edge number

The next group of words is the boundary concentration array on the vertical plane along the edge:

13+	R	Boundary concentrations (ppm, or $\mu g/m^3$ for AEROSOLS) at each vertical level for each cell along the edge. For rows or columns that are not to be simulated and for edges that represent internal segment boundaries, these numbers must be present, but will be ignored.

Table IX-4. BOUNDARY File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span
2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution
3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size
4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data

- For each segment:
  - For each edge:
    - Boundary Definition Record
      - Segment number
      - Edge number
      - Number of cells on edge
      - For each cell:
        - Index of boundary
        - Location of adjacent cell

Table IX-4 (Concluded)

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - For each species:
     - For each edge:
       - Boundary Concentration Record
         - Segment number
         - Species name
         - Edge number
         - Boundary concentrations (z,cell)

4. Chemistry Parameters

The Chemistry Parameters (CHEMPARAM) File contains the definition of species characteristics, reaction properties, and stoichiometric coefficients.

a. Header Records

The CHEMPARAM File begins with two standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'CHEMPARAM ', ten characters, one character per word.
11-70	A	File identifier, 60 characters, one character per word.
71	I	Number of segments For the CHEMPARAM File, which does not vary spatially, this number = 0 and the Segment Definition Header Record is omitted.
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file. For the CHEMPARAM File, which does not vary in time, these numbers are ignored.

2) The Region Description Header Record is omitted for the CHEMPARAM File, which is independent of modeling region.

3) The Segment Description Header Record is omitted for the CHEMPARAM File, which does not vary spatially.

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word.

b. Time-Invariant Data

The CHEMPARAM File contains two to four time-invariant data records.

1) The Chemistry Definition Record contains 62 words:

1-60	A	Mechanism identifier, 60 characters, one character per word
61	I	Number of reactions
62	I	Number of stoichiometric coefficients

2) The Species Parameters Record contains one group of 18 words for each species; the number of species is defined in the File Definition Header Record:

1-10	A	Species name, ten characters, one character per word
11	L	Reactive/Unreactive Flag .TRUE. = Reactive .FALSE. = Unreactive
12	L	Steady-state for initial concentrations .TRUE. = Compute steady-state values .FALSE. = Use input values

13	L	Steady-state for boundary concentrations .TRUE. = Compute steady-state values .FALSE. = Use input values
14	R	Resistance to surface sinks (hr/m)
15	R	Lower bound for steady-state calculations (ppm)
16	R	Upper bound for steady-state calculations (ppm)
17	R	Lower bound for numerical integration (ppm, or $\mu g/m^3$ for AEROSOLS)
18	R	Upper bound for numerical integration (ppm, or $\mu g/m^3$ for AEROSOLS)

3) The Reaction Parameters Record contains one group of five words for each reaction; the number of reactions is specified in the Chemistry Definition Record (if the number of reactions = 0, this record is omitted):

1	R	Rate constant
2	L	Photolysis flag .TRUE. = Photolysis reaction .FALSE. = Not a photolysis reaction
3	L	Temperature-dependence flag .TRUE. = Temperature-dependent reaction .FALSE. = Not a temperature-dependent reaction If this is FALSE the next two words will be ignored
4	R	Activation energy (deg K)
5	R	Reference temperature for activation energy (deg K)

4) The Stoichiometric Coefficients Record contains one group of eleven words for each stoichiometric coefficient; the number of these coefficients is specified in the Chemistry Definition Record (if the number of coefficients = 0, this record is omitted):

1-10	A	Coefficient name, ten characters, one character per word
11	R	Coefficient value

c. Time-Varying Data

The CHEMPARAM File contains no time-varying data.

Table IX-5. CHEMPARAM File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span
2) Region Description Header Record--omitted
3) Segment Description Header Record--omitted
4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data

1) Chemistry Definition Record
   - Mechanism identifier
   - Number of reactions
   - Number of stoichiometric coefficients
2) Species Parameters Record
   - For each species:
     - Name
     - Reactive flag
     - Steady-state flags
     - Steady-state bounds
     - Numerical integration bounds
     - Resistance to surface sinks

Table IX-5 (Concluded)

3) Reaction Parameters Record

For each reaction:

Rate constant

Photolysis flag

Temperature-dependent flags

Activation energy

Reference temperature

4) Stoichiometric Coefficients Record

For each coefficient:

Name

Value

c. Time-Varying Data--none

5. Diffusion Break

The Diffusion Break (DIFFBREAK) File contains the height of the change of diffusion characteristics over the region (x,y).

a. Header Records

The DIFFBREAK File begins with three standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'DIFFBREAK ', ten characters, one character per word.
11-70	A	File identifier, 60 characters, one character per word.
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species For the DIFFBREAK File, which does not vary with species, this number = 0 and the Species Definition Header Record is omitted.
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells. For the DIFFBREAK File, which does not vary vertically, these words are ignored:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)

14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)
2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record is omitted for the DIFFBREAK File, which does not vary with species.

b. Time-Invariant Data

The DIFFBREAK File contains no time-invariant data.

c. Time-Varying Data

The DIFFBREAK File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)

3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region there is one Diffusion Break Record. The first eleven words of the record identify segment and variable:

1	I	Segment
2-11	A	Variable name = 'DIFFBREAK ' ten characters, one character per word

The next group of words is the diffusion break array itself:

12+	R	Diffusion break (meters) at the beginning of the time interval at each cell in the row for each row in the segment

Table IX-6. DIFFBREAK File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record--omitted

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - Diffusion Break Record
     - Segment number
     - Variable name
     - Diffusion break (x,y)

6. Ground-Level Emissions

The Ground-Level Emissions (EMISSIONS) File contains emissions values for each species over the region (x, y).

a. Header Records

The EMISSIONS File begins with four standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'EMISSIONS ', ten characters, one character per word.
11-70	A	File identifier, 60 characters, one character per word.
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells. For the EMISSIONS File, which does not vary vertically, these words are ignored:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)
2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word.

b. Time-Invariant Data

The EMISSIONS File contains no time-invariant data.

c. Time-Varying Data

The EMISSIONS File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region there is a set of Emissions Records, one record for each species. The first eleven words of the record identify segment and species:

1	I	Segment number
2-11	A	Species name, ten characters, one character per word

The next group of words is the emissions array itself:

12+	R	Ground-level emissions (gram-moles/hour, or grams/hour for AEROSOLS)

Table IX-7. EMISSIONS File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

Table IX-7 (Concluded)

2) For each segment:
   For each species:
      Emissions Record
         Segment number
         Species name
         Ground-level emissions (x,y)

7. Instantaneous Concentrations

The Instantaneous Concentrations (INSTANT) File contains calculated concentration values for each species over the region (x,y,z).

a. Header Records

The INSTANT File begins with four standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'AIRQUALITY'; ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file. For the INSTANT File, the ending date and time may not correspond to the last date and time on the file.

2) The Region Description Header Record contains 15 words.
The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)
2	I	y-location of segment origin with respect to origin of modeling region (grid units)

3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word

b. Time-Invariant Data

The INSTANT File contains no time-invariant data.

c. Time-Varying Data

The INSTANT File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region there is a set of Instantaneous Concentration Records, grouped by species, and ordered within each species by vertical level. The first eleven words of the record identify segment and species:

1	I	Segment number
2-11	A	Species name, ten characters, one character per word

The next group of words is the concentration array itself:

12+	R	Concentrations (ppm, or $\mu g/m^3$ for AEROSOLS) at the beginning of the time interval for each cell in one vertical level, varying by x-, then y-directions.

Table IX-8. INSTANT File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - For each species:
     - For each vertical level:
       - Instantaneous Concentration Record
         - Segment number
         - Species name
         - Concentrations (x,y)

8. Meteorological Scalars

The Meteorological Scalars (METSCALARS) File contains values for meteorological parameters that are assumed to be constant over the entire region.

a. Header Records

The METSCALARS File begins with two standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'METSCALARS', ten characters, one character per word.
11-70	A	File identifier, 60 characters, one character per word.
71	I	Number of segments For the METSCALARS File, which does not vary spatially, this number = 0 and the Segment Definition Header Record is omitted.
72	I	Number of chemical species For the METSCALARS File, which does not vary with species, this number = 0 and the Species Definition Header Record is omitted.
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells. For the METSCALARS File, which does not vary vertically, these words are ignored:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region

13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record is omitted for the METSCALARS File, which does not vary spatially.

4) The Species Description Header Record is omitted for the METSCALARS File, which does not vary with species.

b. Time-Invariant Data

The METSCALARS File contains no time-invariant data.

c. Time-Varying Data

The METSCALARS File contains two records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) The Meteorological Scalars Record contains 66 words, or six name-number pairs:

1-10	A	Name = 'TGRADBELOW'
11	R	Temperature gradient (deg K/meter) below the diffusion break

12-21	A	Name = 'TGRADABOVE', ten characters, one character per word
22	R	Temperature gradient (deg K/meter) above the diffusion break
23-32	A	Name = 'EXPCLASS ', ten characters, one character per word
33	R	Exposure Class
34-43	A	Name = 'RADFACTOR ', ten characters, one character per word
44	R	Radiation Factor at end of time interval
45-54	A	Name = 'CONCWATER ', ten characters, one character per word
55	R	Concentration of H_2O (ppm)
56-65	A	Name = 'ATMOSPRESS', ten characters, one character per word
66	R	Atmospheric pressure (atm.)

Table IX-9. METSCALARS File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record--omitted

4) Species Description Header Record--omitted

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) Meteorological Scalars Record
   - For each scalar:
     - Name
     - Value

9. Point Source Emissions

The Point Source Emissions (PTSOURCE) File contains the location of each point source and the emissions for each species.

a. Header Records

The PTSOURCE File begins with four standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'PTSOURCE ', ten characters, one character per word.
11-70	A	File identifier, 60 characters, one character per word.
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)
2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word.

b. Time-Invariant Data

The PTSOURCE File contains the location and other fixed properties of each point source. For each segment there are two records.

1) The Counter Record contains two words:

1	I	Segment number
2	I	Number of point sources in segment

2) The Point Source Definition Record contains the following group of six words for each point source in the segment If there are no point sources in the segment this record does not appear:

1	R	x-coordinate of point source with respect to reference origin (meters)
2	R	y-coordinate of point source with respect to reference origin (meters)
3	I	x-index of grid cell within the segment

4	I	y-index of grid cell within the segment
5	R	Stack height (meters)
6	R	Stack diameter (meters)

c. Time-Varying Data

The PTSOURCE File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment there is a Counter Record, a Point Source Location Record, and one Point Source Emissions Record for each species. The Counter Record contains two words:

1	I	Segment number
2	I	Number of point sources in segment for this time interval. If the number of point sources defined in the corresponding time-invariant Counter Record is greater than zero, this number must also be greater than zero.

The Point Source Location Record contains the following group of five words for each point source in the segment. If there are no point sources in the segment, this record does not appear:

1	I	x-index within segment of cell to receive emissions
2	I	y-index within segment of cell to receive emissions
3	I	z-index of cell to receive emissions
4	R	Flow rate (m^3/hr)
5	R	Effective plume height (meters)

The Point Source Emissions Record contains the following group of words:

1	I	Segment number
2-11	A	Species name, ten characters, one character per word
12+	R	Emissions (gm-moles/hr, or gm/hr for AEROSOLS) from each point source

Table IX-10. PTSOURCE File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data

1) For each segment:
   - Counter Record
     - Segment number
     - Number of point sources

Table IX-10 (Concluded)

Point Source Definition Record
- For each point source:
  - Location in region
  - Stack properties

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time
2) For each segment:
   - Counter Record
     - Segment number
     - Number of point sources
   - Point Source Location Record
     - For each point source:
       - Indices of cell to receive emissions
       - Flow rate
       - Effective plume height
   - For each species:
     - Point Source Emissions Record
       - Segment number
       - Species name
       - For each point source:
         - Emissions

10. Top of Region

The Top of Region (REGIONTOP) File contains the height of the top of the region (x,y).

a. Header Records

The REGIONTOP File begins with three standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'REGIONTOP ', ten characters, one character per word.
11-70	A	File identifier, 60 characters, one character per word.
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species For the REGIONTOP File, which does not vary with species, this number = 0 and the Species Definition Header Record is omitted.
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells. For the REGIONTOP File, which does not vary vertically, these words are ignored:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)
2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record is omitted for the REGIONTOP File, which does not vary with species.

b. Time-Invariant Data

The REGIONTOP File contains no time-invariant data.

c. Time-Varying Data

The REGIONTOP File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region there is one Top of Region Record. The first eleven words of the record identify segment and variable:

1	I	Segment number
2-11	A	Variable name = 'REGIONTOP ', ten characters, one character per word.

The next group of words is the top of region array itself:

12+	R	Top of region (meters) at the beginning of the time interval at each cell in the row for each row in the segment

Table IX-11. REGIONTOP File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record--omitted

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - Top of Region Record
     - Segment number
     - Variable name
     - Top of region (x, y)

11. Roadway Microscale Parameters

The Roadway Microscale Parameters (ROADWAY) File contains values for parameters used in the surface layer microscale computations, varying over the region (x,y).

a. Header Records

The ROADWAY File begins with three standard header records.

1) The File Description Header Records contains 76 words:

1-10	A	File name = 'ROADWAY ', ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin.

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells. For the ROADWAY File, which does not vary vertically, these words are ignored:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)

2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

b. Time-Invariant Data

The ROADWAY File contains no time-invariant data.

c. Time-Varying Data

The ROADWAY File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region there is one Roadway Microscale Parameter (CARM) Record and a set of Roadway Emissions Records, one record for each species:

In the CARM Record the first 11 words identify the segment number and variable name:

1	I	Segment number
2-11	A	Variable name = 'CARM ', ten characters, one character per word

The next group of words is the CARM array itself:

12+	R	CARM over the segment

In each Roadway Emissions Record the first 11 words identify the segment number and species name:

1	I	Segment number
2-11	A	Species name, ten characters, one character per word

The next group of words is the roadway emissions array itself:

12+	R	Roadway emissions (gram-moles/hour, or grams/hour for AEROSOLS) over the segment

Table IX-12. ROADWAY File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - CARM Record
     - Segment number
     - Variable name
     - CARM (x,y)

Table IX-12 (Concluded)

For each species:

- Roadway Emissions Record
  - Segment number
  - Species name
  - Roadway emissions (x,y)

12. Simulation Controls

The Simulation Controls (SIMCONTROL) File contains calculation options, integration parameters, output controls, and other parameters required by the simulation program.

a. Header Records

The SIMCONTROL File begins with one standard header record.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'SIMCONTROL', ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For the SIMCONTROL File, which does not vary spatially, this number = 0 and the Segment Definition Header Record is omitted.
72	I	Number of chemical species For the SIMCONTROL File, which does not vary with species, this number = 0 and the Species Definition Header Record is omitted.
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file. For the SIMCONTROL File, which does not vary in time, these numbers are ignored.

2) The Region Description Header Record is omitted for the SIMCONTROL File, which is independent of region.

3) The Segment Description Header Record is omitted for the SIMCONTROL File, which does not vary spatially.

4) The Species Description Header Record is omitted for the SIMCONTROL File, which does not vary with species.

b. Time-Invariant Data

The SIMCONTROL File contains one Simulation Controls Record that includes all simulation control variables.

The Simulation Controls Record contains 87 words:

1-60	A	Run identifier--sixty characters, one character per word
61	I	Beginning date of the simulation (Julian)
62	R	Beginning time of the simulation (hours)
63	I	Ending date of the simulation (Julian)
64	R	Ending time of the simulation (hours)
65	L	Restart flag
66	L	Surface sink flag
67	L	Point source flag
68	L	ROADWAY file flag
69	L	TEMPERATUR file flag
70	L	TERRAIN file flag
71	L	Concentration variation flag
72	R	Default surface roughness (meters)
73	R	Default vegetation factor (fraction alfalfa)
74	R	Maximum time slice size (hours)
75	I	Maximum number of steps in a slice
76	R	Minimum step size (hours)
77	I	Maximum number of iterations in chemistry
78	R	Relative error tolerance for convergence in chemistry
79	R	Darkness criterion (same units as RADFACTOR on the METSCALARS file)

80	R	Time interval for instantaneous concentrations output
81	R	Time interval for averaging concentrations
82	I	File history print flag
83	I	Core allocation print flag
84	I	Number of levels for instantaneous concentration print
85	I	Number of levels for average concentration print
86	I	Concentration variation print flag
87	I	Print option not implemented

c. Time-Varying Data

The SIMCONTROL File contains no time-varying information.

Table IX-13. SIMCONTROL File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span
2) Region Description Header Record--omitted
3) Segment Description Header Record--omitted
4) Species Description Header Record--omitted

b. Time-Invariant Data

1) Simulation Control Record
   - Run identifier
   - Time span of simulation
   - Option flags
   - Default terrain values
   - Integration controls
   - Output intervals
   - Print options

c. Time-Varying Data--none

13. Temperature

The Temperature (TEMPERATUR) File contains ground-level temperatures over the region (x,y).

a. Header Records

The TEMPERATUR File begins with three standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'TEMPERATUR', ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species For the TEMPERATUR File, which does not vary with species, this number = 0 and the Species Definition Header Record is omitted.
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells. For the TEMPERATUR File, which does not vary vertically, these words are ignored:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)

2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record is omitted for the TEMPERATUR File, which does not vary with species.

b. Time-Invariant Data

The TEMPERATUR File contains no time-invariant data.

c. Time-Varying Data

The TEMPERATUR File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region the TEMPERATUR File contains one Temperature Record. The first eleven words of the record identify segment and variable:

1	I	Segment number
2-11	A	Variable name = 'TEMPERATUR', ten characters, one character per word.

The next group of words is the temperature array itself:

12+	R	Temperature (deg K) at ground level at each cell in the row for each row in the segment

Table IX-14. TEMPERATUR File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record--omitted

b. Time Invariant Data--none

c. Time Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - Temperature Record
     - Segment number
     - Variable name
     - Temperature (x,y)

14. Terrain

The Terrain (TERRAIN) File contains surface roughness values and vegetation factors over the region (x,y).

a. Header Records

The TERRAIN File begins with three standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'TERRAIN ', ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species For the TERRAIN File, which does not vary with species, this number = 0 and the Species Definition Header Record is omitted.
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file. For the TERRAIN File, which does not vary in time, these numbers are ignored.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells. For the TERRAIN File, which does not vary vertically, these words are ignored:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)
2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record is omitted for the TERRAIN File, which does not vary with species.

b. Time-Invariant Data

For each segment of the region the TERRAIN File contains two records.

1) The Surface Roughness Record begins with eleven words that identify segment and variable:

1	I	Segment number
2-11	A	Variable name = 'ROUGHNESS ', ten characters, one character per word

The next group of words is the surface roughness array itself:

12+	R	Surface roughness (meters) over the segment

2) The Vegetation Factor Record begins with eleven words that identify segment and variable:

1	I	Segment number
2-11	A	Variable name = 'VEGFACTOR ', ten characters, one character per word

The next group of words is the vegetation factor array itself:

12+	R	Vegetation factors over the segment (fraction of alfalfa absorption)

c. Time-Varying Data

The TERRAIN File contains no time-varying data.

Table IX-15. TERRAIN File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record--omitted

b. Time-Invariant Data

1) For each segment:
   - Surface Roughness Record
     - Segment number
     - Variable name
     - Surface roughness (x,y)
   - Vegetation Factor Record
     - Segment number
     - Variable name
     - Vegetation factor (x,y)

c. Time-Varying Data--none

15. Concentrations at Top of Region

. The Concentrations at Top of Region (TOPCONC) File contains the concentration values for each species at the top boundary of the region (x,y).

a. Header Records

The TOPCONC File begins with four standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'TOPCONC ', ten characters, one character per word.
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells. For the TOPCONC File, which does not vary vertically, these words are ignored:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)

2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word.

b. Time-Invariant Data

The TOPCONC File contains no time-invariant data.

c. Time-Varying Data

The TOPCONC File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region there is a set of Top Concentration Records, one record for each species. The first eleven words of the record identify segment and species:

1	I	Segment number
2-11	A	Species name, ten characters, one character per word

The next group of words is the concentration array itself:

12+	R	Concentration at top of region (ppm, or $\mu g/m^3$ for AEROSOLS) at each cell in the row for each row in the segment

TABLE IX-16. TOPCONC File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - For each species:
     - Top Concentration Record
       - Segment number
       - Species name
       - Concentration at top (x,y)

16. Concentration Variations

The Concentration Variations (VARIATION) File contains time-averaged subgrid-scale concentration variations in the surface layer for each species.

a. Header Records

The VARIATION File begins with four standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'VARIATION ', ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word
71	I	Number of segments For unsegmented files, this number = 1
72	I	Number of chemical species
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the lcoation of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells.

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)

2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record contains ten words for each species; the number of species is defined in the File Description Header Record:

1-10	A	Species name, ten characters, one character per word.

b. Time-Invariant Data

The VARIATION File contains no time-invariant data.

c. Time-Varying Data

The VARIATION File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region there is a set of Concentration Variation Records, one record for each species. The first eleven words of the record identify segment and species:

1	I	Segment number
2-11	A	Species name, ten characters, one character per word

The next group of words is the concentration variation array itself:

12+	R	Subgrid-scale concentration variation (ppm, or $\mu g/m^3$ for AEROSOLS), averaged over the time interval, for each cell in the surface layer.

Table IX-17. VARIATION File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record
   - For each species:
     - Species name

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:
   - For each species:
     - Concentration Variation Record
       - Segment number
       - Species name
       - Concentration variation (x,y)

17. Wind

The Wind (WIND) File contains the x- and y-components of the wind vector at each cell in the region (x,y,z).

a. Header Records

The WIND File begins with three standard header records.

1) The File Description Header Record contains 76 words:

1-10	A	File name = 'WIND ', ten characters, one character per word
11-70	A	File identifier, 60 characters, one character per word.
71	I	Number of segments For unsegmented files, this number = 1.
72	I	Number of chemical species For the WIND File, which does not vary with species, this number = 0 and the Species Definition Header Record is omitted.
73	I	Beginning date of the file (Julian)
74	R	Beginning time of the file (hours)
75	I	Ending date of the file (Julian)
76	R	Ending time of the file (hours)

Words 73-76 describe the total time span contained on the file.

2) The Region Description Header Record contains 15 words. The first three words define the reference origin:

1	R	x-coordinate (UTM units)
2	R	y-coordinate (UTM units)
3	I	UTM zone

The next two words define the location of the modeling region with respect to the reference origin:

4	R	x-location (meters)
5	R	y-location (meters)

The next two words define the size of each grid cell in the x- and y-directions, respectively:

6	R	Grid cell size--x-direction (meters)
7	R	Grid cell size--y-direction (meters)

The next three words define the size of the modeling region in grid cells:

8	I	Number of grid cells--x-direction
9	I	Number of grid cells--y-direction
10	I	Number of grid cells--z-direction

The last five words describe the vertical distribution of grid cells:

11	I	Number of cells between surface layer and diffusion break
12	I	Number of cells between diffusion break and top of region
13	R	Height of surface layer (meters)
14	R	Minimum height of cells between surface layer and diffusion break (meters)
15	R	Minimum height of cells between diffusion break and top of region (meters)

3) The Segment Description Header Record contains one group of four words for each segment; the number of segments appears in the File Description Header Record:

1	I	x-location of segment origin with respect to origin of modeling region (grid units)
2	I	y-location of segment origin with respect to origin of modeling region (grid units)
3	I	Number of grid cells in segment--x-direction
4	I	Number of grid cells in segment--y-direction

4) The Species Description Header Record is omitted for the WIND File, which does not vary with species.

b. Time-Invariant Data

The WIND File contains no time-invariant data.

c. Time-Varying Data

The WIND File contains one set of the following records for each time interval.

1) The Time Interval Record contains four words:

1	I	Beginning date (Julian)
2	R	Beginning time (hours)
3	I	Ending date (Julian)
4	R	Ending time (hours)

2) For each segment of the region, there is a set of records: one for scalar values of interest, and, for each vertical level, two records containing x- and y-components of wind, respectively.

The Wind Scalars Record contains eight words:

1	I	Segment number
2	R	Reference height of wind measuring stations (meters)
3	R	Maximum absolute value of wind speed in x-direction (meters/hour)

4	R	Maximum absolute value of wind speed in y-direction (meters/hour)
5	R	Average wind speed in x-direction (meters/hour) at west boundary
6	R	Average wind speed in x-direction (meters/hour) at east boundary
7	R	Average wind speed in y-direction (meters/hour) at south boundary
8	R	Average wind speed in y-direction (meters/hour) at north boundary

In the x-Wind Record the first eleven words identify the segment number and variable name:

1	I	Segment number
2-11	A	Variable name = 'WINDX ', ten characters, one character per word

The next group of words is the X-wind array itself:

12+	R	Wind in x-direction (meters/hour) over the segment for one vertical level

In the Y-wind Record the first eleven words identify the segment number and variable name:

1	I	Segment number
2-11	A	Variable name = 'WINDY ', ten characters, one character per word

The next group of words is the Y-wind array itself:

12+	R	Wind in y-direction (meters/hour) over the segment for one vertical level

Table IX-18. WIND File Structure

a. Header Records

1) File Description Header Record
   - File name
   - File identifier
   - Number of segments
   - Number of species
   - Time span

2) Region Description Header Record
   - Reference origin
   - Grid origin
   - Grid cell size
   - Region size
   - Vertical distribution

3) Segment Description Header Record
   - For each segment:
     - Segment origin
     - Segment size

4) Species Description Header Record--omitted

b. Time-Invariant Data--none

c. Time-Varying Data--one set per time interval

1) Time Interval Record
   - Beginning date and time
   - Ending date and time

2) For each segment:

Table IX-18 (Concluded)

Wind Scalars Record

- Segment number
- Maximum absolute wind speeds
- Average wind speeds at each boundary

For each level:

- X-wind Record
  - Segment number
  - Variable name
  - x-wind (x,y)
- Y-wind Record
  - Segment number
  - Variable name
  - y-wind (x,y)

B. INTERNAL FILE STRUCTURE DEFINITIONS

The internal files are used to store data by segment; they are not intended to be used outside the Airshed Simulation Program. With the exception of the INTERINST file, the internal files appear in identically formatted pairs. During the sequential processing of segments, one of each pair will be an input file containing current values and the other will be an output file containing new values. At the end of the segment loop, both files will be rewound and their functions switched, so that the new values become input for the next processing loop. A more detailed description of the usage of these files is given in Chapter VII.

1. Boundary Table Files

The internal Boundary Table files (BON1 and BON2) are binary files containing one record for each segment. Each record is simply a core image of the Boundary Table for a particular segment. (See Section VII.A.3 for the structure of this table.)

The number of words in the record for a given segment is calculated in the following way:

$$8(n_x + n_y) + n_z\, n_s\, n_i \quad ,$$

where n_x = number of cells in x-direction
n_y = number of cells in y-direction
n_z = number of vertical levels
n_s = number of chemical species
n_i = number of interface cells in the segment, i.e., two times the length of all nonexternal edges.

2. Instantaneous Concentration Storage Files

The internal Instantaneous Concentration Storage files (CON1 and CON2) are binary files containing one record for each segment. Each record is simply

a core image of the concentration and variation arrays in the Segment Scratch Table for a particular segment (see Section VII.A.4 for the structure of this table).

The number of words in the record for a given segment is calculated in the following way:

$$n_x \, n_y \, n_s \, (n_z + v),$$

where n_x = number of cells in x-direction,
n_y = number of cells in y-direction,
n_z = number of vertical levels,
n_s = number of species,
$v = \begin{cases} 1, \text{ if concentration variability is calculated;} \\ 0, \text{ otherwise.} \end{cases}$

3. Cumulative Concentration Storage Files

The internal Cumulative Concentration Storage files (CON1 and CON2) are binary files containing one record for each segment. Each record is simply a core image of the cumulative concentration and variation arrays in the Segment Scratch Table for a particular segment (see Section VII.A.4 for the structure of this table).

The number of words in the record for a given segment is calculated in the following way:

$$n_x \, n_y \, n_s \, (n_z + v),$$

where n_x = number of cells in x-direction,
n_y = number of cells in y-direction,
n_z = number of vertical levels,
n_s = number of species,
$v = \begin{cases} 1, \text{ if concentration variability is calculated;} \\ 0, \text{ otherwise.} \end{cases}$

4. Interim Instantaneous Concentrations File

The Interim Instantaneous Concentrations file (INTERINST) is a binary file containing, for each segment, a history of instantaneous concentrations calculated during a single time slice. At the end of the time slice, the file is read, the concentrations are weighted and added into the cumulative concentration arrays, and the file is rewound for the start of the next time slice. A more detailed description of the usage of this file is given in Section VII.D. The INTERINST file contains the following concentration history for each segment: There is one set of records written at the beginning of the time slice, followed by one set of records written at the end of each time step. Each set of records consists of one Time Record, followed by a Concentration Record and a Variation Record (if required) for each species.

a. The Time Record contains three words:

1	I	Segment number
2	I	Date (Julian)
3	R	Time (hours)

b. The Concentration Record contains the concentration array itself:

1+	R	Concentrations (ppm, or $\mu g/m^3$ for AEROSOLS) at the given time for each cell in the segment, varying by x-, then y-, then z-directions.

c. The Variation Record will follow the Concentration Record for each species if concentration variation is calculated. It contains the concentration variation array itself:

1+	R	Concentration variation (ppm, or $\mu g/m^3$ for AEROSOLS) at the given time for each cell in the segment, varying by x-, then y-directions.

5. Segment Scratch Storage Files

The internal Segment Scratch Storage files (SEG1 and SEG2) are binary files containing one record for each segment. Each record is simply a core image of the data arrays (excluding concentrations) in the Segment Scratch Table (see Section VII.A.4 for the structure of this table).

The number of words in the record for a given segment is calculated as follows:

$$n_x\, n_y\, [8 + 2\text{terr} + \text{temp} + \text{road} + 2n_z + n_s(2 + \text{road})] + n_p(2 + n_s) \quad ,$$

where n_x = number of cells in x-direction,

n_y = number of cells in y-direction,

n_z = number of vertical levels,

n_s = number of chemical species,

$$\text{terr} = \begin{cases} 1, & \text{if TERRAIN file input;} \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{temp} = \begin{cases} 1, & \text{if TEMPERATUR file input;} \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{road} = \begin{cases} 1, & \text{if ROADWAY file input;} \\ 0, & \text{otherwise,} \end{cases}$$

n_p = number of point sources.

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO.	2.	3. RECIPIENT'S ACCESSION NO.
4. TITLE AND SUBTITLE SAI AIRSHED MODEL OPERATIONS MANUALS Volume II -- Systems Manual		5. REPORT DATE 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) J. Ames, S.R. Hayes, T.C. Myers, D.D. Whitney		8. PERFORMING ORGANIZATION REPORT NO. EM78-79R2
9. PERFORMING ORGANIZATION NAME AND ADDRESS Systems Applications, Inc. 101 Lucas Valley Road San Rafael, CA 94903		10. PROGRAM ELEMENT NO. CDWA1A/01-5006 (FY-85) 11. CONTRACT/GRANT NO. Contract No. 68-02-2429
12. SPONSORING AGENCY NAME AND ADDRESS Atmospheric Sciences Research Laboratory -- RTP, NC Office of Research and Development U.S. Environmental Protection Agency Research Triangle Park, North Carolina 27711		13. TYPE OF REPORT AND PERIOD COVERED (8/76 - 9/79) 14. SPONSORING AGENCY CODE EPA/600/09

15. SUPPLEMENTARY NOTES

16. ABSTRACT

This report describes the Systems Applications, Inc. (SAI) Airshed Model System from a programmer's point of view. Included are discussions of all subroutines and how they fit together, run-time core allocation techniques, internal methods of segment handling using secondary storage, and detailed definitions of the structure of each file in the system. This manual also includes suggestions for implementing the programs on different computers.

The SAI Airshed Model System consists of 17 programs that communicate with each other through 17 data files; both the programs and files are described in detail in this manual. Chapter II presents an overview of the system, briefly describing the programs used to perform each of the five system functions. Chapter III contains guidelines for implementing the programs, and Chapters IV through VIII contain detailed descriptions of the programs by function, including hierarchical subroutine tables and short module descriptions. Chapter IX contains detailed file structure definitions for each file in the system.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group

18. DISTRIBUTION STATEMENT RELEASE TO PUBLIC	19. SECURITY CLASS *(This Report)* UNCLASSIFIED 20. SECURITY CLASS *(This page)* UNCLASSIFIED	21. NO. OF PAGES 22. PRICE

EPA Form 2220-1 (Rev. 4-77) PREVIOUS EDITION IS OBSOLETE