AD-A048 982 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAI--ETC F/G 13/1
THE BUILDING LOADS ANALYSIS AND SYSTEM THERMO-DYNAMICS (BLAST) --ETC(U)
DEC 77 D C HITTLE

UNCLASSIFIED CERL-TR-E-119-VOL-2 CEEDO-TR-77-35-VOL-2 NL

1 OF 5

ADA048982

CEEDO-TR-77-35

CERL-TR-E-119

12

BLAST

THE BUILDING LOADS ANALYSIS AND SYSTEM THERMODYNAMICS PROGRAM

VOL. II: REFERENCE MANUAL

BY D.C. HITTLE
U.S. ARMY CONSTRUCTION ENGINEERING
RESEARCH LABORATORY , CHAMPAIGN, ILLINOIS

SPONSORED BY:

AIR FORCE CIVIL AND
ENVIRONMENTAL ENGINEERING
DEVELOPMENT OFFICE,
AIR FORCE SYSTEMS COMMAND,
TYNDALL AIR FORCE BASE, FLORIDA

AND

DEPARTMENT OF THE ARMY,
OFFICE OF THE CHIEF
OF ENGINEERS,
WASHINGTON, D.C.

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 CERL-TR-E-119 VOL-2 CEEDO-TR-77-35	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 THE BUILDING LOADS ANALYSIS AND SYSTEM THERMODYNAMICS (BLAST) PROGRAM. VOLUME II. REFERENCE MANUAL.		5. TYPE OF REPORT & PERIOD COVERED 9 FINAL rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 D. C./Hittle		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS CONSTRUCTION ENGINEERING RESEARCH LABORATORY P.O. Box 4005 Champaign, IL 61820		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4A762731AT41-06-012 17 Ø6, Ø1
11. CONTROLLING OFFICE NAME AND ADDRESS CONSTRUCTION ENGINEERING RESEARCH LABORATORY and AIR FORCE CIVIL ENGINEERING CENTER Tyndall AFB, FL 32401		12. REPORT DATE 11 Dec[redacted]977
		13. NUMBER OF PAGES 435 12 437p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

18 CEEDO 19 TR-77-35-V-2

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

16 4A762731AT41, 21Ø2

18. SUPPLEMENTARY NOTES

Copies are obtainable from National Technical Information Service
Springfield, VA 22151

Vol 1 A048734

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

energy consumption
subprograms
central energy plant

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Building Loads Analysis and System Thermodynamics (BLAST) program is a sophisticated set of subprograms for predicting energy consumption in buildings. The four major subprograms are: (1) the input processor, which parses the high-level input language and sets up the building/systems/plant descriptions; (2) the building loads subprogram, which computes the hourly space load in a building or zone based on the user's description of the building/zone and hourly weather data; (3) the air distribution system simulation

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE 4Ø5279 UNCLASSIFIED JW

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 continued.

subprogram, which calculates the coil energy demands, fan power, etc., based on the user's description of the air handling system and the hourly space load data calculated by the previous subprogram; and ~~(4)~~ the central energy plant simulation subprogram, which calculates energy consumption of a central/solar/total energy plant based on the user's description of the plant and the hourly coil loads calculated by the previous subprogram, and performs a life-cycle cost analysis of the plant. In addition to conventional boiler-chiller equipment, the central energy plant subprogram includes solar heating and cooling systems, total energy systems, and commercial utility systems. The program is written in Control Data Corporation (CDC) FORTRAN Extended, Version 4, and can be used on CDC 6000/7000 series computers without major modifications.

Volume I of this report provides detailed user instructions. This volume ~~(Volume II)~~ is the reference manual for BLAST and contains descriptions of all BLAST subprograms, as well as structural algorithm charts where appropriate.

FOREWORD

This research was conducted by the U.S. Army Construction Engineering Research Laboratory (CERL) for the Air Force Civil Engineering Center (AFSC), Tyndall AFB, FL, under RDT&E Program 63723F, Project 2102, "Mission Support," Task 01, "Aerospace Structures," and Work Unit 03, "Optimization of Energy Usage in Military Facilities," and for the Directorate of Military Construction, Office of the Chief of Engineers (OCE), Department of the Army, under RDT&E program 6.27.31A, Project 4A762731AT41, "Design, Construction, and Operation and Maintenance Technology for Military Facilities," Task 06, "Energy Systems," Work Unit 012, "Development of Total Energy Systems for Military Facilities," and Work Unit 021, "Solar Energy for Heating and Cooling of Buildings." The applicable QCRs are 1.05.005 and 1.05.008. The Air Force portion of this effort was accomplished under the auspices of the Air Force Civil Engineering Center (AFCEC). On 8 April 1977, AFCEC was organized into two organizations. AFCEC became part of the Air Force Engineering and Services Agency (AFESA). The R and D Function remains under Air Force Systems Command as Det 1 (CEEDO) HQ ADTC. Both units remain at Tyndall AFB, FL 32403.

Mr. F. Beason was the Air Force Technical Monitor, and Messrs H. Maschke and S. Hiratsuka were the OCE Technical Monitors. Mr. D. C. Hittle was the CERL Principal Investigator. Administrative support provided by Dr. D. J. Leverenz, Chief of the CERL Energy Systems

Branch, and Mr. R. G. Donaghy, Chief of the CERL Energy and Power Division, is acknowledged.

The BLAST program was authored by G. Walton, D. Herron, L. Lawrie, D. Hittle, J. Cameron, M. Knowles, G. Shih, A. Mech, W. Stoecker, K. Morgan, A. Itzkowitz, and R. Blanton, all of CERL, and by the Consultants Computation Bureau, San Francisco, CA. The BLAST program is copyrighted by CERL.

COL J. E. Hays is Commander and Director of CERL, and Dr. L. R. Shaffer is Technical Director.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURES (cont'd)

THE BUILDINGS LOADS ANALYSIS AND SYSTEM THERMODYNAMICS (BLAST) PROGRAM--VOLUME II: PROGRAM REFERENCE MANUAL

1 INTRODUCTION

Background

The Building Loads Analysis and System Thermodynamics (BLAST) program is a comprehensive computer program for estimating (1) hourly space heating and cooling requirements, (2) hourly performance of fan systems, and (3) hourly performance of a conventional heating and cooling plant, total energy plant, and/or solar energy system. The BLAST program, which was developed at the U.S. Army Construction Engineering Research Laboratory (CERL), consists of four major subprograms: the input processor, the building loads subprogram, the air distribution system simulation subprogram, and the central energy plant simulation program. BLAST is used with a separately running program which generates the yearly weather file used by BLAST. The BLAST system of programs is written in Control Data Corporation (CDC) FORTRAN Extended, Version 4, and can be used on CDC 6000/7000 series computers without major modifications.

Purpose and Scope

This reference manual constitutes the external documentation for BLAST. It is not intended to be used alone. Readers are encouraged to read Volume I of this report--the BLAST user instructions[1]--before proceeding, in order to gain the necessary background for the documentation that follows. This reference manual describes the overall organization, the major data structures, and the purpose of each of the subroutines in the program. Users who want to know the details of calculation procedures should study the program code which contains extensive internal documentation. This manual can serve as a guide to the program code.

General Description

The BLAST system consists of two separately running programs--WIFE (Weather Information File Encoder) and BLAST proper. WIFE generates the yearly weather file used by BLAST. BLAST combines four main subprograms: PARSE, SIMBLD, SIMSYS, and SIMTEP.

PARSE accepts the high-level user input language and transforms or generates the data which describe the building/zones, air handlers, or

---

[1]D. C. Hittle, *The Building Loads Analysis and System Thermodynamics (BLAST) Program, Volume I: User Instructions*, CERL Technical Report E-119 and CEEDO-TR-77-35 (U.S. Army Construction Engineering Research Laboratory [CERL], and Air Force Civil and Environmental Engineering Development Office [CEEDO], 1977).

central energy plants to be simulated. SIMBLD uses a linearized set of heat balance equations to simulate the hourly thermodynamic activity of each zone described, and then calculates the corresponding hourly space load. SIMSYS takes the hourly space loads generated by SIMBLD and simulates the hourly thermodynamic activity of each air handling system as it tries to meet the loads for a given set of zones, calculating the corresponding coil and fan system energy demands. SIMTEP takes the hourly energy demands generated by SIMSYS and simulates the hourly thermodynamic activity of each central plant as it tries to meet these demands, calculating the energy needed to do so.

Simulation is attempted for each system in the order the user specifies in the input. However, the simulations also have some interdependence. For example, simulating a central energy plant requires coil energy demands as well as the user's input plant description. These demands may be generated during the same run (but previous to the plant simulation) or attached from a previous run. Each subprogram can be run separately, provided the appropriate load files are available. Thus, two or more central plant configurations can be run without rerunning SIMBLD or SIMSYS.

Figure 1 illustrates the overall program and file structure for BLAST. BLAST (the main program) calls PARSE to process the user input and then PERDS (PERform Desired Simulations), which determines which simulations to perform based on user input. The MSIMDF (Make

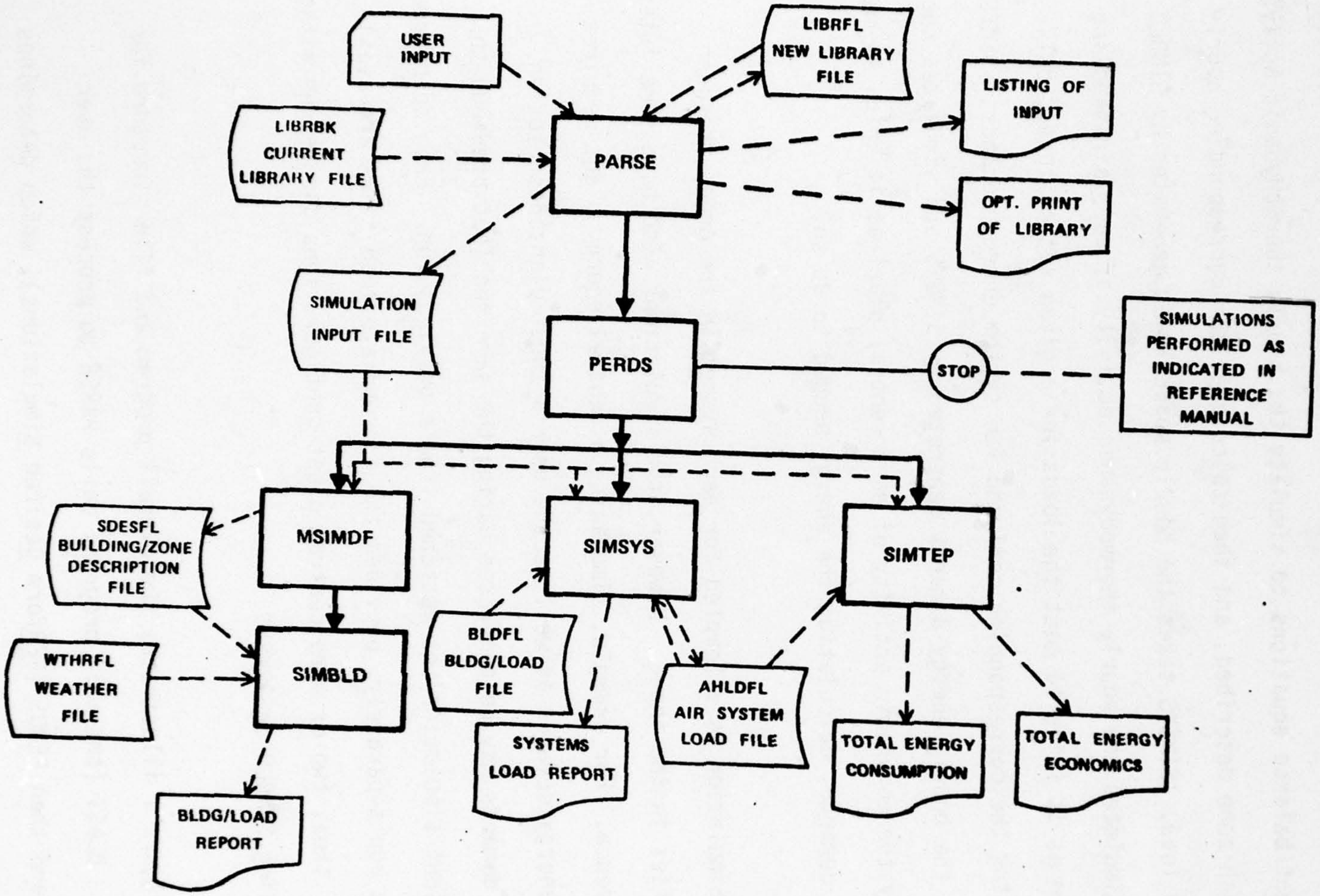


Figure 1. BLAST program and file structure.

SIMulation Description File) routine, if called, uses its subroutines to perform initialization and one-time calculations in preparation for the hourly calculation of space loads. The hourly simulations specified are performed by the routines called by SIMBLD, SIMSYS, and SIMTEP as described above.

Organization of Manual

The next 11 chapters of this reference manual describe the main BLAST routines (Chapter 2), the BLAST input language processing routines and the file they create (Chapters 3 and 4), the BLAST program library (Chapter 5), the routines used and files created by MSIMDF (Chapter 6 and 7), the building load calculation routines and files (Chapters 8 and 9), the air distribution simulation routines and files (Chapters 10 and 11), and the central plant simulation routines (Chapter 12). Chapter 13 describes routines shared by SIMBLD, SIMSYS, and SIMTEP, and Chapter 14 describes the WIFE program used to process weather data tapes.

2 BLAST MAIN ROUTINES

As previously indicated, two simple, high-level routines serve as main drivers for the BLAST program--BLAST and PERDS. BLAST is the primary driver (also the MAIN program) for the BLAST system. It does some trivial initialization and calls PARSE and PERDS. Once PARSE, which contains the outermost program loops, has processed the user input, PERDS is called. PERDS controls the remainder of program execution; it performs some checking and reporting of simulation types and, on the basis of the processed input, determines what (if any) simulations are to be performed.

In addition to the above two major routines, several simple utility routines are used throughout the BLAST system. Table 1 briefly describes these routines.

Several dump routines were also used in the development of the program. They dump intermediate calculations with minimum format or descriptive information. A full program listing is necessary to use dump information. Dumps are initiated by logical flags which are not normally user-controlled. Table 2 lists the existing dump routines.

Table 1

BLAST Basic System Routines

Routine	Description
ZERO	Zeroes an array
ABORT	Fatal error processing routine
ERROR	Prints error message (contained in common block ERRMSG)
ERROR2	Fills error message array with simple message and calls ERROR
INSERT	Inserts characters into array successively on a character-by-character basis
DMPDATA	Dumps successive locations of core (arrays, common blocks, etc.) in a variety of formats
TIMER	Used by certain routines to create trace output
PAGE	Paging routine used by various routines
ZMSG	Resets error message array
BLKFL	Blank fill of zero bytes in words
RECOVRD	Used for automatic recovery from system errors
SETUP	Performs initialization of basic system, reading system sense switches
DMPSINF	Dumps information about simulation input file
FINDNO	Finds occurrence of element in passed array, returns index

Table 1 (cont'd)

ENTRNO	Enters element into passed array at current location
MOVFTN	Copies one array to another
CPYBLK	Copies one array to another

Table 2

Dump Routines

Routine	Description
DUMPBI	Dumps building input file
DUMPHL	Dumps hourly loads data
DUMPHW	Dumps hourly weather data
DUMPITH	Dumps inside temperature histories
DUMPOTH	Dumps outside temperature histories
DUMPQB	Dumps left-side matrix of heat balance
DUMPRS	Dumps right-side matrix of heat balance
DUMPSS	Dumps heating/cooling system status
DUMPST	Dumps surface temperatures
DUMPTC	Dumps time counter information
DUMPZI	Dumps zone input file

3 BLAST INPUT LANGUAGE PROCESSOR

The BLAST input language was defined using formal language techniques. A Backus Naur Form (BNF) of the LALR[2] language was input to LaLonde parser generator which created tables (Fortran arrays) used by the BLAST parser to recognize the user input. The table-driven parser technique allows relatively compact code, relatively easy changes to the language, and some error correction.[3]

The major parts of the BLAST input language processor are: the parser, the scanner, and the synthesizer. The parser controls the interactive execution of the scanner (reading input) and the synthesizer (translating accepted input).

The parser uses the table created by the parser generator and the language types read by the scanner to detect errors or acceptable statements. Detected errors are then passed to a "fix" routine which attempts certain corrections to try to find an acceptable statement (inserting a semicolon, for example). Accepted statements must be "synthesized" into the correct data structures expected by the simulation modules of the BLAST program.

[2]A. V. Aho and S. C. Johnson, "LR Parsing," *Computing Surveys*, Vol 6, No. 2 (June 1974), pp 99-124.
[3]Aho and Johnson; and W. R. LaLonde, *An Efficient LALR Parser Generator*, CSRG-2 (University of Toronto, Computer Systems Research Group, 1971).

The scanner accepts the card input and classifies the words and special characters as to language type. The scanner maintains a name table during each user run. This name table contains some predefined types of names and a large vocabulary recognized by the language, and adds user names and numbers defined during the run - the name for a library item for example. The additional items are not maintained from run to run.

The synthesizer is the "bottom" level of the language processor. It translates items of the input language into the correct common blocks for later simulations. For example, UNITS (IN = ENGLISH, OUT = ENGLISH) sets flags to indicate what succeeding input conversations are to be done (BLAST maintains all numerics in SI units) and to indicate that output reports are to be converted to English units.

The BLAST input language is a block-structured language with major blocks: RUN CONTROL, LIBRARY manipulations, BUILDINGS, ZONES, FAN SYSTEMS, and CENTRAL PLANT. The language is also keyword-oriented, in that certain words have very special meaning (e.g., HEATING, COOLING).

Acceptance of the RUN CONTROL block sets the flags, etc. defined by the user's RUN CONTROL statement, in which the user can indicate reports to be output, types of simulation to be allowed, etc.

Manipulations of the library (DEFINE, DELETE, REDEFINE, TEMPORARY) cause the items to be updated into the new library file. For these changes to become physically permanent, the user must save this file. Items are stored in the library in the same data structure as they will be retrieved and stored into common blocks.

As each of the remaining blocks (BUILDINGS, ZONES, FAN SYSTEMS, CENTRAL PLANT) is accepted, the pertinent common blocks (system defaults plus input data) are written to the simulation input file. Each of these blocks (BUILDINGS, FAN SYSTEMS, CENTRAL PLANT) causes a simulation to be attempted.

Tables 3 through 6 describe the routines used in processing user input.

Table 3

Parser Routines

Routine	Description
PARSE	Initializes language processing, calls initial defaults for simulation input file, calls PARSER
PARSER	Iterates between scanner and synthesizer routines during acceptance of user input; on errors uses PFIX to attempt correction; can internally abort when stacks overflow (calls PMD)
PMD	Post mortem dump routine which provides a dump of stack causing problem and aborts processing
PFIX	Fixing routine; may attempt look-ahead scanning to try to correct error
PARSERR	Prints diagnostic messages for PFIX
FOLLOW	Tests if correction attempted by PFIX can be validly followed by next token in the input stream

Table 4

Scanning Routines

Routine	Description
SCAN	Reads input, determines input token class, builds name table
HASH	Function returning a number between 1 and NHASH depending on the data in the argument array

Table 4 (cont'd)

Routine	Description
TRANSL	Does preliminary translation of input characters into certain special class types - e.g. letter, digit, quote, equals, etc.
PCKSYM	Packs an array of n characters in A1 format into Ln format

Table 5

Synthesis Routines

Routine	Description
SYNTH	Major synthesis routine; most rules are numbered here. As deemed necessary, other routines also number certain parts of the input language
DECSTCK	Decrements symbol stack size by input number of pulls; checks for underflow
INCSTCK	Increments symbol stack size by input number of pushes; checks for overflow
INITUM	Builds, retrieves from name table
INUMB	Takes character representation of an integer and converts to integer
NUMB	Takes character representation of a real and converts to real
SYMTAB	Maintains symbol table at user level
LABLK	Labels assignment block routine used in costs for SYMTEP
SUPTYP	Sets power supply types

Table 5 (cont'd)

Routine	Description
RCHLR	Processes RUN CONTROL block
CNTUPT	Builds and controls data structure for placement into library
CTERR	Processes control error
SETCTL	After accepting control definition, sets status for on/off of heating and cooling
DECKCON	Sets deck control parameters
MDEFLO	Processes materials definition into proper data structure for library
LDEFLD	Processes location definition into proper data structure for library
FILLSCO	Fills a schedule type array
SETSCH	After accepting a schedule definition for the library, sets and reports defaults and errors
DDYLD	Processes design days
CNSTRN	Processes a "construction" in a building, retrieves necessary components from library, fills data structures
WNGHLR	Controls "wing" surface descriptions, creating right, left, or both wings as necessary
DELESUR	Deletes all surfaces of a given type from a zone
SRORNT	Transforms coordinates into global coordinates. Entry ISRORN sets the direction cosines
REOR	Reorients a zone recovered on "same as" statement using the new origin and north axis

Table 5 (cont'd)

Routine	Description
ZNSTMT	Processes a zone statement into proper data structures
FSKEY	Processes fan system keywords
FSNUM	Processes fan system numbers
FSOPR	Processes fan system operation period
FSRAMP	Processes fan system ramp
FFLOPR	Sets on/off dates for fan system
FSSCH	Fills fan system schedule array
PARELT	Processes list of parameters
SPECPAR	Processes special parameters
KEYVAL	Returns keyword value
PACK3	Packs the directory of key names used to build KEYVAL table
UPACK3	Unpacks directory of key name value
SCHED	Fills a schedule array

Table 6

Unit Conversion Routines

Routine	Type of Unit Converted
AREAC	Area
CAPACP	Power capacity
DELTEM	Temperature changes
ENERGC	Energy
ENTHLC	Enthalpy

Table 6 (cont'd)

Routine	Description
FLCAPC	Mass flow per capacity
MASARC	Mass per area
MASFLOC	Mass flow
PRESSR	Pressure
TEMPC	Temperature
VELOC	Velocity
VOLUMEF	Volume

4 BLAST SIMULATION INPUT FILE

The simulation input file is created by the acceptance of the input simulation types. A record is created for each BUILDING, ZONE, FAN SYSTEM, and CENTRAL PLANT accepted. The file maintains the number of simulations, an array of simulations to be attempted, an array of locations for each simulation, and the number of zones, zone number arrays, and locations of the zones for each building. The actual records are subindexed arrays of common blocks necessary to perform the desired simulation.

The simulation input file structure depends on certain pointers maintained in core. These pointers consist of the simulation type array (SIMTYP) and simulation record location array (SFLOC). The simulation type array is the type of simulations to be performed in the sequence (SIMTYP(1) is first simulation of type C(SIMTYP(1)). The maximum number of simulations is currently 20. The simulation record location array is cross referenced (by sequential simulation number) from the SIMTYP array to the location of the simulation record in the master index of the simulation input file. In addition, for each zone within a building, a location array (ZSFLØC) is maintained on the building record. This location array can hold a maximum of 50 zones and points to the master index element for the zone record. Each simulation and zone record consists of the common blocks necessary to perform the simulation. Thus, in actuality, the master index points to a record which is itself an index, each element of which points to the various common block records. Figure 2 shows the general file structure, and Figures 3 through 6 give the subindex pointers for each simulation type. Table 7 indicates the general function of routines used in creating and using the simulation input file.

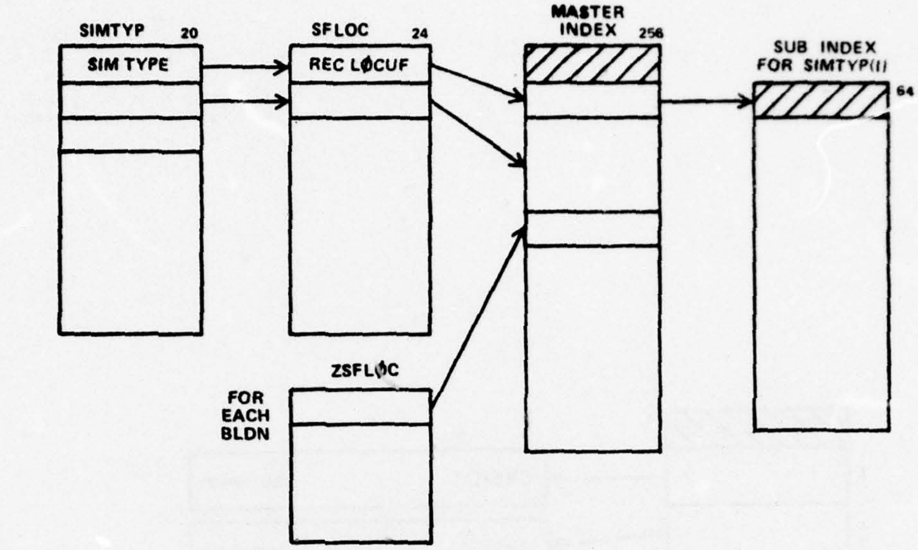


Figure 2. Simulation input file structure.

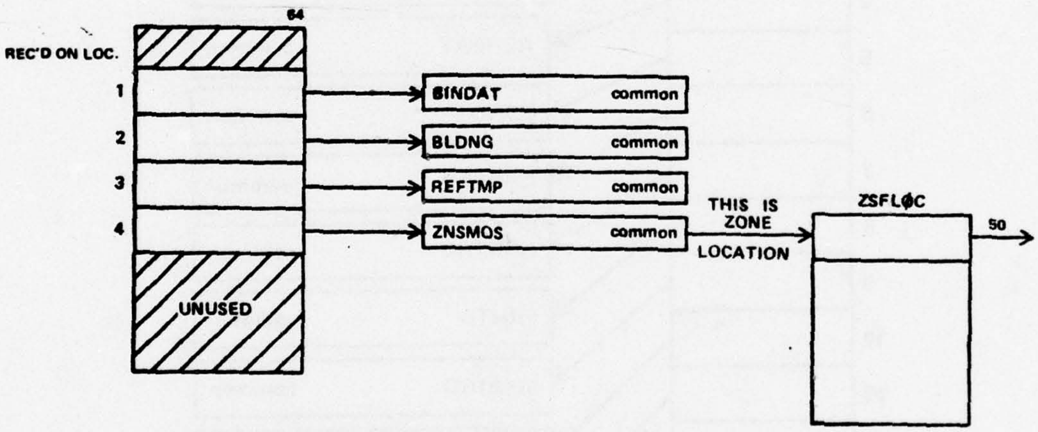


Figure 3. Subindex pointers for buildings.

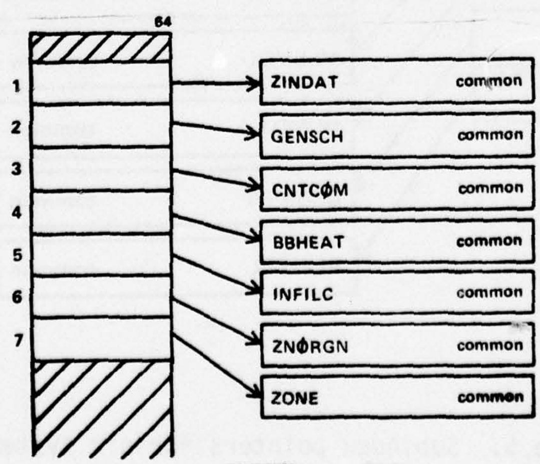


Figure 4. Subindex pointers for zones.

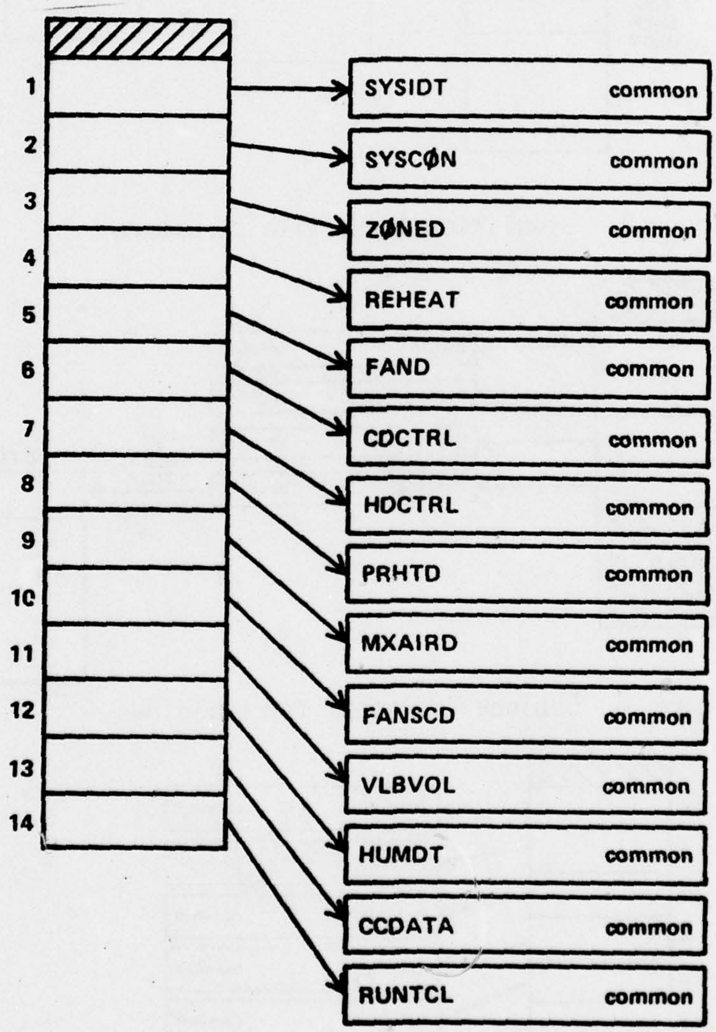


Figure 5. Subindex pointers for air systems.

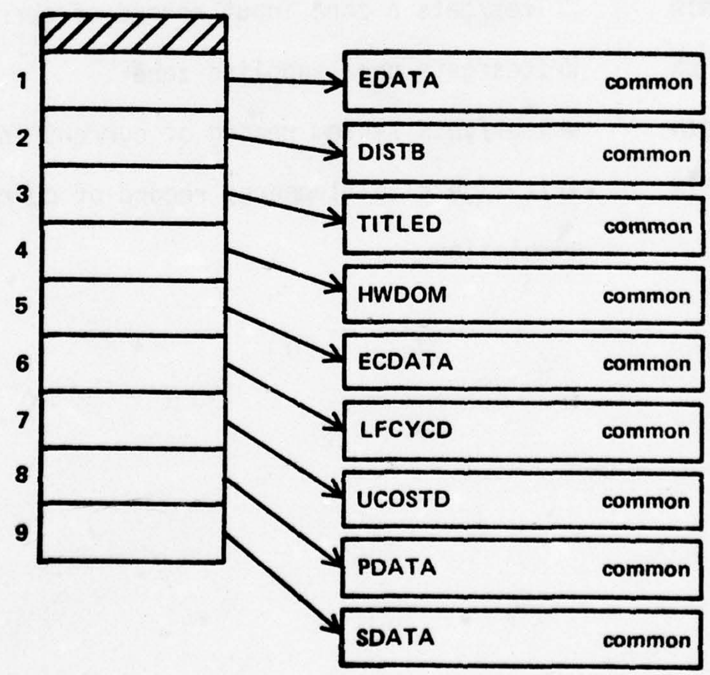


Figure 6. Subindex pointers for total energy plants.

Table 7

Simulation Input File Routines

Routine	Description
OPSINF	Opens file
CLSINF	Closes file
INITDF	Initializes defaults and writes defaults at specific locations for: BUILDINGS, ZONES, SYSTEMS, TOTAL ENERGY PLANT
GTDFIN	Gets a default input for specified type
WRSINF	Writes a record on simulation input file; type is passed
WRBLIN, GTBLIN	Writes/gets a building input record of current input/simulation
WRZNIN, GTZNIN	Writes/gets a zone input record of current user zone
WRULZN, GTULZN	Writes/gets user-supplied zone
WRSYIN, GTSYIN	Writes/gets system record of current input/simulation
WRTEIN, GTTEIN	Writes/gets total energy record of current input/simulation

5 THE BLAST LIBRARY AND LIBRARY UTILITY ROUTINES

The BLAST library utility package consists of 14 subroutines with a total of 17 entry points which allow the user to put, get, replace, and delete library elements, as well as produce a formatted printout or octal dump of the library contents. Table 8 lists the routines and their functions. The library itself is a random access mass storage file consisting of 998 possible library entries (records) with up to 320 words each, plus two header records which are nonaccessible to the user.

The first of the two header records contains two pointers and a counter; the first pointer contains the next available mass storage key for permanent library entries, the second pointer contains the next available mass storage key for temporary entries, and the counter is the number of times which GETLIB (gets data from library) was called in the current run. The second header record contains pointers to each of 20 possible double-linked type lists, plus one for temporary entries, of which 10 are currently being used.

The length of the remaining 998 user-definable entries depends upon type of library entry. Figure 7 shows subdivision of these entries.

Table 8

Library Routines

Routine	Description
SETLIB	Sets up initial parameters, makes working copy of old library; updates will be made to working copy
GETLIB	Gets library of type requested
BADINPT	Determines if input is of invalid type
LOCATE	Locates entry, next available, etc.
DELLIB	Deletes entry
PUTLIB	Defines entry (including temporary entries)
REPLIB	Redefines entry
PRTLIB	Prints library
DDPRT	Prints design days in library
CONPRT	Prints controls in library
GENPRT	Prints schedules in library
MATPRT	Prints materials in library
LOCPRT	Prints locations in library
BLDPRT	Prints walls, roofs, floors, doors, and windows in library

	BACKWARD LINK	FORWARD LINK	N	L	TYPE	ACCESS COUNT	NAME	INFO
ENTRY:	BACKWARD LINK	FORWARD LINK	N	L	TYPE	ACCESS COUNT	NAME	INFO
LENGTH: (words)	1	1	1	1	1	1	10	304

where BACKWARD LINK = mass storage key of the previous item in the type list

FORWARD LINK = mass storage key of the next item in the type list

N = length of the name in words ($1 \leq N \leq 10$)

L = length of the information in words ($1 \leq L \leq 304$)

TYPE = type class of the information ($1 \leq TYPE \leq 20$)

ACCESS COUNT = number of times GETLIB has been called for this item in the current run

NAME = name identifying this entry

INFO = data associated with this entry

Figure 7. Subdivision of library entries.

The 8 subroutines and entry points which are important to the user are briefly described below. The underlined parameters are input parameters, and the remaining are output. No input parameters are changed by the routines.

SETLIB initializes the library and <u>must</u> be called in every run in which the library is used, prior to any library activity. SETLIB creates a working copy of the library on LIBRBR by copying it from LIBRF in a packed form (i.e., performs any condensing made necessary by deletions) with all temporary entries cleared and the access count in each entry set to few.

PUTLIB (T,N,L, NAME, INFO, TFLG) puts an entry in the library; however, if PUTLIB is called for a permanent entry and an entry with the same name and type class already exists, a severe error is signaled. The parameters are:

T = type class ($1 \leq T \leq 20$)

N = number of words in the name ($1 \leq N \leq 10$)

L = number of words in INFO ($1 \leq L \leq 304$)

NAME = name of the entry

INFO = information to be stored

TFLG = 0 for permanent entry and non-zero for a temporary entry.

GETLIB (T, N, L, NAME, INFO, ACNT) retrieves an entry from the library by first searching the temporary list for an entry with the corresponding name and type and then, if one is not found, the permanent entry type list; if the desired entry is not found, a severe error is signaled. T, N, L, NAME, and INFO are as defined previously, and ACNT is the call to GETLIB when the entry was first accessed in the run.

REPLIB (T, N, L, NAME, INFO) replaces the L and INFO fields in the permanent entry specified by T and NAME with L and INFO, respectively.

The access count is cleared. If the entry is not found, PUTLIB is called for a permanent entry with the given data, and a severe error is signaled. The parameters are as previously defined.

DELLIB (T, N, NAME) deletes the permanent entry in the library with type class T, and name NAME. If the entry is not found, a severe error is signaled.

PRTLIB outputs a formatted copy of the contents of the entire library in English and/or SI units.

RSTLIB resets the GETLIB counter in the first header record and each of the library entries to zero.

CLOSLIB must be called after all library activities are completed and prior to the end of any run in which the library was accessed.

The remaining routines are used with the above for information searching and manipulation, and error checking.

6 MSIMDF (MAKE SIMULATION DESCRIPTION FILE)

MSIMDF uses a number of routines to perform major calculations required only once per simulation (e.g., calculation of wall thermal response factor shape factors, glass coefficients, etc.). All such calculations relate to building and zone load calculations only (not air handling system or central plant simulations). Figure 8 illustrates the tree structure of the routines under MSIMDF. The following pages present descriptions of MSIMDF and its subroutines (in alphabetical order), except for the TAG routine, which is described in Chapter 8. Structured algorithm charts are presented where appropriate.

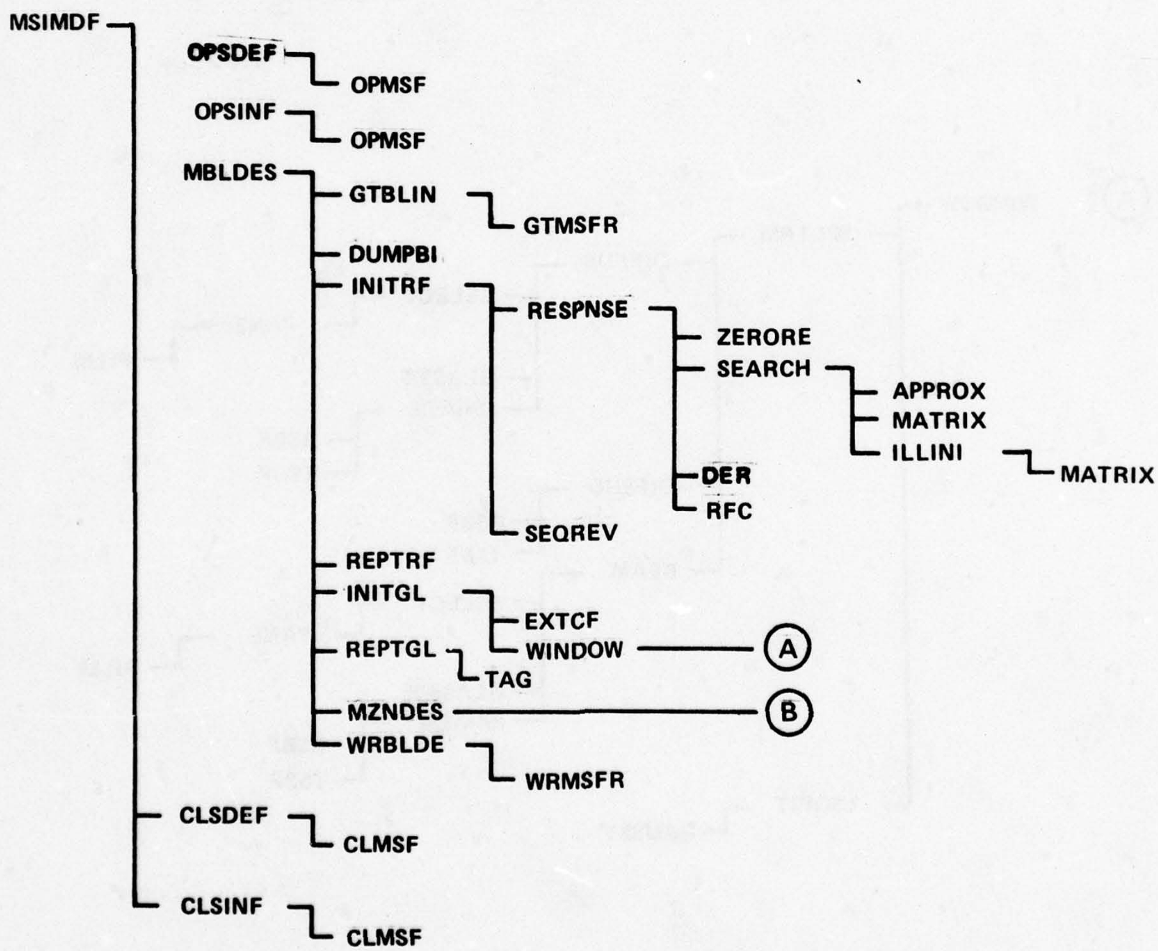


Figure 8. Tree structure for MSIMDF subroutines.

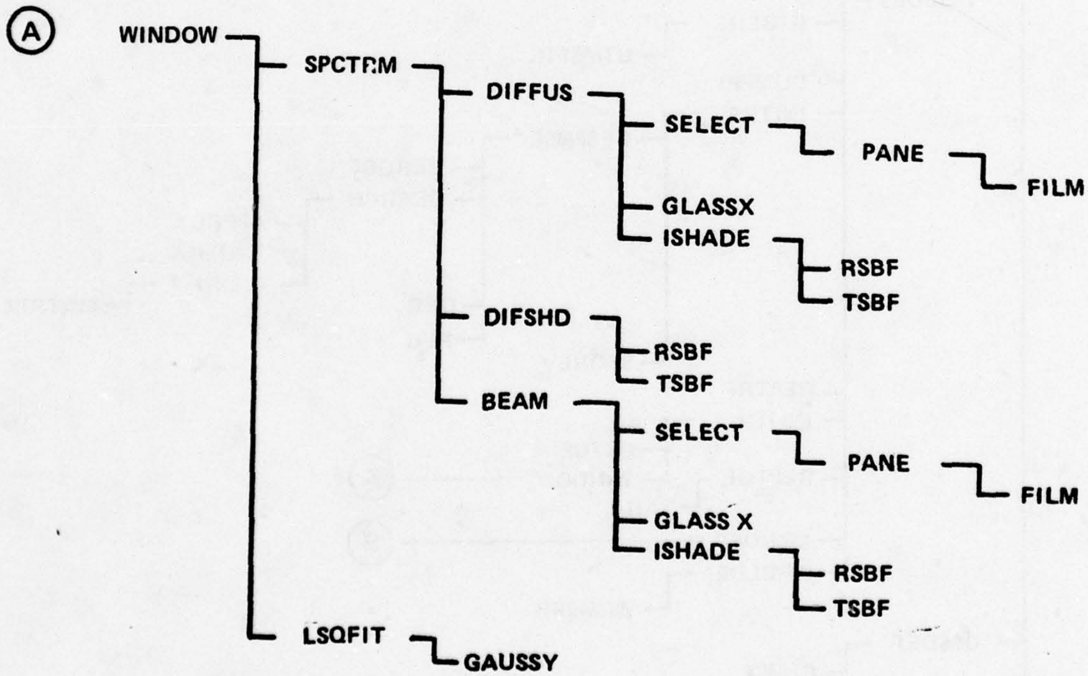


Figure 8 (cont'd).

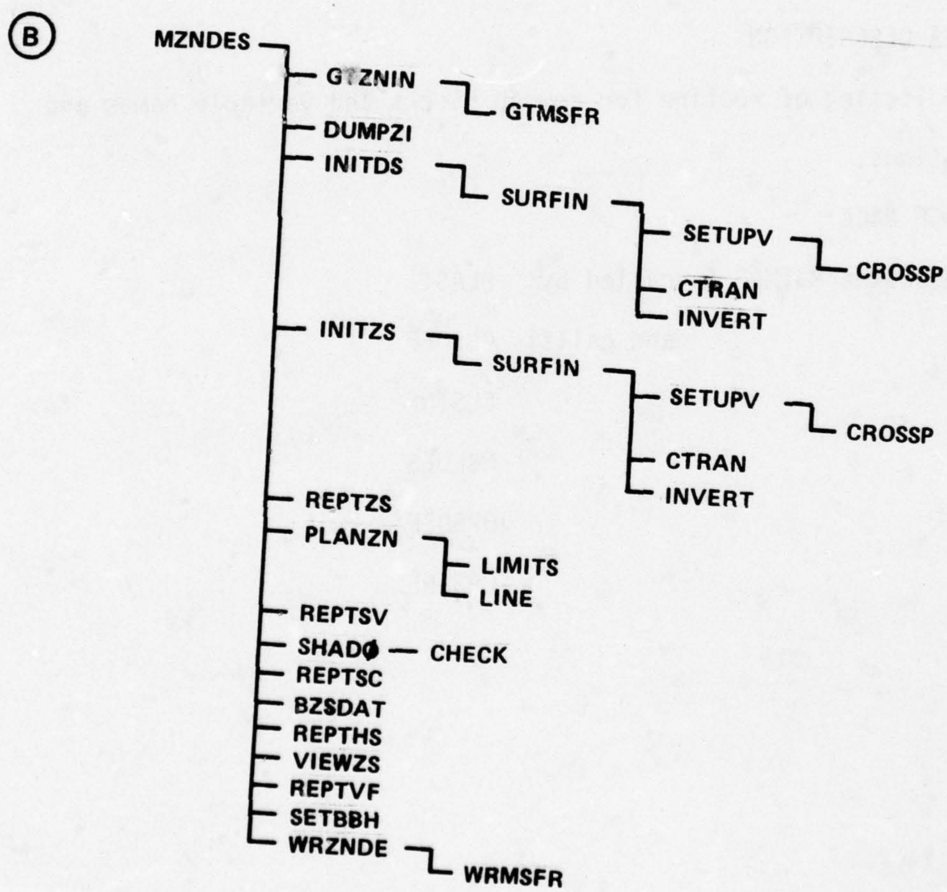


Figure 8 (cont'd).

MSIMDF

a. GENERAL DESCRIPTION

MSIMDF was designed to perform one-time data conversions from data generated from the input language parser to data used by the simulation routines. Since conversions are necessary only for building and zone data, MSIMDF does not apply to air handling system or central plant simulations.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine MSIMDF is called by: BLAST

and calls: CLSDEF
CLSINF
MBLDES
OPSDEF
OPSINF

APPROX

a. GENERAL DESCRIPTION

APPROX is a subroutine which calculates the upper right element of the total construct matrix to find the roots needed to calculate a residue expansion of an integral. The routine, which was originated at CERL, increments needed sines and cosines by using fundamental trigonometric identities. The calculation of the total construct matrix is the matrix product of the layer matrices.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
RF	BETA(J)	Array of layer angles for trigonometric calculations
RF3	CP(J)	Array of present point cosines
RF3	CS(J)	Array of incremental cosines
RF	NL	Number of layers
	PT	Approximate upper bound
RF	R	Layer resistance
RF	RES	Layer R-factors
RF3	SP(J)	Array of present point sines
RF3	SS(J)	Array of incremental sines

2. OUTPUT DATA

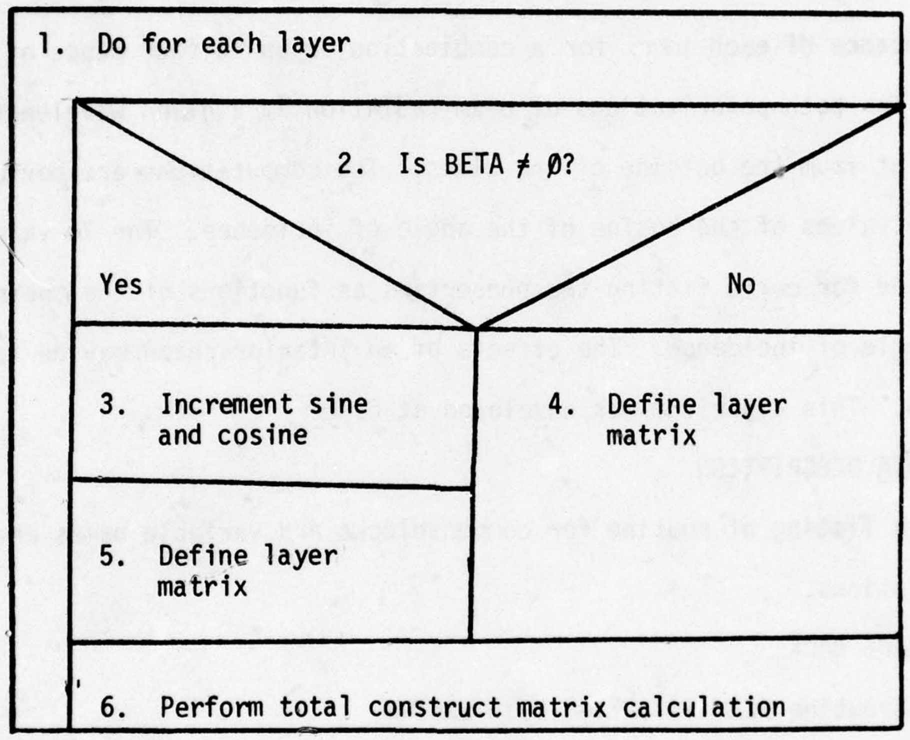
Name	Description
BS	Upper right element of the total construct matrix

c. TRACE BACK

Subroutine APPROX is called by: SEARCH

and calls: No subroutines

APPROX



Note: a. The definitions of layer matrix elements depend on the values of BETA.

b. Sines and cosines are incremented by using formulas for $SIN(\alpha+\beta)$ and $COS(\alpha+\beta)$

BEAM

a. GENERAL DESCRIPTION

This subroutine computes the transmittance, reflectance, and absorptance of each pane for a combination of up to four panes of glass for both polarizations of beam radiation at a given wavelength incident from the outside of the glass. The computations are performed for 16 values of the cosine of the angle of incidence. The 16 values are used for curve fitting the properties as functions of the cosine of the angle of incidence. The effects of an interior shade may be included. This algorithm was developed at CERL.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine BEAM is called by: SPCTRM

and calls: SELECT

GLASSX

ISHADE

BZSDAT

a. GENERAL DESCRIPTION

BZSDAT transfers information from the input arrays into smaller arrays for simplification of load calculations. Pointers are computed for the conduction transfer functions and temperature and flux histories. Surface properties, geometry, and environments are recorded.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine BZSDAT is called by: MZNDES

and calls: No subroutines

CHECK (NGRS, NSS, ZMIN, NABOVE)

a. GENERAL DESCRIPTION

CHECK eliminates surfaces as possible shadowers of other surfaces. The three conditions for elimination are:

1. If the highest point of the shadowing surface is below the lowest point of the receiving surface

2. If the shadowing surface faces straight up (as for a roof)

3. If the shadowing surface is entirely behind the plane of the receiving surface.

The checks are not all-inclusive. There may still be some surfaces which cannot shade the receiving surface.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine CHECK is called by: SHADØ

and calls: No subroutines

CROSSP

a. GENERAL DESCRIPTION

CROSSP uses a cross product to compute the area of nonrectangular figures. It also checks that quadrilaterals are flat using an algorithm described in the *CRC Standard Mathematical Tables.*[4]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine CROSSP is called by: SETUPV

and calls: No subroutines

[4] *CRC Standard Mathematical Tables*, 22nd edition (Chemical Rubber Company, 1972), p 379.

CTRAN (XØ, YØ,ZØ, A)

a. GENERAL DESCRIPTION

CTRAN develops a coordinate transformation such that the X-axis goes through points 2 and 3 and the Y-axis goes through point 1 of a plane figure in three-dimensional space. This is the transformation described in ASHRAE's *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations.*[5] and used in the NASA Energy Cost Analysis Program (NECAP). The transformation is used in computing shading.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine CTRAN is called by: SURFIN

and calls: No subroutines

[5]*Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations* (American Society of Heating, Refrigerating, and Air Conditioning Engineers [ASHRAE], 1975).

DER

a. GENERAL DESCRIPTION

DER is a subroutine which calculates the values of the total construct and total derivative matrices for a given value of the Laplace transform parameter. This routine is also based on basic layer and layer derivative element definitions found in NECAP, but all computational aspects are new. The total construct matrix calculation is merely the matrix product of the layer matrices. The total derivative matrix calculation follows the product rule [d(u v) = (du) v = u dv] of calculus with the added restriction that order must be preserved.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
RF	BETA(J)	Array of layer angles for trigonometric calculations
RF	NL	Number of layers
RF	R(J)	Array of layer resistances
RF	RES(J)	Array of layer R-factors
	SQ	Square root of Laplace transform parameter

2. OUTPUT DATA

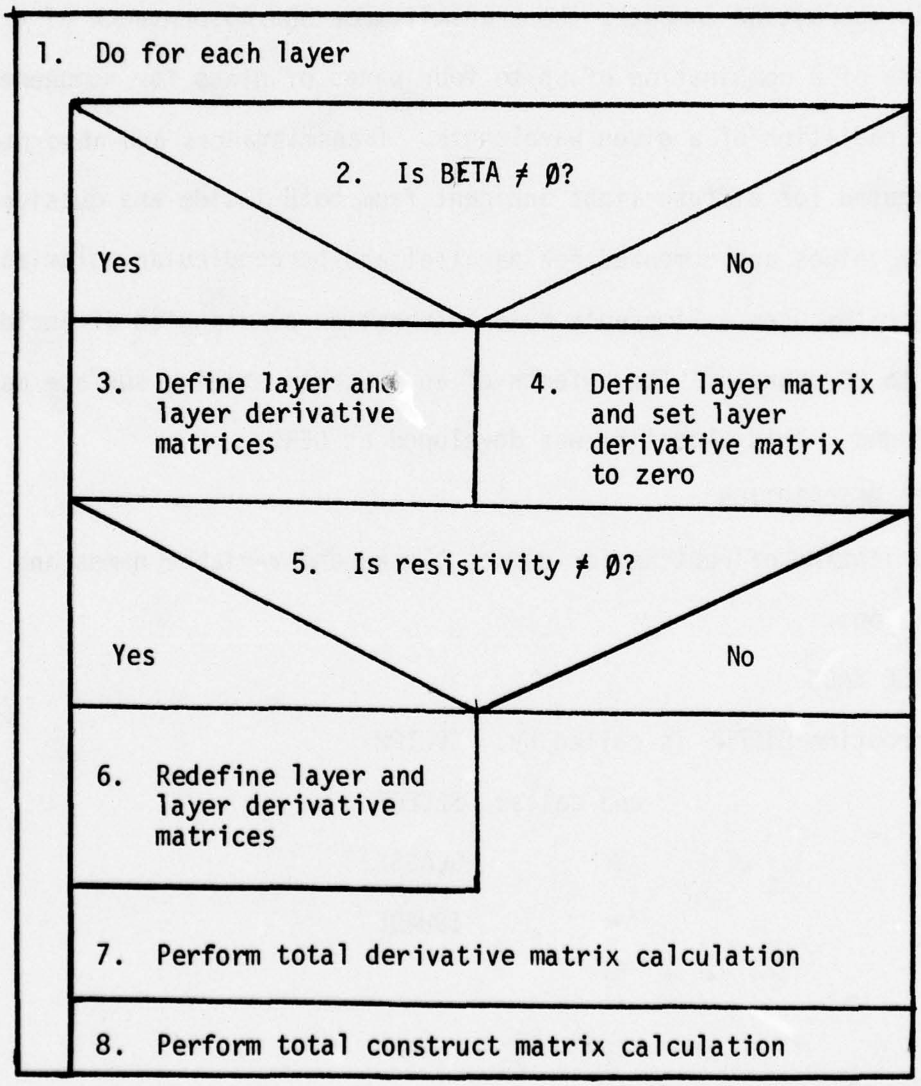
Name	Description
AS	Upper left element of the total construct matrix
BT	Upper right element of the total derivative matrix
DS	Lower right element of the total construct matrix

c. TRACE BACK

Subroutine DER is called by: RESPNSE

and calls: No subroutines

DER



Note: The definitions of layer and layer derivative matrix elements depend on the values of BETA and the R-factor of the layer.

DIFFUS

a. GENERAL DESCRIPTION

This subroutine computes the transmittance and absorptance of each pane of a combination of up to four panes of glass for homogeneous diffuse radiation of a given wavelength. Transmittances and absorptances are computed for diffuse light incident from both inside and outside. Separate values are computed for parallel and perpendicular polarizations The algorithm uses a Simpson's rule integration over angles of incidence from 0 to 90 degrees. The effects of an interior shading surface may be included. This algorithm was developed at CERL.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine DIFFUS is called by: SPCTRM

and calls: SELECT

GLASSX

ISHADE

DIFSHD

a. GENERAL DESCRIPTION

This subroutine computes the reflectance of an interior shading device for diffusely incident radiation. A Simpson's rule integration over angles of incidence from 0 to 90 degrees is used. Functions TSBF and RSBF are designed to contain the angle, wavelength, and polarization properties of the shading surface. This algorithm was developed at CERL.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine DIFSHD is called by: SPCTRM

and calls: TSBF

RSBF

EXTCF

a. GENERAL DESCRIPTION

This function computes the KL product (extinction coefficient* thickness) of a single pane of glass, given the index infraction and normal transmittance. Both inputs must be for the wavelength being considered. The equation is the solution of a quadratic equation which gives transmittance at normal incidence as a function of KL and the index of refraction.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine EXTCF is called by: INITGL

and calls: No subroutines

FILM

a. GENERAL DESCRIPTION

This subroutine calculates the transmittance and reflectance of a thin metallic film on glass. Complex arithmetic is used according to the theory of the optics of metals. Born and Wolf[6] provide an excellent discussion of the theory of optics.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FILM is called by: PANE

and calls: No subroutines

[6]M. Born and E. Wolf, *Principles of Optics* (Pergamon Press, 1959).

GAUSSY

a. GENERAL DESCRIPTION

This is a standard Gaussian elimination routine for the solution of simultaneous linear algebraic equations. It includes full pivoting and testing.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine GAUSSY is called by: LSQFIT

and calls: No subroutines

GLASSX

a. GENERAL DESCRIPTION

This subroutine computes the total transmittance and reflectance of a system of up to four panes of glass and the total absorptance of each pane. Computations are performed for both polarizations of light at a given wavelength and angle of incidence and for light incident from inside or outside the panes. The variables are given subscripts to keep track of polarization, pane number, and side of incidence. The input consists of information for each pane of glass which is computed in PANE. The equations were derived at CERL from a standard radiation balance technique.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine GLASSX is called by: DIFFUS

BEAM

and calls: No subroutines

ILLINI

a. GENERAL DESCRIPTION

ILLINI is a subroutine which uses a modified, bounded secant method[7] to find the roots needed to calculate a residue expansion of an integral. This algorithm was developed at the University of Illinois.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
	A	Lower bound of root
	B	Upper bound of root
RF	BETA(J)	Array of layer angles for trigonometric calculations
	F	Function value at lower bound
	G	Function value at upper bound
RF	NL	Number of layers
RF	R(J)	Array of layer resistances
RF	RES(J)	Array of layer R-factors

2. OUTPUT DATA

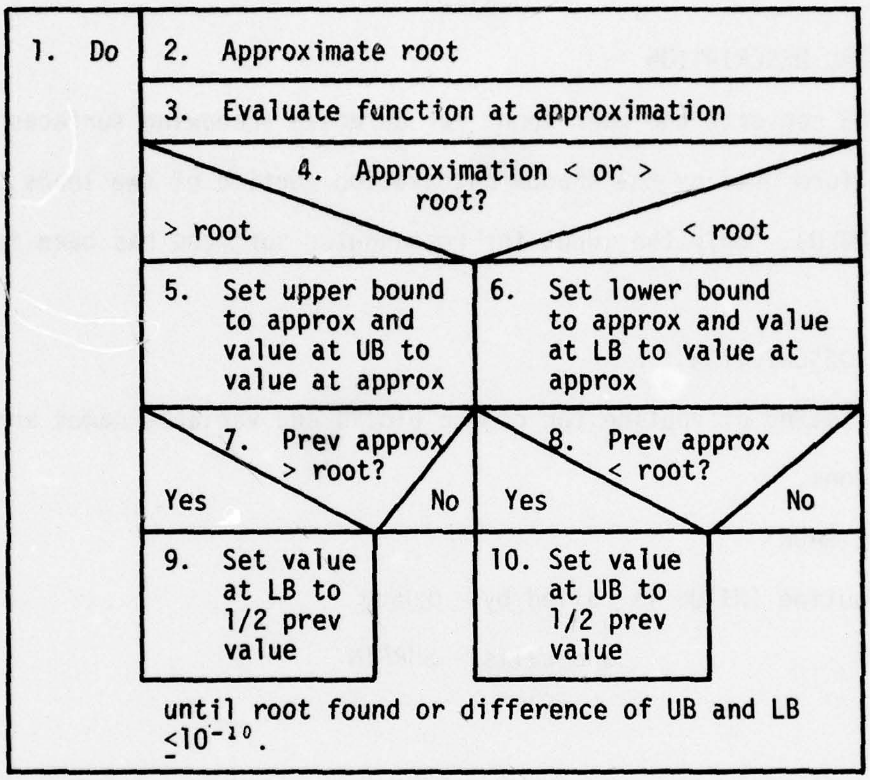
Name	Description
RT	Root

c. TRACE BACK

Subroutine ILLINI is called by: SEARCH

and calls: MATRIX

[7]For an explanation of secant and modified secant methods, see S. D. Conte and C. de Boor, *Elementary Numerical Analysis* (McGraw-Hill, 1972).


1. Do
2. Approximate root
3. Evaluate function at approximation
4. Approximation < or > root?
> root
< root
5. Set upper bound to approx and value at UB to value at approx
6. Set lower bound to approx and value at LB to value at approx
7. Prev approx > root?
Yes
No
8. Prev approx < root?
Yes
No
9. Set value at LB to 1/2 prev value
10. Set value at UB to 1/2 prev value
until root found or difference of UB and LB ≤10⁻¹⁰.

INITDS

a. GENERAL DESCRIPTION

INITDS converts the user input for detached shadowing surfaces into the form used by the shadow calculation portion of the loads program (SIMBLD). Only the input for rectangular surfaces has been implemented.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INITDS is called by: MZNDES

and calls: SURFIN

INITGL

a. GENERAL DESCRIPTION

This subroutine converts the user-supplied data for windows to a set of coefficients used by SIMBLD for calculating heat gain through glass. Material properties are transformed from the format of the input file to the format used by the subroutines which make detailed optical calculations. A section of the subroutine uses a previously determined set of coefficients for standard window glass to implement the shading coefficient description of a window. Use of material properties and the detailed optical calculations should provide more accurate results than shading coefficients. The method for determining the physical properties of a metallic film has not yet been implemented.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INITGL is called by: MBLDES

and calls: EXTCF

WINDOW

INITRF

a. GENERAL DESCRIPTION

INITRF controls the calculation of response factors. Material properties of all layers of each construct are read from the building input file. They are converted from SI to English units because the response factor routines depend on the scale of the numbers used. The response factors and conduction transfer functions (C.T.F.) are recorded. The C.T.F. are recorded in an inverse order to match the order of the temperature histories in the loads calculations.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INITRF is called by: MBLDES

and calls: RSPNSE

SEQREV

INITZS

a. GENERAL DESCRIPTION

INITZS transforms the user input for the geometry of zone surfaces to the form used by SIMBLD. It also resequences the zone heat transfer surfaces so that those with variable inside convection coefficients are last in the heat balance equations.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INITZS is called by: MZNDES

and calls: SURFIN

INVERT (A, B)

a. GENERAL DESCRIPTION

INVERT will invert a 3 x 3 matrix by the cofactor method.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INVERT is called by: SURFIN

and calls: No subroutines

ISHADE

a. GENERAL DESCRIPTION

This subroutine revises the transmittance, reflectance, and absorptances of a system of panes of glass (without an interior shade) to account for an interior shading device. The shade is assumed to be diffusely reflecting. All inter-reflections between the shade and the glass are considered. The equations were derived at CERL.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ISHADE is called by: BEAM
DIFFUS

and calls: RSBF
TSBF

LIMITS

a. GENERAL DESCRIPTION

LIMITS determines a scale which will fit the zone on the page. It checks for maximum and minimum north-south and east-west coordinates, determines which is the critical dimension, and, based on that dimension, finds a scale factor for the elements of the grid on which the drawing is made.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine LIMITS is called by: PLANZN

and calls: No subroutines

LINE (X1, Y1, X2, Y2)

a. GENERAL DESCRIPTION

Given the x and y coordinates of the ends of a line segment, LINE determines which elements of the picture grid are on a straight line between those points.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine LINE is called by: PLANZN

and calls: No subroutines

LSQFIT

a. GENERAL DESCRIPTION

This subroutine does least squares fits to polynomial expressions for transmittance and absorptance. Different polynomials (R) are used depending on the angle of incidence (θ):

For $0.5 \leq \cos\theta \leq 1.0$

R = C1 + C2*X + C3*X**2 + ... + C5*X**4

For $0.0 \leq \cos\theta \leq 0.5$

R = C1*X + C2*X**2 + ... + C6*X**5

The latter expression insures zero transmittance or absorptance at 90 degree incidence.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine LSQFIT is called by: WINDOW

and calls: GAUSSY

MATRIX

a. GENERAL DESCRIPTION

MATRIX is a subroutine which calculates the upper right element of the total construct matrix to find the roots needed to calculate a residue expansion of an integral. The NECAP layer definitions are used in this calculation, which is basically the multiplication of the layer matrices in a reduced form.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
RF	BETA(J)	Array of layer angles for trigonometric calculations
RF	NL	Number of layers
RF	R(J)	Array of layer resistances
RF	RES(J)	Array of layer R-factors
	W	Approximate root

2. OUTPUT DATA

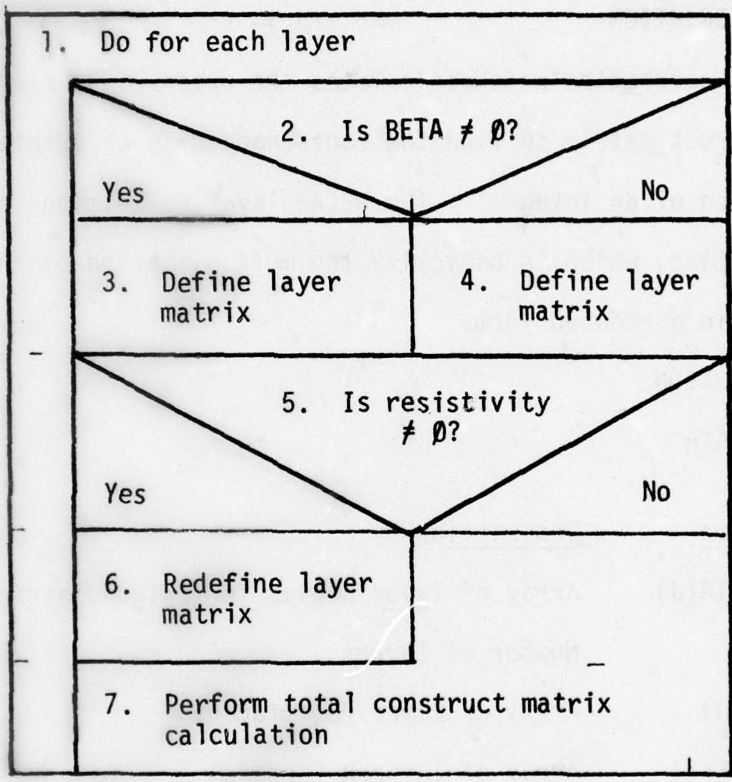
Name	Description
BS	Upper right element of the total construct matrix

c. TRACE BACK

Subroutine MATRIX is called by: ILLINI

SEARCH

and calls: No subroutines



Note: The definitions of layer matrix elements depend on the values of BETA and the R-factor of the layer.

MBLDES(CRSIM)

a. GENERAL DESCRIPTION

MBLDES creates the building description file which is a conversion of data from the file created by the input language parser to a form used by the simulation routines. It calculates response factors and window coefficients for the constructs used in the building. It calls MZNDES, which converts data for each zone.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine MBLDES is called by: MSIMDF

and calls: GTBLIN

INITGL

INITRF

MZNDES

REPTGL

REPTRF

WRBLDE

DUMPBI

MXNDES (CRSMZN)

a. GENERAL DESCRIPTION

MZNDES creates the zone description file. Several elements of zone data are converted, as described by the subroutines called.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine MZNDES is called by: MBLDES

and calls: GTZNIN

DUMPZI

INITDS

INITZS

REPTZS

PLANZN

REPTSV

SHADØ

REPTSC

BZSDAT

REPTHS

VIEWZS

REPTVF

SETBBH

WRZNDE

PANE

a. GENERAL DESCRIPTION

Given physical properties of a pane of glass, this subroutine computes its transmittance, reflectance, and absorptance. Results apply for the wavelength implied by the properties, both polarizations, and with the light incident from either side of the glass at the given angle of incidence. Polarizations, direction of incidence, and pane number are recorded by array indices.

The technique involves computing the transmittance and reflectance of both air-glass interfaces together with the extinction coefficient of the glass to determine its total properties. A special form of the Fresnel equations is used to avoid trigonometric functions. Born and Wolf provide an excellent reference.[8]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine is called by: SELECT

and calls: FILM

[8] M. Born and E. Wolf, *Principles of Optics* (Pergamon Press, 1959).

PLANZN

a. GENERAL DESCRIPTION

PLANZN generates a plan view of the heat transfer surfaces in the zone. The line printer produces this view as a simple line drawing using asterisks. PLANZN calls LIMITS and LINE for most of the calculations in preparing the line drawing.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine PLANZN is called by: MZNDES

and calls: LIMITS

LINE

REPTGL (OUTPUT, UNITS)

a. GENERAL DESCRIPTION

REPTGL generates a report of the optical properties of window constructs. It reports the transmittance and index of refraction of the individual layers of the window, and then the theoretical transmittance of the window and absorptance of each layer compared to the approximate values which will be used in the loads calculations.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTGL is called by: MBLDES

and calls: TAG

REPTHS (OUTPUT, UNITS)

a. GENERAL DESCRIPTION

REPTHS reports various details about the heat transfer surfaces which are part of the simultaneous heat balance. It is primarily useful as a check of the operations that create the zone description file.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTHS is called by: MZNDES

and calls: No subroutines

REPTRF (OUTPUT, UNITS)

a. GENERAL DESCRIPTION

REPTRF generates a report of the conductive properties of all constructs. It reports the thickness, density, conductivity, and specific heat of each layer and then reports the response factor and conduction transfer factors for the entire construct. Certain surface properties of the construct are also reported.

b. DATA DESCRIPTIONS

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTRF is called by: MBLDES

and calls: No subroutines

REPTSC (OUTPUT, UNITS)

a. GENERAL DESCRIPTION

REPTSC reports which surfaces are possible shadowers of other surfaces.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTSC is called by: MZNDES

and calls: No subroutines

REPTSV (OUTPUT, UNITS)

a. GENERAL DESCRIPTION

REPTSV prints the coordinates of the vertices of all surfaces pertaining to the zone. It is useful as an error finder when the zone drawing indicates the surfaces are not placed correctly.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTSV is called by: MZNDES

and calls: No subroutines

REPTVF (OUTPUT, UNITS)

a. GENERAL DESCRIPTION

REPTVF reports the view factors calculated by VIEWZS.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTVF is called by: MZNDES

and calls: No subroutines

REPTZS (OUTPUT, UNITS)

a. GENERAL DESCRIPTION

REPTZS is the primary report on zone surfaces. It lists all surfaces pertaining to the zone, their type, azimuth, tilt, area, and construction.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTZS is called by: MZNDES

and calls: No subroutines

RESPNSE

a. GENERAL DESCRIPTION

RESPNSE calculates the response factor and conduction transfer function. Response factors are calculated as instantaneous heat fluxes by solution of a Laplace-transformed,. Fourier heat transfer equation.[9] A residue expansion[10] is used to perform the inverse transform. Conduction transfer functions are trivially derived from the response factors.

The basic response factor scheme and calculation (see subroutine RFC) are performed as in NECAP, but all of the preliminary residue and most of the root-finding work is new. This new work extends the possible range of cases and decreases program run time. The addition of conduction transfer function calculations makes this routine, at minimum, the equivalent to NBSLD's subroutines RESFX, RESF, RESPTK, and related diminutive routines. The unique root predictor makes case extension and speed superior to NBSLD.

It should be noted that even though a general method is used, inputting values in SI units creates magnitudes outside the range comfortable for the computer techniques used. The calculations would require

---

[9] For a description of the mathematical calculation of response factors, see E. D. Mouen, *Application of the Thermal Response Factor Method to Building Elements with Air Cavities*, PhD. Thesis (University of Illinois, 1973).

[10] For an explanation of residue calculations in basic terms, see M. R. Spiegel, *Theory and Problems of Complex Variables*, Schaum Outline Series (McGraw-Hill, 1964).

many more roots and greater root accuracy. Therefore a conversion to English units and back to SI units would be desirable in order to reduce calculation time.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
RF	BETA(J)	Array of layer angles for trigonometric calculations
RF	NL	Number of layers of construct
RF	R(J)	Array of layer resistances
RF	RES(J)	Array of layer R-factors
	SINC	Search increment
	TINC	Time increment

2. OUTPUT DATA

Source of Data	Name	Description
RF1	CR	Common ratio
RF1	CTX(J)	Array of X conduction transfer functions
RF1	CTY(J)	Array of Y conduction transfer functions
RF1	CYZ(J)	Array of Z conduction transfer functions
	KND	Thermal conductivity of the construct
RF1	NRF	Number of response factors
RF1	RFX(J)	Array of X response factors
RF1	REY(J)	Array of Y response factors
RF1	RFZ(J)	Array of Z response factors

c. TRACE BACK

Subroutine RESPNSE is called by: INITRF

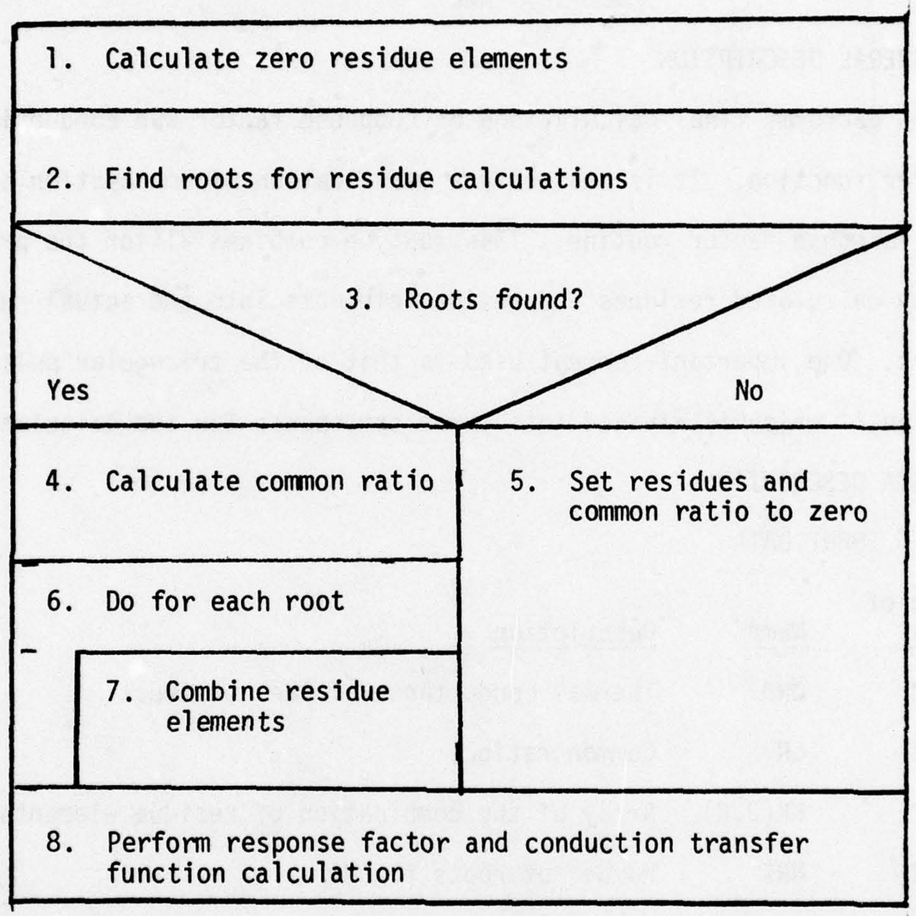
and calls: DER

RFC

SEARCH

ZERORE

RESPNSE



Option: *item 7 can be expressed in two parts:*

7a) Calculate total derivative and construct matrices

7b) Combine residue elements

RFC

a. GENERAL DESCRIPTION

RFC performs final calculations of response factor and conduction transfer function. It is basically a modification of one section of the NECAP response factor routine. This routine combines all of the previously calculated residues and residue elements into the actual response factors. One important concept used is that of the triangular pulse function,[11] which is divided into three components for the calculation.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
RFØ	CND	Thermal conductance of the construct
RF1	CR	Common ratio
RF2	KK(J,K)	Array of the combination of residue elements
RF2	NRT	Number of roots found
RF2	ROOT(J)	Array of values which make the upper right element of total construct matrix zero
	TINC	Time increment
RFØ	ZRX	Zero residue element related to X response factors
RFØ	ZRY	Zero residue element related to Y response factors
RFØ	ZRZ	Zero residue element related to Z response factors

[11]For an explanation of the pulse function, see A. Tustin, "A Method of Analyzing the Behaviour of Linear Systems in Terms of Time Series," *Journal of the Institution of Electrical Engineers*, Vol 94, Part II-A (1947).

2. OUTPUT DATA

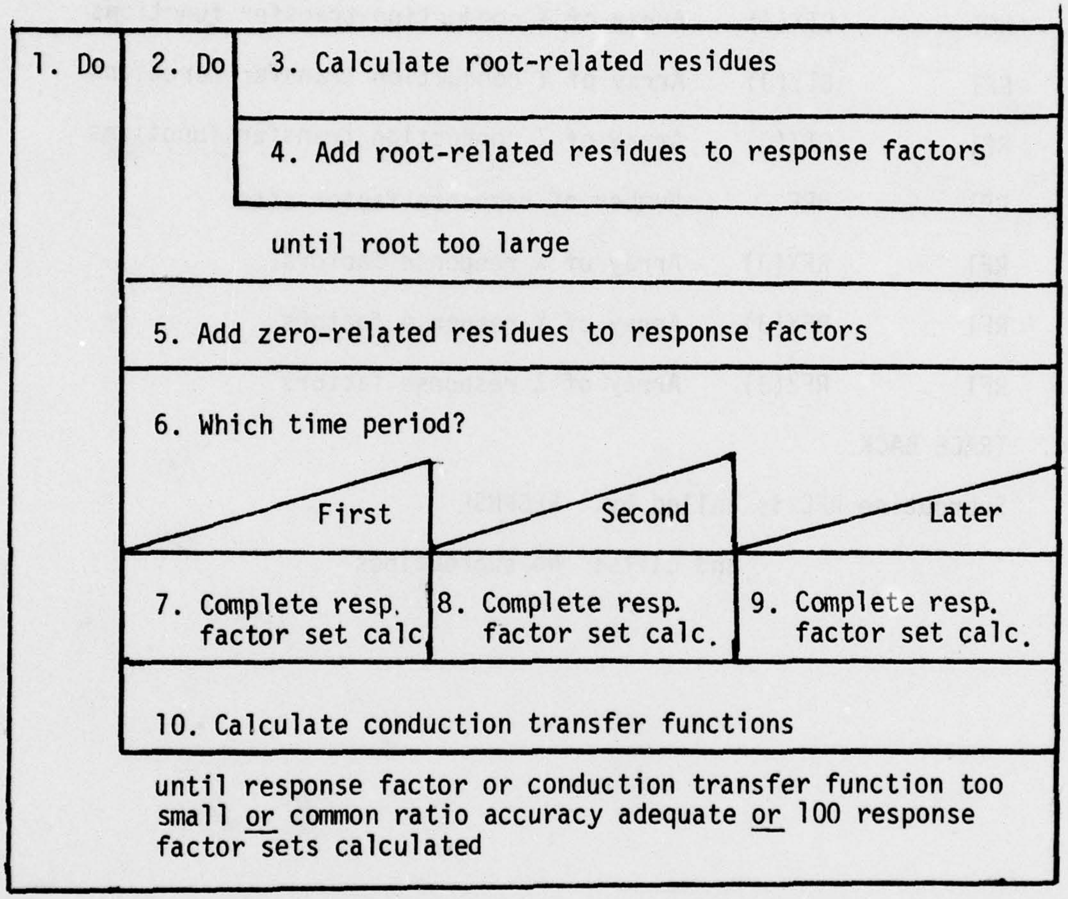
Source of Data	Name	Description
RF1	CTX(J)	Array of X conduction transfer functions
RF1	CTY(J)	Array of Y conduction transfer functions
RF1	CTZ(J)	Array of Z conduction transfer functions
RF1	NRF	Number of response factor sets
RF1	RFX(J)	Array of X response factors
RF1	RFY(J)	Array of Y response factors
RF1	RFZ(J)	Array of Z response factors

c. TRACE BACK

Subroutine RFC is called by: RESPNSE

and calls: No subroutines

RFC



Note: Due to the use of a three-component triangular pulse function, three forms of the final calculation of a response factor exist.

RSBF

a. GENERAL DESCRIPTION

This function was written to compute the reflectance of an interior shading device given the polarization, angle of incidence, and wavelength of the incident light. The function presently returns the reflectance at normal incidence specified by the user because of lack of data on the relationship of reflectance to the specified properties.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function RSBF is called by: DIFSHD

ISHADE

and calls: No subroutines

SEARCH

a. GENERAL DESCRIPTION

SEARCH is a subroutine used to bind (or find) roots for ILLINI. It is designed to do an incremental search using APPROX and will more accurately check the bound before calling ILLINI.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
RF	BETA(J)	Array of layer angles for trigonometric calculations
RF	NL	Number of layers
	KND	Thermal conductivity (for passing purposes)
RF	R(J)	Array of layer resistances
RF	RES(J)	Array of layer R-factors
	SINC	Search increment

2. OUTPUT DATA

Source of Data	Name	Description
RF2	NRT	Number of roots found
RF2	ROOT(J)	Array of values which make upper right element of total construct matrix zero

c. TRACE BACK

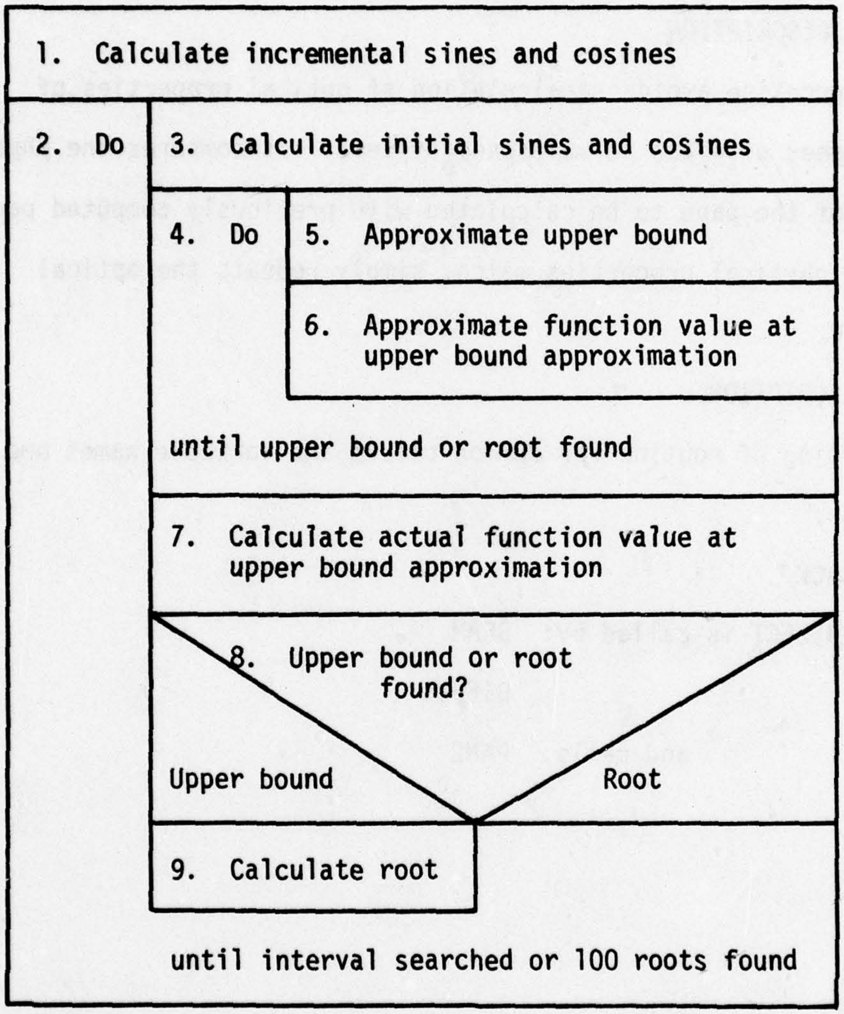
Subroutine SEARCH is called by: RESPNSE

and calls: APPROX

ILLINI

MATRIX

SEARCH


1. Calculate incremental sines and cosines
2. Do
3. Calculate initial sines and cosines
4. Do
5. Approximate upper bound
6. Approximate function value at upper bound approximation
until upper bound or root found
7. Calculate actual function value at upper bound approximation
8. Upper bound or root found?
Upper bound
Root
9. Calculate root
until interval searched or 100 roots found

SELECT

a. GENERAL DESCRIPTION

This subroutine avoids recalculation of optical properties of identical panes of glass in multipane systems. It compares the physical properties of the pane to be calculated with previously computed panes, and, if the physical properties match, simply repeats the optical properties.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SELECT is called by: BEAM
DIFFUS

and calls: PANE

SEQREV (A,N)

a. GENERAL DESCRIPTION

SEQREV reverses the sequence of N elements of array A.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SEQREV is called by: INITRF

and calls: No subroutines

SETBBH

a. GENERAL DESCRIPTION

SETBBH checks the user input values for baseboard heating and sets up the coefficients which compute the amount of baseboard heat as a function of outside temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SETBBH is called by: MZNDES

and calls: No subroutines

SETUPV(ISHAPE)

a. GENERAL DESCRIPTION

SETUPV computes the coordinates of the vertices of a surface. It processes user input according to the shape number which describes the surface.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SETUPV is called by: SURFIN

and calls: CROSSP

AD-A048 982 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAI--ETC F/G 13/1
THE BUILDING LOADS ANALYSIS AND SYSTEM THERMO-DYNAMICS (BLAST) --ETC(U)
DEC 77 D C HITTLE

UNCLASSIFIED CERL-TR-E-119-VOL-2 CEEDO-TR-77-35-VOL-2 NL

2 OF 5
ADA048982

SHADØ

a. GENERAL DESCRIPTION

SHADØ sets up two arrays (ISHD and JSHD) which direct the sequence of the shadowing calculations. These arrays list combinations of shadow-receiving surfaces and shadow-casting surfaces. The purpose of these arrays is to avoid checking every surface shadowing in the SHADOW routine.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SHADØ is called by: MZNDES

and calls: CHECK

SPCTRM

a. GENERAL DESCRIPTION

This subroutine allows integration of the transmittance and absorptance of the window with respect to wavelength. Normally only one wavelength is used, but up to 20 wavelengths may be used if physical properties of the window materials at those wavelengths are available. The one wavelength approximation seems quite good for common window glasses.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions

c. TRACE BACK

Subroutine SPCTRM is called by: WINDOW

and calls: BEAM

DIFFUS

DIFSHD

SURFIN (ITYPE, ISHAPE, NAS, NS, NGRS)

a. GENERAL DESCRIPTION

SURFIN computes zone surface geometric information which is necessary for shadowing calculations. Vertices, area, and direction cosines are computed for each surface.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SURFIN is called by: INITDS

[illegible]S

and calls: [illegible]N

INVERT

SETUPV

TSBF

a. GENERAL DESCRIPTION

This function was written to compute the transmittance of an interior shading device given the polarization, angle of incidence, and wavelength of the incident light. The function presently returns the transmittance at normal incidence specified by the user because of lack of data on the relationship of transmittance to the specified properties.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function TSBF is called by: DIFSHD
ISHADE

and calls: No subroutines

VIEWZS

a. GENERAL DESCRIPTION

VIEWZS computes the view factors between surfaces in the zone. The view factor from surface I to surface J is approximated by (area of J)/(sum of areas visible to I). This approximation is most accurate for a cubic zone. Most, but not all, of the surfaces visible to I are eliminated in computing the sum.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine VIEWZS is called by: MZNDES

and calls: No subroutines

WINDOW

a. GENERAL DESCRIPTION

This subroutine calls the routines which compute the coefficients of the polynomial expressions for transmittance and absorptance of a given window construction. It is in this routine that transmittance and absorptance for perpendicular and parallel light are combined in equal proportions for the assumption of unpolarized light. Values of the calculated transmittance and absorptance are saved for comparison in the glass report.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine WINDOW is called by: INITGL

calls to: SPCTRM

LSQFIT

ZERORE

a. GENERAL DESCRIPTION

ZERORE is a subroutine which calculates the residues related to the double pole occurring at zero. The basic calculation is like that of DER, i.e., calculation of the total construct and total derivative matrices. The layer matrices are multiplied to create the total construct matrix, and an order-preserving chain rule is used in calculating the total derivative matrix. Elements of both of these matrices are then combined to form the zero-related residues. The basic layer and layer derivative matrix element definitions are again from NECAP.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
RF	BETA(J)	Array of layer angles for trigonometric calculations
RF	NL	Number of layers
RF	R(J)	Array of layer resistances
RF	RES(J)	Array of layer R-factors

2. OUTPUT DATA

Source of Data	Name	Description
RFØ	CND	Thermal conductance of the construct
RFØ	ZRX	Zero residue element related to X response factors
RFØ	ZRY	Zero residue element related to Y response factors
RFØ	ZRZ	Zero residue element related to Z response factors

c. TRACE BACK

Subroutine ZERORE is called by: RESPNSE

and calls: No subroutines

ZERORE

1. Do for each layer

   2. Is R-factor ≠ Ø?

   3. Define layer and layer derivative matrices

   4. Define layer and layer derivative matrices

   5. Calculation of total derivative matrix elements

   6. Calculation of total construct matrix element

   7. Calculation of zero-related residue elements

<u>Note</u>: The definitions of the layer and layer derivative matrix elements depend on the value of the R-factor of the layer.

7 BLAST BUILDING DESCRIPTION FILE

The building description file is a transformation of the building and zone records on the simulation input file. The record locations remain the same as on the input file. Among the transformations done are computation of response factors, computation of glass coefficients, setup of shading geometry, setup of zone surfaces geometry, determination of shading surface combinations, recording zone heat transfer data, and determining zone view factors. Table 9 lists the routines used in manipulating the building description file.

Table 9

Description File Routines

Routine	Description
MSIMDF	Makes simulation description file
MZNDES	Makes zone description record
MBLDES	Makes building description record
OPSDEF	Opens file
CLSDEF	Closes file
WRZNDE/GTZNDE	Writes/gets zone description record
WRBLDE/GTBLDE	Writes/gets building description record

The simulation input data must be transferred for the description data; however, the record locations into the master index are the same as for the simulation input file. Figure 9 shows the general description file structure. Each building and zone description record consists of the necessary common blocks for the simulation (Figures 10 and 11).

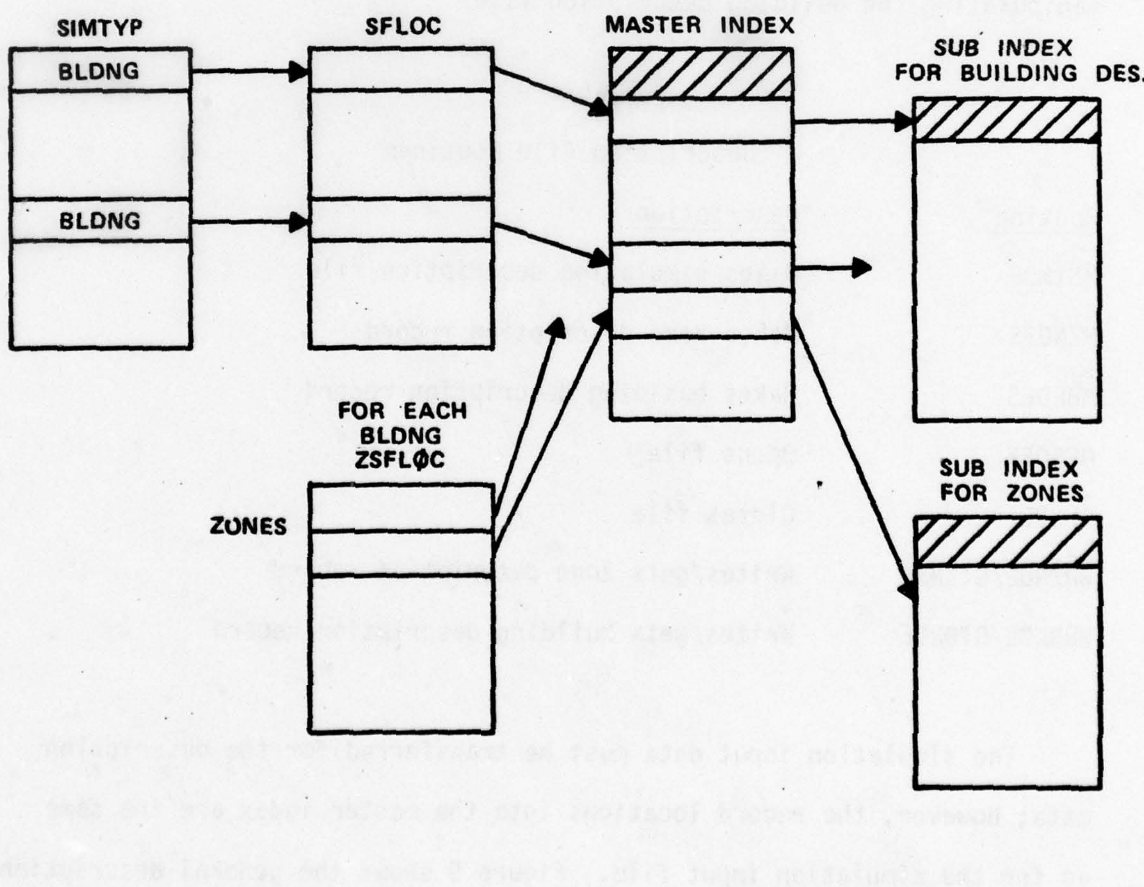


Figure 9. Simulation description file structure.

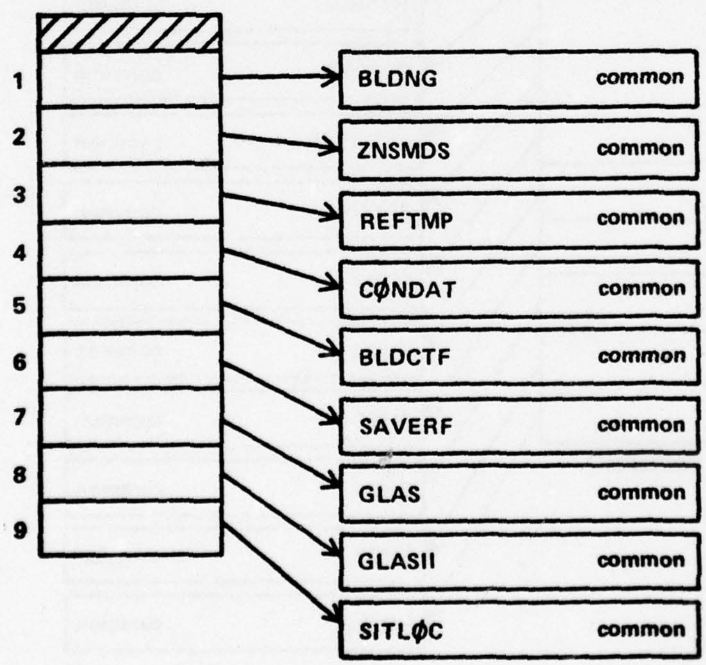


Figure 10. Building description record structure.

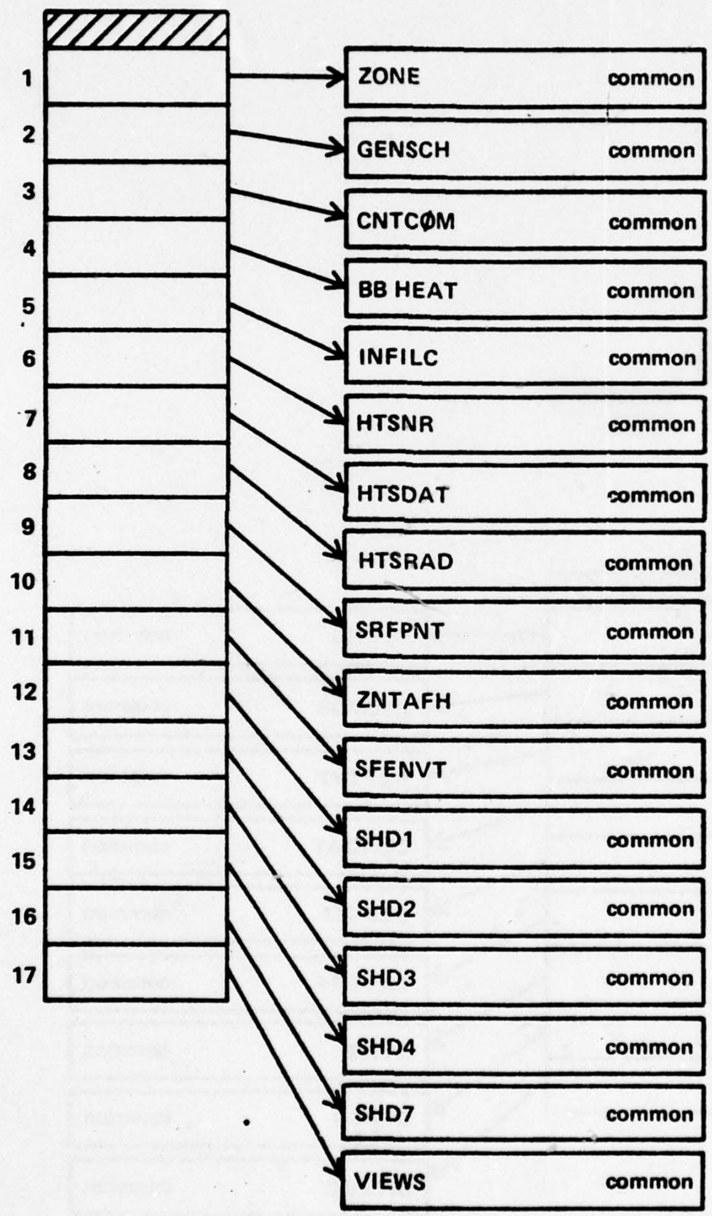


Figure 11. Zone description record structure.

8 SIMBLD (BUILDING LOAD CALCULATION)

The SIMBLD program subsystem computes the hourly zone (space) loads for each user-described space. Figure 12 shows the overall tree structure for the subsystem. The following pages present descriptions of SIMBLD and its subroutines (in alphabetical order). It should be noted that the TAG subroutine is also used by MSIMDF.

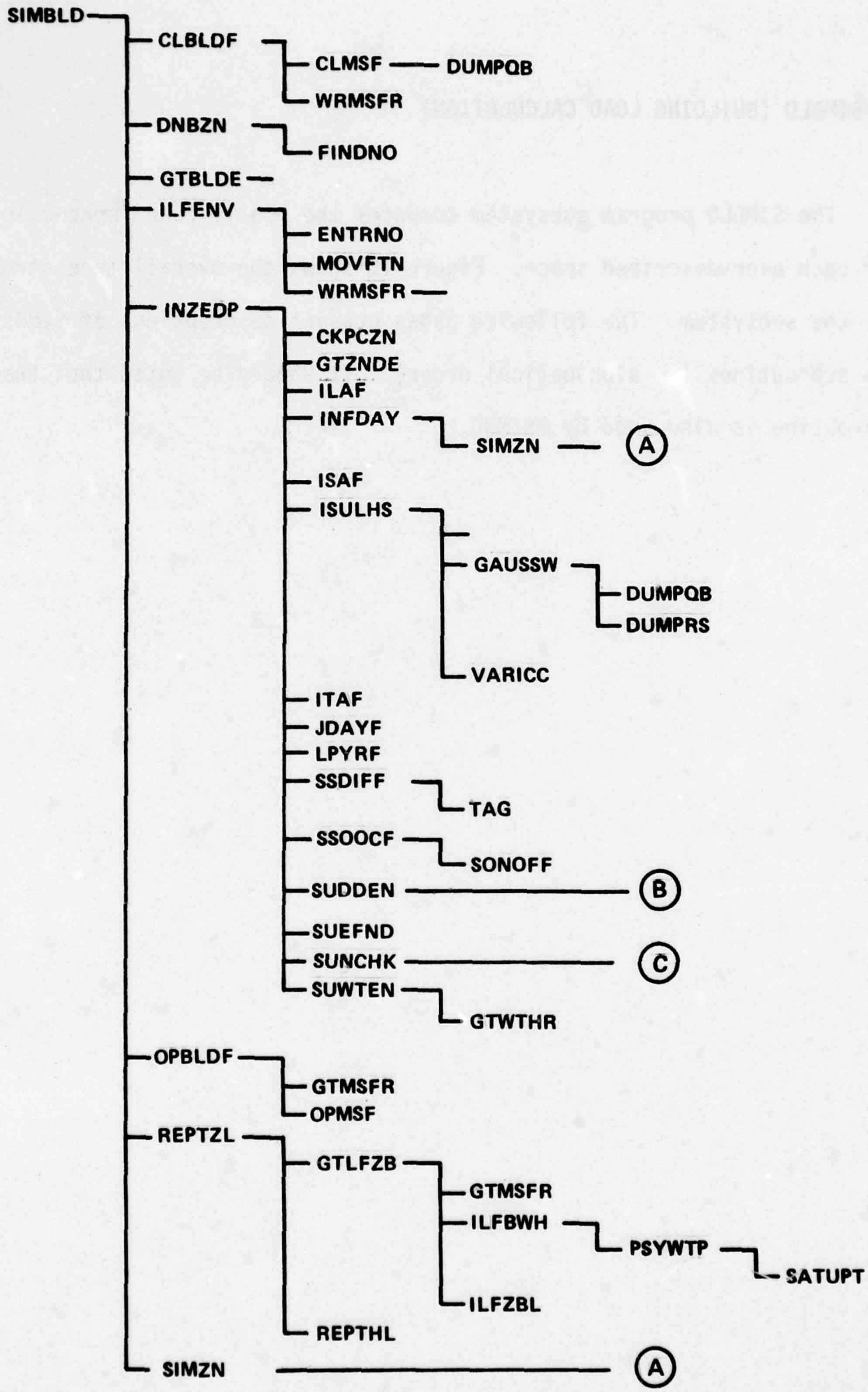


Figure 12. Tree structure for SIMBLD subroutines.

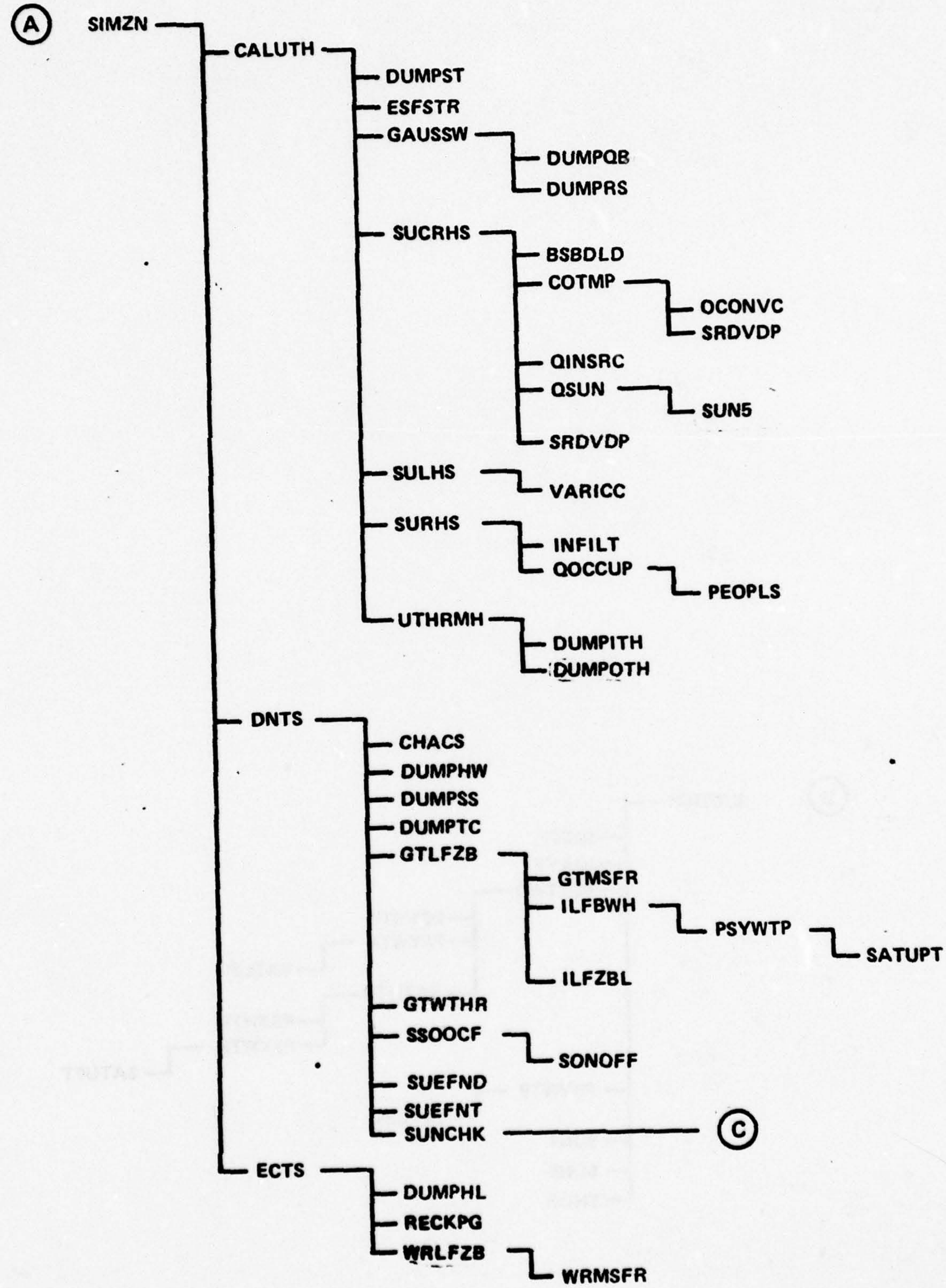


Figure 12 (cont'd).

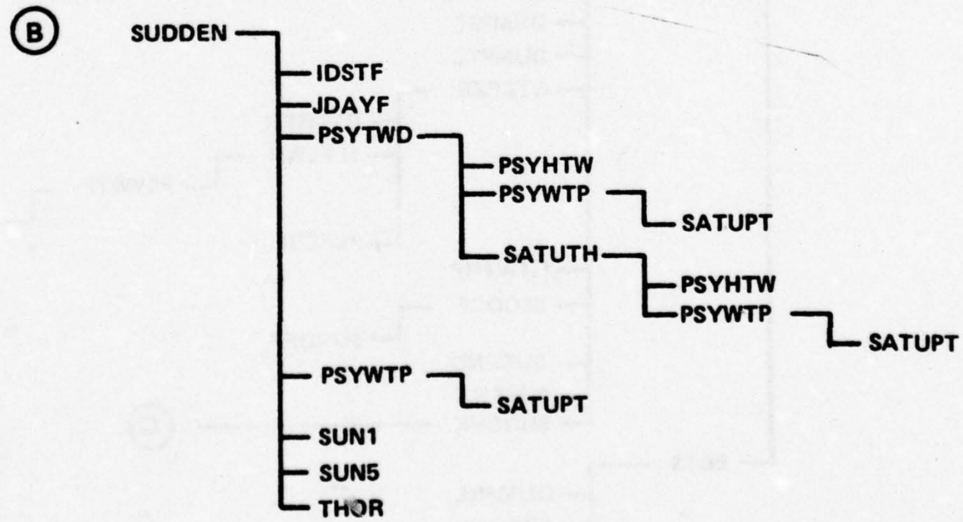


Figure 12 (cont'd).

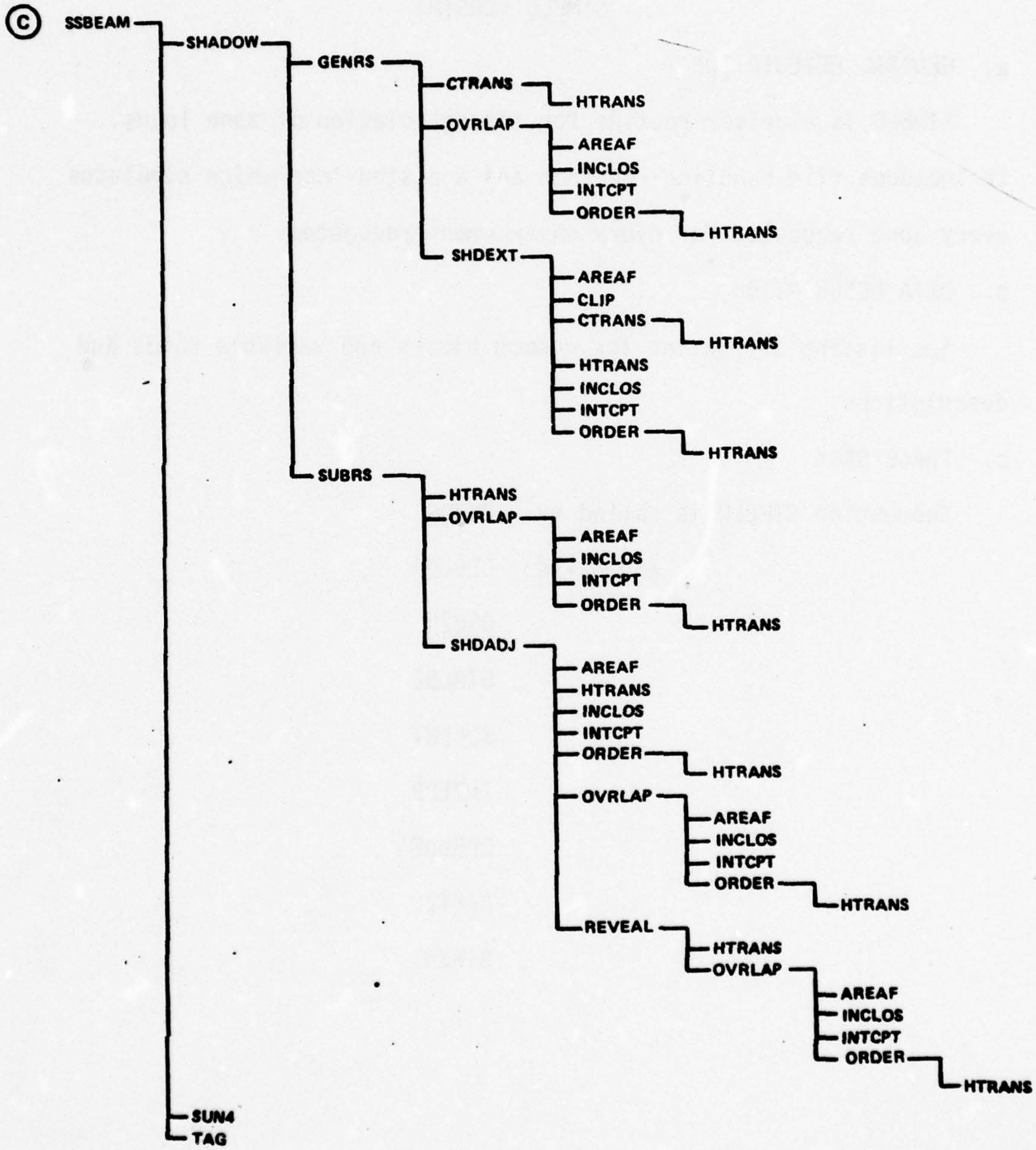


Figure 12 (cont'd).

SIMBLD (CRSIM)

a. GENERAL DESCRIPTION

SIMBLD is a driver routine for the calculation of zone loads. It includes file-handling routines and a nested loop which simulates every zone requested for every environment requested.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SIMBLD is called by: PERDS

and calls: CLBLDF

DNBZN

GTBLDE

ILFENV

INZEDP

OPBLDF

REPTZL

SIMZN

AREAF

a. GENERAL DESCRIPTION

This function computes the area of a plane polygon given the coordinates of its vertices. The algorithm is described in *CRC Standard Mathematical Tables*.[12] The vertices must be in sequence: clockwise sequence gives a negative area while counterclockwise sequence gives a positive area.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Function AREAF is called by: OVRLAP
SHDADJ
SHDEXT

and calls: No subroutines

[12] *CRC Standard Mathematical Tables*, 22nd edition (Chemical Rubber Company, 1972), p 369.

BSBDLD

a. GENERAL DESCRIPTION

BSBDLD simulates baseboard heating, as a heat input into the zone which is a function of outside temperature. It operates according to the heating on and off dates. Capacity is temperature-dependent.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine BSBDLD is called by: SUCRHS

and calls: No subroutines

CALUTH

a. GENERAL DESCRIPTION

CALUTH calculates the zone loads by solving the simultaneous equations which represent a heat balance between all surfaces of the zone. The subroutine is in three parts: (1) calculation of portions of the heat balance equations which are constant for the timestep, (2) an iterative loop which sets up the variable portions, solves the equations, and checks the solved zone air temperature against the heating/cooling control schedule, and (3) evaluation of the zone loads from the solved temperatures.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine CALUTH is called by: SIMZN

and calls: DUMPST

ESFSTR

GAUSSW

SUCRHS

SULHS

SURHS

UTHRMH

CHACS

a. GENERAL DESCRIPTION

By using a decision table to compare annual and daily schedules of heating and cooling, CHACS determines whether heating and/or cooling is on or off. Temperature-dependent and temperature-independent coefficients are computed for use in the air equation in the simultaneous heat balance.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine CHACS is called by: DNTS

and calls: No subroutines

CKPCZN

a. GENERAL DESCRIPTION

CKPCZN checks to see if an attic floor or crawl space ceiling temperature is to be calculated and recorded or used in the current zone simulation. Logical flags are set indicating the necessary actions, and critical surface numbers are recorded.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine CKPCZN is called by: INZEDP

and calls: No subroutines

CLIP

a. GENERAL DESCRIPTION

CLIP removes (clips) that portion of a shadow-casting surface which is below the plane of the shadow-receiving surface to prevent the casting of a false shadow. The method is the same as that used in NECAP and is described in the *NECAP Engineering Manual*[13] and the *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations.*[14]

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine CLIP is called by: SHDEXT

and calls: No subroutines

[13] *NECAP Engineering Manual* (National Aeronautics and Space Administration, 1974).

[14] *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations* (ASHRAE, 1975).

COTMP (HTS)

a. GENERAL DESCRIPTION

COTMP computes the outside surface temperatures of surfaces in the zone heat balance. This outside temperature is used to determine the conductive flux to the inside surface by means of conduction transfer functions, as detailed in *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations*.

Outside temperatures are computed according to the various possible outside environments. The effects of wind speed on the convection coefficient and solar heating gains are considered for surfaces exposed to ambient air. Underground surfaces are given the ground temperature. Surfaces exposed to conditioned zones are given the inside surface temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine COTMP (HTS) is called by: SUCRHS

and calls: OCONVC

SRDVDP

CTRANS

a. GENERAL DESCRIPTION

This subroutine transforms the coordinates of the vertices of a surface from the general coordinate system to a system based on one of the heat transfer surfaces. The transformation is the same as that used in NECAP and is described in the *NECAP Engineering Manual* and the *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations*.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine CTRANS is called by: GENRS

SHDEXT

and calls: HTRANS

DNBZN

a. GENERAL DESCRIPTION

DNBZN determines the next zone of the building to be simulated. It computes the pointers which locate the output in the building loads file.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine DNBZN is called by: SIMBLD

and calls: FINDNO

DNTS

a. GENERAL DESCRIPTION

DNTS increments the time counters and performs several time-dependent actions which must occur at the beginning of a time step. The zone load mass store record is read (GTLFZB) at the beginning of a day. The heating/cooling system status is checked (CHACS) every hour. The environment parameters for the hour are also set up every hour (SUEFNT). Upon change of day the weather file is read (GTWTHR) and changes in shadowing (SUNCHK) and heating/cooling status (SSOOCF) checked.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine DNTS is called by: SIMZN

and calls: CHACS

DUMPHW

DUMPSS

DUMPTC

GTLFZB

GTWTHR

SSOOCF

SUEFND

SUEFNT

SUNCHK

ECTS

a. GENERAL DESCRIPTION

ECTS performs actions necessary at the end of a time step. It records the calculated loads in the loads file. At the end of each day, it writes a load file daily block. It also checks for reaching the end of the zone simulation.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ECTS is called by: SIMZN

and calls: DUMPHL

RECKPG

WRLFZB

ESFSTR (EFSTMP)

a. GENERAL DESCRIPTION

ESFSTR determines what portion of the heating/cooling control schedule is in effect according to the zone air temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ESFSTR is called by: CALUTH

and calls: No subroutines

GAUSSW

a. GENERAL DESCRIPTION

GAUSSW is a routine for solving simultaneous linear algebraic equations where most of the terms in the coefficient matrix are constant. It uses a lower-upper (L-U) decomposition based on a standard Gaussian elimination. All nonconstant terms are grouped in the lower right corner of the coefficient matrix.

GAUSSW can be used to create the L-U decomposition by performing a Gaussian elimination down to the variable elements. It can also finish the elimination of the variable elements and perform the back substitution which solves the equations.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine GAUSSW is called by: CALUTH
ISULHS

and calls: DUMPBQ
DUMPRS

GENRS

a. GENERAL DESCRIPTION

GENRS computes the shadowing on a general receiving surface, i.e., on a base surface which may have subsurfaces. It uses the ISHD array to tell which surfaces to check as possible shadowers of the general receiving surface.

GENRS first checks the angle of incidence of the sun's ray to see whether the surface can be in the sun. It then checks whether there are any possible shadowing surfaces. If the surface cannot be in the sun or there are no possible shadowing surfaces, a simple default for sunlit area occurs. Otherwise, GENRS computes the shadows cast by all possible shadowing surfaces and determines how they overlap the base surface. The sunlit area computed includes the sunlit area of any subsurfaces.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine GENRS is called by: SHADOW

and calls: CTRANS

SHDEXT

OVRLAP

GTWTHR

a. GENERAL DESCRIPTION

GTWTHR reads one day's weather information from the weather file, which is a buffered file having checks for parity and end-of-file errors.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine GTWTHR is called by: DNTS

SUWTEN

and calls: No subroutines

GTZNDE (ZNDE)

a. GENERAL DESCRIPTION

GTZNDE calls the mass store routine which will read the correct zone from the zone description file.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine GTZNDE is called by: INZEDP

and calls: GTMSFR

HTRANS

a. GENERAL DESCRIPTION

HTRANS can be used to convert the cartesian coordinates of the vertices of a figure to homogeneous coordinates and to compute the homogeneous coordinates which describe the sides of the figure. It can also be used to compute the sides, given the vertices.

Homogeneous coordinates are useful for certain geometric calculations. A brief description is given in the listing of HTRANS. Newman and Sproull[15] also discuss the topic.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine HTRANS is called by: CTRANS
ORDER
REVEAL
SHDADJ
SHDEXT
SUBRS

and calls: No subroutines

[15] W. M. Newman and R. F. Sproull, *The Principles of Interactive Graphics* (McGraw-Hill, 1973), Appendix II.

ILAF

a. GENERAL DESCRIPTION

ILAF computes the fraction of the lighting radiant output which is absorbed by each surface in the zone. The fraction absorbed is proportional to the area of the surface and to its absorptance for radiation from lights.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ILAF is called by: INZEDP

and calls: No subroutines

ILFBWH

a. GENERAL DESCRIPTION

ILFBWH records weather data in the load file daily header.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ILFBWH is called by: GTLFZB

and calls: PSYWTP

ILFENV

a. GENERAL DESCRIPTION

ILFENV records header information for the load file.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ILFENV is called by: SIMBLD

and calls: ENTRNO

MOVFTN

WRMSFR

ILFZBL

a. GENERAL DESCRIPTION

ILFZBL sets all data in the zone block on the load file to zero.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ILFZBL is called by: GTLFZB

and calls: No subroutines

INCLOS

a. GENERAL DESCRIPTION

INCLOS determines which vertices of one figure are enclosed by another figure. The algorithm requires that the figures be convex and the vertices ordered clockwise. The test for enclosure is that the point lie to the right of all sides of the enclosing figure. Homogeneous coordinate techniques are used. A check is also made to prevent duplication of previously computed enclosed points.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine INCLOS is called by: OVRLAP
SHDADJ
SHDEXT

and calls: No subroutines

INFDAY (WTENV)

a. GENERAL DESCRIPTION

INFDAY initializes the temperature and flux histories of the zone surfaces by performing the zone load calculations several times for the first day of the simulation. Flags are set which prevent writing of output or advancing to the next day.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INFDAY is called by: INZEDP

and calls: SIMZN

INFILT

a. GENERAL DESCRIPTION

INFILT computes the infiltration mass flow by multiplying the scheduled infiltration by the Achenback and Coblenz factor.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INFILT is called by: SURHS

and calls: No subroutines

INTCPT

a. GENERAL DESCRIPTION

INTCPT computes all intercepts between the sides of two figures using homogeneous coordinate techniques. The procedure first eliminates cases where two sides (line segments) do not intersect and then computes the points of intersection. The points of intersection are stored with the enclosed points. Together they describe the vertices of the overlap between two convex polygons.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine INTCPT is called by: OVRLAP
SHDADJ
SHDEXT

and calls: No subroutines

INZEDP

a. GENERAL DESCRIPTION

INZEDP performs initialization calculations which are needed before the hourly zone load calculations are made. The purpose of this routine is to remove all one-time calculations from the hourly loop. These one-time calculations are described by the subroutines called.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INZEDP is called by: SIMBLD

and calls: CKPCZN

GTZNDE

ILAF

INFDAY

ISAF

ISULHS

ITAF

JDAYF

LPYRF

SSDIFF

SSOOCF

SUDDEN

SUEFND

SUNCHK

SUWTEN

ISAF

a. GENERAL DESCRIPTION

ISAF computes the fraction of the solar radiation transmitted into the zone which is absorbed by each surface. The fraction absorbed is assumed to be proportional to the area of the surface and to its absorptance for solar radiation. This is a good approximation for diffuse radiation, but not for beam radiation.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ISAF is called by: INZEDP

and calls: No subroutines

ISULHS

a. GENERAL DESCRIPTION

ISULHS sets up the constant portion of the left-hand side of the heat balance matrix. The form of the matrix is described in the Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations. An L-U decomposition is then performed (GAUSSW) on the matrix and the results are saved for the hourly load calculations.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ISULHS is called by: INZEDP
and calls: DUMPQB
GAUSSW
VARICC

ITAF

a. GENERAL DESCRIPTION

ITAF computes the fraction of thermal radiation from occupants and equipment which is absorbed by each surface in the zone. The fraction absorbed is proportional to the area of the surface and to its absorptance for thermal radiation.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ITAF is called by: INZEDP

and calls: No subroutines

JDAYF (MO, IDAY, LPYR)

a. GENERAL DESCRIPTION

This function calculates the day of the year from the month, day of the month, and the leap year indicator.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function JDAYF is called by: INZEDP
SUDDEN
and calls: *No subroutines*

LPYRF (YEAR)

a. GENERAL DESCRIPTION

LPYRF calculates the leap year indicator, which is 1 if the year is a leap year and Ø if it is not.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function LPYRF is called by: INZEDP

and calls: No subroutines

OCONVC (HTS)

a. GENERAL DESCRIPTION

OCONVC computes the outside convection coefficient of a surface based on the wind speed, wind and wall directions, and wall roughness. It uses the method presented in the *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations.*

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine OCONVC is called by: COTMP

and calls: No subroutines

ORDER

a. GENERAL DESCRIPTION

ORDER sorts the vertices defining an overlap (vertices computed by INCLOS and INTCPT) into clockwise sequence. The overlap is a convex polygon. Its left-most vertex is found first, and then the slopes from that vertex to all others are computed. The slopes are sorted into decreasing order; the accompanying points are then in clockwise sequence.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine ORDER is called by: OVRLAP

SHDADJ

SHDEXT

and calls: HTRANS

OVRLAP

a. GENERAL DESCRIPTION

OVRLAP determines the overlaps between the current figure and a sequence of previous figures. The vertices which define an overlap consist of all vertices of figure 1 enclosed by figure 2 plus all vertices of figure 2 enclosed by figure 1 plus all intercepts of figure 1 and figure 2. These are then sorted into clockwise sequence. The area of the overlap is computed using the following sign convention:

base surface - positive

shadow - negative

overlap of two shadows - positive

etc.

such that the sum of the areas equals the sunlit area of the base surface.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine OVRLAP is called from: GENRS
REVEAL
SHDADJ
SUBRS

and calls: AREAF
INCLOS
INTCPT
ORDER

PEOPLS (T, Q)

a. GENERAL DESCRIPTION

PEOPLS computes the sensible heat gain from one person given his/her activity level (Q) and the air temperature (T). The equation was derived from a curve fit at 75°F and 80°F of data on total heat transfer obtained from Table 29 of the ASHRAE Handbook of Fundamentals.[16]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PEOPLS is called by: QOCCUP

and calls: No subroutines

[16] *ASHRAE Handbook of Fundamentals* (ASHRAE, 1972) Chapter 22, Table 29 See the "Total Heat Adjusted" column.

QINSRC

a. GENERAL DESCRIPTION

QINSRC determines heat gains from lighting and equipment according to their schedules and the type of day (weekday, weekend, or holiday). The number of occupants is also computed.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine QINSRC is called by: SUCRHS

and calls: No subroutines

QOCCUP (ZNTMP)

a. GENERAL DESCRIPTION

QOCCUP computes the sensible and latent heat gains from occupants as functions of zone air temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine QOCCUP is called by: SURHS

and calls: PEOPLS

QSUN

a. GENERAL DESCRIPTION

QSUN computes solar heat gains and all heat transfer surfaces. It evaluates solar intensities and multiplies these by coefficients computed in SSBEAM to compute the solar heat gain on each surface, and the sunlight transmitted and convected into the zone.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine QSUN is called by: SUCRHS

and calls: SUN5

RECKPG

a. GENERAL DESCRIPTION

RECKPG is called every hour to record zone loads information. It records certain weather information, sums the zone loads into the building loads, and records the zone loads.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RECKPG is called by:ECTS

and calls: No subroutines

REPTHL (ZNR, OUTPUT, ENGLSH, HOURLY)

a. GENERAL DESCRIPTION

REPTHL computes data for the zone load report (REPTZL) and prints data if hourly loads are requested. Positive and negative loads are divided into heating and cooling loads. Certain maximum and minimum loads and temperatures are also computed.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTHL is called by: REPTZL

and calls: No subroutines

REPTZL (NE, NZ)

a. GENERAL DESCRIPTION

REPTZL reports the loafs calculated by SIMZN for environment number NE and zone number NZ. Hourly, daily, and monthly reports are possible, but hourly reports are limited to 10 days to prevent excessive output.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine REPTZL is called by: SIMBLD

and calls: REPTHL

GTLFZB

REVEAL

a. GENERAL DESCRIPTION

This subroutine computes the shadowing effects of a reveal on a subsurface (window or door). It projects the window up to the plane of the wall and determines the overlap of the projected window surfaces with all previously computed shadows on the window.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine REVEAL is called by: SHDADJ

and calls: HTRANS

OVRLAP

SHADOW

a. GENERAL DESCRIPTION

SHADOW steps through two previously created arrays to determine which surfaces will be checked for shadowing. It also determines the cosine of the angle of incidence of the sun's rays on all surfaces. It determines the shadows on the base surface (GENRS) and then checks each subsurface of the base surface (SUBRS).

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine SHADOW is called by: SSBEAM

and calls: GENRS

SUBRS

SHDADJ

a. GENERAL DESCRIPTION

SHDADJ computes the shadows from five types of shadowing surfaces whicn can only shade subsurfaces. Only one type (reveals) has been implemented in the user language. It uses the same overlap technique as SHDEXT and OVRLAP.

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine SHDADJ is called by: SUBRS

and calls: AREAF

HTRANS

INCLOS

INTCPT

ORDER

OVRLAP

REVEAL

SHDEXT

a. GENERAL DESCRIPTION

SHDEXT reduces the shadow cast into the plane of a general receiving surface to only that portion which overlaps the surface. First the coordinates of the shadowing surface are transformed (CTRANS) to a system relative to the receiving surface. The shadowing surface is then clipped (CLIP) and projected onto the receiving plane. The cartesian coordinates are transformed (HTRANS) to homogeneous coordinates. The vertices of the overlap are computed (INCLOS and INTCPT) and put in clockwise sequence (ORDER). Finally, the shaded area is computed (AREAF).

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine SHDEXT is called by: GENRS

and calls: AREAF

CLIP

CTRANS

HTRANS

INCLOS

INTCPT

ORDER

SIMZN

a. GENERAL DESCRIPTION

SIMZN is a simple driver program for the hourly calculation of zone loads. It calls subroutines which advance the time counters and perform appropriate beginning-of-hour operations (DNTS), calculate the loads for the hour (CALUTH), and perform operations and checks at the end of the hour (ECTS). An end-of-simulation parameter is checked.

SIMZN is called to initialize the thermal histories and for the actual load calculations.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SIMZN is called by: INFDAY
SIMBLD

and calls: CALUTH
DNTS
ECTS

SONOFF (DAYOFY, ONDAY, OFFDAY, ONOROF)

a. GENERAL DESCRIPTION

SONOFF determines whether the current day of the year is between the day a system is turned on and the day it is turned off.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SONOFF is called by: SSOOCF

and calls: No subroutines

SRDVDP (A, B, C, D, N)

a. GENERAL DESCRIPTION

SRDVDP evaluates the portion of a conductive heat flux which is due to the temperature history and the conduction transfer functions. The *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations* provides details. SRDVDP can be expressed mathematically as:

$$SRDVDP = \sum_{i=1}^{N} A_i \cdot B_i - \sum_{i=1}^{N-1} C_i \bullet D_i$$

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SRDVDP is called by: COTMP

SUCRHS

and calls: No subroutines

SSBEAM

a. GENERAL DESCRIPTION

SSBEAM calculates the elements of COMMON /SHDSRF/ which apply to beam radiation for the given solar declination. These elements contain enough information so that only the hourly beam and diffuse intensities of solar radiation are needed to compute solar radiation absorbed on each surface.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SSBEAM is called by: SUNCHK

and calls: SHADOW

SUN4

TAG

SSDIFF

a. GENERAL DESCRIPTION

SSDIFF computes factors for the inside and outside of each heat transfer surface which, when multiplied by the intensity of diffuse solar radiation, give the amount of diffuse radiation on each surface.

b. DATA DESCRIPTION

See listing of routing for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SSDIFF is called by: INZEDP

and calls: TAG

SSOOCF

a. GENERAL DESCRIPTION

SSOOCF sets the flags indicating heating and cooling system status. Status is based on the day of the year on which each system is turned on or turned off.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SSOOCF is called by: DNTS

INZEDP

and calls: SONOFF

SUBRS

a. GENERAL DESCRIPTION

SUBRS determines the sunlit areas of all subsurfaces of a given base surface. It also revises the sunlit area of the base surface by subtracting the sunlit area of the subsurfaces. The routine first checks the subsurface with all previously computed shadows (OVRLAP) and revises the sunlit area of the base surface. It then checks shadowing from surfaces designated as only subsurface shadowers (SHDADJ).

b. DATA DESCRIPTION

See listing of routine for variable names and descriptions.

c. TRACE BACK

Subroutine SUBRS is called by: SHADOW

and calls: HTRANS

OVRLAP

SHDADJ

SUCRHS

a. GENERAL DESCRIPTION

SUCRHS sets up the constant portion of the right-hand side of the zone heat balance equations, which are described in the *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations*. Heat gains which are independent, or nearly independent, of the zone inside temperatures are computed. This includes solar gains (QSUN), equipment and lighting (QINSRC), baseboard heat (BSBDLD), and conduction from the outside of surfaces (COTMP). Conduction is handled by means of conduction transfer functions which are also described in *Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUCRHS is called by: CALUTH

and calls: BSBDLD

COTMP

QINSRC

QSUN

SRDVDP

SUDDEN (DDENV)

a. GENERAL DESCRIPTION

SUDDEN converts user input describing a design day to a full set of weather information for the day. The weather information is in the same common block that is used for the weather file. SUDDEN and the program for generating weather files use the same algorithms.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUDDEN is called by: INZEDP
and calls: IDSTF
JDAYF
PSYTWD
PSYWTP
SUNI
SUN5
THOR

SUEFND

a. GENERAL DESCRIPTION

SUEFND is called once per day to transfer daily data from COMMON /WTHRFL/ to other common blocks. The day type for indexing schedules is computed.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUEFND is called by: DNTS
INZEDP
and calls: No subroutines

SUEFNT

a. GENERAL DESCRIPTION

SUEFNT adjusts the temperature and flux history arrays by moving each value for hour H to hour H + 1. It also transfers several elements of weather information from 24-hour arrays to single-element variables.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUEFNT is called by: DNTS

and calls: No subroutines

SULHS

a. GENERAL DESCRIPTION

SULHS sets up the variable portion of the left-hand side of the heat balance equations. The variable elements are those with variable internal convection coefficients and the equation for the air heat balance. These elements have been moved to the bottom portion of the heat balance matrix to effectively use the L-U decomposition solution technique.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SULHS is called by: CALUTH

and calls: VARICC

SUN1 (DAYYR, AA, BB, CC, AVSC)

a. GENERAL DESCRIPTION

Subroutine SUN1 computer coefficients for determining solar position and intensity for design day environments. This routine is almost identical to subroutine SUN in the WIFE program.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUN is called by: SUDDEN

and calls: No subroutines

SUN4 (HOUR)

a. GENERAL DESCRIPTION

SUN4 computes the direction cosines of the sun for the declination and equation of time which are set in subroutine SUNCHK. The direction cosines are used to evaluate shadowing. The algorithm is the same as that used in SUN[illegible].

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUN4 is called by: SSBEAM

and calls: No subroutine

SUN5(HOUR)

a. GENERAL DESCRIPTION

SUN5 evaluates the direction cosines of the sun for the declination and equation of time of that day and the solar time (HOUR). For details, see *Thermal Environmental Engineering* by Threlkheld[17] and the *NECAP Engineering Manual*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUN5 is called by: QSUN

SUDDEN

and calls: No subroutines

[17] J. L. Threlkheld, *Thermal Environmental Engineering* (Prentice-Hall, 1970).

SUNCHK

a. GENERAL DESCRIPTION

SUNCHK determines if a new set of sunlit surface areas must be calculated because of a change in solar declination. Fourteen time-spans have been selected to minimize the maximum error in declination angle for shadowing calculations. The timespans center on the declinations in the DEC array. The equation of time value is the mean value for the timespan. J and JOLD determine when a new timespan has been entered and another shadowing calculation must be made.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUNCHK is called by: DNTS

INZEDP

and calls: SSBEAM

SURHS

a. GENERAL DESCRIPTION

SURHS sets up portions of the right-hand side of the heat balance equations which depend on the zone temperature. These include heat gains from occupants, infiltration, and the heating/cooling system.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SURHS is called by: CALUTH

and calls: INFILT

QOCCUP

SUWTEN(WTENV)

a. GENERAL DESCRIPTION

SUWTEN reads the weather file and positions the file at the record desired by the user. Error checks are included for no weather file or reading past the end of the file.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUWTEN is called by: INZEDP

and calls: GTWTHR

TAG

a. GENERAL DESCRIPTION

This subroutine computes the approximate transmittance and absorptances of a particular window type as a function of the cosine of the angle of incidence. It uses the coefficient recorded by subroutine WINDOW. The polynomial expressions are evaluated by Horner's rule.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine TAG is called by: REPTGL
SSBEAM
SSDIFF

and calls: No subroutines

UTHRMH

a. GENERAL DESCRIPTION

UTHRMH records the current hour's temperatures and conductive fluxes for the surfaces of the zone.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine UTHRMH is called by: CALUTH

and calls: DUMPITH

DUMPOTH

VARICC (HTS)

a. GENERAL DESCRIPTION

VARICC computes the inside surface convection coefficient as a function of surface tilt and temperature relative to zone air. Vertical surfaces have a constant coefficient. Upward- or downward-facing surfaces have variable coefficients depending on relative temperature. The coefficients are taken from the NBSLD program.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine VARICC is called by: ISULHS
SULHS

abd calls: No subroutines

9 BLAST BUILDING LOADS FILE

Use

The building loads file (BLDLF) is used to pass zone heating/cooling load information between SIMBLD and SIMSYS. It also passes weather information that is used by SIMSYS and SIMTEP. It allows up to five environments (i.e., design days and weather data runs), which cannot be changed after initial setup, and up to 100 zones, which can be added or replaced from run to run.

Type and Structure

The building loads file is a mass store file--a special type of file under the CDC SCOPE operating system.

The first record of the file is a header record containing a table of zones and a table of the environments present in the file. The zone table contains user-supplied logical zone numbers and titles. The *environment* table contains the record number of the environment header record which is read in before accessing (or created before generating) daily load information.

The environment header record contains a description of the environment time period: starting date and number of days. After each environment header are N records(where N is the number of days). Each of these records contains building summary and weather information and the sub-index array which allows access to blocks of daily zone load data. Each block of zone data contains 24 hours of 7 load data/hour/zone, and (currently) 10 zones; there is also a safety indicator to verify that information for any zone in that block was output for that day.

Common Blocks

The common blocks associated with the building loads file are:

/LFHEDR/ - the common block buffer that holds the load file header

/LFEHDR/ - the common block buffer that holds the current environment header

/LFBDWB/ - the common block buffer that holds the building daily weather block

/LFZDLB/ - the common block buffer that holds the zone daily load block

/LFSMDS/ - the common block that contains the current access state of the loads file (current environment, current zone position, etc.)

/SIMBLF/ - the common block that contains file information such as the master index array (LFMIA) records length, etc.

/MSFIOP/ - the common block that contains general mass store file parameters.

Access

Accessing of the building loads file is embedded in two layers of subroutines with carefully designed calling sequences. The outermost subroutines are:

a. OPBLDF, which opens the load file and gets the header record

b. CLBLDF, which writes the (potentially modified) header record and closes the file

c. GTLFZB (env, lsp, dos), which gets the load file zone block for environment (env), load file zone position (lzp), and day of simulation (dos)

d. WRLFZB (env, lzp, dos), which writes the load file zone block for env, lzp, and dos.

The calling sequences here are designed to be as simple as possible to minimize error in usage.

The innermost layer of subroutines consists of:

a. OPMSF (msfl, msfmia, lmsfm, cmsfia), which opens the mass store file, msfl (BLDFL), using msfmia (LFMIA) as its master index array of length lmsfm (LLFMIA) and the current mass store file index array indicator cmsfia (CBLFIA), which it checks to see if it is null (file closed) and sets to indicate the master index array.

b. CLMSF (msfl, msfmia, lmsfm, cmsfia), which closes the mass store file, msfl (BLDFL), switching to msfmia (LFMIA) which is the master index array of length lmsfm; if the current index array indicator cmsfia indicates that the file is currently on a subindex array, cmsfia is set to null.

c. GTMSFR (msfl, msfia, lmsfia, cmsfia, inxtyp, bufwa, bufln, recloc), which gets record number recloc from the mass store file msfl using the index array msfia of length lmsfia designated by inxtyp which is compared to the current index array indicator cmsfia (and switched to, if necessary); the record is read into a buffer of length bufln starting at first word address bufwa; if the actual record length read in differs from bufln, it flags an error.

d. WRMSFR (msfl, msfia, lmsfia, cmsfia, inxtyp, bufwa, bufln, recloc, imf), which writes record number reeloc from the mass store file msfl using the index array msfia of length lmsfia designated by inxtyp which is compared to the current index array indicator cmsfia (and switched to, if necessary); the record is read from a buffer of length bufln starting at first word address bufwa; the index marker flag imf serves no function except to allow for compatibility with planned extensions of CDC SCOPE file editing routines.

Usage Scenario

Before input processing begins, the building load mass store file is opened; if it was attached by the user, the header record describing environments and already existing zones is read in. Otherwise, it remains at the default values showing an empty file. During input processing of building descriptions, this table is checked/modified for replacing/adding zones. Before the file is closed, this header record is written back to it.

During building load calculations, the building load file is updated. This means that at the beginning of the simulated day, the proper zone block must be read in (by GTLFZB) and at the end of the day the modified block must be written out (by WRLFZB). The data for a single zone are modified for each day of the current environment and the process repeated for each zone simulated. To read in a zone block, GTLFZB reads in the building/weather block for that day (initializes if no record currently exists), switches to the zone subindex array contained in that block, and uses that to read in the proper zone block. To write out the zone block, the block is written and then the building/weather block with the updated subindex array is written out.

During air handler load calculations, the building load file is not updated. However, for each simulation day any number of zones are accessed simultaneously. In this case, GTLFZB reads the building/weather block for that day only once and reads a new zone load data block in only when a zone not in the resident block is requested.

Record Structure Descriptions

The building load file header record - /LFHEDR/ - contains:

a. Version number

b. Maximum number of zones in load file (currently 100)

c. Number of zones currently in load file

d. Maximum number of environments (currently five)

e. Number of environments currently in load file

f. Maximum number of records in load file (currently 376)

g. Number of records currently in load file

h. Table of environment base (header) record pointers

i. Table of user-supplied logical zone numbers (relating them to position in load file)

j. Table of zone titles

k. Number of words in zone title

l. Project title

m. Number of words in project title

The *environment* header - /LFEHDR/ - contains:

a. Environment number

b. Environment type

c. Number of days in environment

d. Title of environment

e. Number of environment title words

f. Starting year

g. Starting month

h. Starting day of month

i. Starting day of year

The building summary/daily weather block -/LFBDWB/- contains:

a. Environment number

b. Weather station

c. Year

d. Month

e. Day of month

f. Daylight savings time indicator

g. 24 hours of:

    1. Outdoor dry bulb

    2. Outdoor wet bulb

    3. Outdoor barometric pressure

    4. Outdoor humidity ratio

    5. Solar radiation - beam

       - diffuse

       - ground reflected

    6. Solar direction cosines - east

       - north

       - vertical

    7. Building load summaries (all zones)

h. Subindex array for zone daily load blocks

i. 24 hours of:

    1. Attic surface temperatures

    2. Crawl space surface temperatures

The zone daily load block - /LFZDLB/ - contains:

a. Zone indicator (indicates which zones in block have been written for that day)

b. Number of zone load data (currently seven)

c. 24 hours of load data for each zone in block, including:

1. Sensible load
2. Latent load
3. Space air temperature
4. Return air load
5. Baseboard load
6. Electric load
7. Infiltration mass flow

Figure 13 diagrams the basic structure of the building loads file.

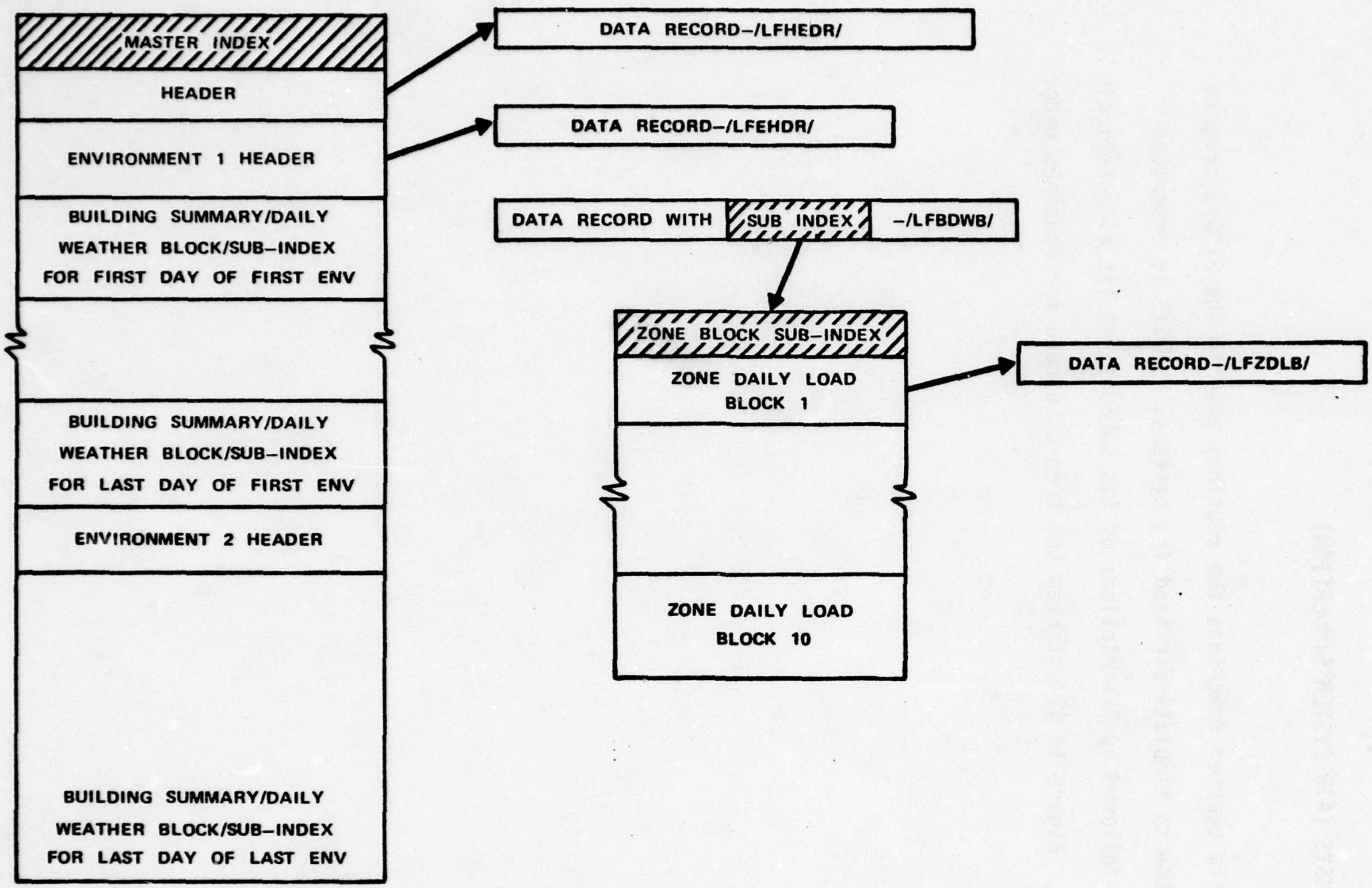


Figure 13. Building loads file structure.

10 SIMSYS (AIR SYSTEM SIMULATION)

This chapter describes the routines used in the SIMSYS program subsystem to simulate air handling systems. SIMSYS is described first, followed by descriptions of its subroutines (in alphabetical order). Figure 14 illustrates the tree structure for routines under SIMSYS.

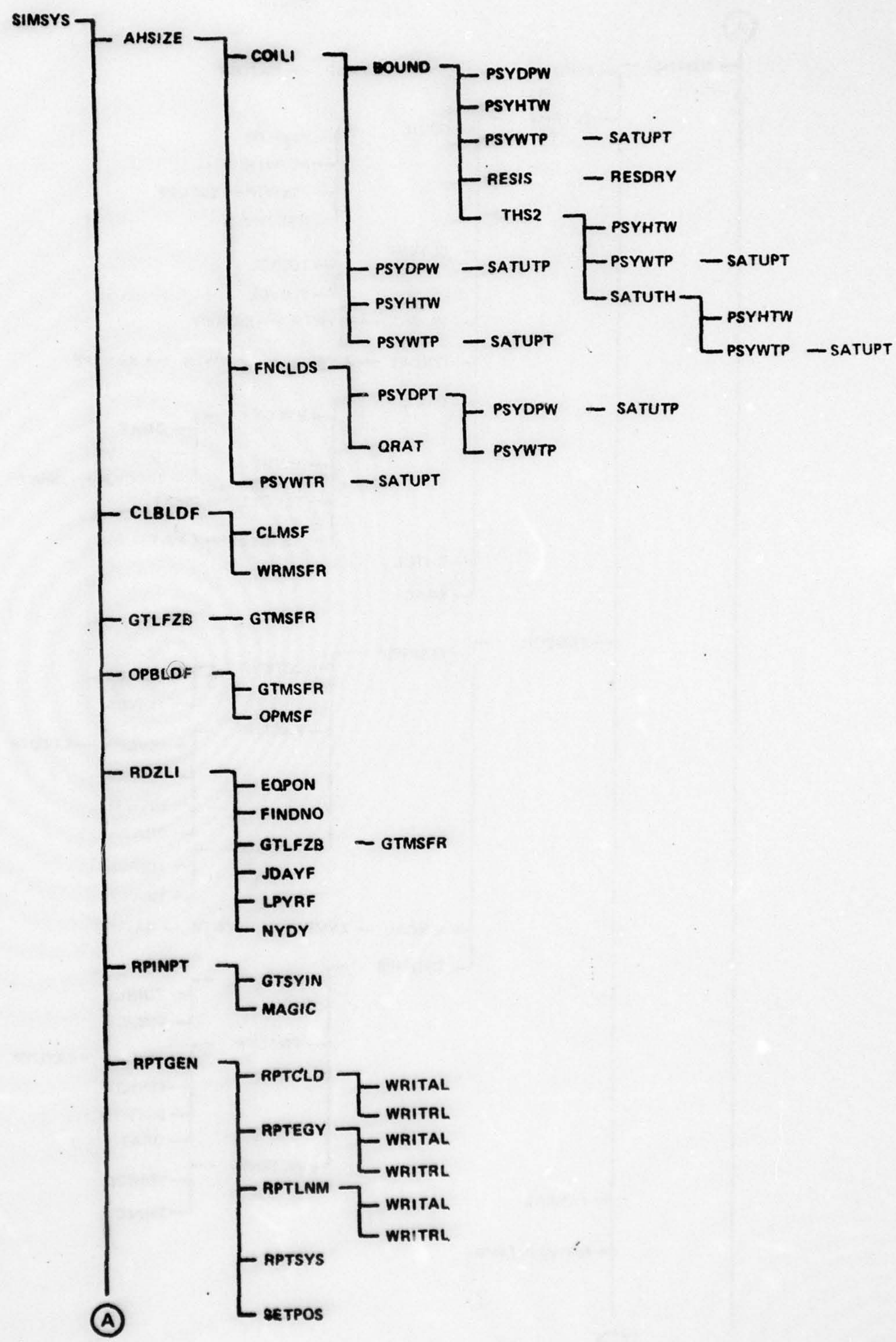


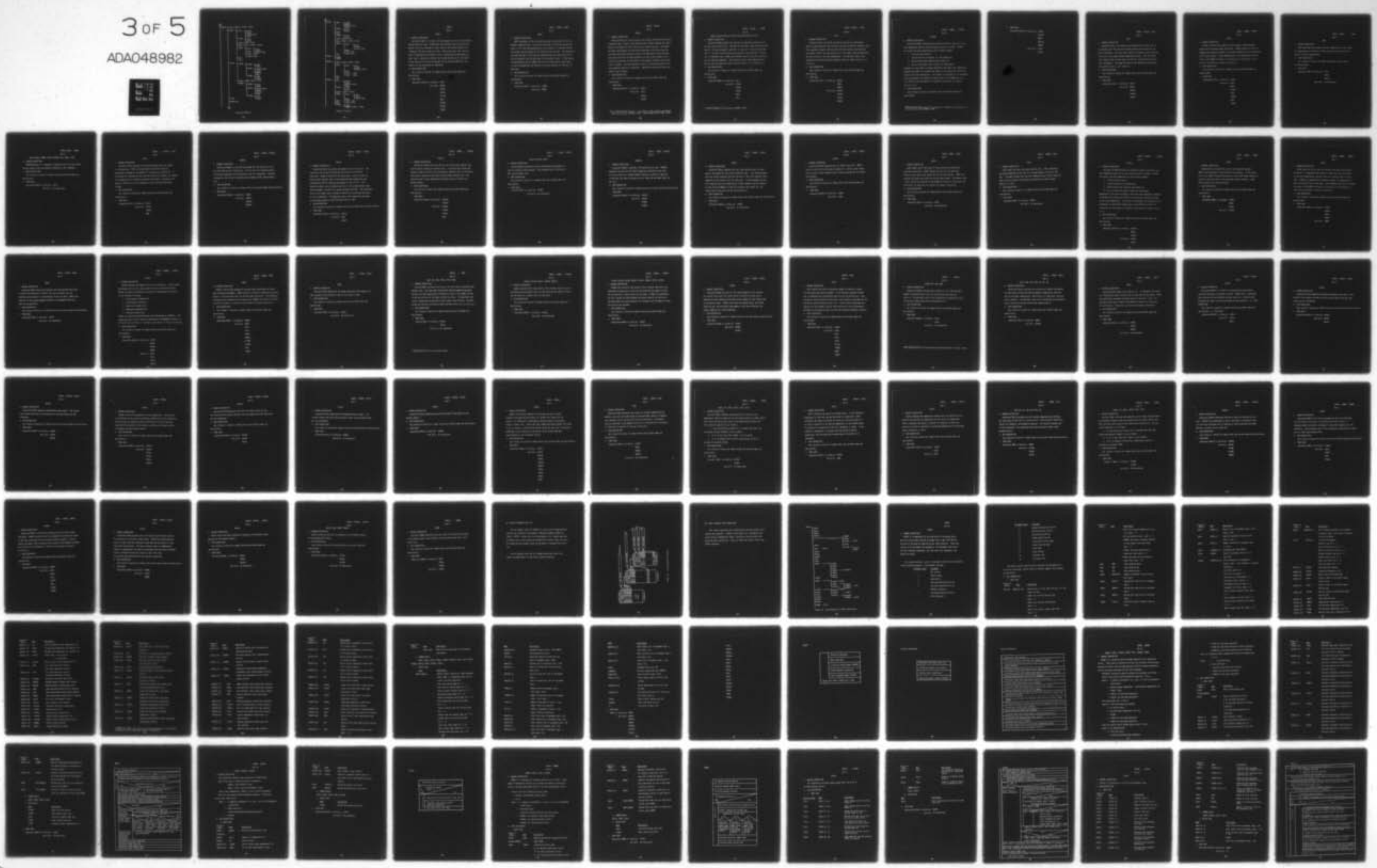
Figure 14. Tree structure for SIMSYS subroutines.

AD-A048 982 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN--ETC F/G 13/1
THE BUILDING LOADS ANALYSIS AND SYSTEM THERMO-DYNAMICS (BLAST) --ETC(U)
DEC 77 D C HITTLE

UNCLASSIFIED CERL-TR-E-119-VOL-2 CEEDO-TR-77-35-VOL-2 NL

3 OF 5

ADA048982



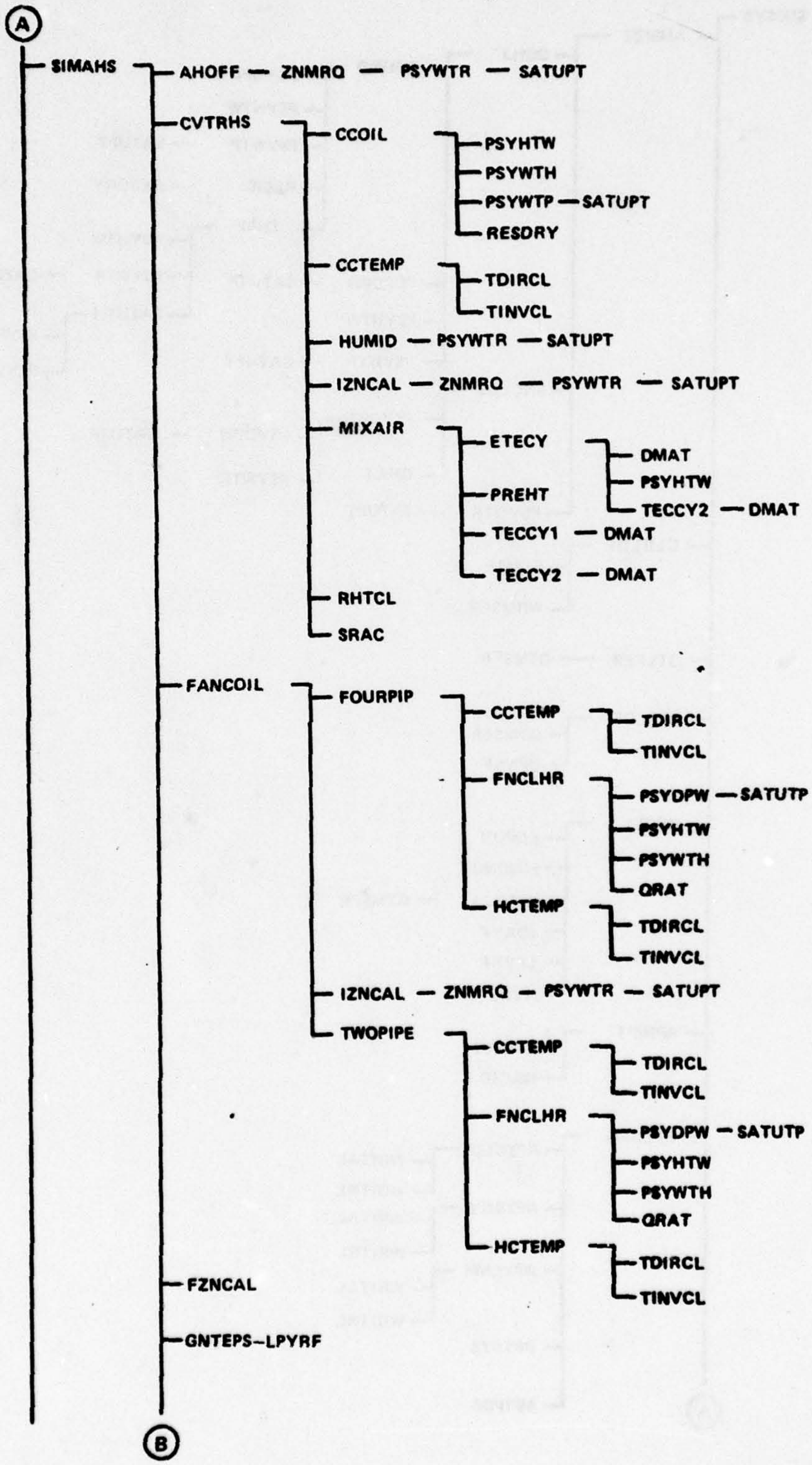


Figure 14 (cont'd).

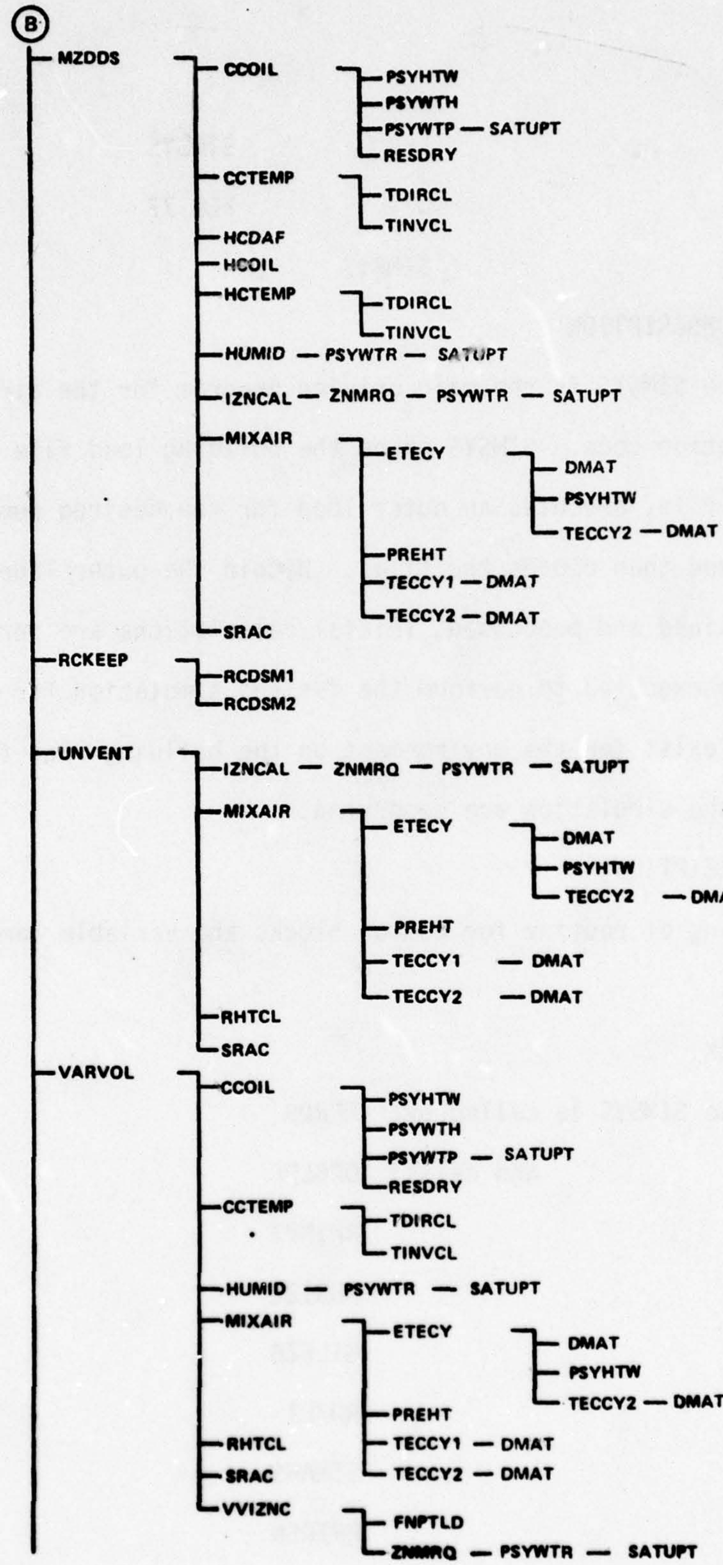


Figure 14 (cont'd).

SIMSYS

a. GENERAL DESCRIPTION

Subroutine SIMSYS is the main driving program for the air distribution system simulation code. SIMSYS opens the building load file and the air handler load file, executes an outer loop for the desired number of environments, and then closes the files. Within the outer loop, the user input is obtained and processed, initial calculations are performed, an inner loop is executed to perform the desired simulation for the number of days that exist for the environment on the building load file, and reports for the simulation are generated.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SIMSYS is called by: PERDS

and calls: OPBLDF

RPINPT

AHSIZE

GTLFZB

RDZLI

SIMAHS

RPTGEN

CLBLDF

AHOFF

a. GENERAL DESCRIPTION

AHOFF determines if the air handling system can be shut off for the present simulation hour. If the user specifies (1) that the system must be on, or (2) that the system may be off and there is a sensible load on the system, the fan system will be turned on for the hour. The system will be turned off ONLY IF the user has specified that the system may be off and the sensible load for each zone on the system is zero. If the system is determined to be off, AHOFF sets all coil loads and all unmet loads to zero and determines the effects of moisture addition for the hour on each zone.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine AHOFF is called by: SIMAHS

and calls: ZNMRQ

AHSIZE

a. GENERAL DESCRIPTION

AHSIZE performs the initialization calculations required by the hourly simulation code. Based on user-supplied data, AHSIZE computes the supply, return, and exhaust air mass flow for each zone and system. The system fan power requirements and the design temperature rise across each fan are determined using procedures used in the CERL Thermal Loads Analysis and System Simulation Program (presented in CERL Interim Report E-81).[18] Initial calculations are performed for the preheat, heating, and cooling coils if needed. The zone humidity ratios are initialized based on the user-supplied humidifier information. AHSIZE also zeroes the hourly and monthly load variables.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine AHSIZE is called by: SIMSYS

and calls: COIL1

FNCLDS

PSYWTR

[18] D. C. Hittle and B. Sliwinski, *CERL Thermal Loads Analysis and Systems Simulation Program*, Volumes 1 and 2, Interim Report E-81 (CERL, 1975).

BOUND (TDBE,WE,TDBL,HL,TWE,TWL,VA,VW,WFA,QT,AOS,ISN,PB)

a. GENERAL DESCRIPTION

Subroutine BOUND computes the cooling coil operating conditions and the coil heat transfer area. Variable ISN indicates under which condition the coil is being operated: when ISN = 0, the coil is totally dry; when ISN = 1, the coil is partially wet and partially dry; when ISN = 2, the coil is completely wet. BOUND also computes the air and water conditions at this operating boundary. The equations used in this subroutine are derived from the equations given in *ASHRAE Handbook, 1975 Equipment*.[19]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine BOUND is called by: COIL1

and calls: PSYDPW

PSYHTW

PSYWTP

RESIS

THS2

[19] *ASHRAE Handbook, 1975 Equipment* (ASHRAE, 1975).

CCOIL

a. GENERAL DESCRIPTION

Subroutine CCOIL calculates the hourly cooling coil performance, which includes the total heat transfer rate and leaving-air humidity ratio. This subroutine requires the data of coil heat transfer area which was calculated in COIL1 and also the hourly air and water entering conditions, and the leaving-air dry-bulb temperature. The equations used in this subroutine are derived from the equations given in *ASHRAE Handbook, 1975 Equipment*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine CCOIL is called by: CVTRHS
MZDDS
VARVOL

and calls: PSYHTW
PSYWTH
PSYWTP
RESDRY

CCTEMP

a. GENERAL DESCRIPTION

Subroutine CCTEMP determines the desired cold deck, leaving-air dry-bulb temperature based on the user-supplied control strategy. CCTEMP can control the deck temperature by the following methods:

1. Fixed set point control
2. Inverse function of outside air (OA) temperature
3. Controlled by zone requiring the coldest air
4. Set to a particular zone's supply air temperature

Regardless of the control strategy specified, the actual deck temperature computed by CCTEMP also includes the effect of the controller throttling range on the final temperature. This effect is accounted for in accordance with the procedure for throttling ranges given in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations.*[20]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

---

[20] *Procedures for Simulating the Performance of Components and Systems for Energy Calculations* (ASHRAE, 1974).

c. TRACE BACK

Subroutine CCTEMP is called by: CVTRHS

FOURPIP

MZDDS

TWOPIPE

VARVOL

and calls: TDIRCL

TINVCL

COIL1

a. GENERAL DESCRIPTION

Subroutine COIL1 calculates the equivalent heat transfer area of the cooling coil from the set of given entering and leaving air and water conditions. This coil heat transfer area needs to be calculated only once in the annual energy computation program. The calculated heat transfer area is then input to CCOIL for calculating the hourly coil performance. The equations used in this subroutine are derived from *ASHRAE Handbook, 1975 Equipment.*

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine COIL1 is called by: AHSIZE

and calls: BOUND

PSYDPW

PSYHTW

PSYWTP

CVTRHS

a. GENERAL DESCRIPTION

CVTRHS is the driving program for the constant volume terminal reheat and the subzone reheat simulation. CVTRHS consists of calls to component and controller models in the order necessary for the simulation of the above systems. The procedures used in the simulations are outlined in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations.*

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine CVTRS is called by: SIMAHS

and calls: RHTCL

HUMID

CCTEMP

CCOIL

MIXAIR

SRAC

IZNCAL

DMAT

a. GENERAL DESCRIPTION

DMAT determines the desired mixed-air temperature for the system. The temperature can either be a fixed set point or be set to the desired cold deck temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine DMAT is called by: ETECY

TECCY1

TECCY2

and calls: No subroutines

EQPON (ONDAY, ONMON, OFFDAY, OFFMON, DAY, MONTH, FLAG)

a. GENERAL DESCRIPTION

EQPON determines if a component is operating for the current month and day based on the user-supplied schedule for that component.

b. DATA DESCRIPTIONS

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine EQPON is called by: RDZLI

and calls: No subroutines

ETECY

a. GENERAL DESCRIPTION

Subroutine ETECY simulates the enthalpy economy cycle for outdoor air introduction. ETECY first determines the desired mixed-air temperature followed by the outdoor air and return air enthalpies. If the outdoor air enthalpy is less than the return air enthalpy, the temperature economy cycle is allowed to operate (see TECCY2). Otherwise, the amount of outdoor air to be introduced is set to the user-specified minimum.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ETECY is called by: MIXAIR

and calls: TECCY2

PSYHTW

DMAT

FANCOIL

a. GENERAL DESCRIPTION

Subroutine FANCOIL is the driving program for the two-pipe fan coil and four-pipe fan coil simulations. The routine calls component models in the order necessary for the particular fan coil simulation. The basic strategy for the fan coil simulation can be found in CERL Interim Report E-81.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FANCOIL is called by: SIMAHS

and calls: IZNCAL

TWOPIPE

FOURPIP

FNCLDS

a. GENERAL DESCRIPTION

Subroutine FNCLDS calculates the number of rows of fan coil. The input data include both the entering and leaving air and water conditions. Based on one set of the input data, a ratio of total to sensible heat transfer rate is calculated and then compared to that calculated by QRAT to determine the number of rows of coil. This subroutine needs to be calculated only once in the annual energy computation program if the fan coil system has been specified. The calculated fan coil row number then is passed to FNCLHR for calculating the hourly fan coil performance. The equations used in this subroutine are based on the energy balance on both the water and air sides.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FNCLDS is called by: AHSIZE

and calls: QRAT

PSYDPT

FNCLHR

a. GENERAL DESCRIPTION

Subroutine FNCLHR calculates the fan coil hourly heat transfer rate and the hourly leaving-air humidity ratio. The input data include the number of rows of the fan coil calculated in FNCLDS as well as the hourly entering-air conditions and hourly sensible heat transfer rate. The equations used in this subroutine are based on the air side of the energy balance.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FNCLHR is called by: FOURPIP

TWOPIPE

and calls: PSYDPW

PSYHTW

PSYWTH

QRAT

FNPTLD (VVTYPE, FRAC)

a. GENERAL DESCRIPTION

Function FNPTLD calculates the part-load energy requirements of fans for variable volume systems. The procedure used is outlined in CERL Interim Report E-81.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function FNPTLD is called by: VVIZNC

and calls: No subroutines

FOURPIP

a. GENERAL DESCRIPTION

Subroutine FOURPIP simulates a four-pipe fan coil unit. FOURPIP determines the hot and cold water temperature supplied to the coils for the current hour, decides whether heating or cooling is required in the zone, and then calculates the coil loads and the unmet loads for the unit.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FOURPIP is called by: FANCOIL

and calls: FNCLHR

HCTEMP

CCTEMP

FZNCAL

a. GENERAL DESCRIPTION

Subroutine FZNCAL computes the final room humidity ratio for each zone on the system for the current simulation hour. The routine assumes that the initial humidity ratio for the hour is the final humidity ratio computed for the previous hour for that zone. It then computes the new final humidity ratio by adjusting the initial humidity ratio to account for the moisture changes in that hour caused by the supply air, the latent load in the zone, and the infiltration air.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FZNCAL is called by: SIMAHS

and calls: No subroutines

GNTEPS

a. GENERAL DESCRIPTION

Subroutine GNTEPS generates the air handler load file. GNTEPS stores on the file the information needed for a central plant simulation, as well as that needed to add or replace systems and for reporting purposes.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine GNTEPS is called by: SIMAHS

and calls: LPYRF

HCDAF

a. GENERAL DESCRIPTION

HCDAF models the mixing boxes for the multizone and three-deck multizone simulations. HCDAF computes the hot and cold deck mass flows for each zone on the system and for the total system. HCDAF also computes the unmet zone loads for the multizone system. The methods used in this routine are outlined in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine HCDAF is called by: MZDDS

and calls: No subroutines

HCOIL

a. GENERAL DESCRIPTION

HCOIL models hot water and steam heating coils. HCOIL determines the energy demanded by each coil by an energy balance on the air side only. The model assumes that the coil essentially has infinite capacity. HCOIL also calculates the load that the heating coil does not meet because it is off.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine HCOIL is called by: MZDDS

and calls: No subroutines

HCTEMP

a. GENERAL DESCRIPTION

Subroutine HCTEMP determines the desired hot deck, leaving-air dry-bulb temperature based on the user-supplied control strategy. HCTEMP can control the deck temperature by the following methods:

1. Fixed set point control
2. Inverse function of OA temperature
3. Controlled by zone requiring the warmest air
4. Set to a particular zone's supply air temperature

Regardless of the control strategy, the actual deck temperature computed by HCTEMP also includes the effect of the controller throttling range on the final temperature. This effect is accounted for according to the procedures for throttling ranges given in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine HCTEMP is called by: FOURPIP

MZDDS

TWOPIPE

and calls: TDIRCL

TINVCL

HUMID

a. GENERAL DESCRIPTION

HUMID models the humidifiers used in the simulated systems. HUMID first determines if the humidifier can operate. If the humidifier is on, HUMID calculates the amount of water and energy required to keep the controlled zone at the specified relative humidity. CERL Interim Report E-81 provides details.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine HUMID is called by: CVTRHS
MZDDS
VARVOL
and calls: PSYWTR

IZNCAL

a. GENERAL DESCRIPTION

IZNCAL does the initial zone calculations required by each hourly simulation. It determines the return air mass flow rate, the return air dry-bulb temperature, and the return air humidity ratio for each zone. IZNCAL also determines the necessary supply air temperature to meet the sensible load in each zone. The procedures used in this routine are found in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations.*

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine IZNCAL is called by: CVTRHS
FANCOIL
MZDDS
UNVENT

and calls: ZNMRQ

MAGIC

a. GENERAL DESCRIPTION

Subroutine MAGIC checks and processes the user-supplied input data. It checks the magnitudes of several key input parameters and can terminate the simulation if unrecoverable errors are found. MAGIC also supplies all the system-dependent defaults for parameters which the user has not specified.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine MAGIC is called by: RPINPT

and calls: No subroutines

MIXAIR

a. GENERAL DESCRIPTION

MIXAIR simulates the preheat coil and air mixing box. It will allow the preheat coil to be in the outside air duct or the mixed-air duct. MIXAIR can control the amount of outside air introduced by the system in the following five ways:

1. Fixed volume of outside air
2. Fixed percent outside air
3. Return air economy cycle
4. Temperature economy cycle
5. Enthalpy economy cycle

MIXAIR can simulate both blow-through and draw-through air handlers. The procedures used in this routine are described in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine MIXAIR is called by: CVTRHS
MZDDS
UNVENT
VARVOL

and calls: PREHT
ETECY
TECCY1
TECCY2

MZDDS

a. GENERAL DESCRIPTION

MZDDS is the driving program for the dual duct, multizone, and three-deck multizone simulations. MZDDS consists of calls to component and control models in the prescribed order for the multizone simulation. The procedures followed in the simulation are outlined in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations.*

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine MZDDS is called by: SIMAHS

and calls: HUMID
HCOIL
CCOIL
HCDAF
MIXAIR
HCTEMP
CCTEMP
SRAC
IZNCAL

PREHT

a. GENERAL DESCRIPTION

Subroutine PREHT determines the energy required by the preheat coil when located in the outside air duct or the mixed air duct.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine PREHT is called by: MIXAIR

and calls: No subroutines

QRAT (NR, VFRA, TDPTW, TDTW, VFRW)

a. GENERAL DESCRIPTION

Function QRAT calculates the ratio of fan coil total to sensible heat transfer rate. The input data include both entering dewpoint and water dry-bulb and water temperature differences as well as the number of rows of the fan coil, and air and water volume flow rates. The equations used in this subroutine are obtained by least square curve fitting. The data points for the curve fitting are taken from the *Carrier Fan Coil Catalog*.[21]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function QRAT is called by: FNCLDS

FNCLHR

and calls: No subroutines

[21] *Carrier Fan Coil Catalog* (Carrier Corp).

RCDSM1 (TOTLSM, PEAKVL, HOURSM, HRDATA)

a. GENERAL DESCRIPTION

RCDSM1 sums an hourly load variable into a monthly load array, and determines the peak hourly value for the month and the number of hours the variable has a nonzero value for the month.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RCDSM1 is called by: RCKEEP

and calls: No subroutines

RCDSM2 (QTOTAL, QPEAK, QHOURS, CEPEAK, CEHOUR, CAPCTY, HRVAL)

a. GENERAL DESCRIPTION

RCDSM2 sums an hourly load variable into a monthly load array, and determines the peak hourly value for the month and the number of hours the variable has a nonzero value for the month. RCDSM2 also determines the total amount the load exceeded the design capacity for the month, the peak amount the design capacity was exceeded, and the number of hours the capacity was exceeded.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RCDSM2 is called by: RCKEEP

and calls: No subroutines

RCKEEP

a. GENERAL DESCRIPTION

Subroutine RCKEEP performs all the record keeping for the program. The routine first sorts the system loads according to energy source required and then determines the total hourly demand for each energy type on the central plant. RCKEEP also stores the system loads, the system loads not met, and the energy requirements in the form of monthly and annual summaries for future reporting.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RCKEEP is called by: SIMAHS

and calls: RCDSM1

RCDSM2

RDZL1

a. GENERAL DESCRIPTION

RDZLI obtains the hourly information needed to simulate a system for the current day and environment. First the hourly weather information is obtained from the building load file for the desired day. Then the hourly load variables for each zone on the system are obtained for the desired day from the building load file. Finally, RDZLI determines the operation of the preheat, heating, and cooling coils using information obtained in the preceding steps and the user-supplied equipment schedules.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RDZLI is called by: SIMSYS

and calls: EQPON

NYDY

LPYRF

JDAYF

GTLFZB

FINDNO

ERROR

INSERT

RESDRY (VA, RAD, RMD)

a. GENERAL DESCRIPTION

Subroutine RESDRY calculates the cooling coil air side and metal thermal resistances with a dry coil surface using the given air velocity. The equations used in this subroutine are obtained by curve fitting the figures given by ARI Standard 410-72.[22]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RESDRY is called by: CCOIL
RESIS

and calls: No subroutines

[22] ARI Standard 410-72 (Air Conditioning and Refrigeration Institute, 1972).

RESIS (TWE, TWL, TDPE, VA, VW, PB)

a. GENERAL DESCRIPTION

Subroutines RESIS calculates the cooling coil thermal resistances. The input data include the entering dewpoint temperature, entering and leaving water temperatures, velocities of air and water, and barometric pressure. The equations used in this subroutine are obtained by curve fitting the figures given by ARI Standard 410-72.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RESIS is called by: BOUND

and calls: RESDRY

RHTCL

a. GENERAL DESCRIPTION

Subroutine RHTCL models zone reheat coils. It determines the reheat coil load and the reheat load not met for each zone on the system. RHTCL also determines the unmet recooling loads for the zone, if any. The procedures used can be found in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RHTCL is called by: CVTRHS

UNVENT

VARVOL

and calls: No subroutines

RPINPT

a. GENERAL DESCRIPTION

Subroutine RPINPT is a driver for obtaining the user-supplied input data necessary to simulate a system. RPINPT calls a subroutine which obtains the user-supplied data and then calls a routine which processes the input into the form needed for the simulation. b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions. c. TRACE BACK

Subroutine RPINPT is called by: SIMSYS

and calls: GTSYIN

MAGIC

RPTCLD

a. GENERAL DESCRIPTION

Subroutine RPTCLD generates the monthly equipment load report. This routine first converts the data into the desired output units and then writes out the information.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RPTCLD is called by: RPTGEN

and calls: WRITRL

WRITAL

RPTEGY

a. GENERAL DESCRIPTION

Subroutine RPTEGY generates the monthly energy report. The routine first converts the data to the proper units and then writes out the information.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RPTEGY is called by: RPTGEN

and calls: WRITRL

WRITAL

RPTGEN

a. GENERAL DESCRIPTION

RPTGEN is the driving program for report generation. For each system simulated for the current environment, RPTGEN sets up the conversion factor arrays for converting results from basic SI units to the user-specified units desired for the reports, and then calls report writers to generate the user-desired reports.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RPTGEN is called by: SIMSYS

and calls: RPTLNM

RPTCLD

RPTEGY

RPTSYS

SETPOS

RPTLNM

a. GENERAL DESCRIPTION

Subroutine RPTLNM generates the report of monthly loads not met. This routine first converts the data into the proper units and then writes out the information.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RPTLNM is called by: RPTGEN

and calls: WRITRL

WRITAL

RPTSYS

a. GENERAL DESCRIPTION

Subroutine RPTSYS generates the system-description report. The routine converts the data into the proper output units and then writes out the information.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine RPTSYS is called by: RPTGEN

and calls: No subroutines

SETPOS

a. GENERAL DESCRIPTION

Subroutine SETPOS determines the print positions of the months in the monthly reports.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SETPOS is called by: RPTGEN

and calls: No subroutines

SIMAHS

a. GENERAL DESCRIPTION

SIMAHS is the driver program for performing the daily system simulation and generating the daily air handler file record for the system. The simulation is performed by executing hourly calculations inside a 24-hour loop. Within the loop, SIMAHS determines whether the system is on for the hour, simulates the desired system for that hour if necessary, determines final conditions for each zone on the system, and does record keeping on the simulation results.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SIMAHS is called by: SIMSYS

and calls: GNTEPS

RCKEEP

FZNCAL

FANCOIL

UNVENT

VARVOL

CVTRHS

MZDDS

AHOFF

SRAC

a. GENERAL DESCRIPTION

Subroutine SRAC determines the return air dry-bulb temperature and humidity ratio for the system based on the individual return air temperature, humidity ratio, and mass flow rate from each zone. The methods used are described in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations.*

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SRAC is called by: CVTRHS
MZDDS
UNVENT
VARVOL

and calls: No subroutines

TDIRCL (CT, DATHI, DATLO, CTHI, CTLO)

a. GENERAL DESCRIPTION

Function TDIRCL simulates the operation of a direct acting temperature controller based on the user-supplied data for DATHI, DATLO, CTHI, CTLO, and the current value of the controlling variable (CT). The controller operation is as follows:

1. If the controlling variable (CT) is greater than CTHI, the controlled variable (TDIRCL) is set to DATHI.

2. If CT is less than CTLO, TDIRCL is set to DATLO.

3. If CT is between CTHI and CTLO, proportional control is executed to determine TDIRCL.

c. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function TDIRCL is called by: HCTEMP

CCTEMP

and calls: No subroutines

TECCY1

a. GENERAL DESCRIPTION

TECCY1 simulates the return air economy cycle. If the outside air temperature is less than the desired mixed-air temperature, TECCY1 introduces the amount of outside air which will bring the mixed air as close as possible to the desired temperature or the minimum amount of outside air as specified by the user, whichever is larger. If the outside air temperature is greater than the desired mixed-air temperature, the user-specified minimum amount of outside air is introduced.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine TECCY1 is called by: MIXAIR

and calls: DMAT

TECCY2

a. GENERAL DESCRIPTION

TECCY2 simulates the temperature economy cycle for ventilation air. Based on the outside air temperature and the return air temperature, TECCY2 introduces the amount of outside air necessary to bring the mixed air as close as possible to the desired temperature or the minimum amount, whichever is larger.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine TECCY2 is called by: MIXAIR
ETECY
and calls: DMAT

THS2 (CH, HL, TWE, TS2, HS2, PB)

a. GENERAL DESCRIPTION

Subroutine THS2 calculates the coil surface temperature and enthalpy. The input data include the coil characteristic, entering-water temperature, leaving air enthalpy, and barometric pressure. The iterative method used in this subroutine is to substitute the figure (Figure 9) shown in ARI Standard 410-72.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine THS2 is called by: BOUND

and calls: PSYHTW

PSYWTP

SATUTH

TINVCL (CT, DATHI, DATLO, CTHI, CTLO)

a. GENERAL DESCRIPTION

Function TINVCL simulates the operation of an inverse acting temperature controller based on the user-supplied data for DATHI, DATLO, CTHI, CTLO, and the current value of the controlling variable (CT). The controller operation is as follows:

1. If the controlling variable (CT) is greater than CTHI, the controlled variable (TINVCL) is set to DATLO.

2. If CT is less than CTLO, TINVCL is set to DATHI.

3. If CT is between CTHI and CTLO, proportional control is executed to determine TDIRCL.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function TINVCL is called by: HCTEMP

CCTEMP

and calls: No subroutines

TWOPIPE

a. GENERAL DESCRIPTION

Subroutine TWOPIPE determines the hourly load on a two-pipe fan coil unit. TWOPIPE first determines if hot or cold water is supplied to the coil and then calculates the coil heating or cooling load and the unmet heating or cooling load for that unit.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine TWOPIPE is called by: FANCOIL

and calls: FNCLHR

HCTEMP

CCTEMP

ERROR2

UNVENT

a. GENERAL DESCRIPTION

Subroutine UNVENT is the driving program for the unit ventilation and unit heater simulations. The routine consists of calls to component models and control strategies in the order necessary for the simulation. The procedures used in this simulation closely parallel those given in CERL Interim Report E-81.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine UNVENT is called by: SIMAHS

and calls: RHTCL

MIXAIR

SRAC

IZNCAL

VARVOL

a. GENERAL DESCRIPTION

Subroutine VARVOL is the driving program for the variable volume simulation. VARVOL consists of calls to component and controller models in the order prescribed for the variable volume simulation. The procedures followed in this subroutine are outlined in the ASHRAE *Procedures for Simulating the Performance of Components and Systems for Energy Calculations.*

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine VARVOL is called by: SIMAHS

and calls: RHTCL

HUMID

CCOIL

MIXAIR

SRAC

VVIZNC

CCTEMP

VVIZNC

a. GENERAL DESCRIPTION

Subroutine VVIZNC performs the initial hourly calculations required for each zone on a variable volume system. VVIZNC first determines the supply air mass flow rate required in each zone and the return air mass flow rate from each zone. The zone's desired supply air temperature, return air temperature, and return air humidity ratio are also calculated. Finally, VVIZNC determines the system air mass flow rates and the part-load performance of the variable volume fans.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine VVIZNC is called by: VARVOL

and calls: FNPTLD

ZNMRQ

WRITAL

a. GENERAL DESCRIPTION

WRITAL writes the alpha information contained in the monthly column headings for the monthly reports.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine WRITAL is called by: RPTCLD

RPTEGY

RPTLNM

and calls: No subroutines

WRITRL (VAR, VARNM, VARUNT)

a. GENERAL DESCRIPTION

WRITRL writes out the rows of information in the monthly reports in the diagonalized format.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine WRITRL is called by: RPTCLD

RPTEGY

RPTLNM

and calls: No subroutines

ZNMRQ

a. GENERAL DESCRIPTION

Subroutine ZNMRQ determines the zone return air humidity ratio based on the previous hour's zone humidity ratio and the current hour's zone latent load.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine ZNMRQ is called by: AHOFF

IZNCAL

VVIZNC

and calls: PSYWTR

11 BLAST AIR HANDLER LOAD FILE

The air handler load file (AHLDFL) is used to pass system heating/cooling load information and weather information between SIMSYS and SIMTEP. AHLDFL allows up to five environments (i.e., design days and/or weather runs), which cannot be changed after initial setup, and up to 60 systems per environment, which can be added or replaced from run to run.

The air handler load file is a random access mass store file. Figure 15 shows details of the file's internal structure.

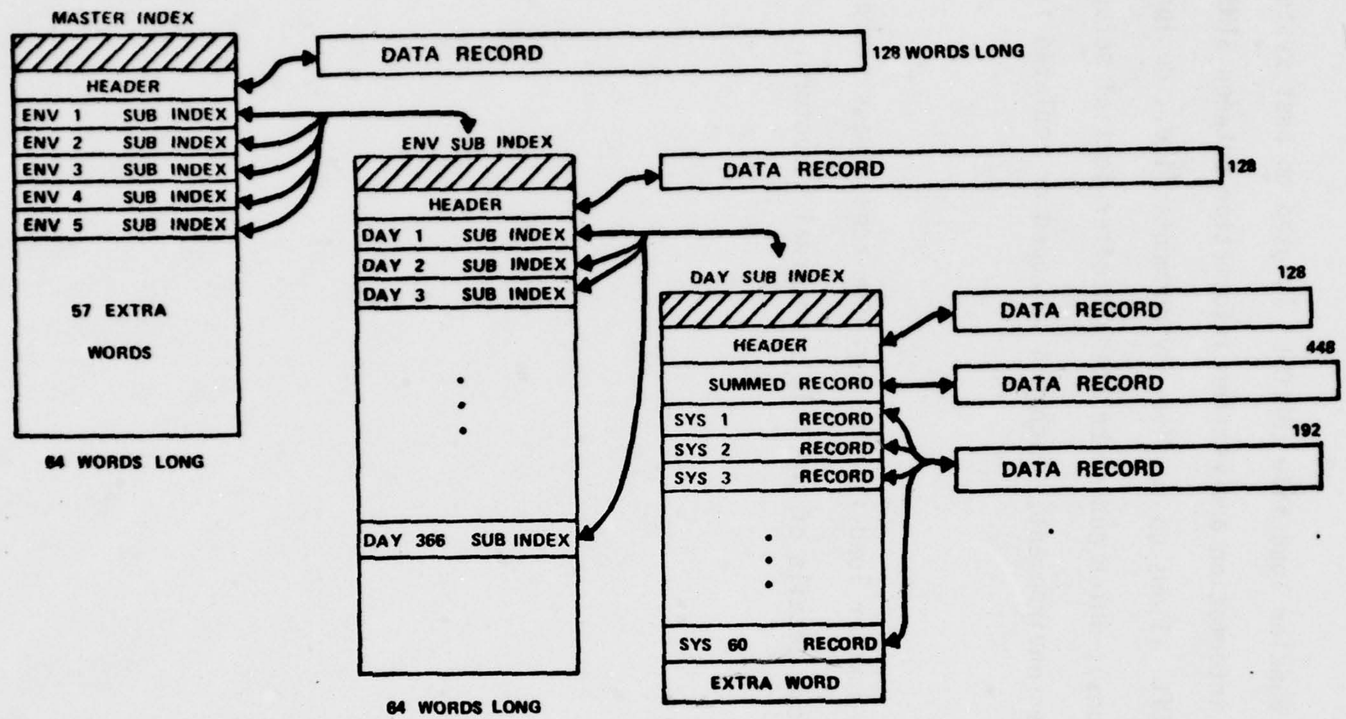


Figure 15. File structure for AHLDFL.

12 SIMTEP (CENTRAL PLANT SIMULATION)

This chapter describes the routines used to simulate central plant systems and components. SIMTEP is described first, followed by the subroutines (in alphabetical order). Structured algorithm charts are presented where appropriate. Figure 16 shows the overall flow of the SIMTEP subsystem.

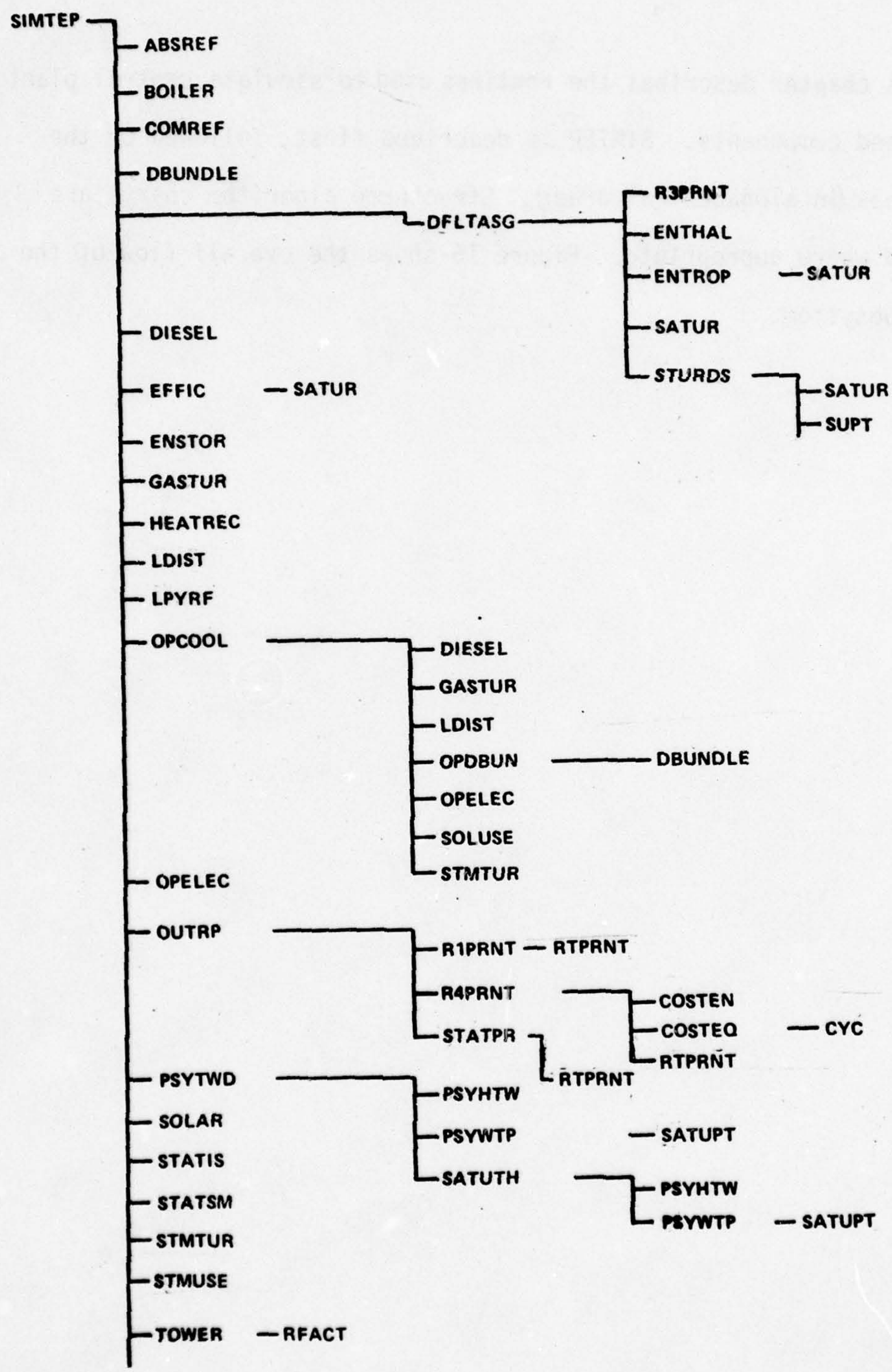


Figure 16. Tree structure of SIMTEP subroutines.

SIMTEP

a. GENERAL DESCRIPTION

SIMTEP is a subprogram for the simulation of the energy plant. The first set of data relating to energy load is read from TAPE 20. The rest of the data are supplied by the input processor. These data consist of the equipment size parameters, the equipment load ratios, the cost reference parameters, the life-cycle cost parameters, and special variables.

The program performs a series of equipment simulations and generates output-related parameters. The equipment includes:

Equipment Index	Equipment
1	Gas Turbine
2	Diesel Engine
4	Steam Boiler
5	One-stage Absorption Chiller
6	Two-stage Absorption Chiller Without Economizer
7	Two-stage Absorption Chiller With Economizer

Equipment Index	Equipment
9	Hermetic Compression Chiller
10	Open Centrifugal Chiller
11	Reciprocating Chiller
13	Double-bundle Chiller
14	Traditional Cooling Tower
15	Ceramic Cooling Tower
16	Solar Panel
17	Steam Turbine
18	Hot Water Tank
19	Cold Water Tank

The output results consist of cost reference for equipment, cost of utility and energy, central plant utilization summary, and equipment use statistics.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
TAPE 20	ENGYLD(I,J,K)	Energy load; K is the index for day, J is the index for hour; = EHEAT, the required heating (kW), when I = 1 = ECOOL, the required cooling (kW), when I = 2 = EELEC, the electric energy input (kW), when I = 3

Source of Data	Name	Description
		= TAIR, the dry-bulb temperature of air (°C), when I = 4
		= HR, the humidity ratio, when I = 5
		= EHWDOM, the amount of energy required to heat domestic hot water (kW), when I = 6
		= ESTUSE, the steam energy load of steam user (kW), when I = 7
		= RWTR, the ratio of return water to steam flow, when I = 8
DATE	IMON	Index denoting month
DATE	IDAY	Index denoting day
DATE	IHR	Index denoting hour
EDATA	NEQSIZE(I)	Number of different sizes of equipment type I
EDATA	ROPT(I)	Optimum part load ratio of equipment type I
EDATA	RMIN(I)	Minimum part load ratio of equipment type I
EDATA	RMAX(I)	Maximum part load ratio of equipment type I
EDATA	PEL(I)	Electrical input to nominal capacity ratio.

Source of Data	Name	Description
EDATA	CNOM(J,I)	Nominal size of equipment type I, size index J (kW)
EDATA	KINS(J,I)	Number of equipment installed with type I, size J
EDATA	KAV(J,I)	Number available for equipment type I, size J
EDATA	IENAME(J,I)	Equipment name (30H FORMAT)
EDATA	NEDATA	Range of equipment with size I as specified
ECDATA	EQCOSD(K,J,I)	Cost parameters for equipment of generic type I, size category J; variable represents: Size (kW), when K = 1 Unit cost ($), when K = 2 Installed cost factor, when K = 3 Consumable ($/hr), when K = 4 Maintenance (hrs/yr), when K = 5 Equipment life (hrs), when K = 6 Hours to minor overhaul (hrs), when K = 7 Minor overhaul cost ($), when K = 8 Hours to major overhaul (hrs), when K = 9 Major overhaul cost ($), when K = 10

Source of Data	Name	Description
ECDATA	EQCOSR(K,I)	Cost reference parameters for equipment of generic type I, with index K identical to those in EQCOSD
LFCYCD	ALFCYC(I)	Life-cycle parameters:
		Interest rate for I = 1
		Labor inflation rate for I = 2
		Material inflation rate for I = 3
		Energy inflation rate for I = 4
		Project life (yr) for I = 5
		Labor cost ($/hr) for I = 6
		Site cost factor for I = 7
SDATA(1,1)	HSTEAM	Steam enthalpy (kWh/kg)
SDATA(1,2)	TSATUR	Saturation temperature (°C)
SDATA(1,3)	RFLASH	Boiler flash water/steam feed
SDATA(1,4)	PELCL	Electric input to circulation pump/cooling load
SDATA(1,5)	PELHT	Electric input to circulation pump/heating load
SDATA(1,6)	PELTWR	Electric input to cooling tower/tower cooling load
SDATA(1,7)	TOWOPR	Tower operation type
SDATA(1,9)	TWMAKE	Makeup water temperature (°C)
SDATA(1,10)	TCOOL	Chilled water temperature (°C)
SDATA(1,11)	DTCOOL	Chilled water temperature rise (°C)
SDATA(1,12)	TTOWR	Entering tower water temperature (°C)

Source of Data	Name	Description
SDATA(1,13)	TCW	Leaving condenser water temperature (°C)
SDATA(1,14)	TMINH	Minimum tank temperature for heating (°C)
SDATA(1,15)	TMINC	Minimum tank temperature for cooling (°C)
SDATA(1,16)	CPTYPE	Plant type 1 = utility only 2 = mixed plant
SDATA(1,17)	TLEAVE	Boiler stack leaving temperature (°C)
SDATA(1,18)	SR2A	Full load steam rate (kg/sec) (two-stage absorption chiller)
SDATA(1,19)	SR1A	Full load steam rate (kg/sec) (one-stage absorption chiller)
SDATA(1,20)	RAVRHDB	Available recoverable heat ratio
SDATA(1,22)	RMXKWD	Maximum exhaust flow/kW input (diesel)
SDATA(1,24)	RMXKWG	Maximum exhaust flow/kW input (gas)
SDATA(1,25)	RMCA	Tower water/absorption chiller capacity
SDATA(1,26)	RWCC	Tower water/compression chiller capacity
SDATA(1,27)	RWCDB	Tower water/double-bundle chiller capacity
SDATA(1,28)	SRATB	Air, fuel stoichiometric ratio
SDATA(1,29)	HFUELB	Heat content of fuel (kW/kg)
SDATA(1,30)	RHFLASH	Recovered heat/flash steam energy
SDATA(1,31)	PSTEAM	Steam pressure (Pa)
SDATA(1,32)	PSTMTUR	Entering steam pressure (Pa)
SDATA(1,33)	TSTMTUR	Entering steam temperature (°C)
SDATA(1,34)	PEXSTUR	Nominal exhaust steam pressure (Pa)
SDATA(1,35)	RPMNOM	Nominal speed (rad/sec)
SDATA(1,36)	RWSTUR	Condensate/entering steam

Source of Data	Name	Description
SDATA(1,37)	TOTUEF	Total efficiency of utility electric generation
SDATA(1,38)	TILT	Collector tilt from horizontal (degree)
SDATA(1,39)	AZMUTH	Collector azimuthal angle (degree)
SDATA(1,42)	FLH2O	Mass flow rate/unit area through collector (kg/sec-m^2)
SDATA(1,43)	HXEFF	Heat exchanger effectiveness
PDATA(1,1)*	CAVL1A	Available capacity (one-stage absorption chiller)
PDATA(1,2)	CAVL2A	Available capacity (two-stage absorption chiller)
PDATA(1,3)	REN1A	Energy input-output (I/O) coefficients (one-stage absorption chiller)
PDATA(1,4)	REN2A	Energy I/O coefficients (two-stage absorption chiller)
PDATA(1,5)	REN2AE	Energy I/O coefficients (two-stage absorption chiller with economizer)
PDATA(1,6)	TCON1A	Condensate temperature coefficient (one-stage absorption chiller)
PDATA(1,7)	RPWR1C	Energy I/O coefficient (hermetic compression chiller)
PDATA(1,8)	RPWR2C	Energy I/O coefficients (open centrifugal compression chiller)

*PPDATA(1,I); (2,N); and (3,N) are the three coefficients of the quadratic polynomial function whose name appears in PDATA(4,N).

Source of Data	Name	Description
PDATA(1,9)	RPWR3C	Energy I/O coefficients (reciprocating compression chiller)
PDATA(1,10)	RCAVDB	Available capacity ratio (double-bundle chiller)
PDATA(1,11)	RPWRDB	Energy I/O coefficients (double-bundle chiller)
PDATA(1,12)	ADJTDB	Condensate cooling water temperature adjustment factor (double-bundle chiller)
PDATA(1,13)	ADJEDB	Energy ratio adjustment factor (double-bundle chiller)
PDATA(1,14)	RELD	Power out/fuel input coefficients (diesel)
PDATA(1,15)	RJACK	Jack heat/fuel input coefficients (diesel)
PDATA(1,16)	RLUBD	Lube heat/fuel input coefficients (diesel)
PDATA(1,17)	REXD	Exhaust heat/fuel input coefficients (diesel)
PDATA(1,18)	TEXD	Exhaust temperature coefficients (diesel)
PDATA(1,19)	FUEL1G	Fuel I/O coefficients 1-3 (gas turbine)
PDATA(1,20)	FUEL2G	Fuel I/O coefficients 4-6 (gas turbine)
PDATA(1,22)	FEXG	Exhaust flow coefficients (gas turbine)
PDATA(1,23)	TEX1G	Exhaust temperature coefficients 1-3 (gas turbine)
PDATA(1,24)	TEX2G	Exhaust temperature coefficients 4-6 (gas turbine)
PDATA(1,25)	FLUBG	Lube oil coefficients (gas turbine)

Source of Data	Name	Description
PDATA(1,26)	RF1	Rating factor temperature coefficients 1-3 (cooling tower)
PDATA(1,27)	RF2	Rating factor temperature coefficients 4-6 (cooling tower)
PDATA(1,28)	RF3	Rating factor temperature coefficients 7-9 (cooling tower)
PDATA(1,29)	RF4	Rating factor temperature coefficients 10-12 (cooling tower)
PDATA(1,30)	RF5	Rating factor temperature coefficients 13-15 (cooling tower)
PDATA(1,31)	RF6	Rating factor temperature coefficients 16-18 (cooling tower)
PDATA(1,32)	RFUELB	Energy I/O coefficients (steam boiler)
PDATA(1,33)	SR1DTA	Steam rate coefficients (one-stage absorption chiller)
PDATA(1,34)	SR2DTA	Steam rate coefficients (two-stage absorption chiller)
PDATA(1,35)	TCON2A	Condensate temperature coefficient (two-stage absorption chiller)
PDATA(1,36)	RFSTUR	Steam flow coefficients (steam turbine)
PDATA(1,37)	UACD	Stack U-factor* area coefficients (diesel)
PDATA(1,38)	UACG	Stack U-factor* area coefficients (gas turbine
PDATA(1,39)	RFR	Rating factor range coefficients (cooling tower)
PDATA(2,21)	FRUL	Slope of collector performance curve (W/m^2 - °C)

Source of Data	Name	Description
	Other data	Other data are described in the related subroutines.

2. COMMON BLOCKS

DATE, EDATA, EFFICD, EPARS, HOURTOT, MONTOT, SDATA, STM, STMTUR, TOWERD, WEATHR, PDATA, AFCYCD, ECDATA

3. OUTPUT DATA

Name	Description
PRNTA1(IMON,I)	Monthly output information; IMON represents month, IMON = 13 represents total sum in a year; variable represents: Total heat energy (kWh) for I = 1 Total electric energy (kWh) for I = 2 Cooling electric energy (kWh) for I = 3 Recovered energy (kWh) for I = 4 Wasted recoverable energy (kWh) for I = 5 Heat energy input for cooling (kWh) for I = 6 Electric energy input for cooling (kWh) for I = 7 Energy input for heating (kWh) for I = 8 Energy input for electricity (kWh) for I = 9 Total fuel input (kWh) for I = 10 Total energy input (kWh) for I = 11 Average plant efficiency for I = 12

Name	Description
IENAME(2-4,I)	Equipment name of type I (30H FORMAT)
AVGOPR	Average operation ratio
OPCAPY(I)	Operating capacity totaled over the year of equipment type I (GWh)
AMAXLD(I)	Maximum load of equipment type I (kWh)
MAXTIM(1,I)	Month of maximum part load for equipment type I
MAXTIM(2,I)	Day of maximum part load for equipment type I
MAXTIM(3,I)	Hour of maximum part load for equipment type I
CNOM(J,I)	Nominal size of equipment type I, size index J (kW)
IOPRHR(J,I)	Number of operation hours of equipment type I, size index J
KIN(J,I)	Number of equipment of type I, size index J that are installed
KAV(J,I)	Number of equipment of type I, size index J that are available
EQCHT(1,I)	Total first cost of equipment type I ($)
EQCHT(2,I)	Total annual cost of equipment type I ($)
EQCHT(3,I)	Total cyclical cost of equipment type I ($)
EQCHT(5,I)	Total cost of equipment type I ($)
EQCOST(1,J,I)	Total first cost of equipment type I, size index J ($)

Name	Description
EQCOST(2,J,I)	Total annual cost of equipment type I, size index J ($)
EQCOST(3,I)	Total cyclical cost of equipment type I, size index J ($)
EQCOST(5,I)	Total cost of equipment type I, size index J ($)
TOTECS	Equipment total cost ($)
IUNAM(IU)	Utility, energy name (6H FORMAT)
ENCOST(IU)	Cost of utility type IU ($)
ENUSE (13,IU)	Yearly energy used for utility type IU (MW).
ENPEAK(13,IU)	Yearly energy peak for utility type IU (kWh)
UDATA(3,IU)	Cost escalation factor for life cycle for utility type IU
TOTUCS	Total utility, energy cost ($)
TOTCOST	Total life-cycle cost ($)
IY	Life cycle in years (yr)

c. TRACE BACK

SIMTEP is called by: PERDS

and calls: ABSREF
BOILER
COMREF
DBUNDLE
DFLTASG
DIESEL

EFFIC

ENSTOR

GASTUR

HEATREC

LDIST

LPYRF

OPCOOL

OPELEC

OUTRP

SOLAR

STATIS

STATSM

STMTUR

STMUSE

TOWER

PSYTWD

SIMTEP

Initialize variables

Read input data

Initialize stored energy (ENSTOR)

Perform monthly simulations

Print requested report (OUTRP)

Repeat until END or STOP card is read

Monthly Simulations

Read data from energy load file
Initialize monthly sum variables
Perform hourly simulations
Calculate monthly results (STATSM)

Hourly Simulations

Define hour load variables
Calculate available solar heat and temperature (SOLAR)
Calculate the cooling and heating loads satisfied by the stored energy (ENSTOR)
Distribute cooling loads among chiller types (OPCOOL)
Distribute cooling loads among double-bundle chiller units (LDIST)
Simulate double-bundle chiller units (DBUNDLE)
Distribute loads among absorption chiller units (LDIST)
Simulate absorption chiller units (ABSREF)
Distribute loads among compression chiller units (LDIST)
Simulate compression chiller units (COMREF)
Calculate total cooling tower load and total electrical energy input required for cooling
Simulate cooling tower and calculate total electrical energy output (TOWER)
Distribute electrical energy loads among prime generator types (LDIST)
Simulate gas turbine units (GASTUR)
Store energy from waste heat (ENSTOR)
Calculate total heat energy output including energy stored
Distribute loads among boiler units (LDIST)
Simulate boiler units (BOILER)
Calculate total efficiencies and heat consumption (EFFIC)
Calculate monthly sums and maximums (STATIS)

ABSREF (ECOOL, FSTEAM, ESTEAM, EELEC, ETOWER, TCOND)

a. GENERAL DESCRIPTION

ABSREF is a subroutine which simulates operation of an absorption chiller. Three types of absorption chillers are included--the one-stage absorption chiller, the two-stage absorption chiller without economizer, and the two-stage absorption chiller with economizer.

Parameters relating to output variables are evaluated as follows:

Ratio of (available capacity)/(nominal capacity) = f(c,x)

where f = a quadratic polynomial in x, and c is a set of polynomial coefficients

x = entering water temperature - leaving water temperature (°C)

= TTOWR - TCOOL

c = CAVL1A, for one-stage absorption

= CAVL2A, for two-stage absorption

Part-load steam rate = A f(c,x)

where A = full-load steam rate (kg/kW)

f = as defined above

x = chilled water temperature rise (°C)

= DTCOOL

c = SR1DTA for one-stage absorption

= SR2DTA for two-stage absorption

Ratio of (energy input) (design energy input) = f(c,x)

where f = as defined above

x = part-load ratio

= (cooling load)/(available capacity)

c = REN1A for one-stage absorption

= REN2A for two-stage absorption without economizer

= REN2AE for two-stage absorption with economizer

Ratio of (condensate water temperature)/(saturation temperature)

= f(c,x)

where f = as defined above

x = part-load ratio

= (cooling load)/(available capacity)

c = TCON1A for one-stage absorption

= TCON2A for two-stage absorption

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP (PLOAD(IABSOR))	ECOOL	Total cooling energy (kW)
EPARS	IABSOR	Types of absorption chillers: 5 for one-stage absorption 6 for two-stage absorption without economizer 7 for two-stage absorption with economizer
SDATA(1,1)	HSTEAM	Steam enthalpy (kJ/kg)
SDATA(1,2)	TSATUR	Steam saturation temperature (°C)
SDATA(1,10)	TCOOL	Chilled water temperature (°C)
SDATA(1,11)	DTCOOL	Chilled water temperature rise (°C)

Source of Data	Name	Description
SDATA(1,12)	TTOWR	Entering tower water temperature (°C)
SDATA(1,19)	SR1A	Full-load steam rate for one-stage absorption chiller (kg/kW)
SDATA(1,18)	SR2A	Full-load steam rate for two-stage absorption chiller (kg/kW)
PDATA(1,1)	CAVL1A	Quadratic polynomial coefficients for one-stage absorption as described in previous section
PDATA(1,3)	REN1A	Quadratic polynomial coefficients for one-stage absorption as described in previous section
PDATA(1,6)	TCON1A	Quadratic polynomial coefficients for one-stage absorption as described in previous section
PDATA(1,33)	SR1DTA	Quadratic polynomial coefficients for one-stage absorption as described in previous section
PDATA(1,2)	CAVL2A	Quadratic polynomial coefficients for two-stage absorption as described in previous section
PDATA(1,4)	REN2A	Quadratic polynomial coefficients for two-stage absorption as described in previous section
PDATA(1,5)	REN2AE	Quadratic polynomial coefficients for two-stage absorption as described in previous section

Source of Data	Name	Description
PDATA(1,35)	TCON2A	Quadratic polynomial coefficients for two-stage absorption as described in previous section
PDATA(1,34)	SR2DTA	Quadratic polynomial coefficients for two-stage absorption as described in previous section
EDATA	RMIN(IABSOR)	Minimum part load ratio for absorption chiller type IABSOR
EDATA	PEL(IABSOR)	Electrical input to nominal capacity ratio for absorption chiller type IABSOR

2. COMMON BLOCKS

EDATA, EPARS, PDATA, SDATA

3. OUTPUT DATA

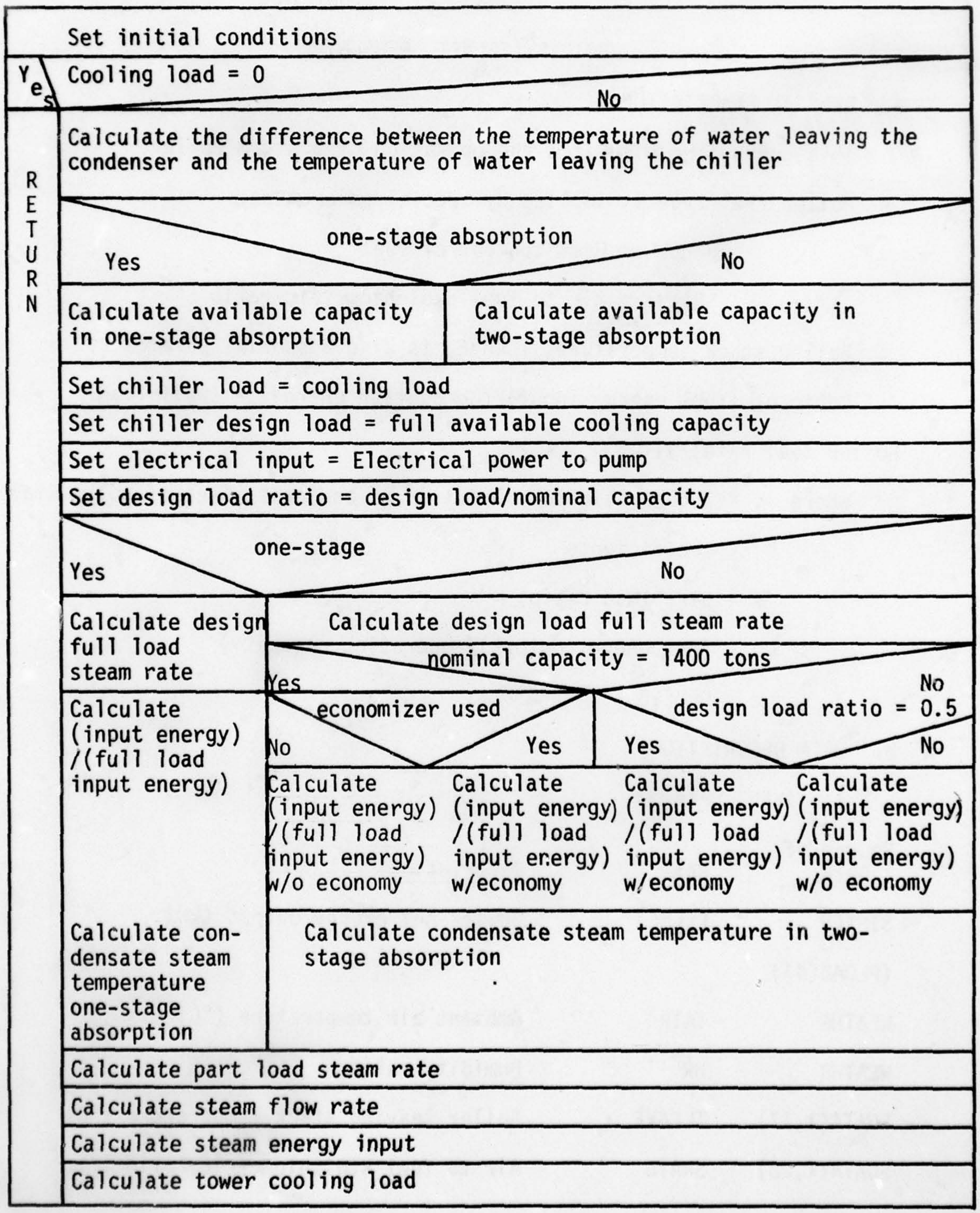
Name	Description
FSTEAM	Steam flow rate (kg/hr)
ESTEAM	Steam energy input (kW)
EELEC	Electrical energy input (kW)
ETOWR	Tower cooling load (kW)
TCOND	Condensate water temperature (°C)

c. TRACE BACK

Subroutine ABSREF is called by: SIMTEP

and calls: No subroutines

ABSREF



Set initial conditions
Yes
Cooling load = 0
No
RETURN
Calculate the difference between the temperature of water leaving the condenser and the temperature of water leaving the chiller
one-stage absorption
Yes
No
Calculate available capacity in one-stage absorption
Calculate available capacity in two-stage absorption
Set chiller load = cooling load
Set chiller design load = full available cooling capacity
Set electrical input = Electrical power to pump
Set design load ratio = design load/nominal capacity
one-stage
Yes
No
Calculate design full load steam rate
Calculate design load full steam rate
nominal capacity = 1400 tons
Yes
No
Calculate (input energy) /(full load input energy)
economizer used
No
Yes
design load ratio = 0.5
Yes
No
Calculate (input energy) /(full load input energy) w/o economy
Calculate (input energy) /(full load input energy) w/economy
Calculate (input energy) /(full load input energy) w/economy
Calculate (input energy) /(full load input energy) w/o economy
Calculate condensate steam temperature one-stage absorption
Calculate condensate steam temperature in two-stage absorption
Calculate part load steam rate
Calculate steam flow rate
Calculate steam energy input
Calculate tower cooling load

BOILER (EBLNET, EFUELB)

a. GENERAL DESCRIPTION

This subroutine simulates the operation of a steam boiler.

Boiler fuel type is implied by special parameters:

HFUELB = Heat content of fuel

SRATB = Air to fuel stoichiometric ratio

Boiler stack temperature, TLEAVE, is also a special parameter.

Ratio of (fuel energy input)/(combustion energy) = (Full-load boiler fuel rate)/f(c,x)

where f = a quadratic polynomial in x, and c is a set of polynomial coefficients

x = part-load ratio

= (net energy output)/(operating capacity)

c = RFUELB

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP (PLOAD(4))	EBLNET	Boiler net energy output (kW)
WEATHR	TAIR	Ambient air temperature (°C)
WEATHR	HR	Humidity ratio
SDATA(1,17)	TLEAVE	Boiler leaving stack temperature (°C)
SDATA(1,28)	SRATB	Air to fuel stoichiometric ratio

Source of Data	Name	Description
SDATA(1,29)	HFUELB	Heat content of fuel (kW/kg)
PDATA(1,32)	RFUELB	Quadratic polynomial coefficients for part-load ratio as described in previous section
EPARS	OPCAP(4)	Operating capacity for boiler
EDATA	RMIN(4)	Minimum part-load ratio for boiler

2. COMMON BLOCKS

EDATA, EPARS, PDATA, SDATA, WEATHR

3. OUTPUT DATA

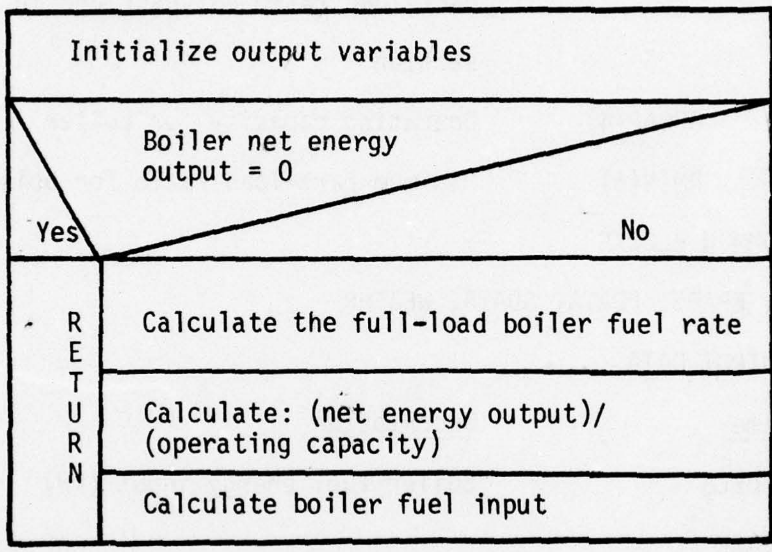
Name	Description
EFUELB	Boiler fuel energy input (kW)

c. TRACE BACK

Subroutine BOILER is called by: SIMTEP

and calls: No subroutines

BOILER


Initialize output variables
Boiler net energy output = 0
Yes
No
RETURN
Calculate the full-load boiler fuel rate
Calculate: (net energy output)/ (operating capacity)
Calculate boiler fuel input

COMREF (ECOOL, EELEC, ETOWER)

a. GENERAL DESCRIPTION

COMREF is a subroutine to simulate operation of a chiller. Three types of compression chillers are included--the hermetic centrifugal chiller, the open centrifugal chiller, and the reciprocating chiller.

Fraction of chiller electrical energy input

= (energy input)/(design energy input)

= f(c,x)

where f = a quadratic polynomial in x, and c is a set of polynomial coefficients

x = part-load ratio

= (required cooling)/(nominal cooling capacity

c = RPWR1C, for hermetic centrifugal chiller

= RPWR2C, for open centrifugal chiller

= RPWR3C, for reciprocating chiller

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP	ECOOL	Required cooling for compression chiller
(PLOAD(ICOMPR))		type ICOMPR
EPARS	ICOMPR	Compression chiller type:
		9, for hermetic centrifugal chiller
		10, for open centrifugal chiller
		11, for reciprocating centrifugal chiller

Source of Data	Name	Description
PDATA(1,7)	RPWR1C	Quadratic polynomial coefficients for hermetic centrifugal chiller as described in previous section
PDATA(1,8)	RPWR2C	Quadratic polynomial coefficients for open centrifugal chiller as described in previous section
PDATA(1,9)	RPWR3C	Quadratic polynomial coefficients for reciprocating chiller as described in previous section
EDATA	RMIN(ICOMPR)	Minimum part-load ratio of compression chiller type ICOMPR
EDATA	RMAX(ICOMPR)	Maximum part-load ratio of compression chiller type ICOMPR

2. COMMON BLOCKS

EDATA, EPARS, PDATA

3. OUTPUT DATA

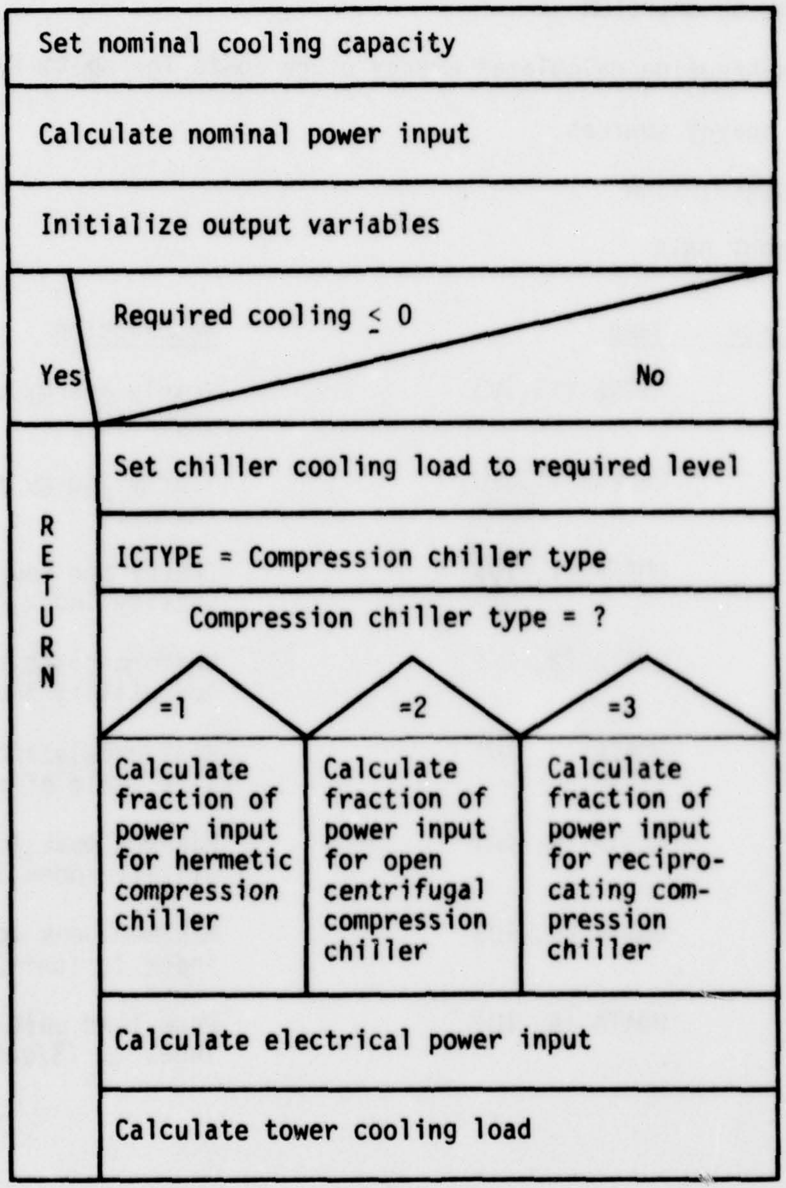
Name	Description
EELEC	Electrical energy input (kW)
ETOWER	Tower cooling load (kW)

c. TRACE BACK

Subroutine COMREF is called by: SIMTEP

and calls: *No subroutines*

COMREF


Set nominal cooling capacity
Calculate nominal power input
Initialize output variables
Required cooling ≤ 0
Yes
No
RETURN
Set chiller cooling load to required level
ICTYPE = Compression chiller type
Compression chiller type = ?
=1
=2
=3
Calculate fraction of power input for hermetic compression chiller
Calculate fraction of power input for open centrifugal compression chiller
Calculate fraction of power input for recipro- cating com- pression chiller
Calculate electrical power input
Calculate tower cooling load

COSTEN

a. GENERAL DESCRIPTION

This subroutine calculates energy usage costs for up to 10 different energy sources.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
STATD	ENUSE (13,IU)	Yearly energy used for utility index IU
STATD	ENPEAK (13,IU)	Yearly energy peak for utility index IU
STATD	UDATA (1, IU)	Energy per source unit for utility index IU (kW)
STATD	UDATA (2, IU)	Uniform costs per source unit for utility index IU ($)
STATD	UDATA (3, IU)	Cost escalation factor for life cycle of utility index IU
STATD	UDATA (4, IU)	Minimum peak load charge for utility index IU
STATD	UDATA (5, IU)	Minimum peak load for utility index IU (unit)
STATD	UDATA (6, IU)	Peak load unit cost for utility index IU ($/unit).

Source of Data	Name	Description
UCOSTD	UBLK	Two parameters specifying each block of a graduated change
UCOSTD	NUTLTY	Number of different energy sources
UCOSTD	NBULK	Number of different blocks of an incremental charge

2. COMMON BLOCKS

STATD, UCOSTD

3. OUTPUT DATA

Name	Description
ENCOST	Total escalated yearly charge for each source ($)

c. TRACE BACK

Subroutine COSTEN is called by: R4PRNT

and calls: No subroutines

COSTEN

Do through different energy sources, (IU=1, NUTLTY)

NB = Number of charge blocks

Initialize intermediate cost variables

Do through 12 months of the years

Units of consumption = energy consumption/energy per unit

NB > 0

No

CSTM = uniform cost* units of energy consumption

Yes

Do through charge blocks, (IB=1, NB)

Charge block size = block multiplier (UBLK(1))

Uniform cost > 0

No

Continue

Yes

Block size = block multiplier* the highest peak of energy consumption units in month

Block ≥ EMON

Yes

OUT = TRUE

No

CSTM = CSTM + block size* charge per unit in block

Decrease energy consumption units chargeable by these units accounted for in charge block, EMON = EMON - BLOK

Index of charge block = last charge block specified

OUT

Yes

No

CSTM = CSTM + remaining energy consumption units* charge/unit

CSTM = CSTM + (charge/peak load unit)*(maximum of either the average of the month's highest peak and the year's highest peak or the base peak load)

CST = CST + (Maximum of either the maximum monthly charge or the total monthly energy usage cost)

Total escalated yearly charge = escalation factor* total yearly charge

COSTEQ

a. GENERAL DESCRIPTION

COSTEQ is a subroutine to calculate the equipment costs.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
LFCYCD	ALFCYC (1)	Interest rate (%)
LFCYCD	ALFCYC (2)	Labor inflation rate (%)
LFCYCD	ALFCYC (3)	Material inflation rate (%)
LFCYCD	ALFCYC (4)	Energy inflation rate (%)
LFCYCD	ALFCYC (5)	Project life (yrs)
LFYCYD	ALFCYC (6)	Labor cost ($/hr)
LFCYCD	ALFCYC (7)	Site cost factor
ECDATA	EQCOSR (1,I)	Reference data equipment Size (kW)
ECDATA	EQCOSR (2,I)	Reference data equipment unit cost ($)
ECDATA	EQCOSR (3,I)	Reference data equipment installed cost factor
ECDATA	EQCOSR (4,I)	Reference data equipment consumables ($/hr)
ECDATA	EQCOSR (5,I)	Reference data equipment maintenance
ECDATA	EQCOSR (6,I)	Reference data equipment life (hrs/yr)

Source of Data	Name	Description
ECDATA	EQCOSR (7,I)	Reference data equipment hours to minor overhaul (hrs)
ECDATA	EQCOSR (8,I)	Reference data equipment minor overhaul cost ($)
ECDATA	EQCOSR (9,I)	Reference data equipment hours to major overhaul (hrs)
ECDATA	EQCOSR (10,I)	Reference data equipment major overhaul cost ($)
ECDATA	EQCOSD (1-10, J.I)	Similar to EQCOSR (1-10), except that data is for given equipment I and size J
EDATA	KINS	Number of units installed
EDATA	KAV	Number of units available
EDATA	NEDATA	Number of different equipment sizes
EDATA	NEQSIZE	Number of different sizes of each equipment type

2. COMMON BLOCKS

ECDATA, EDATA, LFCYCD, STATD

3. OUTPUT DATA

Name	Description
EQCHT (1, I)	Total first cost of equipment index I ($)
EQCHT (2, I)	Total annual cost of equipment index I ($)
EQCHT (3, I)	Total cyclical cost of equipment index I ($)
EQCHT (4,I)	Unused
EQCHT (5,I)	Total cost of equipment index I ($)

c. TRACE BACK

Subroutine COSTEQ is called by: R4PRNT

and calls: CYC

COSTEQ

If site cost factor < 1, set site cost factor = 1

Set A1 = Interest rate/100

R = Material inflation rate/100

E = Project life (in years)

Calculate first cost factor, using interest rate and project life

FCF = E*A1*A [(A1+1)^{E}/(A1+1)$^{E-1}$]

Calculate inflation multiplier on annual costs, using labor inflation rate, site factor, and project life

ACM = E* (1 + (E-1)* ALFCYC (2)/200)* ALFCYC(7)

Do through equipment types (I=1, NEDATA)

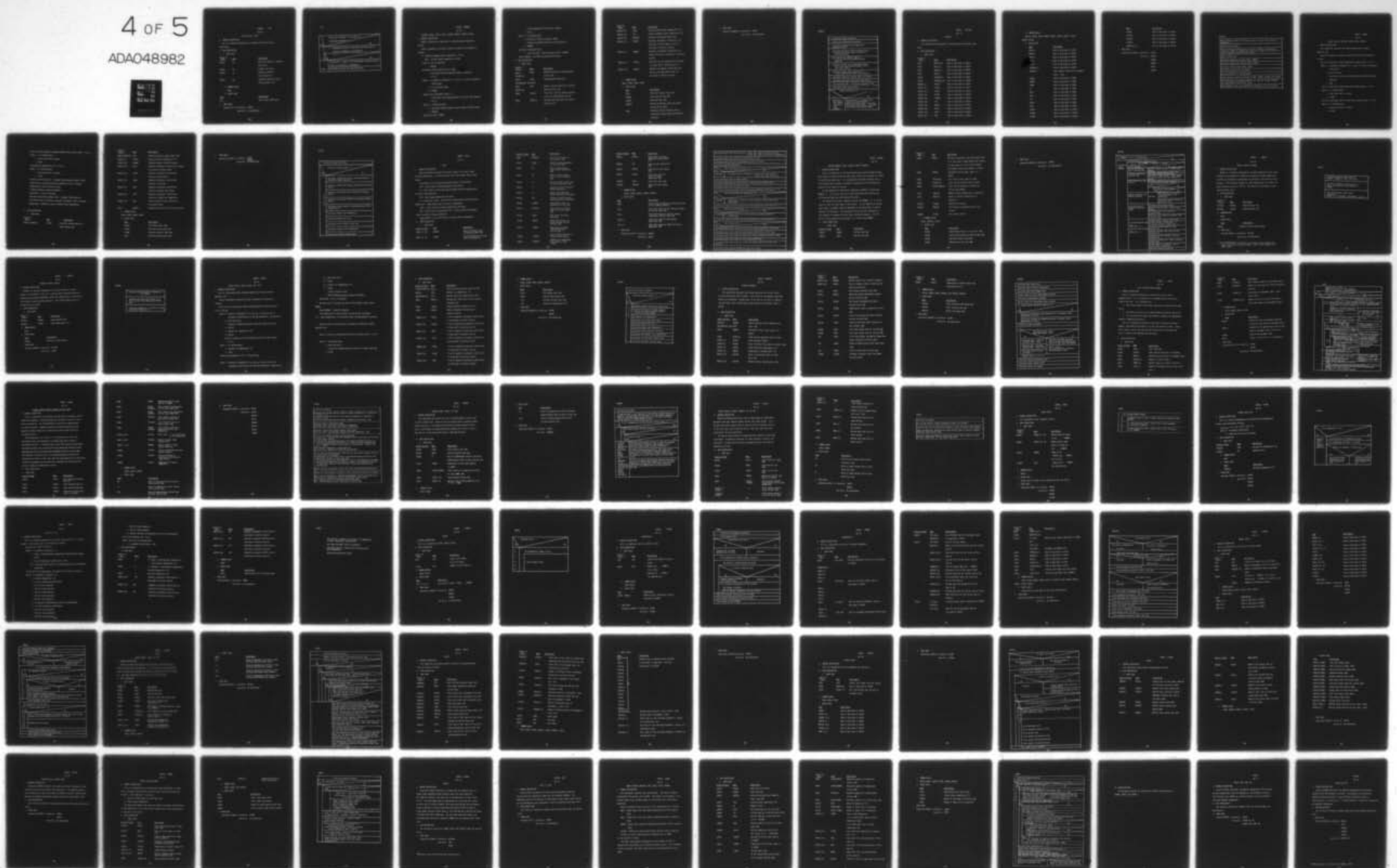
- Initialize output variables
- NS = number of equipment sizes for each equipment type
- NS $\leq$ 0
  - Yes: RETURN
  - No:
    - Do through different equipment sizes (J=1,NS)
      - Equipment first cost = number of units installed *uniform cost* installed cost factor* first cost factor
      - Annual cost = ACM* (consumables cost* operating hrs/yr + number of units installed* maintenance hrs/yr* labor cost/hr)
      - Total operating hours during project life = (operating hrs/yr)* (project life (yrs)/number of units installed)
      - Total cyclical cost = cyclical cost coefficient *number of units installed* uniform cost *installed cost factor
      - Total cyclical cost = present cyclical cost + cyclical cost coefficient* number of units installed* cost of minor overhaul
      - Repeat the calculation of cyclical cost with major overhaul
      - Total cost = first cost + annual cost + cyclical cost
      - Calculate total horizontally first, annual, cyclical, and total costs

AD-A048 982 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAI--ETC F/G 13/1
THE BUILDING LOADS ANALYSIS AND SYSTEM THERMO-DYNAMICS (BLAST) --ETC(U)
DEC 77 D C HITTLE

UNCLASSIFIED CERL-TR-E-119-VOL-2 CEEDO-TR-77-35-VOL-2 NL

4 OF 5

ADA048982



CYC (HO, CN, R, RH)

a. GENERAL DESCRIPTION

CYC is a function subroutine to calculate the cyclical cost coefficient.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
COSTEQ	HO	Hours to overhaul or replacement (hrs)
COSTEQ	CN	Project life (yrs)
COSTEQ	R	Material inflation rate (fraction)
COSTEQ	RH	Equipment operation hours during project life

2. COMMON BLOCKS

None

3. OUTPUT DATA

Name	Description
CYC	Cyclic cost coefficient

c. TRACE BACK

Function CYC is called by: COSTEQ

and calls: No subroutines

CYC

Hours until overhaul = 0 or hours of equipment operation = 0

Yes | No

CYC=0

No:

F = Rounded-off value of (equipment operation hours)/(hours to overhaul or replacement)

F < 1

Yes | No

Life period = (project life)/(equipment operation hours)/(hours to overhaul or replacement)

Calculate cyclical cost coefficient CYC

DBUNDLE (ECOOL, EHEAT, EELEC, ETOWER, ERCVCD, EWASTCD, RCAV)

a. GENERAL DESCRIPTION

DBUNDLE simulates the operation of a double-bundle compression chiller.

Several parameters relating tc output variables are evaluated as follows:

Z = (leaving condenser water temperature - A(1))/
A(2) - (chilled water temperature - A(3))

where A = set of constants
= ADJTDB

Available cooling capacity ratio, RCAV
= (available cooling capacity)/(nominal capacity)
= f(c,x)

where f = a quadratic polynomial in x, and c is a set of polynomial coefficients
x = Z, as defined above
c = RCAVDB

Energy ratio adjustment factor, G
= (full-load input energy)/nominal full-load input energy)
= f(c,x)

where f = as defined above
x = available cooling capacity ratio, RCAV,as defined above
c = ADJEDB

Energy I/O ratio, RPOWER

= (input energy)/(full-load input energy)

= f(c,x)

where f = as defined above

x = fraction of nominal capacity, RLOAD

= (required cooling)/(nominal cooling capacity)

c = RPWRDB

Available recoverable heat

= (cooling load + electric energy input) RAVRHDB

where RAVRHDB = available recoverable heat ratio

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP (PLOAD(13))	ECOOL	Required cooling for double-bundle chiller (kW)
SIMTEP (ABS(ENGYLDC1,IHR,IDAY))	EHEAT	Required heat energy (kW)
EPARS (OPCAP(13))	CNCD	Nominal cooling capacity for double-bundle chiller (kW)
EDATA	PEL(13)	Electrical input to nominal capacity ratio for double-bundle chiller
EDATA	RMIN(13)	Minimum part-load ratio for double-bundle chiller

Source of Data	Name	Description
SDATA(1,10)	TCOOL	Leaving chilled water temperature (°C)
SDATA(1,13)	TCW	Leaving condenser water temperature (°C)
SDATA(1,20)	RAVRHDB	Available recoverable heat ratio
PDATA(1,10)	RCAVDB	Quadratic polynomial coefficients for available cooling capacity ratio as described in previous section
PDATA(1,11)	RPWRDB	Quadratic polynomial coefficients for energy I/O ratio as described in previous section
PDATA(1,12)	ADJTDB	Constants for the evaluation of variable Z as described in previous section
PDATA(1,13)	ADJEDB	Quadratic polynomial coefficients for energy ratio adjustment factor as described in previous section

2. COMMON BLOCKS

EDATA, EPARS, PDATA, SDATA

3. OUTPUT DATA

Name	Description
EELEC	Electrical energy input (kW)
ETOWER	Tower cooling load (kW)
ERCVCD	Recovered heat (kW)
EWASTCP	Wasted recoverable heat from double-bundle chiller (kW)
RCAV	Available cooling capacity ratio = (available cooling capacity)/(nominal capacity)

c. TRACE BACK

Subroutine DBUNDLE is called by: SIMTEP

and calls: No subroutines

DBUNDLE

Initialize output variables
Initialize intermediate variables
Calculate condenser water temperature adjustment factor
Calculate available cooling capacity ratio = (available cooling capacity)/(nominal cooling capacity)
Set fraction of nominal capacity = (required cooling)/(nominal cooling capacity)
Calculate energy ratio adjustment factor = (full load input energy)/(nominal full-load input energy)
Calculate energy input output ratio = (input energy)/(full-load input energy)
Calculate electrical energy input to chiller
Calculate available recoverable heat energy
Set recovered heat = available recoverable heat
Calculate tower cooling load

Required heating > 0

Yes	No
Recovered heat energy from double-bundle is used elsewhere	Recovered heat = recovered heat energy from double-bundle Waste recoverable heat = available recoverable heat energy - recovered heat energy from double-bundle

DFLTASG

a. GENERAL DESCRIPTION

This subroutine assigns default values and posts processes input data.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
EDATA	NEQSIZE(I)	Same as described in SIMTEP
EDATA	KAV(J,I)	Same as described in SIMTEP
EDATA	CNOM(J,I)	Same as described in SIMTEP
EDATA	ROPT(I)	Same as described in SIMTEP
ECDATA	EQCOSR(K,I)	Same as described in SIMTEP
ECDATA	EQCOSD(K,J,I)	Same as described in SIMTEP
SDATA(1,1)	HSTEAM	Same as described in SIMTEP
SDATA(1,2)	TSATUR	Same as described in SIMTEP
SDATA(1,17)	TLEAVE	Same as described in SIMTEP
SDATA(1,31)	PSTEAM	Same as described in SIMTEP
SDATA(1,33)	TSTMTUR	Same as described in SIMTEP
SDATA(1,38)	TILT	Same as described in SIMTEP
SDATA(1,39)	AZMUTH	Same as described in SIMTEP
SDATA(1,42)	FLH2O	Same as described in SIMTEP
SDATA(1,43)	HXEFF	Same as described in SIMTEP
PDATA(2,21)	FRUL	Same as described in SIMTEP

2. COMMON BLOCKS

ECDATA, EDATA, PDATA, EPARS, REPOPT, SDATA, STMTUR, TITLED, TOWERD, SOLARD

3. OUTPUT DATA

Name	Description
RMIN(I)	Same as described in SIMTEP
RMAX(I)	Same as described in SIMTEP
ROPT(I)	Same as described in SIMTEP
PEL(I)	Same as described in SIMTEP
NEDATA	Same as described in SIMTEP
IENAME(J,I)	Same as described in SIMTEP
TOTCAP(I)	Total nominal capacity of equipment type I (kW)
IABSOR	Same as described in TOWER
ICOMPR	Same as described in TOWER
ITOWR	Same as described in TOWER
KT	Same as described in TOWER
PNTK	Same as described in TOWER
CNTU	Same as described in TOWER
CNT	Same as described in TOWER
CNTUH	Same as described in TOWER
CNTH	Same as described in TOWER
TLEAVE	Same as described in SIMTEP
PSTEAM	Same as described in SIMTEP
TSATUR	Same as described in SIMTEP

Name	Description
HSTEAM	Same as described in SIMTEP
TSTMTUR	Same as described in SIMTEP
HSTMTUR	Same as described in STMTUR
SSTMTUR	Same as described in STMTUR
EQCOSD(K,J,I)	Same as described in COSTEQ

c. TRACE BACK

Subroutine DFLTASG is called by: SIMTEP

and calls: R3PRNT

ENTHAL

ENTROP

SATUR

STURDS

DFLTASG

Assign default values for reference cost to actual cost conversion parameters
Define variable names
Assign values for minimum part load ratio, RMIN(I), maximum part load ratio, RMAX(I), optimum part load ratio, ROPT(I), and electrical input to nominal capacity ratio, PEL(I).
Default SOLAR data
Calculate absorption chiller index, IABSOR
Calculate compression chiller index, ICOMPR
Calculate cooling tower index, ITOWR
Fill total capacity array, TOTCAP
Set up variables for cooling tower
Assign default values for TLEAVE, PSTEAM, TSATUR, HSTEAM, and TSTMTUR
Calculate specific enthalpy HSTMTUR, and specific entropy SSTMTUR for superheated high pressure steam
Fill equipment cost array EQCOSD from reference cost array EQCOSR

DIESEL (EELECD, EJACKD, ELUBED, EEXD, EFUELD)

a. GENERAL DESCRIPTION

This subroutine simulates the super-charged diesel engine-generator set.

Several parameters relating to output variables are evaluated as follows:

Ratio of (electrical energy output)/fuel energy input) = f(c,x)

where f = a quadratic polynomial in x, and c is a set of polynomial coefficients

x = part-load ratio

= (electrical output required)/(nominal electrical output capacity)

c = RELD

Ratio of (available jacket heat)/(fuel energy input) = f(c,x)

where f = as defined above

x = part-load ratio, as above

c = RJACD

Ratio of (available lube-oil heat)/(fuel energy input) = f(c,x)

where f = as defined above

x = part-load ratio, as above

c = RLUBD

Ratio of (heat energy of exhausted gases)/(fuel energy input) = f(c,x)

where f = as defined above

x = part-load ratio as above

c = REXD

Exhausted gas temperature (°C) = f(c,x)

where f = as defined above

x = part-load ratio, as above

c = TEXD

Exhaust gas flow (kg/sec) = (exhaust heat energy)/((specific heat coefficient at constant pressure)*(enthalpy of gas at exhaust temperature)) (stack U-factor)* area

= UACD(1)*(nominal capacity)**UACD(2)

where UACD = a set of constants

Available recoverable exhaust heat = (exhaust flow)*(specific heat coefficient at constant pressure)* (enthalpy of gas at exhaust temperature - enthalpy of gas at exhaust stack temperature)

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP (PLOAD(2))	EELECD	Electrical energy output for diesel engine (kW)

Source of Data	Name	Description
EPARS (OPCAP(2))	PND	Nominal electrical power output (kW)
SDATA(1,2)	TSATUR	Steam saturation temperature (°C)
SDATA(1,22)	RMXKWD	(Maximum exhaust flow)/(kW output)
PDATA(1,14)	RELD	Quadratic polynomial coefficients related to electrical energy output
PDATA(1,15)	RJACD	Quadratic polynomial coefficients related to jacket heat
PDATA(1,16)	RLUBD	Quadratic polynomial coefficients related to lube heat
PDATA(1,17)	REXD	Quadratic polynomial coefficients related to exhaust heat energy
PDATA(1,18)	TEXD	Quadratic polynomial coefficients related to exhaust gas temperature
PDATA(1,37)	UACD	(Stack U-factor)* area coefficient for diesel engine
EDATA	RMIN(2)	Minimum part load ratio for diesel engine

2. COMMON BLOCKS

EDATA, EPARS, PDATA, SDATA

3. OUTPUT DATA

Name	Description
EFUELD	Fuel energy input (kW)
EJACKD	Available jacket heat (kW)
ELUBED	Available lube-oil heat (kW)
EEXD	Available exhaust heat (kW)

c. TRACE BACK

Subroutine DIESEL is called by: SIMTEP
OPCOOL
and calls: No Subroutines

DIESEL

Initialize output variables.

Electrical energy output < 0

YES: RETURN

NO:

- Calculate: (electrical output required)/(nominal electrical output capacity)
- Calculate: (electrical energy output)/(fuel energy input)
- Calculate: (available jacket heat)/(fuel energy input)
- Calculate: (available lube-oil heat)/(fuel energy input)
- Calculate: (heat energy of exhausted gases)/(fuel energy input)
- Calculate exhaust gas temperature
- Calculate exhaust gas flow
- Calculate stack U-factor* area term
- Calculate exhaust stack temperature
- Calculate available exhaust heat

EFFIC

a. GENERAL DESCRIPTION

Subroutine EFFIC calculates efficiency factors for several types of energy production. It also calculates total fuel energy input to the system.

Calculations are straightforward applications of definitions:

1/FE = plant electric energy generation efficiency

FE = (fuel input to the prime movers)/(plant electric energy generation plus recovered heat)

1/FH = heat energy generation efficiency

FH = (plant fuel input - plant electric generation* FE)/(total heat generation - steam energy used for electric generation)

1/FUE = total electronic energy generation efficiency

FUE = (plant electric generation *FE + utility electricity/TOTUEF)/(total electrical energy generation)

where TOTUEF = total efficiency of utility electric generation

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SDATA (1, 36)	RWSTUR	Ratio of exhaust steam to entering steam flow
SDATA (1, 37)	TOTUEF	Total efficiency of utility electric generation

Source of Data	Name	Description
EPARS	PLOAD(4)	Part load of boiler at current hour (kW)
EFFICD	EPLANT	Electric energy generated in the plant (kW)
EFFICD	PG	Ratio of gas turbine load to total electrical load
EFFICD	PD	Ratio of diesel engine load to total electrical load
EFFICD	PS	Ratio of steam turbine load to total electrical load
EFFICD	FE	Inverse of plant electrical energy generation
EFFICD	FH	Inverse of plant heat energy generation efficiency
EFFICD	FUE	Inverse of total electric energy generation efficiency
EFFICD	ESTMAB	Steam energy input to absorption chiller (kW)
EFFICD	EELECCO	Total electrical energy input to cooling stage (kW)
EFFICD	EHEAT	Heat energy (building load (kW)
EFFICD	EWASTE	Waste heat energy from diesel engines and gas turbines (kW)
EFFICD	ERCVCD	Heat energy recovered from double-bundle chiller (kW)
STMTUR	FSTMTUR	Flow of steam entering to steam turbine (kg/hr)
STMTUR	HSTMTUR	Enthalpy of superheated high-pressure steam (kJ/kg)

Source of Data	Name	Description
STMTUR	TEXSTM	Temperature of exhaust steam from steam turbine (°C)
HOURTOT	EUT	Total utility electricity (kW)
HOURTOT	EFUELB	Total boiler fuel energy (kW)
HOURTOT	EFUELE	Total fuel input for electric energy generation (kW)
HOURTOT	EFUEL	Total fuel input (kW)
HOURTOT	EBOILER	Boiler net heat output (kW)

2. COMMON BLOCKS

EFFICD, EPARS, SDATA, STMTUR, HOURTOT

3. OUTPUT DATA

Name	Description
ENTOT	Total energy consumed by the plant and the utility company (Btu/hr)
EFIHC	Total fuel input for heat energy consumed by cooling stage (kW)
EFIEC	Total fuel input for electric energy consumed by cooling stage (kW)
EFUELTH	Total fuel input for heat energy generation (kW)
EFUELIE	Total fuel input for electrical energy generation (kW)

c. TRACE BACK

Subroutine EFFIC is called by: SIMTEP

and calls: SATUR

Calculate total fuel input per hour = total fuel input to boiler per hour
+ total fuel input for electrical generation

Calculate total plant electrical generation per hour

Calculate total electricity supplied by utility per hour

Initialize boiler efficiency

Total fuel input to boiler > 0

NO	YES
CONTINUE	Calculate boiler efficiency = (boiler part load ratio)/ (total fuel input to boiler)

Calculate waste heat of steam turbine

Set heat recovered from steam turbine = waste heat of steam turbine

Calculate total heat recovered in plant

Initialize inverse of plant electrical energy generation efficiency FE = 0

Total plant electrical generation > 0

NO	YES
CONTINUE	Recalculate inverse of plant electrical energy generation efficiency

Initialize inverse of total electrical energy generation efficiency FUE = 0

Total electrical energy > 0

NO	YES
CONTINUE	Recalculate inverse of total electrical energy generation coefficient

Initialize inverse of plant heat energy generation coefficient FH = 0

Heat energy load > 0

NO	YES
CONTINUE	Recalculate inverse of plant heat energy generation coefficient

Calculate total fuel input for heat energy consumed by cooling stage

Calculate total fuel input for electrical energy consumed by cooling stage

Calculate total fuel input for heat energy generation

Calculate total fuel input for electrical energy generation (plant and utility)

Calculate total energy consumed by plant and utility

ENSTOR (OPMODE, EHEAT, ECOOLD, EHEATR, ECOOLR)

a. GENERAL DESCRIPTION

This is a subroutine to calculate heating and cooling energy storage. Heat energy is stored in the hot water tank, while cooling energy is stored in the chilled water tank. It is assumed that the heat transfer rate is sufficiently high to permit any amount of energy up to the storage tank capacity to be stored in an hour.

At the beginning of each hourly simulation, ENSTOR is called with OPMODE < 0. It tries to satisfy or lower the heating and cooling load. The remaining loads are satisfied by the plant.

At the end of the hour, ENSTOR is called with OPMODE > 0. It first tries to store all waste heat as heat energy. If the capacity of the hot water tank is exceeded, then the remainder of the waste heat is used to generate chilled water. The absorption chiller capacity assigned for this purpose is assumed to be the useful operating capacity. Initialization of the subroutine occurs when it is called using OPMODE.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP	ECOOLD	Cooling load (kW)
SIMTEP	EHEAT	Heating load (kW)

Source of Date	Name	Description
SIMTEP	OPMODE	Variable representing the operational mode; it is less than 0, when energy from storage is used; equal to 0 for initialization; and greater than 0 when energy is stored.
EPARS	IABSOR	Absorption chiller type, same as in ABSREF
EPARS	TOTCAP(18)	Total heat storage capacity (kWh)
EPARS	TOTCAP(19)	Total cooling storage capacity (kWh)
EPARS	TOTCAP(IABSOR)	Total cooling capacity of absorption chiller type IABSOR
LDIST	IOPR(18)	Number of heat storage tank in operation
LDIST	IOPR(19)	Number of cooling storage tank in operation
HOURTOT	ESTORBL	Storable heat energy
HOURTOT	EWASTED	Total wasted recoverable heat at end of hour
HOURTOT	ESTRED	Total energy stored

2. COMMON BLOCKS

EPARS, HOURTOT, LDISTD

3. OUTPUT DATA

Name	Description
EHSTOR	Stored heat energy at end of hour (kW)
ECSTOR	Stored cooling energy at end of hour (kW)
EHEATR	Remaining heating load (kW)
ECOOLR	Remaining cooling load (kW)

c. TRACE BACK

Subroutine ENSTOR is called by: SIMTEP

and calls: No Subroutines

ENSTOR

OPMODE ≠ 0 (no initialization)

NO | YES

NO:

Initialize energy storage

YES:

Initialization of remaining Loads

OPMODE > 0 (Energy is stored)

NO (Energy used for storage) | YES (energy stored)

NO (Energy used for storage)	YES (energy stored)
Set stored cooling used = cooling load	Set heat energy stored = storable heat energy
	Calculate stored heat at end of hour
Calculate remaining cooling load	Calculate remaining wasted heat
	Set ratio of heat energy = (heat energy in/cooling energy out) for absorption chiller
Store cooling at end of hour	Initialize cooling to be stored at 0
	Total cooling storage capacity of chiller 0 = 0 — Yes / No
	No: Remaining waste heat = 0 — Yes / No
Repeat procedures for heating storage	Yes: Continue
	No: Set cooling to be stored = (remaining heat to be stored)/SA
	No: Calculate stored cooling at the end of hour
Initialization of equipment operation index	Calculate total energy stored
	Remaining heating load = total energy stored
	Calculate total heat recovered at end of hour
	Calculate total wasted recoverable heat at end of hour
If stored heating used >1. IOPR(1,18) = 1 If stored cooling used >1. IOPR(1,19) = 1	Calculate total heat at end of hour
	If heat stored>1, IOPR(1,18) = 1 If cooling stored>1, IOPR(1,19) = 1
	Calculate present demand on heat storage tank
	Calculate present part load on heat storage tank
	Calculate present demand on cooling storage tank
	Calculate present part load on cooling storage tank
	Assign number of different sizes of storage tank in operation
	Calculate operating capacity

ENTHAL (PSTEAM, TSTEAM)

a. GENERAL DESCRIPTION

ENTHAL is a function subprogram to calculate enthalpy of pure steam. Enthalpy is calculated as a quadratic polynomial of steam temperature, in which the polynomial coefficients are related to the absolute steam pressure. This function applies for steam temperatures up to 556°C and steam pressure up to 6.9×10^6 Pa. The function is explained in CERL Interim Report E-81.[23]

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
DFLTASG	PSTEAM	Steam pressure (Pa)
DFLTASG	TSTEAM	Steam pressure (°C)

2. COMMON BLOCKS

None

3. OUTPUT DATA

Name	Description
ENTHAL	Enthalpy of pure steam (kJ/kg)

c. TRACE BACK

Function ENTHAL is called by: DFLTASG

and calls: No subroutines

[23] D. C. Hittle and B. Sliwinski, *CERL Thermal Loads Analysis and Systems Simulation Program*, Volumes 1 and 2, Interim Report E-81 (CERL, 1975).

ENTHAL

Calculate absolute steam pressure: PSTMAB = PSTEAM + 1.013×10^5, (Pa)
Determine quadratic coefficients A, B, and C using absolute steam pressure PSTMAB
Calculate enthalpy using steam temperature: ENTHAL = A + B*TSTEAM + C*TSTEAM**2, (kJ/kg)

ENTROP (PSTEAM, TSTEAM)

a. GENERAL DESCRIPTION

ENTROP is a function subprogram to calculate entropy of steam. Entropy is calculated in terms of a function which involves saturation temperature and steam temperature, while the coefficients in the function are related to absolute steam pressure. CERL Interim Report E-81 describes the calculation.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
DFLTASG	PSTEAM	Steam pressure (Pa)
DFLTASG	TSTEAM	Steam temperature (°C)

2. COMMON BLOCKS

None

3. OUTPUT DATA

Name	Description
ENTROP	Entropy of steam (kW/kg)

c. TRACE BACK

Function ENTROP is called by: DFLTASG

and calls: SATUR

ENTROP

Calculate steam saturation temperature (by SATUR)

Determine function coefficients A, B, and C, using steam temperature PSTEAM (Pa)

Calculate entropy as a function of steam saturation temperature

GASTUR (EELEC, EFUEL, ELUBE, EEX, TEX)

a. GENERAL DESCRIPTION

This is a subroutine that simulates operation of the gas turbine-generator set.

Several parameters relating to output variables are evaluated as follows:

Ratio of (gas turbine fuel energy input)/(electrical energy output) = $f(c,x_1)\ f(B,X_2)$

where f = quadratic polynomials in x_1 and x_2; c and B are sets of polynomial coefficients for the two polynomials, respectively

x_1 = part-load ratio

= (electrical demand)/(nominal electrical output capacity)

c = FUEL1G

T = ambient air temperature (°C)

B = FUEL2G

Ratio of (exhaust gas flow)/(nominal electrical power output)

= f(c,x)

where f = as defined above

x = ambient air temperature (°C)

c = FEXG

Exhaust gas temperature (°C) = $f(D,x_3)f(E,x_4)$

where f = quadratic polynomials in x_3 and x_4; D and E are sets of polynomial coefficients for the two polynomials, respectively

x_3 = part-load ratio

D = TEX1G

x_4 = ambient air temperature (°C)

E = TEX2G

(Stack U-factor)* area

= UACG(1)*nominal capacity (Btu/hr))**UACG(2)

where UACG = a set of constants

Maximum value of (exhaust gas flow (lb/hr))/(power output (kW)) = RMXKWG

where RMXKWG = a special parameter

The temperature of exhaust gases leaving the heat exchanger, i.e., stack temperature, is calculated using the thermodynamic relation.

Default values of coefficients corresponse to SOLAR gas turbine-generator set.

Ratio of (lube-oil heat)/(nominal electrical energy output) = f(c,x)

where f = as defined above

x = part-load ratio

= (electrical demand)/(nominal electrical output capacity)

c = ELUBG

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP(PLOAD(1))	EELEC	Electrical energy output capacity (kW)
WEATHR	TAIR	Ambient air temperature (°C)
EPARS(OPCAP(1))	PNG	Nominal electrical power output (kW)
EDATA	RMIN(1)	Minimum part load ratio for gas turbine-generator set
SDATA	TSATUR	Steam saturation temperature (°C)
SDATA	RMXKWG	Special parameter as described in previous section
PDATA(1,19)	FUEL1G	A set of quadratic polynomial coefficients as described in previous section
PDATA(1,20)	FUEL2G	A set of quadratic polynomial coefficients as described in previous section
PDATA(1,22)	FEXG	A set of quadratic polynomial coefficients as described in previous section
PDATA(1,23)	TEX1G	A set of quadratic polynomial coefficients as described in previous section
PDATA(1,24)	TEX2G	A set of quadratic polynomial coefficients as described in previous section
PDATA(1,25)	ELUBG	A set of quadratic polynomial coefficients as described in previous section
PDATA(1,28)	UACG	A set of quadratic polynomial coefficients as described in previous section

2. COMMON BLOCKS

EDATA, EPARS, PDATA, SDATA, WEATHR

3. OUTPUT DATA

Name	Description
EFUEL	Fuel energy input (kW)
ELUBE	Lube-oil heat energy (kW)
EEX	Available exhaust heat (kW)
TEX	Exhaust gas temperature (°C)

c. TRACE BACK

Subroutine GASTUR is called by: SIMTEP

OPCOOL

and calls: No subroutines

GASTUR

Initialize output variables

Electrical energy output < 0

Yes: RETURN

No:

- Calculate nominal electrical output (kW)
- Calculate fraction of nominal capacity = (electrical energy output)/(nominal electrical energy output)
- Set ambient air temperature
- Calculate gas turbine fuel input
- Calculate exhaust gas flow
- Calculate exhaust gas temperature
- Calculate stack U-factor* area term
- Calculate the exhaust stack temperature
- Calculate the available exhaust heat
- Calculate available lube-oil heat

HEATREC (EHWDOM)

a. GENERAL DESCRIPTION

This subroutine simulates heat energy recoveries at various levels and calculates total boiler output. Four levels of recoverable waste heat energy are considered. Energy which is not used at any level is added to the next lower level; energy that cannot be used at the lowest level is wasted.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
CCBSTEPS ABS(ENGYLD(6,IHR,IDAY))	EHWDOM	Energy required to heat domestic hot water (kW)
EPARS	IABSOR	Absorption chiller type, same as in ABSREF
EPARS	OPCAP(17)	Steam turbine operating capacity (kW)
SDATA(1,1)	HSTEAM	Steam enthalpy (kJ/kg)
SDATA(1,3)	RFLASH	Ratio of boiler flash water to steam losses
SDATA(1,2)	TSATUR	Steam saturation temperature (°C)
SDATA(1,9)	TWMAKE	Temperature of makeup water (°C)
SDATA(1,30)	RHFLASH	Ratio of recoverable heat to steam flash heat
SDATA(1,34)	PEXSTUR	Nominal exhaust steam pressure (Pa)

Source of Data	Name	Description
SDATA(1,35)	RPMNOM	Nominal speed of gas turbine (rad/sec)
SDATA(1,36)	RWSTUR	Ratio of exhaust steam to steam turbine entering steam flow
EFFICD	EHEAT	Heat energy (building) load (kW)
EFFICD	EWASTE	Waste heat energy from diesel engines and gas turbines (kW)
EFFICD	ERCVCD	Heat energy recovered from double-bundle chiller (kW)
EFFICD	ESTMAB	Steam energy input to absorption chiller (kW)
EFFICD	EJACKD	Jacket heat energy from diesel engines and gas turbines (kW)
EFFICD	ELUBE	Lube-oil heat from diesel engines and gas turbines (kW)
SOLARD	ESOLC	Solar heat energy used for cooling (kW)
SOLARD	ESOLH	Solar heat energy used for heating (kW)
STM	ESTMS	Total steam energy consumed by steam users except absorption chillers (kW)
STM	EWTRM	Energy of water mixture from steam users (kW)
STM	FWTRM	Flow of return water mixture (kW)
STMTUR	HEXSTM	Enthalpy of exhaust steam from steam turbine (kJ/kg)

Source of Data	Name	Description
STMTUR	TEXSTM	Temperature of exhaust steam from steam turbine (°C)

2. COMMON BLOCKS

EFFICD, EPARS, SDATA, SOLARD, STM, STMTUR, HOURTOT

3. OUTPUT DATA

Name	Description
ERECOVR	Total recovered heat energy (kW)
ESTORBL	Total storable heat (kW)
EBOILER	Boiler net output (kW)

c. TRACE BACK

Subroutine HEATREC is called by: SIMTEP

and calls: No subroutines

HEATREC

- Calculate total steam loss
- Calculate flow of boiler flush water
- Calculate water heat makeup flow
- Assign values for heat requirement variables
- Initialize intermediate variables
- Assign values for recoverable heat variables
- Steam turbine operating capacity ≤ 0
  - Yes
  - No
    - Calculate exhausting steam enthalpy ES(1) if exhaust steam temperature $\geq$ 82.2°C
    - Calculate rejected low temperature heat if exhaust steam temperature $\geq$ 82.2°C
- Initialize input variables
- I = 1
  - Recoverable heat, ES(I) $\geq$ heat requirement, ED(I)
    - Yes
      - Recovered heat = previous recovered heat + ED(I)
      - Rejected heat = previous rejected heat + ES(I)-ED(I)
      - Calculate wasted recoverable heat
    - No
      - Subtract available recovered heat from heat required and add result to next heat requirement
      - ED(I) = ES(I)
        Set recovered heat = previous recovered heat + ED(I)
  - Repeat until (I + 1) > 4
- Calculate wasted recoverable heat
- Calculate total storable heat
- Calculate boiler net output

LDIST (IEQTYPE,TLOAD,PLOADE)

a. GENERAL DESCRIPTION

LDIST is a default subroutine to distribute load to similar equipment units. It is called only if assignment tables are not provided by the user. It is assumed that:

1. Units have identical efficiency functions (in terms of part-load ratio).

2. Efficiency functions can be approximated by quadratic functions.

3. The rule should be simple and practical enough to be implemented without a computerized control.

This is a combination algorithm involving an iterative procedure. However, the number of iterations is at most the number of units. Allocation is done in such a way that the part-load ratio of the operating units is close to the optimum part-load ratio.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	NAME	Description
OPCOOL	TLOAD	Total load (kW)
OPCOOL	IEQTYPE	Index denoting the type of equipment
EDATA	RMAX(J)	Maximum part load ratio of equipment type J
EDATA	NEQSIZE(J)	Number of sizes of type J
EDATA	CNOM(I,J)	Nominal capacity of type J and size I
EDATA	KAV(I,J)	Number of available units of type J and size I

Source of Data	Name	Description
EPARS	TOTCAP(J)	Total nominal capacity of equipment type J
DISTB	NDISTB(K)	Number of load ranges for load distribution by table
DISTB	DISTB(I,J)	Load range for equipment type J, load range index I
DISTB	IDISTB(I,K,J)	Number of units in use of size index I, load range index K, and equipment type J

2. COMMON BLOCKS

DISTB, EDATA, EPARS, LDISTD

3. OUTPUT DATA

Name	Description
NOPR(J)	Number of sizes of equipment operating at current time step for equipment type J
IOPR(J)	Number of units operating at current time step for equipment type J with size I
PLOADE	Load of the equipment (kW).
KOP(I)	Number of units of size I in operation

c. TRACE BACK

Subroutine LDIST is called by: OPCOOL
SIMTEP

and calls: No subroutines

LDIST

Equipment type = IEQTYPE
Optimum part-load ratio = ROPT (IEQTYPE)
Number of equipment size = NEQSIZE(IEQTYPE)

Total capacity = 0 for IEQTYPE equipment
YES | NO

Total load < 0
YES | NO

Number of load ranges NDISTB(IEQTYPE) for load distribution = 0
NO | YES

NO:

Operating capacity = TOTCAP (IEQTYPE)

DO I through 100 different equipment sizes

Number of I size units operating, KOP(I)=Number of I size units available, KAV(IEQTYPE)

IT = 1

RNEW = total load/total capacity operating

DNEW = absolute value of (optimum part load ratio - RNEW)

Set ROLD = RNEW
DOLD = DNEW

No. of II size operating < 0
YES | NO

Reset operating capacity = operating capacity-Nominal capacity of type IEQTYPE with size II

Operating Capacity < total load
YES | NO

Decrease number of units

Recalculate RNEW and DNEW

DNEW < DOLD
NO | YES

Increase number of units operating by one

OUT = TRUE

II =II+1

Repeat until II + 1 > number of equipment, NEQ or OUT = TRUE

Set operating capacity = 0

DO J through NEQ equipment sizes for type IEQTYPE

Recalculate capacity operating = capacity operating + (No. of units operating of size J for type IEQTYPE

Load of equipment type IEQTYPE + ROLD* capacity operating

YES:

Equipment selection by table look up, 12=NDISTB(IEQTYPE)

Do I through 12 load ranges

Total load < load range of type IEQTYPE size I
YES | NO

OUT = TRUE | I = 12

Number of equipment size NEQ = NEQSIZE(IEQTYPE)

Set operating capacity = 0

DO J through NEQ equipment sizes

Number of units operating of size J, type IEQTYPE = IDISTB(J,I,IEQTYPE)

Recalculate operating capacity = operating capacity + (number of units operating of size J) *(nominal capacity of size J type IEQTYPE)

OPCAP(IEQTYPE) = operating capacity of equipment type IEQTYPE

Set equipment load = OPCAP (IEQTYPE)

Then:

Number of sizes operating of the IEQTYPE = NEQ

DO I through NEQ equipment sizes for type IEQTYPE

Number of units operating at current time step for equipment type IEQTYPE, size I IOP(I, IEQTYPE) = number of I size units operating,KOP(I)

OPCOOL (EELECT, EHEATT, ECOOLT, PA, PC, PCD)

a. GENERAL DESCRIPTION

This is a subroutine to distribute cooling load to different chiller types near optimally. Some approximations are done to avoid an interactive calculation. I/O relationships of chillers are approximated by linear functions. However, quantities relating prime movers are computed through simulation of prime movers using primary electrial and heating load.

The methodology is as follows. If the waste heat is more than the heating load, the remainder of the waste heat can be used by absorption chillers. If heating load is more than available waste heat, the remaining heat load can be satisfied by generating electricity for compression chillers and using the incremental portion of waste heat. The remainder of cooling load is distributed between absorption and compression chillers in such a way that the steam generated for absorption chillers is generated by the waste heat released while generating the electric energy for compression chillers.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP	EELECT	Total electrical energy load (kW)
SIMTEP	EHEATT	Total heating load (kW)
SIMTEP	ECOOLT	Total cooling load (kW)
EPARS	IABSOR	Absorption chiller type, same as in ABSREF

EPARS	ICOMPR	Compression chiller type, same as in COMREF
EPARS	TOTCAP (IABSOR)	Total capacity of absorption chiller type IABSOR (kW)
EPARS	TOTCAP (ICOMPR)	Total capacity of compression chiller type ICOMPR (kW)
EPARS	TOTCAP(1)	Total nominal capacity of gas-turbine engine (kW)
EPARS	TOTCAP(2)	Total nominal capacity of diesel engine (kW)
EPARS	TOTCAP (IABSOR)	Total nominal capacity of absorption chiller type IABSOR (kW)
SDATA (1,16)	CYPTYPE	Plant Type = 1 for utility only = 2 for mixed plant
SDATA (1,34)	PEXSTUR	Nominal exhaust steam pressure (Pa)
SDATA (1, 35)	RPMNOM	Nominal speed of steam Turbine (Rad/sec)
STMTUR	FSTMTUR	Flow of steam entering steam turbine (kg/sec)
STMTUR	HSTMTUR	Specific enthalpy of superheated high-pressure steam (kJ/kg)
STMTUR	TEXSTM	Temperature of exhaust steam (°C)

2. COMMON BLOCKS

EPARS, SDATA, STMTUR

3. OUTPUT DATA

Name	Description
PA	Ratio of absorption chiller load to total cooling load
PC	Ratio of compression chiller load to total cooling load
PCD	Ratio of double-bundle chiller load to total cooling load

c. TRACE BACK

Subroutine OPCOOL is called by: SIMTEP

and calls: DIESEL

GASTUR

LDIST

OPDBUN

OPELEC

SOLUSE

STMTUR

OPCOOL

Initialize variables
Calculate the average specific electric energy consumption of compression chillers, (sc) Calculate the average specific heat energy consumption of absorption chillers, (SA) Calculate the average specific energy consumption of boilers, (SB)
Simulate electric generation stage Calculate total fuel input for electric generation Calculate total waste heat from electric generation
Calculate (waste heat)/(power output) ratio, (CEX) Calculate (fuel input)/(power output) ratio for generators, (SG)
First try to satisfy heating load by waste heat If waste heat > heating load, the remainder of the waste heat will be used to give "free cooling" If heating load > available waste heat, the remaining heating load will be satisfied by generating electricity for "cheap cooling" and applying this incremental portion of waste heat to the remaining heating load up to a limit Free cooling = Remainder of waste heat/ SA Cheap cooling = Remainder of heating load/(SC*CEX)
Try to satisfy cooling load by "free cooling" and then by "cheap cooling," (within limit of component) The remainder of the cooling load is distributed between absorption and compression chiller in such a way that: The system required for the absorption chiller is generated by the waste released while generating the electric input to the compression chiller to supply their share of the remaining cooling load (unconstrained optimization)
Impose capacity constraints, adjust the ratio (constrained optimization)
Adjust load distribution if there is solar cooling available (call SOLUSE)
Ajust load distribution is there are double-bundle chillers and heat load Calculate PA = (absorption chiller load)/(total cooling load) PC = (compression chiller load)/(total cooling load) PCD = (double-bundle chiller load)/(total cooling load)

OPDBUN (ECOOL, EHEAT, PC, PCD)

a. GENERAL DESCRIPTION

This subroutine calculates the ratio of double-bundler chiller load to total cooling load. Given cooling and heating load, available double-bundle capacity is calculated by simulating the double-bundle chiller. Load is distributed to double-bundle and compression chillers in such a way that all of the recoverable heat is used when possible.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
OPCOOL	ECOOL	Total cooling load (kW)
OPCOOL	EHEAT	Building heating load (kW)
OPCOOL	PC	Ratio of compression chiller (including double-bundle) load to total cooling load
EPARS	ICOMPR	Compression chiller type, same as in COMREF
EPARS	TOTCAP(ICOMPR)	Total capacity of compression chiller of type ICOMPR (kW)
EPARS	TOTCAP (13)	Double-bundle chiller (kW)
EDATA	CNOM(1,13)	Nominal size of double-bundle chiller of size 1 (kWh)

2. COMMON BLOCKS:

EDATA, EPARS

3. OUTPUT DATA

Name	Description
PC	Ratio of compression chiller (excluding double-bundle) load to total cooling load
PCD	Ratio of double-bundle chiller load to total cooling load

c. TRACE BACK

Subroutine OPDBUN is called by: OPCOOL

and calls: DBUNDLE

OPDBUM

Initialize variables

If heating load is less than one of the smallest double-bundle unit capacity, set available cooling capacity to the nominal capacity; otherwise simulate double-bundle chiller to calculate available cooling capacity of double-bundle chiller

IF PC = (compression chiller)/(total cooling load) < 0

YES | NO

No double-bundle chiller available

YES | NO

Heating load = 0

YES | NO

Allocate regular compression chillers, then allocate double-bundle chillers to the remainder of the compression refrigeration load (if any double-bundle chiller available)	Allocate double-bundle chillers so that the rejected heat satisfies all heating load subject to the available double-bundle capacity constraint If double-bundle cooling load is less than 45 percent of total double-bundle capacity (empirical relation), try to allocate all double-bundle capacity, if necessary, to avoid inefficiently low part-load ratios If total allocated double-bundle capacity and regular compression chiller total capacity is less than the compression refrigeration load, redistribute the load such that double-bundle chillers are fully utilized

OPELEC (EELECT, EHEATT, ECOOLT, PG, PD, PS)

a. GENERAL DESCRIPTION

OPELEC distributes electrical load to three types of prime mover-generator sets--gas turbine, diesel engine, and steam turbine. It is assumed that diesel engines and gas turbines will not normally coexist in the same plant, and if they do coexist, they will share the load porportional to total capacities.

The load of steam turbine is determined with respect to waste heat requirement. An empirical relation for steam turbines is used in the subroutine. Average specific energy consumption factors are used for absorption chillers.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	*Name*	*Description*
OPCOOL	EELECT	Total electrical loads (kW)
OPCOOL	EHEAT	Total heating load (kW)
OPCOOL	ECOOLT	Total cooling load (kW)
EPARS	IABSOR	Absorption chiller type, same as in ABSREF
EPARS	TOTCAP (IABSOR)	Total nominal capacity of absorption chiller type IABSOR (kW)
TOTCAP (1) (EPARS)	G	Total nominal capacity of gas turbine (kW)
TOTCAP(2) (EPARS)	D	Total nominal capacity of diesel engine (kW)

Source of Data	Name	Description
TOTCAP (17)	S	Total nominal capacity of steam turbine (kW)
EDATA	CNOM(1,17)	Nominal size of steam turbine with size 1 (kW)
EDATA	ROPT(17)	Optimum part-load ratio of steam turbine
EDATA	RMX (1)	Maximum part-load ratio of gas turbine
EDATA	RMAX (2)	Maximum part-load ratio of diesel engine
EDATA	RMAX (17)	Maximum part-load ratio of steam turbine

2. COMMON BLOCKS

EDATA, EPARS

3. OUTPUT DATA

Name	Description
PG	Ratio of gas turbine load to total electrical load
PD	Ratio of diesel engine load to total electrical load
PS	Ratio of steam turbine load to total electrical load

c. TRACE BACK

Subroutine OPELEC is called by: SIMTEP
OPCOOL

and calls: No subroutines

OPELEC

Initialize variables
Set average specific energy consumption factor for chillers
Calculate the ratio of waste heat needed to total electrical load (CWASTE)
Impose an empirical constraint on steam turbine load (in terms of CWASTE) Determine steam turbine load
Distribut remaining electrical generation load to diesel engines and gas turbines proportional to their total capacities

OUTRP (UNIT)

a. GENERAL DESCRIPTION

This subroutine prints requested results.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
REPOPT	IREPOP (1, IR)	Report option of index IR (2H . . . . FORMAT)
REPOPT	IREPOP (2, IR)	Report option range control of index IR (3H . . . .FORMAT)
REPOPT	NREPOP	Range of IR
		= METRIC (6H . . FORMAT) for SI units
SIMTEP	UNIT	= ENGLISH (7H . . FORMAT) for English units

2. COMMON BLOCKS

REPOPT

3. OUTPUT DATA

Output data are shown in the subroutines that are called.

c. TRACE BACK:

Subroutine REPORT is called by: SIMTEP

and calls: R1PRNT

R4PRNT

STATPR

OUTRP

Do I through NREPOP ranges

- If report option of index I equals 2H/M, print monthly values (R1PRNT)
- If report option of index I equals 2H/S, print equipment use statistics (STATPR)
- If report option of index I equals 2H/C, print life-cycle report (R4PRNT)

PSYTWD (TAIR, HR)

a. GENERAL DESCRIPTION

This is a function subprogram to calculate wet-bulb temperature using a cubic polynomial fitting:

$$T_{wet\ bulb} = C_1 + C_2Y + C_3Y^2 + C_4Y^3 \ (°C)$$

where C = polynomial coefficients

Y = logarithm of moist air enthalpy

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
CCBTEPS	TAIR	Dry-bulb air temperature (°C)
CCBTEPS	HR	Humidity ratio

2. COMMON BLOCKS

None

3. OUTPUT DATA

Name	Description
WETBULB	Wet-bulb temperature (°C)

c. TRACE BACK

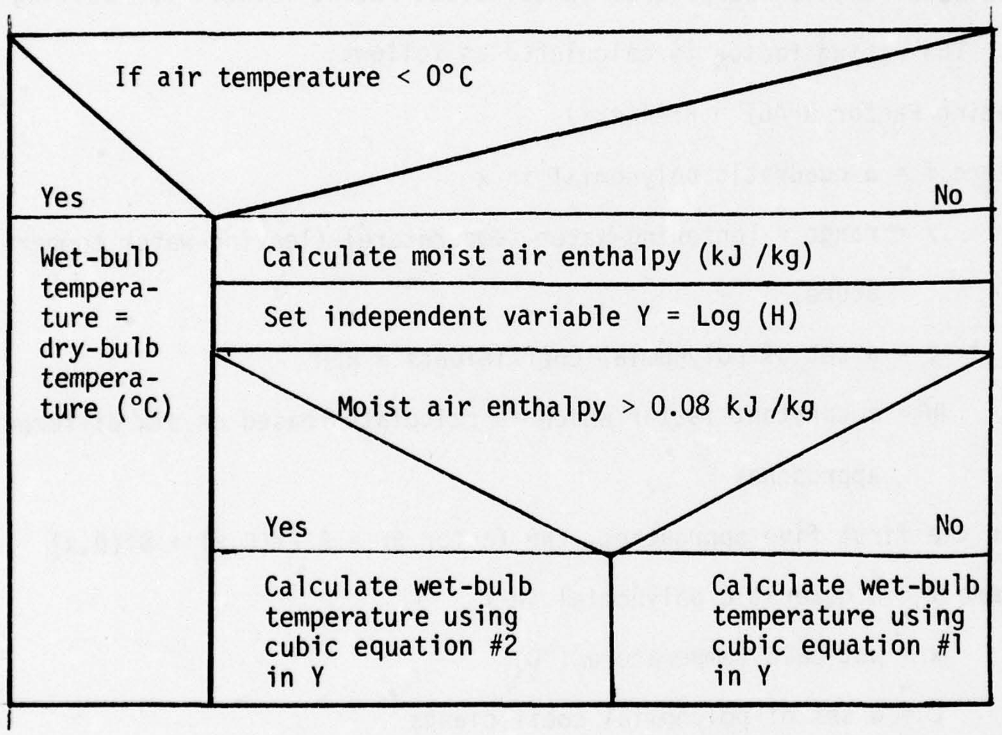
Function PSYTWD is called by: SIMTEP

and calls: PSYHTW

PSYWTP

SATUTH

PSYTWD



If air temperature < 0°C
Yes
No
Wet-bulb tempera-ture = dry-bulb tempera-ture (°C)
Calculate moist air enthalpy (kJ /kg)
Set independent variable Y = Log (H)
Moist air enthalpy > 0.08 kJ /kg
Yes
No
Calculate wet-bulb temperature using cubic equation #2 in Y
Calculate wet-bulb temperature using cubic equation #1 in Y

RFACT (R, A, T)

a. GENERAL DESCRIPTION

This is a function subprogram to calculate rating factors for cooling towers. The rating factor is calculated as follows:

Rating Factor RFACT = RF*f(c,x)

where f = a quadratic polynomial in x

x = range = (entering-water temperature)-(leaving-water temperature)

c = a set of polynomial coefficients = RFR

RF = a constant factor which is calculated based on six different approaches

For the first five approaches, the factor RF = A (f(C,x) + Bf(D,x)

where f = a quadratic polynomial in x

x = wet-bulb temperature (°C)

C = a set of polynomial coefficients

= RF1 for first approach

= RF2 for second approach

= RF3 for third approach

= RF4 for fourth approach

= RF5 for fifth approach

A = a constant corresponding to each of the approaches

D = a set of polynomial coefficients

= RF2 for first approach

= RF3 for second approach

= RF4 for third approach

= RF5 for fourth approach

= RF6 for fifth approach

B = another constant corresponding to each of the approaches

In the last approach, RF = f(c,x)

where f and z are as defined above

c = polynomial coefficients = RF6

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
TOWER	R	R = Range = entering-water temperature - leaving-water temperature (°C)
TOWER	A	A = Approach = leaving-water temperature - wet-bulb temperature (°C)
TOWER	T	Wet-bulb temperature (°C)
PDATA(1,26)	RF1	Quadratic polynomial coefficients as described in previous section
PDATA(1,27)	RF2	Quadratic polynomial coefficients as described in previous section
PDATA(1,28)	RF3	Quadratic polynomial coefficients as described in previous section

Source of Data	Name	Description
PDATA(1,29)	RF4	Quadratic polynomial coefficients as described in previous section
PDATA(1,30)	RF5	Quadratic polynomial coefficients as described in previous section
PDATA(1,31)	RF6	Quadratic polynomial coefficients as described in previous section
PDATA(1,39)	RFR	Quadratic polynomial coefficients as described in previous section

2. COMMON BLOCKS

PDATA

3. OUTPUT DATA

Name	Description
RFACT	Rating factor for the cooling tower

c. TRACE BACK

Function RFACT is called by: TOWER

and calls: No subroutines

RFACT

Calculate AP = Approach of wet-bulb air temperature to water temperature leaving tower

Set lower and upper limits on approach

Determine quadratic coefficients RF based on wet-bulb temperature

Calculate rating factor RFACT

RTPRNT

a. GENERAL DESCRIPTION

This is a subroutine to print report titles.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
CDPR	IPRT	Logical unit number for print output
TITLED	NTIT	Number of title index IT

2. COMMON BLOCKS

CDPR, TITLED

3. OUTPUT DATA

Name	Description
ITIT (1-8, IT)	Title data of index IT (80H. . . FORMAT)

c. TRACE BACK

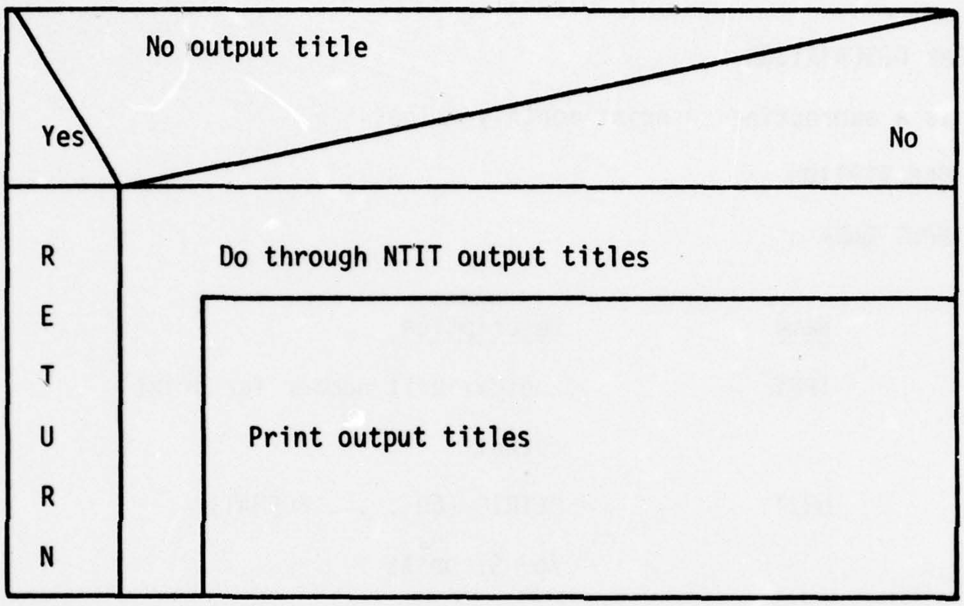
Subroutine RTPRNT is called by: R1PRNT

R4PRNT

STATPR

and calls: No subroutines

RTPRNT


No output title
Yes
No
RETURN
Do through NTIT output titles
Print output titles

R1PRNT(UNIT)

a. GENERAL DESCRIPTION

This is a subroutine to print monthly values.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
CDPR	IPRT	Logical unit number for print output
REPORT	UNIT	= METRIC (6H . . . FORMAT) for SI units
		= ENGLISH (7H . . FORMAT) for ENGLISH units

2. COMMON BLOCKS

CDPR, PRNTA1

3. OUTPUT DATA

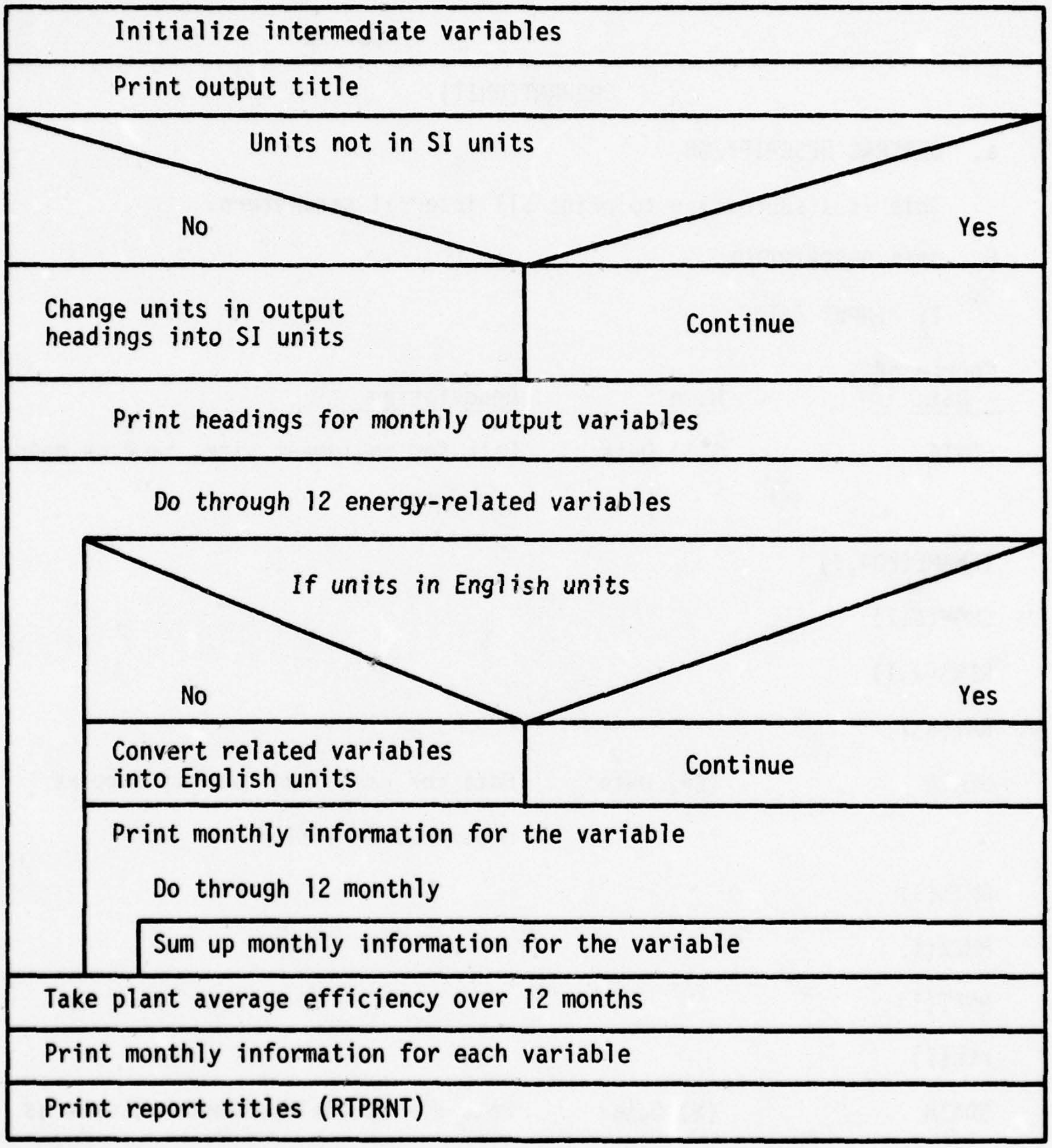
Name	Description
PRNTA1 (IMON,I)	Monthly output information; same as described in SIMTEP

c. TRACE BACK

Subroutine R1PRNT is called by: OUTRP

and calls: RTPRNT

R1PRNT


Initialize intermediate variables
Print output title
Units not in SI units
No
Yes
Change units in output headings into SI units
Continue
Print headings for monthly output variables
Do through 12 energy-related variables
If units in English units
No
Yes
Convert related variables into English units
Continue
Print monthly information for the variable
Do through 12 monthly
Sum up monthly information for the variable
Take plant average efficiency over 12 months
Print monthly information for each variable
Print report titles (RTPRNT)

R3PRNT(UNIT)

a. GENERAL DESCRIPTION

This is a subroutine to print all internal parameters.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
EDATA	(ES) Data:	Data for equipment size; same as described in SIMTEP
IENAME(204,I)		
CMOM(J,I)		
KINS(J,I)		
KAV(J,I)		
DEATA	(ER) Data:	Data for part-load ratios; same as described in SIMTEP
RMIN(I)		
RMZX(I)		
ROPT(I)		
PEL(I)		
SDATA	(S) Data:	Data for special parameters; same as described in SIMTEP
SDATA(J,I)		
PDATA	(EP) Data:	Data for equipment performance coefficients
PDATA(J,I)		

Source of Data	Name	Descriptions
ECDATA	(CR) Data:	Cost reference data for equipment; same
	EQCOSR(J,I)	as described in SIMTEP
UCOSTD	(CU) Data:	Cost of utility, energy
	NBULK(IU)	Number of block multipliers used for utility type IU
	UBLK(1,K,IU)	Block multiplier for Kth block, utility type IU
	UBLD(2,K,IU)	Cost per unit for Kth block, utility type IU
	IUNAME(IU)	Utility, energy name (6H.....FORMAT)
	UDATA(1,IU)	Energy/unit for utility type IU (kW)
	UDATA(2,IU)	Uniform cost/unit for utility type IU ($)
	UDATA (3,IU)	Cost escalation factor for life cycle for utility type IU
	UDATA(4,IU)	Minimum peak load charge for utility type IU ($)
	UDATA(5,IU)	Minimum peak load for utility type IU (unit)
	UDATA(6,IU)	Peak load unit cost for utility type IU ($/unit)
LFCYCD	(L) Data:	Life-cycle data; same as described in SIMTEP
	ALFCYC(I)	
	(CE) Data:	Data for cost of equipment; same as described in SIMTEP

Source of Data	Name	Description
EDATA	IENAME (J,I)	
ECDATA	EQCOSD(K,J,I)	
TILTED	(T) Data:	Report title; same as described in RTPRNT
	NTIT	
	ITIT(I,IT)	
	(EA) Data:	Equipment assignment data
EDATA	IENAME(J,I)	Same as described in SIMTEP
EDATA	NEQSIZ(I)	Same as described in SIMTEP
DISTB	NDISTB	Same as described in LDIST
DISTB	DISTB(J,I)	Same as described in LDIST
DISTB	IDISTB(J,K,I)	Same as described in LDIST
CDPR	IPRT	Logical unit number for print output
KEYLST	KEYLST(J,I)	Data card type name (30H...FORMAT)

2. COMMON BLOCKS:

CDPR, DISTB, ECDATA, EDATA, KEYLST, LDISTB, LFCYCD, PDATA, REPOPT, SDATA, TILTED, UCOSTD.

3. OUTPUT DATA

Output data are the same as input data (printed out).

c. TRACE BACK:

Subroutine R3PRNT is called by: DFLTASG

and calls: No subroutines

R3PRINT

Use English units

No | Yes

Set units in headings in SI units | Set units in headings in English units

Check data card type code . .

Use English units

No | Yes

Continue | Convert equipment size data into English units

Print equipment size,(ES) data

Print equipment operation ratios, (ER) data

Use English units

No | Yes

Continue | Convert special parameters into English units

Print special parameter data, (S) data

Print equipment performance, (EP) data

Print equipment cost reference, (CR) data

Print cost of utility, energy, (CU) data

Print life cycle, (L) data

Print cost of equipment, (CE) data

Print report title, (T) data

Print equipment assignment table, (EA) data

R4PRNT (UNIT)

a. GENERAL DESCRIPTION

This is a subroutine to print the life-cycle report.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
CDPR	IPRT	Logical unit number for print output
EDATA	NEDATA	Range of equipment of size I as specified
EDATA	NEQSIZE (I)	Number of different sizes of equipment type I
REPORT	UNIT	METRIC (6H. . FORMAT), for SI units ENGLISH (7H . . FORMAT), for English units
	NUTLTY	Number of different utilities

2. COMMON BLOCKS

CDPR, ECDATA, EDATA, LFCYCD, STATD, UCOSTD

3. OUTPUT DATA

Name	Description
IENAME (J,I)	Same as described in SIMTEP
KINS (J,I)	Same as described in SIMTEP
CNOM (J,I)	Same as described in SIMTEP

Name	Description
AMAXLD (I)	Same as described in SIMTEP
MAXTIM (J, I)	Same as described in SIMTEP
EQCHT (J,I)	Same as described in SIMTEP
EQCOST (K, J, I)	Same as described in SIMTEP
TOTECS	Same as described in SIMTEP
IUNAME (IU)	Same as described in SIMTEP
ENCOST (IU)	Same as described in SIMTEP
ENUSE (13, IU)	Same as described in SIMTEP
ENPEAK (13, IU)	Same as described in SIMTEP
UDATA (3, IU)	Same as described in SIMTEP
TOTUCS	Same as described in SIMTEP
TOTCOST	Same as described in SIMTEP

c. TRACE BACK:

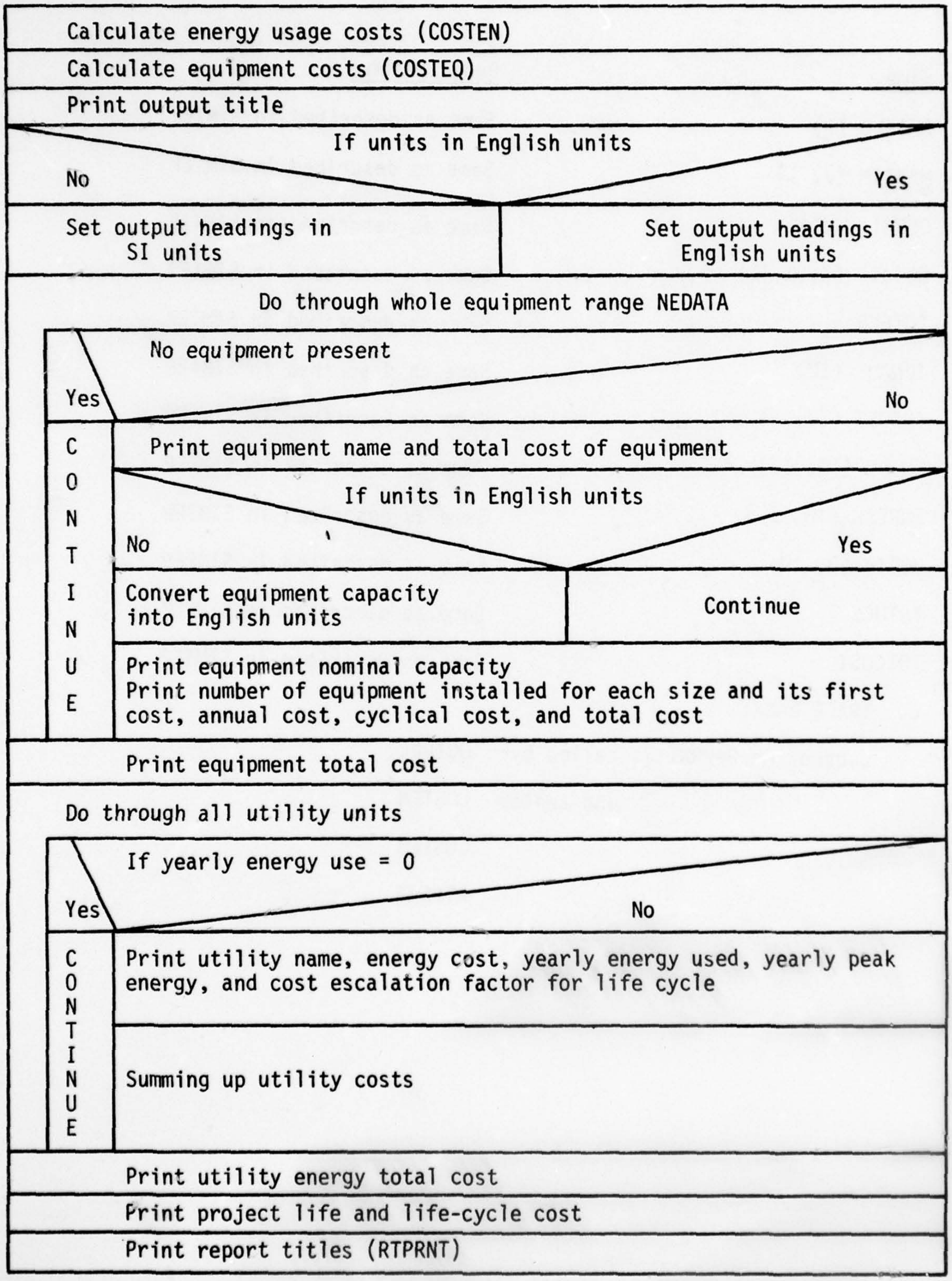
Subroutine R4PRNT is called by: OUTRP

and calls: COSTEN

COSTEQ

RTPRNT

R4PRNT


Calculate energy usage costs (COSTEN)
Calculate equipment costs (COSTEQ)
Print output title
If units in English units
No
Yes
Set output headings in SI units
Set output headings in English units
Do through whole equipment range NEDATA
No equipment present
Yes
No
CONTINUE
Print equipment name and total cost of equipment
If units in English units
No
Yes
Convert equipment capacity into English units
Continue
Print equipment nominal capacity
Print number of equipment installed for each size and its first cost, annual cost, cyclical cost, and total cost
Print equipment total cost
Do through all utility units
If yearly energy use = 0
Yes
No
CONTINUE
Print utility name, energy cost, yearly energy used, yearly peak energy, and cost escalation factor for life cycle
Summing up utility costs
Print utility energy total cost
Print project life and life-cycle cost
Print report titles (RTPRNT)

SOLUSE (EHEAT, ECOOL, PA, PC)

a. GENERAL DESCRIPTION

SOLUSE calculates heat energy used for heating and cooling and adjusts cooling load distribution. It first tries to satisfy heating load, and then checks whether the remaining solar heat can be utilized in a one-stage absorption chiller to satisfy cooling load.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
OPCOOL	EHEAT	Heating load (kw)
OPCOOL	ECOOL	Cooling load (kW)
SOLARD	ESOLAR	Available solar heat energy (kW)
SOLARD	TSOLAR	Solar heat storage tank temperature (°C)
EPARS	IABSOR	Type index of absorption chiller, same as in ABSREF
EPARS	TOTCAP(16)	Total capacity of solar energy (kW)
EPARS	TOTCAP(5)	Total capacity of absorption chiller type 5
SDATA (1,14)	TMINH	Minimum tank temperature required for heating (°C)
SDATA (1,15)	TMINC	Minimum tank temperature required for cooling (°C)

2. COMMON BLOCKS

EPARS, SDATA, SOLARD

3. OUTPUT DATA

Name	Description
PA	Ratio of absorption chilling to total cooling load with solar energy
PC	Ratio of compression chilling to total cooling load with solar energy
PAO	Ratio of absorption chilling to total cooling load without solar energy
PCO	Ratio of compression chilling to total cooling load without solar energy

c. TRACE BACK

Subroutine SOLUSE is called by: OPCOOL

and calls: No subroutines

SOLUSE

Initialize output variables

ESOLAR = Absolute value of the available solar heat

No solar equipment installed

Yes | No

No absorption cooling equipment installed

Yes | No

Solar heat storage tank temp. < minimum temperature for heating or available solar heat = 0

Yes | No

Set solar heat used for heating to present heating load subject to an upper bound

Yes | No cooling present | No

Remaining available solar heat = available solar heat - solar heat used for heating
Remaining available solar heat < 0 or solar heat storage tank temperature < minimum temperature for absorption cooling

Yes | No

Set free-of-load absorption cooling capacity to: the total absorption cooling capacity - the absorption cooling capacity to satisfy its assigned share of the present cooling load without solar heat

Set SA = average ratio of heat energy in to cooling energy out for absorption chiller = 1.5

Heat energy input required = free-of-load absorption cooling capacity*SA

Set solar heat used for cooling = heat input required for absorption cooling

Set cooling energy provided by solar = solar heat used for cooling/SA

Set the absorption fraction of total cooling load with solar to: the absorption fraction of total cooling load without solar + cooling energy provided by solar/ present cooling load, (PAO+SOLCOOL/ECOOL)

The compression fraction of total cooling load with solar = PAO+PCO-the absorption fraction of total cooling load with solar

STATIS

a. GENERAL DESCRIPTION

This subroutine calculates monthly statistics for energy-related input and output variables.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
/HOURTOT/	EUT	Total utility electrical power (kW)
/HOURTOT/	ENTOT	Total power required by plant and utility (kW)
/HOURTOT/	EFUELB	Total boiler fuel consumption rate (kW)
/HOURTOT/	EFUELD	Total diesel fuel consumption rate (kW)
/HOURTOT/	EFUELG	Total gas turbine fuel consumption (kW)
/HOURTOT/	EHEATT	Total heat power (kW)
/HOURTOT/	ECOOL	Total cooling load (kW)
/HOURTOT/	ERECOVR	Rate of total recovered heat energy (kW)
/HOURTOT/	EWASTED	Total wasted energy (kW)
/HOURTOT/	EFIHC	Total rate of fuel input for heat energy consumed by cooling stage (kW)
/HOURTOT/	EFIEC	Total rate of fuel input for electrical energy consumed by cooling stage (kW)
/HOURTOT/	EFUELIH	Total rate of fuel input for heat energy generation (kW)

Source of Data	Name	Description
/HOURTOT/	EFUELIE	Total rate of fuel input for electricity generation by the plant and utility (kW)
/HOURTOT/	EFUEL	Total rate of fuel energy input, including to utility (kW)
/LDISTD/	NOPR(I)	Number of different sizes of equipment operating at current time step
/EPARS/	PLOAD(I)	Part load of equipment I for current hour (kW)
/STATD/	PLOADY(I)	Part load totaled over the year for equipment I (kWh)
/EPARS/	OPCAP(I)	Operating capacity of equipment I (kW)
/STATD/	OPCAPY(I)	Operating capacity totaled over the year for equipment I (kWh)
/LDISTD/	IOPR(J,I)	Number of operating hours of equipment I, size J (hr)
/STATD/	IOPRHR(J,I)	Number of operating hours of equipment I, size J (hr)
/DATE/	IMON	Month index
/DATE/	IDAY	Day index
/DATE/	IHR	Hour index

2. COMMON BLOCKS

DATE, EDATA, EPARS, HOURTOT, LDISTD, MONTOT, STATD

3. OUTPUT DATA

Name	Description
EUTS ENTOTS EFUELBS EFUELDS EFUELGS EHEATTS EELECTS ECOOLS ERECOVS EWASTES EFIHCS EFIECS EFUELHS EFUELES EFUELS	Monthly total of energy-related variables as described in Input Data. Also see descriptions in STATSM
ENPEAK(IMON,I)	Monthly energy peak for utility index I (kWh)
AMAXLD(I)	Maximum load of equipment I (kWh)
MAXTIM(1,I)	Month index for the time when equipment I reaches its maximum part load
MAXTIM (2,I)	Day index for the time when equipment I reaches its maximum part load
MAXTIM(3,I)	Hour index for the time when equipment I reaches its maximum part load

c. TRACE BACK

Subroutine STATIS is called by: SIMTEP

and calls: No subroutines

STATPR (UNIT)

a. GENERAL DESCRIPTION

This is a subroutine to print equipment use statistics.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
CDPR	IPRT	Logical unit number for print output
EDATA	NEQSIZE(I)	Same as described in SIMTEP
STATD	PLOADY (I)	Part load totaled over the year of equipment type I

2. COMMON BLOCKS

CDPR, EDATA, STATD

3. OUTPUT DATA

Name	Description
AVGOPR	Same as described in SIMTEP
OPCAPY (I)	Same as described in SIMTEP
IENAME (J,I)	Same as described in SIMTEP
AMAXLD (I)	Same as described in SIMTEP
MAXTIM (J,I)	Same as described in SIMTEP
IOPRHR (J,I)	Same as described in SIMTEP
CNOM (J,I)	Same as described in SIMTEP

c. TRACE BACK

Subroutine STATPR is called by: OUTRP

and calls: RTPRNT

STATPR

If units in SI units

No

Set output headings in English units

Yes

Set output headings in SI units

Print title for equipment use statistics

Do through whole equipment range NEDATA

If units in English units

No

Continue

Yes

Convert maximum part load into English units

Convert operating capacity into English units

No equipment present

Yes

CONTINUE

No

Total operating capacity = 0

Yes

CONTINUE

No

Average operating ratio = (total yearly part load/(total yearly operating capacity

Print equipment name

Print equipment operation ratio

Print maximum load

Print monthly maximum part load

Print nominal operating capacity

Print number of operating hours

Print report titles (RTPRNT)

STATSM

a. GENERAL DESCRIPTION

This subroutine stores energy-related monthly results.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
/MONTOT/	ENTOTS	Monthly rate of total power required by the plant and utility (kWh)
/MONTOT/	EHEATTS	Monthly total heat energy (kWh)
/MONTOT/	EELECTS	Monthly total electrical energy demanded including that used by the plant (kWh)
/MONTOT/	ECOOLS	Monthly cooling load (kWh)
/MONTOT/	ERECOVS	Monthly total recovered heat energy (kWh)
/MONTOT/	EWASTES	Monthly total wasted heat (kWh)

Source of Data	Name	Description
/MONTOT/	EFIHCS	Monthly fuel energy input for heat energy consumed by cooling stage (kWh)
/MONTOT/	EFIECS	Monthly fuel energy input for electrical energy consumed by cooling stage (kWh)
/MONTOT/	EFUELHS	Monthly total input for heat energy generation (kWh)
/MONTOT/	EFUELES	Monthly total input for electrical energy generation (kWh)
/MONTOT/	EFUELS	Monthly fuel energy input including to utility (kWh)

2. COMMON BLOCKS

DATE, HOURTOT, MONTOT, PRNTAI, STATD

3. OUTPUT DATA

Name	Description
PRNTA1(1,IMON)	Total heat energy (kWh)
PRNTA1(2,IMON)	Total electrical energy (kWh)
PRNTA1(3,IMON)	Cooling electrical energy (kWh)
PRNTA1(4,IMON)	Recovered energy (kWh)
PRNTA1(5,IMON)	Wasted recovered energy (kWh)
PRNTA1(6,IMON)	Heat energy input for cooling (kWh)
PRNTA1(7,IMON)	Electrical energy input for cooling (kWh)
PRNTA1(8,IMON)	Energy input for heating (kWh)
PRNTA1(9,IMON)	Energy input for electricity (kWh)
PRNTA1(10,IMON)	Total fuel energy input (kWh)
PRNTA1(11,IMON)	Total energy input (kWh)
PRNTA1(12,IMON)	Average plant efficiency
ENUSE (IMON,I)	Monthly energy used for utility index I (kWh)
ENPEAK(13,I)	Monthly energy peak for utility index I (kWh)

c. TRACE BACK

Subroutine STATSM is called by: SIMTEP

and calls: No subroutines

STMTUR (EELEC, PEXSTM, RPM)

a. GENERAL DESCRIPTION

Subroutine STMTUR calculates the hourly steam rate consumed by steam turbines under the given hourly load conditions. The equations used in this subroutine are taken from SSTUR of SYSSIM in the CERL Thermal Loads Analysis and Systems Simulation Program (see CERL Interim Report E-81).

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine STMTUR is called by: SIMTEP

OPCOOL

and calls: No subroutines

STMUSE (ESTUSE, RWTR)

a. GENERAL DESCRIPTION

This is a subroutine to calculate total steam consumption of steam users (including steam turbine excluding space heating and absorption chiller). Each steam user is given by:

1. Ratio of return water to steam flow input
2. Return water temperature

The energy consumed by the steam user equals the energy of the portion of the steam which is not returned plus energy loss of the portion which is returned to the boiler.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
CCBTEPS	ESTUSE	Total steam energy load of steam users (kW)
CCBTEPS	RWTR	Ratio of return water to steam flow
STMTUR	FSTMTUR	Flow of steam entering to steam turbine (kg/sec)
STMTUR	HSTMTUR	Enthalphy of superheated high-pressure steam (kJ/kg)
STMTUR	TEXSTM	Temperature of exhaust steam (°C)
SDATA (1,1)	HSTEAM	Steam enthalpy (kJ/kg)
SDATA (1,36)	RWSTUR	Ratio of exhaust steam to steam turbine entering flow
EPARS	TOTCAP (4)	Total capacity of boiler (kW)

EPARS	OPCAP(17)	Operating capacity of steam turbine (kW)

2. COMMON BLOCKS

EPARS, SDATA, STM, STMTUR

3. OUTPUT DATA

Name	Description
ESTMS	Total team energy (kW)
FSTMS	Total steam flow (kg/hr)
EWTRM	Energy of return water mixture (kW)
FWTRM	Flow of return water mixture (kg/hr)

c. TRACE BACK

Subroutine STMUSE is called by: SIMTEP

and calls: No subroutines

STMUSE

Initialize output variables

No steam boiler installed

Yes | No

Steam turbine operating capacity ≤ 0

Yes | No

Steam flow = steam flow to turbine
Steam energy = (steam flow to turbine) *(enthalpy of superheated steam)

Return condensate flow = (exhaust steam to entering steam flow ratio)* (entering steam flow)
Return condensate energy = (return condensate flow) *(enthalpy of condensate at exhaust steam temperature)

Flow to steam users = (total steam load) /(steam enthalpy - enthalpy of return condensate)

Energy of steam to users = (flow to steam users)*(steam enthalpy)

Return condensate flow = (return water to steam flow ratio)*(steam flow)
Return condensate energy = (return condensate flow)* (enthalpy of water @ 100°C)

Total steam energy = energy of turbine steam + energy of steam to users

Repeat addition for total steam flow, total return condensate energy, and total return condensate flow

STURDS

a. GENERAL DESCRIPTION

Subroutine STURDS calculates the steam rate of different sizes of single-stage condensing steam turbines under full-load conditions. The number of turbine sizes which can be considered has an upper limit of six. The algorithms used for developing this subroutine are similar to those used in SSTUR of SYSSIM which were derived from the performance curves presented in Carrier Corporation Bulletin H-31E[24] for Type YR single-stage turbines, which range in size from 596 kW to 5220 kW and range in speeds from 1750 to 6000 rpm. The full-load steam rate needs to be calculated only once and is passed to STMTUR for calculating hourly steam rate.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine STURDS is called by: DFLTASG

and calls: SUPT

SATUR

[24]Bulletin H-31E (Elliott Division, Carrier Corp.).

SUPT (T, TSR)

a. GENERAL DESCRIPTION

Function SUPT calculates the steam turbine superheat correction factor given the theoretical steam rate and superheat degrees. The equations used in this function are obtained by least square curve fitting of the performance curves presented in Carrier Corporation Bulletin H-31E.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function SUPT is called by: STURDS

and calls: No subroutines

TOWER (ETOWER, EWASTED, TENT, EELEC, TCOLD, PLOADT)

a. GENERAL DESCRIPTION

This subroutine simulates the cooling tower. Two types of towers--conventional and ceramic--are included. The towers can be operated under variable water rate, variable range, or fixed water rate, fixed range conditions.

Special parameters which are used in this subroutine are as follows:

RWCA = (tower water flow rate (kgPS))/(absorption chiller capacity (kW))

RWCC = (tower water flow rate (kgPS))/(compression chiller capacity (kW))

RWCDB = (tower water flow rate (kgPS))/(double-bundle chiller capacity (kW))

PELTWR = (electrical energy input)/(tower nominal cooling capacity)

PELTWR is used for approximating consumption only if TOWER is not specific in input.

The tower leaving-water temperature and the number of cells in operation are calculated by an iterative process using a 1.11°C increment. At each iteration, the tower rating factor is recalculated by calling RFACT.

b. DATA DESCRIPTION

1. INPUT DATA

Source of Data	Name	Description
SIMTEP	HSTEAM	Steam enthalpy (kJ/kg)
SIMTEP	ETOWER	Tower load (kW)
SIMTEP	EWASTED	Wasted recoverable heat (added to tower load) (kW)
SIMTEP	TENT	Entering water temperature (°C)
TOWERD	KT	Number of cells
TOWERD	PNTK	Fan motor power for one cell (kW)
TOWERD	CNT	Cooling capacity at 90-80-70 point (kW)
TOWERD	CNTU	Cooling capacity of one tower cell, in TU = CNT*5000
TOWERD	CNTH	Cooling capacity of one cell at half speed (kW)
TOWERD	CNTUH	Cooling capacity of one cell at half speed, in TU = CNTH*5000
EPARS	IABSOR	Absorption chiller type, same as in ABSREF
EPARS	ICOMPR	Compression chiller type, same as in COMREF
EPARS	ITOWR	Cooling tower type: 14 for conventional cooling tower 15 for ceramic cooling tower

Source of Data	Name	Description
EPARS	OPCAP(IABSOR	Operating capacity of absorption chiller (kW)
EPARS	OPCAP(ICOMPR)	Operating capacity of compression chiller (kW)
EPARS	OPCAP(13)	Operating capacity of double-bundle chiller (kW)
EPARS	OPCAP(ITOWR)	Operating capacity of cooling tower (kW)
WEATHR	TWET	Wet-bulb temperature (°C)
LDISTD	NOPR(ITOWR)	Number of tower units in operation
LDISTD	IOPR(1,ITOWR)	Number of tower cells in operation
SDATA(1,7)	TOWOPR	Type of tower operation: 1 for variable water rate, variable temperature range 2 for fixed water rate, variable temperature rate
SDATA(1,12)	TTOWER	Lower bound for temperature of leaving water (°C)
SDATA(1,25)	RWCA	Tower water flow rate/absorption chiller capacity
SDATA(1,26)	RWCC	Tower water flow rate/compression chiller capacity
SDATA(1,27)	RWCDB	Tower water flow rate/double-bundle chiller capacity
SDATA(1,6)	PELTWR	Electrical input to tower/tower cooling load

2. COMMON BLOCKS

EDATA, EPARS, LDISTD, SDATA, TOWERD, WEATHR

3. OUTPUT DATA

Name	Description
EELEC	Required electric energy (kW)
TCOLD	Leaving water temperature (°C)
PLOADT	Amount of load tower is handling (kW)
NCELL	Number of tower cells in operation

c. TRACE BACK

Subroutine TOWER is called by: SIMTEP

and calls: RFACT

TOWER

Initialize output variables

Tower Load = 0

RETURN

Set leaving water temperature = wet-bulb temperature + 1.11 (°C)

Total capacity of cooling tower = 0

Yes | No

- Set electrical input = 0
- Tower operating at fixed water rate
  - Yes: Calculate flow rate and temperature drop through cooling tower at fixed water rate
  - No: Calculate temperature drop and flow rate through cooling tower at variable water rate
- Calculate wet-bulb air temperature
- (leaving water temp wet-bulb temp)<2.78°C
  - Yes
  - No:
    - Calculate rating factor (from RFACT)
    - Calculate rated area of tower
    - Set number of tower cells operating = 1
      - Initialize intermediate variable
      - If cooling capacity of one cell operating at half speed CNTUH>0, recalculate the electrical input
      - Capacity required ≤ cooling capacity provided
        - Yes
        - No:
          - Calculate electrical input
          - Capacity required = cooling cap of 1 cell
            - Yes
            - No
          - Cells in operation = total no. of tower cells
            - Yes
            - No
      - Go OUT
      - OPCAP = TRUE
      - Decrease the capacity required by the capacity provided by NCELL no. of cells
    - Increase cell unit by one; repeat until OUT = TRUE
- Increase water temperature by 1.11°C
- Repeat until temperature drop through cooling tower ≤ 0
- <ORCAP> calculate operating capacity
- Set cooling load in water = operating capacity
- Set required electrical energy = input electrical energy
- Tower operating at fixed water rate
  - Yes: Recalculate required electrical energy at fixed water rate
  - No: If temperature drop through cooling tower >0, recalculate required electrical energy at variable water rate

13 SHARED ROUTINES

The following routines are engineering routines used variously in SIMBLD, SIMSYS, and SIMTEP.

PSYDPT (TDB, TWB, PB)

a. GENERAL DESCRIPTION

Function PSYDPT calculates the dewpoint temperature from the given dry-bulb temperature, wet-bulb temperature, and barometric pressure. This function calls PSYWTP to calculate humidity ratio and PSYDPW to calculate dewpoint temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYDPT is called by: FNCLDS

and calls: PSYDPW (W, PB)

PSYWTP (TDB, TWB, PB)

PSYDPW (W, PB)

a. GENERAL DESCRIPTION

Function PSYDPW calculates the dewpoint temperature from the given humidity ratio and barometric pressure. The equation for calculating dewpoint saturation pressure is derived from the equation given in *ASHRAE Handbook of Fundamentals*.[25] Function SATUTP is called for calculating the dewpoint temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYDPW is called by: BOUND

COIL1

FNCLHR

PSYDPT

PSYRTW

and calls: SATUTP (P)

[25] *ASHRAE Handbook of Fundamentals* (ASHRAE, 1972).

AD-A048 982 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAI--ETC F/G 13/1
THE BUILDING LOADS ANALYSIS AND SYSTEM THERMO-DYNAMICS (BLAST) --ETC(U)
DEC 77 D C HITTLE

UNCLASSIFIED CERL-TR-E-119-VOL-2 CEEDO-TR-77-35-VOL-2 NL

5 OF 5

ADA048982

END

DATE FILMED

2-78

DDC

PSYHTW (TDB, W)

a. GENERAL DESCRIPTION

Function PSYHTW calculates enthalpy from the given dry-bulb temperature and humidity ratio. The equation used in this function for calculating the enthalpy is derived from that given in *ASHRAE Handbook of Fundamentals*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYHTW is called by: BOUND
CCOIL
COIL1
ETECY
FNCLHR
PSYTWD
SATUTH
THS2

and calls: No subroutines

PSYRHT (TDB, TDP)

a. GENERAL DESCRIPTION

Function PSYRHT calculates relative humidity from the given dry-bulb temperature and dewpoint temperature. It calls SATUPT to calculate dewpoint saturation pressure and dry-bulb saturation pressure. The relative humidity then is computed as the ratio of those two pressures using the equation given in the *ASHRAE Handbook of Fundamentals*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYRHT is called by: PSYRTW

and calls: SATUPT (T)

PSYRTW (TDB, W, PB)

a. GENERAL DESCRIPTION

Function PSYRTW calculates the relative humidity from the given dry-bulb temperature, humidity ratio, and barometric pressure. It calls PSYDPW to calculate dewpoint temperature and PSYRHT to calculate relative humidity.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYRTW is not called by any routines in BLAST, but is available should the user require it.

It calls: PSYDPW (W, PB)

PSYRHT (TDB, TDP)

PSYTWD (TDB, W, PB)

a. GENERAL DESCRIPTION

Function PSYTWD calculates wet-bulb temperature from the given dry-bulb temperature, humidity ratio, and barometric pressure. An iterative method is used in this function to calculate the wet-bulb temperature. The iterations are limited to 30. If the desired result has not been obtained after 30 iterations, the wet-bulb temperature is set equal to the saturation enthalpy temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYTWD is called by: SIMTEP
SUDDEN

and calls: PSYHTW (TDB, W)
PSYWTP (TDB, TWB, PB)
SATUTH (H, PB)

PSYVTW (TDB, W, PB)

a. GENERAL DESCRIPTION

Function PSYVTW calculates specific volume from the given dry-bulb temperature, humidity ratio, and barometric ratio. The equation for calculating the specific volume in this function is derived from the equation given in the *ASHRAE Handbook of Fundamentals*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYVTW is not called by any routines in BLAST, but is available should the user require it. It calls no subroutines.

PSYWDP (TDP, PB)

a. GENERAL DESCRIPTION

Function PSYWDP calculates humidity ratio from the given dewpoint temperature and barometric pressure. This function calls SATUPT to calculate dewpoint saturation pressure and then uses an equation derived from the *ASHRAE Handbook of Fundamentals* to calculate the humidity ratio.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYWDP is not called by any routines in BLAST, but is available should the user require it. It calls: SATUPT (T).

PSYWTH (TDB, H)

a. GENERAL DESCRIPTION

Function PSYWTH calculates humidity ratio from the given dry-bulb temperature and enthalpy. The equation used in this function for calculating the humidity ratio is derived from the equation given in the *ASHRAE Handbook of Fundamentals*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYWTH is called by: CCOIL

FNCLHR

and calls: No subroutines

PSYWTP (TDB, TWB, PB)

a. GENERAL DESCRIPTION

Function PSYWTP calculates humidity ratio from the given dry-bulb temperature, wet-bulb temperature, and barometric pressure. In this function, the wet-bulb saturation pressure is first calculated by SATUPT. The humidity ratio is then calculated using the equations derived from the *ASHRAE Handbook of Fundamentals*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function PSYWTP is called by: BOUND

ILFBWH

SUDDEN

CCOIL

COIL1

PSYDPT

PSYTWD

SATUTH

THS2

and calls: SATUPT (TWB)

PSYWTR (TDB, RH, PB)

a. GENERAL DESCRIPTION

Function PSYWTR calculates humidity ratio from the given dry-bulb temperature, relative humidity, and barometric pressure. This function calls SATUPT to calculate dry-bulb saturation pressure. The equations used for calculating the humidity ratio are derived from the *ASHRAE Handbook of Fundamentals*.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACEBACK

Function PSYWTR is called by: AHSIZE

HUMID

ZNMRQ

and calls: SATUPT(T)

SATUPT (T)

a. GENERAL DESCRIPTION

Function SATUPT calculates saturation pressure at a given temperature. The equations used in this function are obtained by least square curve fitting. The data for curve fitting are derived from ASME Steam Tables[26] and *ASHRAE Handbook of Fundamentals*. The temperature range for this function is from −50° to 100°C. Outside of this range the pressure is still computed by this function, but the accuracy will no longer be within the 1 percent limit.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function SATUPT is called by: PSYRHT

PSYWDP

PSYWTP

PSYWTR

and calls: No subroutines

[26] *ASME Steam Tables* (American Society of Mechanical Engineers, 1967).

SATUR (PSTEAM)

a. GENERAL DESCRIPTION

SATUR calculates the saturation temperature as a function of steam pressure.

b. DATA DESCRIPTION

See listing for common blocks and variable names and descriptions.

c. TRACE BACK

Function SATUR is called by: DFLTASG
EFFIC
ENTROP
STURDS

and calls: No subroutines

SATUTH (H, PB)

a. GENERAL DESCRIPTION

Function SATUTH calculates saturation temperature from the given enthalpy and barometric pressure. If the difference between the given pressure and 101 330 N/m^2 is within 1 percent, the temperature is calculated by a set of equations which are obtained by least square curve fitting with a temperature range from -40° to 90°C. Otherwise, the temperature is computed by an iterative method. The maximum iterations are 30. If the desired result has not been obtained after 30 iterations, the temperature is set equal to the temperature calculated by the above curve-fitted equations.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function SATUTH is called by: PSYTWD

THS2

and calls: PSYHTW (TDB, W)

PSYWTP (TDB, TWB, PB)

SATUTP (P)

a. GENERAL DESCRIPTION

Function SATUTP calculates saturation temperature at a given pressure. The equations in this function are obtained by least square curve fitting. The data used for curve fitting are computed by SATUPT. The pressure range for those equations is from 3.93 to 1.0133×10^5 N/m^2. If the given pressure is out of this range, temperature is still calculated by this function, but the accuracy is no longer within 1 percent.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function SATUTP is called by: PSYDPW

and calls: No subroutines

14 WEATHER INFORMATION FILE ENCODER

This chapter describes the Weather Information File Encoder (WIFE) program and its subroutines. Figure 17 shows the WIFE tree structure.

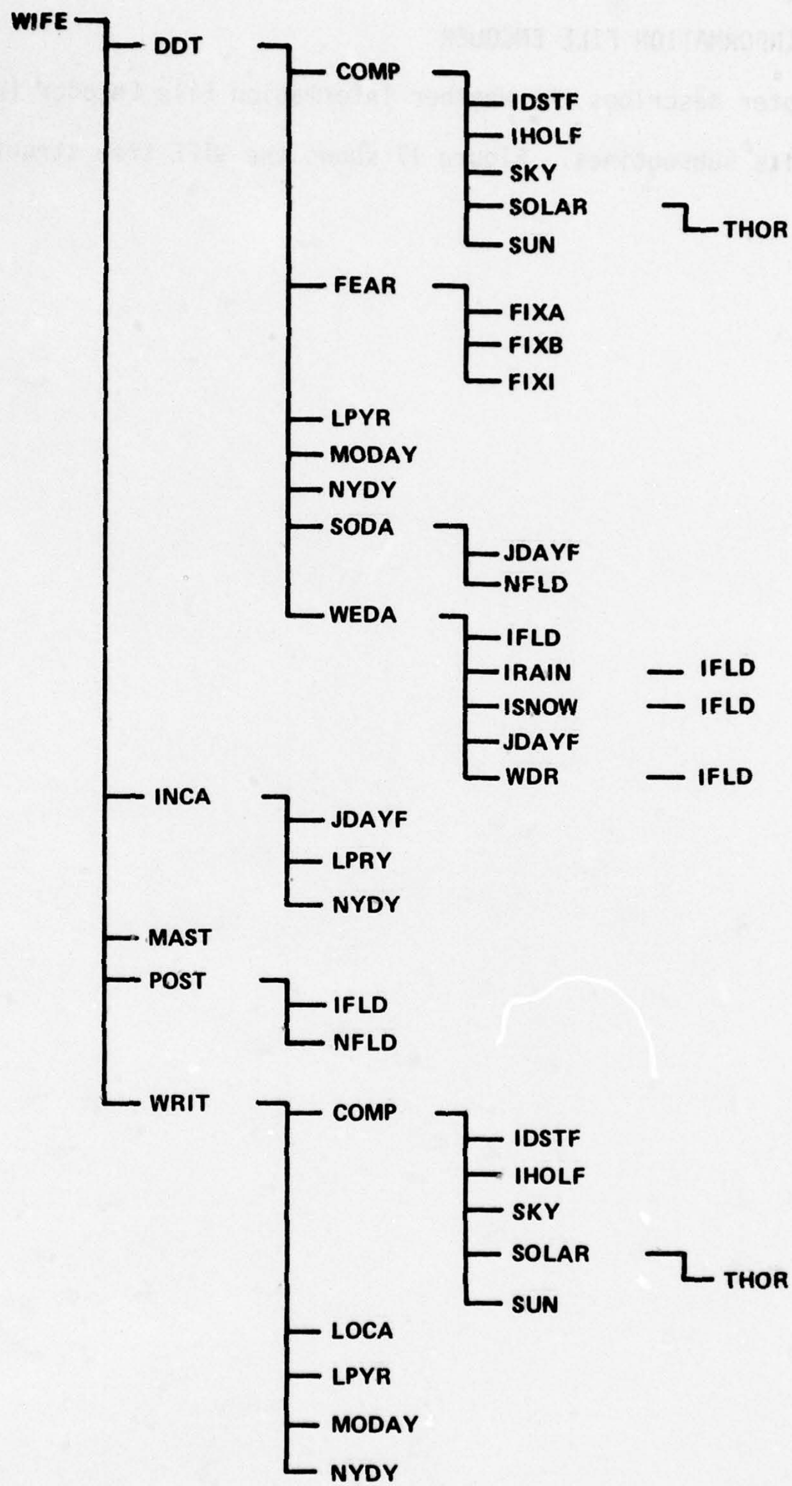


Figure 17. Tree structure of WIFE subroutines.

WIFE

a. GENERAL DESCRIPTION

WIFE is a program which creates and encodes weather data based on data taken from the 1440 weather and 280 solar radiation tapes available from the National Weather Records Center.

Input to WIFE consists of the 1440 weather tape labeled TAPE1, the 280 solar radiation tape labeled TAPE3 (when available), and three input cards. The first of these cards is an 80-character alphanumeric run identifier or description, which is the user's choice. The second card contains the station latitude, longitude, time zone, weather station number, solar radiation station number (0 indicates no solar tape), start year, start month, start day, number of days, and report type indicator; the entries are separated by blanks. The third input card contains the monthly Celsius ground temperatures for the location from January to December separated by blanks.

Output to WIFE consists of a clean weather file labeled TAPE2, a list of missing weather days, and the choice of two output reports. If the report type indicator is 1, a daily report is designated which prints all the data on the tape, one day per page. Any other value will result in printing of a summary report containing one line per day; each line will contain high, low, and mean temperatures, heating and cooling degree days, and total radiation. One month is printed per page. Included are monthly and total tape summaries.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Program WIFE calls: DDT
INCA
MAST
POST
WRIT

COMP

a. GENERAL DESCRIPTION

Subroutine COMP computes the remainder of the data elements needed for a record from existing data. These data consist of the day of the week, daylight savings and holiday indicators, beam and diffuse solar radiation values, sky temperature, and ground temperature.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine COMP is called by: DDT

WRIT

and calls: IDSTF

IHOLF

SKY

SOLAR

SUN

DDT

a. GENERAL DESCRIPTION

Subroutine DDT decodes data and derives replacement data for bad data from the weather and solar tapes. The data are translated into SI units, structured into final record form, and written to a random access (mass storage) file for later use.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine DDT is called by: WIFE

and calls: COMP

FEAR

LPYR

MODAY

NYDY

SODA

WEDA

FEAR

a. GENERAL DESCRIPTION

Subroutine FEAR finds those data elements whose values are not within a reasonable range. Those elements are then replaced with data based on surrounding values. Eight consecutive bad values result in the classification of a bad day; the day will be replaced later in program WIFE.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FEAR is called by: DDT

and calls: FIXA

FIXB

FIXI

FIXA

a. GENERAL DESCRIPTION

Subroutine FIXA replaces missing data after good data have been found based on a trigonometric curve fit of previous good data points.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FIXA is called by: FEAR

and calls: No subroutines

FIXB

a. GENERAL DESCRIPTION

Subroutine FIXB replaces data missing at the beginning of a record by using a trigonometric curve fit based on good data found later in the record.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FIXB is called by: FEAR

and calls: No subroutines

FIXI

a. GENERAL DESCRIPTION

Subroutine FIXI replaces data missing between existing data elements by using a trigonometric curve fit based on existing points.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine FIXI is called by: FEAR

and calls: No subroutines

IDSTF

a. GENERAL DESCRIPTION

Function IDSTF checks to see if the present day is between the last Sunday in April and the last Sunday in October. If so, daylight savings time is in effect and the function returns 1. If not, 0 is returned.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function IDSTF is called by: COMP

and calls: No subroutines

IFLD

a. GENERAL DESCRIPTION

Function IFLD converts fields from the 1440 weather tape presently in character format into integer format.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function IFLD is called by: IRAIN

ISNOW

POST

WDR

WEDA

and calls: No subroutines

IHOLF

a. GENERAL DESCRIPTION

Function IHOLF checks to see if the present day is a holiday. If it is, a 1 is returned; if not, 0 is returned.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function IHOLF is called by: COMP

and calls: No subroutines

INCA

a. GENERAL DESCRIPTION

Subroutine INCA reads the necessary input card data and converts this data into a usable form for the main routine, WIFE.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine INCA is called by: WIFE

and calls: JDAYF

LPYR

NYDY

IRAIN

a. GENERAL DESCRIPTION

Function IRAIN checks the 1440 weather data for rain information. If rain is present, 1 is returned. If not, 0 is returned.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function IRAIN is called by: WEDA

and calls: IFLD

ISNOW

a. GENERAL DESCRIPTION

Function ISNOW checks the 1440 weather tape for the presence of a medium of heavy snowfall. If after a snowfall temperatures remain at or below freezing, the snow indicator is left on. The return of a 1 indicates snow is present. The return of a 0 indicates it is not.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function ISNOW is called by: WEDA

and calls: IFLD

JDAYF

a. GENERAL DESCRIPTION

Function JDAYF calculates the day of the year (Julian date) given the month, the day of the month, and the leap year indicator.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function JDAYF is called by: INCA
SODA
WEDA

and calls: No subroutines

LOCA

a. GENERAL DESCRIPTION

Function LOCA locates the record number of the best replacement for a bad (missing) day's data. The replacement must be within 15 days of the missing day. A table of dry-bulb temperatures is used for comparison purposes. If more than one "best" replacement day is found, the day closest to the missing one is chosen. If two days are equivalently close, the one previous to the missing record is used.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function LOCA is called by: WRIT

and calls: No subroutines

LPYR

a. GENERAL DESCRIPTION

Function LPYR calculates the leap year indicator. If the year is a leap year, 1 is returned; if not, 0 is returned. Although the function is set up for the general case, the weather tapes store only the last two digits of the year. This coupled with the fact that the other two checks are not necessary or cancel each other through the year 2100 has prompted use of only the check for years divisible by four for future versions of this code.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function LPYR is called by: DDT

INCA

WRIT

and calls: No subroutines

MAST

a. GENERAL DESCRIPTION

Subroutine MAST initializes the program's mass storage (random access) file, the file index, and the daily record status flag array.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine MAST is called by: WIFE

and calls: No subroutines

MODAY

a. GENERAL DESCRIPTION

Subroutine MODAY calculates the month and day, given a Julian date (day of the year) and the appropriate leap year indicator.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine MODAY is called by: DDT

WRIT

and calls: No subroutines

NFLD

a. GENERAL DESCRIPTION

Function NFLD converts fields of data read from the 280 solar radiation tape into integer format.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function NFLD is called by: POST

SODA

and calls: No subroutines

NYDY

a. GENERAL DESCRIPTION

Function NYDY calculates the day of the week of New Year's Day (1 = Sunday). Some of the checks and calculations can be omitted (as in LPYR), but the function has been left in general form.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function NYDY is called by: DDT

INCA

WRIT

and calls: No subroutines

POST

a. GENERAL DESCRIPTION

Subroutine POST positions the tapes for data reading after it checks the years and station numbers against those desired. Flags are set to describe the status of each tape.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine POST is called by: WIFE

and calls: IFLD

NFLD

SKY

a. GENERAL DESCRIPTION

Subroutine SKY calculates sky temperatures from a relationship between dry-bulb and dewpoint temperatures.[27]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SKY is called by: COMP

and calls: No subroutines

[27] For a detailed explanation of the relationship used in SKY, see Raymond W. Bliss, Jr., "Atmospheric Radiation Near the Surface of the Ground: A Summary for Engineers," *Solar Energy* (1961).

SODA

a. GENERAL DESCRIPTION

Subroutine SODA reads and decodes the total horizontal radiation data from the 280 solar radiation tape.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SODA is called by: DDT

and calls: JDAYF

NFLD

SOLAR

a. GENERAL DESCRIPTION

Subroutine SOLAR calculates the horizontal diffuse and normal beam radiation amounts. The total horizontal radiation read from the 280 solar radiation tape (when available) and ASHRAE data from the *ASHRAE Handbook of Fundamentals*[28] are used to separate horizontal and vertical components. If no solar data are available, simulated data are derived from cloud cover information from the 1440 weather tape.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SOLAR is called by: COMP

and calls: THOR

[28] *ASHRAE Handbook of Fundamentals* (ASHRAE, 1972).

SUN

a. GENERAL DESCRIPTION

Subroutine SUN computes coefficients for determining solar position and intensity. The expressions are based on least-square fits of data from Threlkheld[29] and the *ASHRAE Handbook of Fundamentals*.[30]

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine SUN is called by: COMP

and calls: *No subroutines*

[29]J.L. Threlkheld, *Thermal Environmental Engineering* (Prentice-Hall, 1970), p 316.

[30]*ASHRAE Handbook of Fundamentals* (ASHRAE, 1972), p 387.

THOR

a. GENERAL DESCRIPTION

Function THOR uses ASHRAE and cloud cover data from the 1440 weather tape to create a total horizontal radiation amount for replacement of bad or missing solar data.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function THOR is called by: SOLAR

and calls: No subroutines

WDR

a. GENERAL DESCRIPTION

Function WDR converts the two-digit wind direction field into a degree measurement.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Function WDR is called by: WEDA

and calls: IFLD

WEDA

a. GENERAL DESCRIPTION

Subroutine WEDA reads and decodes the weather data accessed from the 1440 weather tape and converts it to SI units.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine WEDA is called by: DDT

and calls: IFLD

IRAIN

ISNOW

JDAYF

WDR

WRIT

a. GENERAL DESCRIPTION

Subroutine WRIT writes the final weather tape and desired report. It replaces missing days with the most equivalent day (in terms of dry-bulb temperature) of good data found within 15 days of the missing day.

b. DATA DESCRIPTION

See listing of routine for common blocks and variable names and descriptions.

c. TRACE BACK

Subroutine WRIT is called by: WIFE

and calls: COMP

LOCA

LPYR

MODAY

NYDY

REFERENCES

Aho, A. V. and S. C. Johnson, "LR Parsing," *Computing Surveys*, Vol 6, No. 2 (June 1974), pp 99-124.

ARI Standard 410-72 (Air Conditioning and Refrigeration Institute, 1972).

ASHRAE Handbook of Fundamentals (American Society of Heating, Refrigerating, and Air Conditioning Engineers [ASHRAE], 1972).

ASHRAE Handbook, 1975 Equipment (ASHRAE, 1975).

ASME Steam Tables (American Society of Meahcnical Engineers, 1967).

Bliss, Raymond W., Jr., "Atmospheric Radiation Near the Surface of the Ground: A Summary for Engineers," *Solar Energy* (1961).

Born, M. and E. Wolf, *Principles of Optics* (Pergamon Press, 1959).

Bulletin H-31E (Elliott Division, Carrier Corp.).

Carrier Fan Coil Catalog (Carrier Corp.)

Conte, S. D. and C. deBoor, *Elementary Numerical Analysis* (McGraw-Hill, 1972).

CRC Standard Mathematical Tables, 22nd edition (Chemical Rubber Company, 1972).

Hittle, D. C., *The Building Loads Analysis and System Thermodynamics (BLAST) Program, Volume I: User Instructions*, CERL Technical Report E-119 and CEEDO-TR-77-35 (U.S. Army Construction Engineering Research Laboratory [CERL], and Air Force Civil and Environmental Engineering Development Office [CEEDO], 1977).

Hittle, D. C. and B. Sliwinski, *CERL Thermal Loads Analysis and Systems Simulation Program*, Volumes 1 and 2, Interim Report E-81 (CERL, 1975).

LaLonde, W. R., *An Efficient LALR Parser Generator*, CSRG-2 (University of Toronto, Computer Systems Research Group, 1971).

Mouen, E. D., *Application of the Thermal Response Factor Method to Building Elements with Air Cavities*, Ph. D. Thesis (University of Illinois, 1973.

NECAP Engineering Manual (National Aeronautics and Space Administration, 1974).

Newman, W. M., and R. F. Sproull, *The Principles of Interactive Graphics* (McGraw-Hill, 1973).

Procedure for Determining Heating and Cooling Loads for Computerized Energy Calculations (ASHRAE, 1975).

Procedures for Simulating the Performance of Components and Systems for Energy Calculations (ASHRAE, 1974).

Spiegel, M. R., *Theory and Problems of Complex Variables*, Schaum Outline Series (McGraw-Hill, 1964).

Threlkheld, J. L., *Thermal Environmental Engineering* (Prentice-Hall, 1970).

Tustin, A., "A Method of Analyzing the Behavior of Linear Systems in Terms of Time Series," *Journal of the Institution of Electrical Engineers*, Vol 94, Part II-A (1947).

INDEX OF ROUTINES

INDEX (cont'd)

INDEX (cont'd)

INDEX (Cont'd)

INDEX (cont'd)

CERL DISTRIBUTION

Chief of Engineers
ATTN: DAEN-MCE-U (2)
ATTN: DAEN-ASI-L (2)
ATTN: DAEN-RDL

US Army Engineer District
  New York
    ATTN: Chief, NANEN-E
    ATTN: Chief, Design Br.
  Baltimore
    ATTN: Chief, Engr Div
  Norfolk
    ATTN: Library
    ATTN: Chief, NAOEN-M
  Savannah
    ATTN: Library
    ATTN: Chief, SASAS-L
  Mobile
    ATTN: Chief, SAMEN-C
  Kansas City
    ATTN: Library (2)
    ATTN: Chief, Engr Div
  Omaha
    ATTN: Chief, Engr Div
  Fort Worth
    ATTN: Library
    ATTN: Chief, SWFED-D
    ATTN: Chief, SWFED-MA/MR
  Sacramento
    ATTN: Library, Room 8307
    ATTN: Chief, SPKED-D
  Far East
    ATTN: Chief, Engr Div
  Japan
    ATTN: Library
  Los Angeles
    ATTN: Library
  Alaska
    ATTN: Library
    ATTN: Chief, NPADE-R

Each Military Construction
  Division Office

Civil and Env. Engr. Dev. Office (6)
Air Force Systems Command
Tyndall AFB, FL 32401

FESA
Ft Belvoir, VA 22060

Defense Documentation Center (12)

Hittle, Douglas C
The building loads analysis and system thermodynamics (BLAST) program / by D. C. Hittle. -- Champaign, Ill. : Construction Engineering Research Laboratory ; Springfield, Va. : for sale by National Technical Information Service, 1977-   .
v. ; 27 cm. -- (Technical report - Construction Engineering Research Laboratory ; E-119)

Contents: v.1. Users manual. -- v.2. Program reference manual.

continued on next card.

Hittle, Douglas C
The building loads... (card 2)

1. Energy consumption. 2. BLAST (computer program). 3. Buildings - fuel consumption. I. U.S. Construction Engineering Research Laboratory. II. Title. III. Series: U.S. Construction Engineering Research Laboratory. Technical report ; E-119.