Contract NAS9-13033
DRL Number T-815
DRL Line Item 7
DRD Number DM-055TB
MCR-73-105

THERMAL RADIATION ANALYSIS SYSTEM

T R A S Y S

USER'S MANUAL

MAY 1973

Approved by:

Carl L. Jensen
Program Manager

MARTIN MARIETTA

Prepared for:

National Aeronautics and Space Administration
Lyndon B. Johnson Spacecraft Center
Contract NAS9-13033

Prepared by:

Carl L. Jensen

Richard G. Goble

Concurrence:

James H. Kidd
Chief, Thermophysics Section

FOREWORD

The Martin Marietta Thermal Radiation Analyzer System (TRASYS) program marks the first instance that thermal radiation analysis has been put on the same basis as thermal analysis using program systems such as MITAS and SINDA. As with these thermal analyzer programs, the user is provided the powerful options of writing his own executive, or driver logic and choosing, among several available options, the most desirable solution technique(s) for the problem at hand. In addition, many features never before available in a single radiation analysis program are provided.

Among the more important are:

- 1000 node problem size capability with shadowing by intervening opaque or semi-transparent surfaces;
- choice of diffuse, specular or diffuse/specular radiant interchange solutions;
- capability for time variant geometry in orbit;
- choice of analytically determined or externally supplied shadow data for environmental flux calculations;
- form factors and environmental fluxes computed using an internally-optimized number of surface grid elements, selected on the basis of user-supplied accuracy criteria;
- A general edit capability for updating thermal radiation model data stored on tape.
- A plot package that provides a pictorial representation of the user's geometry.

TRASYS is indebted to a number of predecessor programs in the thermal radiation analysis field. The major contributors were HEATRATE, MTRAP version 2.0, RADFAC and the MRI Computer Program for Determining External Radiation absorbed by the Apollo Spacecraft.

This User's Manual represents a concerted effort to document the capabilities of TRASYS and will, hopefully, serve the twofold purpose of instructing the user in all applications and serve as a convenient reference book that presents the features and capabilities in a concise, easy-to-find manner.

This User's Manual was generated under NASA Contract NAS9-13033, "Development of a Thermal Radiation Analysis/Heat Rate Computer Program System." The technical monitoring for this program was provided by Mr. Robert A. Vogt of the Thermal Technology Branch of the Structures and Mechanics Division, NASA Lyndon B. Johnson Space Center. His helpful suggestions during the development of TRASYS are gratefully acknowledged. TRASYS would not exist without the superb design and programming efforts of Messrs. R. E. Paulson and R. J. Connor, who were responsible for generating the majority of the TRASYS code. Their efforts are gratefully acknowledged. Extensive thanks are also due Mr. G. M. Holmstead for his efforts in developing the direct irradiation program segment and for the valuable consulting effort he performed during the course of program development Mr. R. G. Goble is also recognized for his praiseworthy efforts in developing the specular-diffuse radiation interchange segment, the orbit plotter segment, and for his solutions of many knotty problems that cropped up during program checkout.

CONTENTS

*In a separate volume.

Figure

Table

1. INTRODUCTION

1.1 WHAT IS TRASYS?

The Thermal Radiation Analysis System is a digital computer software system with generalized capability to solve the radiation related aspects of thermal analysis problems. When used in conjunction with a generalized thermal analysis program such as the Systems Improved Numerical Differencing Analyzer (SINDA) program, any thermal problem that can be expressed in terms of a lumped parameter R-C thermal network can be solved. The function of TRASYS is twofold. It provides:

a. Internode radiation interchange data; and
b. Incident and absorbed heat rate data from environmental radiant heat sources.

Data of both types is provided in a format directly usable by the thermal analyzer programs.

One of the primary features of TRASYS is that it allows the user to write his own executive or driver program which organizes and directs the program library routines toward solution of each specific problem in the most expeditious manner. The user also may write his own output routines, thus the system data output can directly interface with any thermal analyzer using the R-C network concept.

Other outstanding features of TRASYS include:

a. 1000 node allowable problem size.
b. Time variable problem geometry allowed.
c. Edit capability allowing the combination or separation of multiple thermal radiation models.
d. A plot package that provides pictorial plots of input geometry and orbit data as well as output data.

The TRASYS system consists of two major components: (1) the preprocessor, and (2) the processor library. The preprocessor has two major functions. First, it reads and converts the user's geometry input data into the form used by the processor library routines. Second, it accepts the users driving logic written in the TRASYS modified FORTRAN language that directs user-provided and/or library routines in the solution of the problem. The processor library consists of FORTRAN language routines that perform the functions commonly needed by the user. The user has, in some cases, a choice of solution techniques to perform the same function.

1.2 SYSTEM STRUCTURE

In the usual engineering environment, a programmer is commissioned to prepare an applications program which is subsequently made available to the engineer on a production basis. The engineer supplies input data and receives output data, as shown in Figure 1-1.

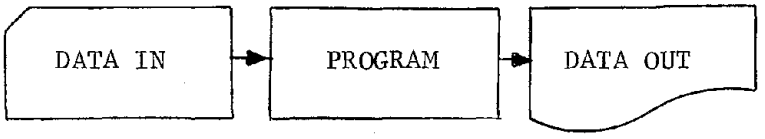


FIGURE 1-1: BASIC FLOW IN USING AN APPLICATIONS PROGRAM

Changes to the logic and equations are difficult for the program user to implement conveniently since they must be written in a computer-oriented language and submittal may be required through a formal programming organization. When TRASYS is used, however, the engineer need only call on the programmer to supply a standard deck of computer oriented "control cards" which will call the various elements of the system into action in the proper sequence. The engineer then formulates his problem in the engineering-oriented TRASYS

language, assembling both data and solution techniques (i.e., logic and equations) into this card deck, which then serves as the complete input to the TRASYS system. Programmer support has been minimized since the bulk of the programming effort is already built into the TRASYS preprocessor and processor library. The engineering user need only specify the data and the order and type of "program building blocks" which he deems necessary for the solution of his problem, as illustrated in Figure 1-2.

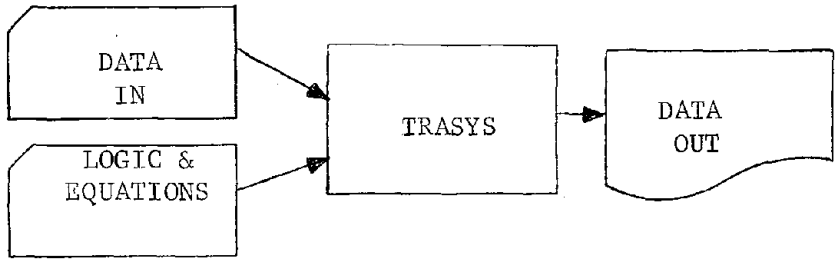


FIGURE 1-2: BASIC FLOW IN USING THE TRASYS

It should then be evident that TRASYS is much more than an applications program. It has, in fact, all of the functions and capabilities of a special purpose operating system. Since most computers in current use in engineering environments already have operating systems built around a FORTRAN compiler, TRASYS is designed to augment the existing FORTRAN system. Hence, the TRASYS library serves as an extension to the existing FORTRAN library, and the TRASYS program serves as a preprocessor to (i.e., it preceeds) the existing FORTRAN compiler. This augmentation arrangement is illustrated in Figure 1-3.

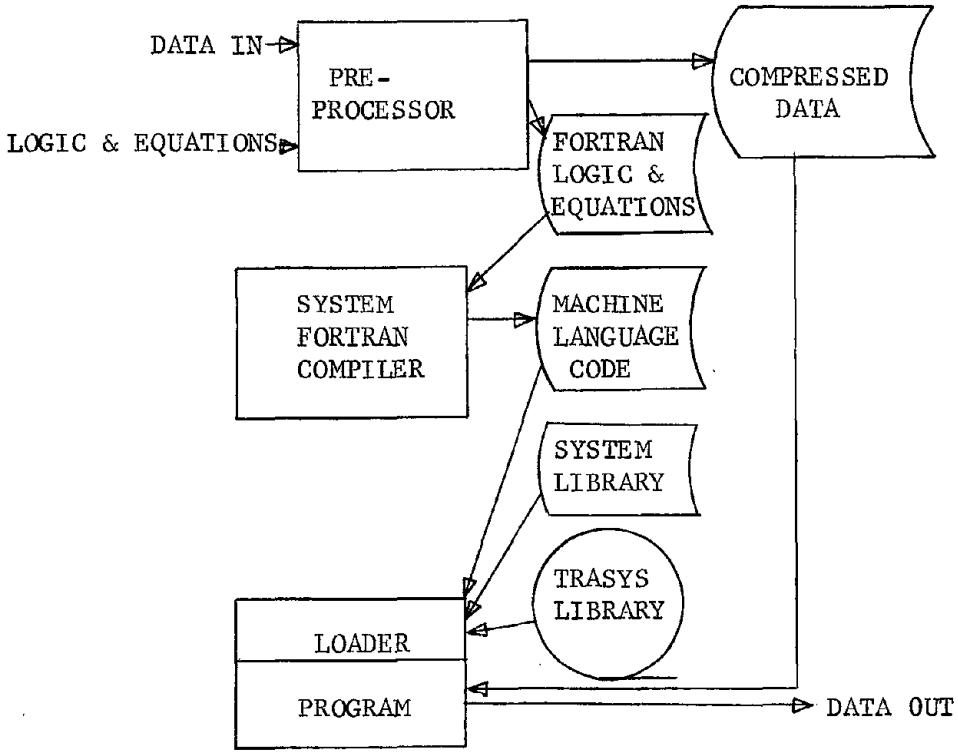


FIGURE 1-3: DETAILED INTERNAL FLOW OF TRASYS

When using the full capability of TRASYS, the engineer will be required to exert a programming effort of sorts, in a language consisting of FORTRAN statements and problem oriented TRASYS statements that are FORTRAN related. This, together with the wide variety of options and features offered by the system, suggests an appropriate word of caution: TRASYS is a comprehensive system which cannot be mastered overnight. The prospective user should not assume that a cursory review of the Instruction Manual will lead to immediate success, nor should he assume that this manual represents a "cookbook" which will eventually yield to a plodding and rigid adherence to each and every rule. In presenting instructions on the use of a computer program, it is not possible to completely avoid some "cookbook-like" sections;

however, every effort has been made to explain the "why" and "how" behind each rule, option, and feature, with the intent of encouraging the reader to think about and understand TRASYS in depth. To help the novice user, an attempt has been made to default much of the required input to normally used values so that the user need not define them.

2. BACKGROUND INFORMATION

2.1 TYPOGRAPHICAL CONVENTIONS

2.1.1 Punched Cards

The reader is (or soon will be) familiar with the standard 80 column punched card. It is the user's primary means of conveying his input data and logic to TRASYS.

The program input format design is predicated on minimum dependence upon data/card column relationships. Most card input is covered by one column rule: card columns 1 thru 6 inclusive comprise the control field and columns 7 through 72 comprise the data field. Data in the control field are used by read routines to identify the type of data to expect in the cards' data field. In this manual, the typographical convention shown in Figure 2.1 will be used to indicate the card columns of interest. (Card columns 1,7 and 12 in this case).

```
CC1        CC7        CC1
                        2
```

FIGURE 2-1: SAMPLE CARD COLUMN DESIGNATIONS

Throughout the rest of the manual (in contrast to Figure 2.1) punched cards will not be identified as figures. Whenever material is presented with one or more indicators with the format CCX directly above, a punched card is indicated. The card data will always be presented as Gothic capitals and/or numerals. For example, a card format might be shown as follows:

```
CC1        CC7                              CC7
  TITLE      THIS IS A SAMPLE TITLE CARD     3 0012
```

In general, the character directly below the column number begins the relevant data field.

2.2 FILE AND TAPE CONVENTIONS

Since TRASYS can be implemented on a variety of computers, it is necessary to refer to data storage media by some nomenclature which will be independent of the particular system

configuration. FORTRAN "logical unit numbers" are often used for this purpose, but were rejected for use in this manual because certain installations impose restrictions on the type of physical storage device which may be assigned to a given unit. Instead, each serial access storage device referenced by TRASYS is given a proper name as follows:

"Purpose TAPE"

Hence, for example, the processed Data Tape contains the result of processing the user's data, and Edit Input Tape contains input for the Edit routine. "Tape" is used as part of the name only because a reel of magnetic tape is normally associated with computer storage. However, any "Tape" may, in fact, be a disk file, a drum file, a punched paper tape, or a magnetic tape, at the option of the user. Appendix G contains a list of the system-oriented unit designations for each of the "Tapes" mentioned in this manual, along with the recommended type of storage device to which these units should be assigned.

On the other hand, when speaking in general about saving or retrieving data on or from a serial access storage device, the generic term "file" will be used.

2.3 TERMS AND DATA CONVENTIONS

The words SUBROUTINE and ROUTINE are generally used interchangeably. A program SEGMENT is specific collection of routines used to do a specific processing job, such as the calculation of radiation interchange factors. Generally, the routines comprising a segment are brought into core together, and in this sense, a segment can be thought of as an OVERLAY, where that concept is applicable to a particular computer operating system. INTEGER and FIXED POINT mean the same thing, as do REAL and FLOATING POINT.

The term HOLLERITH* is applied to strings of alphanumeric characters. The term DATA VALUE, or LITERAL, will be taken to mean one element of the set of all integers, floating point numbers, and 6-character** Hollerith strings.

A data value may also be any arithmetic FORTRAN expression, which may contain variable names. For example:

ANAME = 6.8*4.317/CON1

is allowed. On the other hand, function calls such as:

DNAME = 4.7* SIN (1.73)

are not. The user is also cautioned to avoid mixed-mode expressions.

Integers will be shown in print as a sequence of digits preceeded, optionally, by a plus or minus sign. Floating point numbers will appear in print as a sequence of digits with a leading, trailing, or imbedded decimal point, prefixed, optionally, by a plus or minus sign, and suffixed, optionally, by an exponent (to the base 10) denoted as the letter E followed by an integer. Hollerith strings of characters will be delineated in print by asterisks. These are necessary because blanks are valid characters and have a specific binary code (i.e., they do not appear on the printed page, but they do appear explicitly in the computer).

- -

*The Hollerith code is actually a binary code for representing alphanumeric characters on punched cards. Other common binary codes for representing alphanumeric characters include BCD, ASCII, EBDIC, and FIELDATA. The use of Hollerith to denote character strings in general is purely arbitrary.

**A 6-character string may be stored in one UNIVAC 1108 computer word. TRASYS implementations on other computers may provide more or less characters per word. In the general case, a Hollerith data value would contain as many characters as will fit in one computer word.

In addition to DATA VALUES, another entity, called an IDENTIFIER, REFERENCE FORM, or VARIABLE, will be used (in a programming sense). For example, consider the following statement:

```
PI = 3.14
```

In this case, 3.14 is a floating point data value, and PI is an identifier. Note that PI is different from *PI* which is a Hollerith string.

The data field of any card may be terminated by the character $. This terminates any further data read operations for that card, and allows the user to enter comment data to the right of the $. This may tempt the user to enter a comment to the right of it in an otherwise blank card. This results in an empty data field and a fatal error. Instead, comment cards are formatted in the classic FORTRAN manner, that is, with a C in card column 1. Such comment cards may be used in any of the data blocks.

3. INPUT DECK

3.1 Introduction to the Input Deck

3.1.1 Basic Concepts

TRASYS input decks consist of two fundamental parts. Part I consists of the EDIT/CONTROL blocks. These blocks do not participate at all in the definition of the mathematical model of the thermal radiation problem. This part provides basic program control and provides the user with his edit capability. Part II is referred to hereinafter as the TRASYS MODEL. This part is made up of the data blocks that describe the user's problem in terms of geometry definition and drive logic. The Options, Model Collect, and Source Edit data blocks comprise the EDIT/CONTROL portion of the input data. Examples of options data are problem title information, edit tape identification, input data punch/no punch, list/no list flags and a documentation data list/no list flag. A one line (CC7-72) problem title is entered in the options data block. This title identifies the MODEL portion of the input deck on any tapes or storage units it may be transferred to. This title will also appear on each page of output printed by the standard library output routines during execution. The Model Collect data block allows the user to collect the input data defining two or more thermal radiation analysis models and combine them into one. These models must have been previously stored on tape, under model name identifiers. The Edit Data block allows the user to do a line by line edit on previously taped input data. This input data may be a single model found on tape, or the result of the model collect operation.

The MODEL portion of the input deck consists of the following blocks:

Documentation Data
Array Data
Quantities Data
Surface Data
Block Coordinate System (BCS) Data
Form Factor Data
Shadow Data
Flux Data
Correspondence Data
Operations Data
Subroutine Data

In general, the largest block is the surface data. This block is the user's means to describe the geometry of the surfaces that participate in the radiant interchange of his problem. Because of the surface data block's size, a number of input options, differing in format and concept, are provided. This allows the user to choose the most convenient means of defining the different parts of his geometry; thus easing his most laborious task.

Another type of data that may comprise a large portion of the user's input may consist of information normally considered to be program output or interim output. A user may have, for example, a large portion of the form factors needed for his solution available from some external source. Using the form factor data block, he may enter this data and save much processing time. The most frequent application of this type of input is for restarting runs that were not completed for one reason or another. Input blocks are available for form factor and direct flux data.

Another data block that allows the user to take advantage of large blocks of previously known data is the shadow data block. If shadow factor tables are known for a portion of this model, the user may enter them through this block and avoid computing them in a shadow factor generating run.

The remaining data blocks used for general alphanumeric input are the correspondence data, array data and quantities data blocks. The correspondence data provides the capability for the user to redesignate node numbers, combine a number of nodes into single nodes, or subdivide nodes into subnodes. The array data block provides a convenient input point for any array data that the user may require. Array data may be integer or floating point data value strings, or Hollerith strings. The quantities data block performs the same function as the array data block except that single values are entered rather than strings. If the user desires, he may enter an extended written description of his problem in the documentation block. This will appear at the user's option at the beginning of his printed output and will be stored with the remainder of his input data on his edit out tape.

The user's driver logic is entered in the operations data and subroutines data blocks. The operations data block consists of a series of calls to user-addressable subroutines and computation segments arranged in a series of steps that are used for orderly handling of the output data in out-of-core storage. Operations block subroutine calls are primarily used to input and update appropriate problem parameters. The calls to the computation segments are what actually result in the generation of output data. The operations block subroutine calls are in classic FORTRAN format, and the user has at his disposal the FORTRAN V language for coding specialized operations block logic. In the operations block, the user has access to all variables he identified in his array and quantities data, plus an extensive list of program variables located in labeled common.

The subroutines data block contains FORTRAN language subroutines that are either user-called or called by the various computation segments. Routines found in the subroutines block bearing the same name as processor library routines will compile in place of the library routine, thus giving the user the capability to override any program function he desires.

3.1.2 Basic Structure

The basic structure of the TRASYS input deck is shown in Figure 3-1. This figure illustrates the three EDIT/CONTROL blocks and the 11 MODEL blocks in their correct input sequence. Format of the header cards that lead each

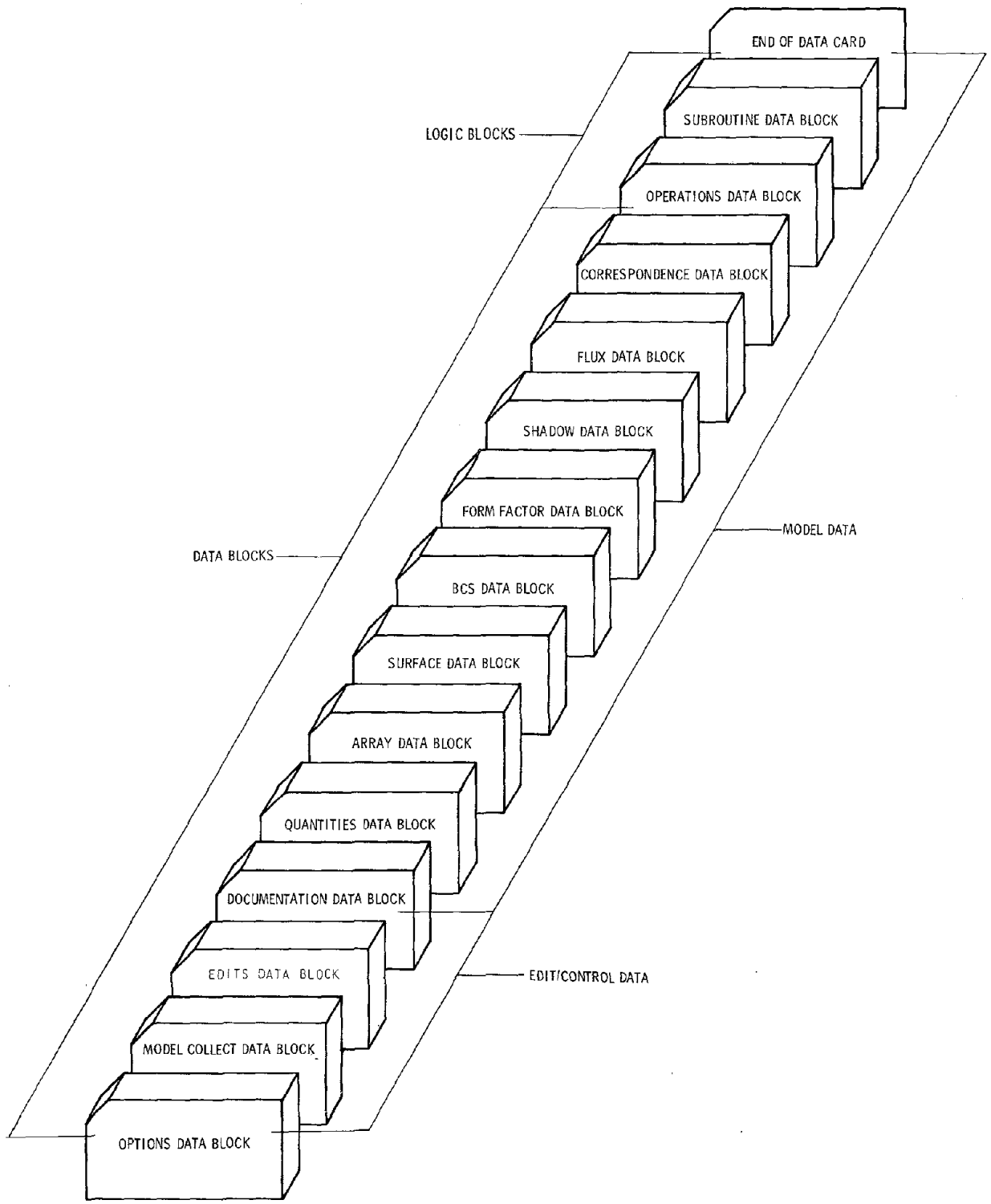


Figure 3-1 Input Deck Structure

block is defined in Figure 3-2. The data blocks must appear in the order shown in Figure 3-1. Any block except the options block may be omitted, along with its header card if it is not required for the problem at hand.

The EDIT/CONTROL blocks are discussed in Section 3.2; the model data blocks in Section 3.3 and the operations and subroutines blocks in Section 3.4.

FORTRAN CODING FORM

	Punching Instructions		Page of
Program	Graphic	Card Form # *	Identification 73 80
Programmer Date	Punch		

C FOR COMMENT

STATEMENT NUMBER 1 5	Cont. 6	FORTRAN STATEMENT 7 10 15 20 25 30 35 40 45 50 55 60 65 70 72
		HEADER OPTIONS DATA
		(Options data cards)
		HEADER MODEL COLLECT DATA
		(Model Collect data cards)
		HEADER EDIT DATA
		(Source edit cards)
		HEADER DOCUMENTATION DATA
		(Documentation data cards)
		HEADER QUANTITIES DATA
		(Quantities data cards)
		HEADER ARRAY DATA
		(Array data cards)
		HEADER SURFACE DATA
		(Surface data cards)
		HEADER BCS DATA
		(BCS data cards)
		HEADER FORM FACTOR DATA
		(Form factor data cards)
		HEADER SHADOW DATA
		(Shadow data cards)
		HEADER FLUX DATA

Figure 3-2. Header Card Formats

FORTRAN CODING FORM

	Punching Instructions		Page of
Program	Graphic	Card Form # *	Identification 73 80
Programmer / Date	Punch		

C FOR COMMENT

STATEMENT NUMBER 1 5	Cont. 6	FORTRAN STATEMENT 7 10 15 20 25 30 35 40 45 50 55 60 65 70 72
		HEADER CORRESPONDENCE DATA
		(Correspondence data cards)
		HEADER OPERATIONS DATA
		(Operations data cards)
		HEADER SUBROUTINE DATA
		(Subroutine data cards)
		END OF DATA

Figure 3-2. (concl)

3.2 EDIT/CONTROL Data Blocks

3.2.1 Options Data Block

3.2.1.1 Basic Concepts

The Options data block provides the user with the following capabilities and operating options:

1) An entry point for his problem title and model name

2) Error plot option control

3) Source deck list/no list control

4) Source deck punch/no punch control

5) Go/No-Go option (No Go for edit only, no execution)

6) Print/no print for edit directives

7) Relabel edit directives

8) Print/no print of documentation data block

9) Read/write directions for all input and output tapes.

3.2.1.2 Options Data Block Variables

Table 3-I lists the options data block variables together with their options, default values, and descriptions.

3.2.1.3 Options Data Block Example

Figure 3-3 is an example of an options data block.

3.2.1.4 Automatic Node Plots Option

When surface data input rules are violated to the point where any surface is insufficiently defined to be usable, a fatal error flag is set and the run is terminated at the end of preprocessor execution. This is oftentimes undesirable because the primary objective of the first run on a newly defined model is usually to obtain plots of the problem geometry so that the user can visually verify his input. Time and effort can be saved if the surfaces that are usable to the node plotter are plotted in spite of the fatal errors.

Table 3-I Options Data Input Detail

Options Data Input CC1	CC7	OPTIONS	DEFAULT VALUE	DESCRIPTION
TITLE		PROBLEM TITLE IN CARD COLUMNS 7-72, INCLUSIVE	NONE	PROBLEM TITLE
	MODEL	= ANY 1-6 CHARACTER MODEL NAME	****	PRIMARY MODEL NAME
		= NAME1 - NAME2	NONE	CHANGE MODEL NAME FROM NAME1 TO NAME2
	LIST SOURCE	- ACTIVE	NONE	LIST ACTIVE CARDS IN MODEL
		- INACTIVE	NONE	LIST INACTIVE CARDS IN MODEL
		- ALL	NONE	LIST ALL ACTIVE AND INACTIVE CARDS
		(NOT INPUT)	NONE	NO LIST
	PUNCH SOURCE	- ACTIVE	NONE	PUNCH ACTIVE CARDS IN MODEL
		- INACTIVE	NONE	PUNCH INACTIVE CARDS IN MODEL
		- ALL	NONE	PUNCH ALL ACTIVE AND INACTIVE CARDS
		(NOT INPUT)	NONE	NO PUNCH
	NOGO	(INPUT)	NONE	EDIT, BUT DO NOT PREPROCESS OR PROCESS
		(NOT INPUT)	NONE	EDIT, PREPROCESS, AND PROCESS
	NO PRINT	- EDIT	NONE	DO NOT PRINT EDIT DIRECTIVES
		(NOT INPUT)	NONE	PRINT EDIT DIRECTIVES
	RELABEL	(INPUT)	NONE	CHANGE MODIFIER LABEL TO (AA) AND DELETE
		(NOT INPUT)	NONE	ALL INACTIVE CARDS
	DMPDOC	(INPUT)	NONE	PRINT DOCUMENTATION DATA BLOCK
		(NOT INPUT)	NONE	NO PRINT
	EDITI	- TXXXX	NONE	DIRECTS EDITOR TO READ EDITI TAPE TXXXX
	EDITO	- TXXXX	NONE	DIRECTS EDITOR TO WRITE EDITO TAPE TXXXX
	CMERG	- TXXXX	NONE	DIRECTS EDITOR TO READ CMERG TAPE TXXXX
	BCDOU	- TXXXX	NONE	DIRECTS PROGRAM TO WRITE BCDOU TAPE TXXXX
	ERPLOT	(INPUT)	NONE	ACTIVATES AUTOMATIC NODE PLOT
		(NOT INPUT)	NONE	NO PLOT
	TAPENAME	- TXXXX	NONE	DIRECTS PROGRAM TO READ OR WRITE TAPE TXXXX TO GENERAL UNIT TAPENAME*

*See Appendix G for allowable TAPENAME OPTIONS

FORTRAN CODING FORM

	Punching Instructions		Page of
Program	Graphic	Card Form #	Identification
Programmer / Date	Punch		73 80

C FOR COMMENT

STATEMENT NUMBER 1–5	Cont. 6	FORTRAN STATEMENT 7–72
HEADER		OPTIONS DATA
TITLE		SAMPLE OPTIONS DATA BLOCK INPUT
		MODEL = MODEL1
		LIST SOURCE - ALL
		PUNCH SOURCE ACTIVE
		NOGO
		DMPDOC

Figure 3-3 Options Data Block Sample

The automatic node plot routines eliminate this problem. This capability functions as follows: when the word **ERPLOT** appears in the options data block and fatal errors result from the surface data, an operations data block is generated by the preprocessor. An example of such an operations data block is shown in Figure 3-4. This example is for a model that uses three block coordinate systems. The operations data block generated is then executed and the job terminates. Note that four automatically scaled plots are generated for each BCS. The views are from the x, y, and z axes, plus a 3-D. Also note that any operations data entered by the user is ignored.

```
HEADER OPERATIONS DATA
STEP 1
        CALL BUILDC(BCS1)
        CALL NDATAS(1,3HALL,0)
L       NPLOT
        CALL BUILDC(BCS2)
L       NPLOT
        CALL BUILDC(BCS3)
L       NPLOT
END OF DATA
```

Figure 3-4 Sample Operations Data Block Generated for Automatic Node Plots

3.2.2 Model Collect and Edit Data Blocks

3.2.2.1 Basic Concepts

Figure 3-5 is a block diagram of the edit portion of the preprocessor. Edit logic flow is shown, together with its relationship with the remainder of the program.

The edit portion consists of two basic parts:

1) The MODEL COLLECTOR which transfers files consisting of complete TRASYS models from one or more input units to a single unit which may be output as is or edited.

2) The SOURCE EDITOR which deletes, inserts, merges, and yanks records and blocks of records to form a model.

The model collector uses information contained in the MODEL COLLECT block (input on cards) to pick files from any number of input units and generate from them an EMERG tape. Each input tape may contain one or more TRASYS models, in TRASYS source edit format, each identified by a six character model name. These units are all in binary mode, having been generated as EDITO tpes. One input unit, the EDITI tape, contains the primary model. The EDITI model may be passed directly on to the source editor in lieu of the EMERG tape if no model collect operations are required.

The source editor's function is to generate a complete model and pass it on to the data and logic preprocessors on the DATAI unit. If desired, the

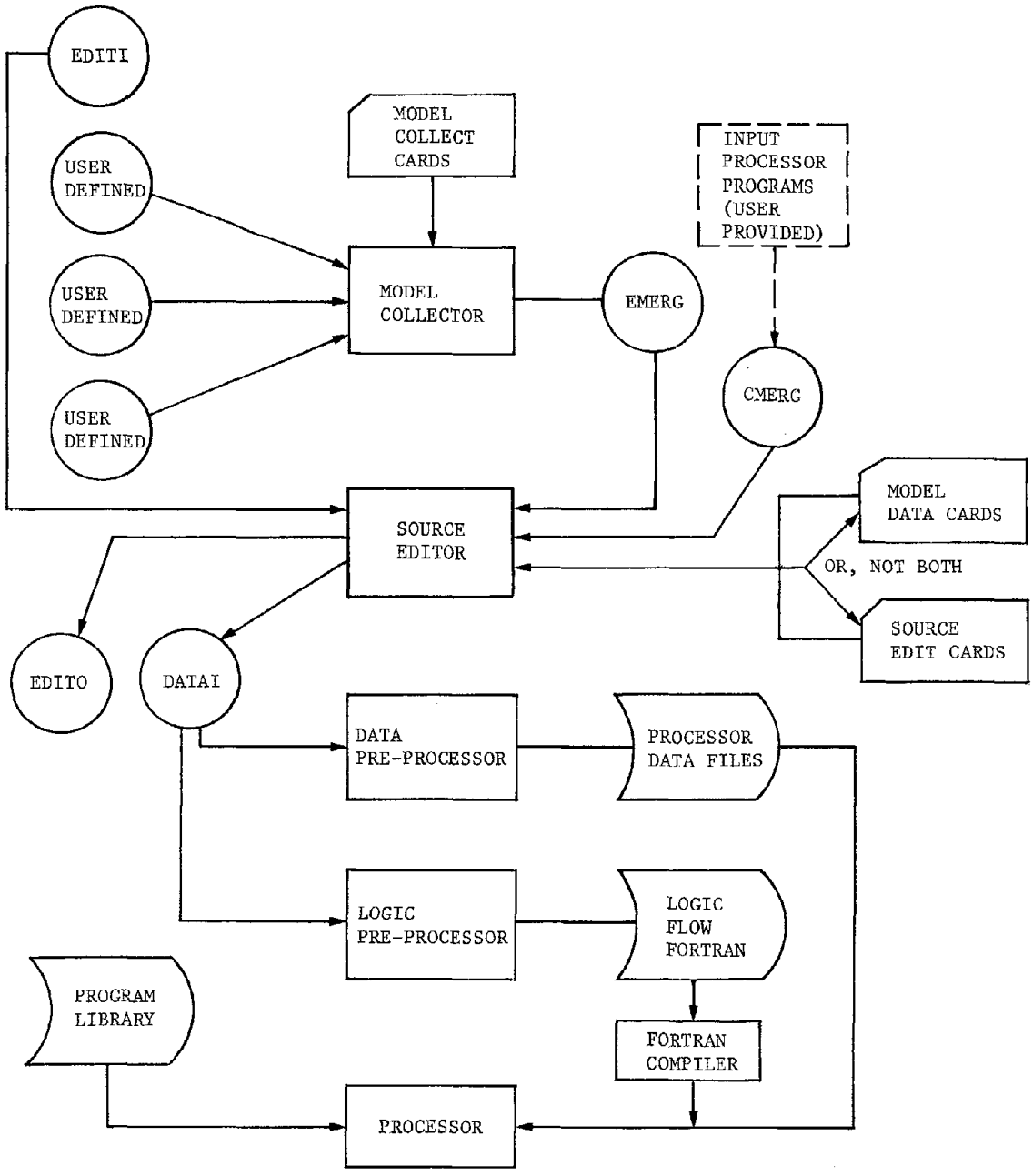


Figure 3-5 Edit Segment Logic Flow

user may obtain a permanent copy of the DATAI model on an EDITO tape. Source editing can be done in several ways. The simplest mode is to read the user's cards and pass them on as a complete model on the DATAI unit. Alternatively, the user supplied CMERG tape can be passed along directly if the CMERG tape is nothing more than the user's cards previously transferred to tape. The CMERG unit also provides an interface between any user supplied input processing routine(s) and the program. In the general edit case, the source-edit cards are used to generate an executable model from input data found in card form and in the CMERG, EDITI, or EMERG units.

The product of the TRASYS editor is a single TRASYS model. This model, designated the primary model, is either a model selected from an EDITI tape, or a model in the card input stream. Edits can be in the form of record deletions and record insertions. Records for insertions can be obtained from these sources: (1) the card input unit (edit data block); (2) the card merge unit - CMERG; and (3) the edit merge unit - EMERG.

The CMERG unit can be single or multifile tape, disk, or drum. The data contained on the CMERG unit must be in BCD mode, TRASYS card image form. This type of data is usually generated by: (1) card-to-tape by an off-line computer; (2) TRASYS processor output; or (3) TRASYS input data conversion programs.

The EMERG unit can be a single or multifile tape, disk, or drum. The data contained on the EMERG unit must be in binary mode, TRASYS generated format. All EMERG files are generated by an EDITO output file from a previous run, or a combination of the files combined onto one EMERG unit by the use of the model collector.

Any record or group of records contained in any file or model on the EMERG or CMERG units can be merged into the primary model. Cards to be merged into the primary model can be in random order on the EMERG and CMERG unit. Note that inserting data from cards in the edit data block is possible only when the primary model comes from an EDITI unit. In fact, a Header Edit Data card results in an error abort if no EDITI tape is specified in the options data.

Besides deleting, inserting, and merging cards into the primary model, the source editor has the capability of yanking modifications. Each time a primary model is edited, the inserted and deleted cards are tagged with a modifier label and deleted cards are maintained on an inactive status. A "yank" provides a simple means of returning a primary model to the condition it was before a given modification. For instance, yanking modification "A" will insert all cards deleted by "A" and delete all cards inserted by "A." More than one label can be yanked in one run. Cards deleted by a yank are not maintained on inactive status.

3.2.2.2 Model Collect Data

Cards in the model collect data block enable the user to select entire models from a single or multifile tape containing TRASYS models and transfer them to a single EMERG unit.

The following are the allowable commands from the model collect data block:

Format	Function
CC7	
UNIT*, MODNAM	Selects a model named MODNAM from UNIT (Defined in options data block) and writes it to EMERG.
UNIT,MODNAM1,CHANGE, MODNAM2	Same as UNIT,MODNAM, except model is renamed on EMERG.
UNIT,DIRECTORY	Prints a directory of model names found on UNIT

*Choices are USER1, USER2

3.2.2.3 Edit Data Block

After a valid EDITO tape has been generated, the user will have a source listing with edit numbers and edit labels. This tape may then become an EDITI or an EMERG tape that can be edited according to the source listing using edit directives in the edit data block. Table 3-II presents details of the edit data block directives together with format information. Figure 3-6(a) is an example of an edit data block.

3.2.2.4 Edit Operations

Edit operations are illustrated by the following examples, see Fig. 3-6(b), 3-6(c), and 3-6(d):

1) Primary model on cards, edits directly from cards and from EMERG and CMERG tapes.

2) Primary model on an EDITI tape, edits from cards, EMERG and CMERG tapes.

3) Restart from CMERG after DI calculations.

Table 3-II Edit Data Block Input Details

EDIT DATA INPUT (SEE NOTES AT END OF TABLE)	DESCRIPTION
*I, N1 OR *INSERT, N1	THE CARDS FOLLOWING THIS CARD WILL BE INSERTED AFTER EDIT NUMBER -N1-
*D, N1 OR *DELETE, N1	DELETE PRIMARY MODEL CARD WITH EDIT NUMBER -N1-
*D, N1, N2 OR DELETE, N1, N2	DELETE PRIMARY MODEL CARDS WITH EIDT NUMBERS -N1- THROUGH -N2-
*C, F1 OR *CMERG, F1	MERGE THE ENTIRE CARD FILE -F1- FROM UNIT -CMERG-
*C, F1, N1, N2 OR *CMERG, F1, N1, N2	MERGE LINES N2 THROUGH N2 FROM CMERG FILE F1
*C, F1, N1, ALL OR *CMERG, F1, N1, ALL	MERGE ALL LINES FROM N1 TO END OF CMERGE FILE F1
*E, NAME OR *EMERGE, NAME	MERGE THE ENTIRE EDIT MODEL, -NAME- FROM UNIT -EMERG-
*P OR *PUNCH	PUNCH ACTIVE CARDS FROM THE UPDATED MODEL (SEE NOTE 3)
*P, INACTIVE OR *PUNCH, INACTIVE	PUNCH INACTIVE CARDS FROM THE UPDATED MODEL (SEE NOTE 3)
*P, ALL OR *PUNCH, ALL	PUNCH ALL ACTIVE AND INACTIVE CARDS FROM THE UPDATED MODEL (SEE NOTE 3)
*L OR *LIST	LIST ACTIVE CARDS FROM THE UPDATED MODEL

Table 3-II (concl)

EDIT DATA INPUT (SEE NOTES AT END OF TABLE)	DESCRIPTION
*L, INACTIVE OR *LIST, INACTIVE	LIST INACTIVE CARDS FROM THE UPDATED MODEL
*L, ALL OR *LIST, ALL	LIST ALL ACTIVE AND INACTIVE CARDS FROM THE UPDATED MODEL
*S OR *SEQUENCE	NUMERICALLY SEQUENCE PUNCHED CARDS. THIS CARD MAY APPEAR ANYWHERE IN THE EDIT DATA BLOCK
*Y, AB OR *YANK, AB	DELETE ALL CARDS FROM PRIMARY MODEL THAT WERE INSERTED BY EDIT DIRECTIVE -AB- (SEE NOTE 4)
*Y, AB, AD OR *YANK, AB, AD	DELETE ALL CARDS FROM PRIMARY MODEL THAT WERE INSERTED BY SOURCE EDIT DIRECTIVES -AB- THROUGH -AD- (SEE NOTE 4)

NOTES:

1. AN ASTERISK (*) IN CARD COLUMN 1 DESIGNATES AN EDIT CONTROL CARD.
2. EDIT CONTROL CARDS ARE FREE FIELD FORMATED IN CARD COLUMNS 2 THROUGH 72 WITH BLANK COLUMNS AND COLUMNS 73 THROUGH 80 IGNORED.
3. THIS CARD MAY BE USED IN PLACE OF THE -PUNCH SOURCE- OPTION IN THE OPTIONS DATA BLOCK. IF BOTH CARDS ARE INPUT, THIS CARD OVERRIDES THE OPTIONS DATA INPUT. THIS CARD MAY APPEAR ANYWHERE IN THE EDIT DATA BLOCK.
4. ONLY ONE YANK DIRECTIVE CAN APPEAR IN THE EDIT DATA BLOCK, AND THAT CARD MUST IMMEDIATELY FOLLOW THE -HEADER EDIT DATA- CARD.

FORTRAN CODING FORM

		Punching Instructions	Page of
Program		Graphic	Card Form # *	Identification 73 80
Programmer	Date	Punch	

C FOR COMMENT

STATEMENT NUMBER 1 5	Cont. 6	FORTRAN STATEMENT 7 72
		HEADER EDIT DATA
		*YANK,AC
		*LIST,ALL
		*PUNCH
		*DELETE,12 (DELETES CARD 12)
		*INSERT,11
		ICS 103,-1.,8.,0.,ROTZ=90. (THIS CARD INSERTED AFTER CARD 11)
		*DELETE,214,223 (DELETES CARDS 214 THROUGH 223)
		*SEQUENCE

Figure 3-6(a) Edit Data Block Example

```
HEADER OPTIONS DATA
TITLE  EDITOR CHECK OUT-EXAMPLE 1
       MODEL - DATA1 $ DATA1 = MODEL NAME THAT WILL APPEAR ON EDITO TAPE
       EMERG - TXXXX
       CMERG - TXXXX
       EDITO - TXXXX
HEADER SURFACE DATA
                       (SURFACE DATA CARDS)
C       INSERT CARDS 199 THRU 998 FROM EMERGE MODEL SURFD1 INTO SURFACE
C       DATA BLOCK
*EMERG,SURFD1,199,998
                       (ADDITIONAL SURFACE DATA CARDS)
HEADER FORM FACTOR DATA
C       INSERT FILE 4 FROM CMERG TAPE
*CMERG,4
HEADER FLUX DATA
                       (FLUX DATA CARDS)
HEADER OPERATIONS DATA
C       INSERT PART OF FILE 2 FROM CMERG TAPE
*C,2,1,39
END OF DATA
```

(b) Edit Operations Example - Primary Model on Cards

```
HEADER OPTIONS DATA
TITLE  EDITOR CHECK OUT EXAMPLE 2
       MODEL = DOCK1 - DOCK2  $ DOCK1 = EDITI (PRIMARY) MODEL NAME
C                               DOCK2 = EDITO MODEL NAME
       EMERG - TXXXX
       CMERG - TXXXX
       EDITI - TXXXX
       EDITO - TXXXX
HEADER EDIT DATA
*D,2,6
                       (CARDS TO BE INSERTED IN LIEU OF CARDS 2 THRU 6)
*I,11
                       (CARDS TO BE INSERTED AFTER CARD 11)
*I,340      $CARD FOLLOWING INSERTS CARDS 312 THRU 450 FROM MODEL DOCKA
C           ON EMERG AFTER CARD 340
*E,DOCKA,312,450
*C,3,2,111  $ INSERTS CARDS 2 THRU 111 FROM CMERG FILE 3
END OF DATA
```

(c) Edit Operations Example - Primary Model on EDITI Tape

Figure 3-6 (cont)

```
HEADER OPTIONS DATA
TITLE  RESTART FROM CMERG EXAMPLE - FIRST RUN
       RTO-T3815
                          (DATA BLOCKS AS REQUIRED)
HEADER OPERATIONS DATA
STEP 1
                       (BUILDC & ADD CALLS AS REQUIRED)
C      COMPUTE FORM FACTORS & WRITE THEM TO TAPE (ONE FILE)
       FFPNCH = 4HTAPE
L      FFCAL
STEP 2
                       (ORBIT DEFINITION AS REQUIRED)
       DIPNCH = 4HTAPE
C      FOLLOWING CARD COMPUTES FLUXES FOR
C      11 POINTS IN ORBIT-WRITES 11 FILES
C      TO TAPE.  STEPS 10000 THRU 10010 ARE GENERATED
ORBGEN INER, 0., 360., 10, 0, 0, 0
END OF DATA


HEADER OPTIONS DATA
TITLE  RESTART FROM CMERG EXAMPLE - RESTART RUN
       BCDOU - TXXX
       CMERG - T3815

                          (DATA BLOCKS AS REQUIRED)
HEADER FORM FACTOR DATA
*C, 1  $ FORM FACTORS ON FIRST FILE OF CMERG
HEADER FLUX DATA
*C, 2  $ FLUX DATA, STEP 10000
*C, 3  $ FLUX DATA, STEP 10001

*C, 12 $ FLUX DATA, STEP 10010
HEADER OPERATIONS DATA
STEP 1
                       (BUILDC & ADD CALLS AS REQUIRED)
       FFPNCH = 2HNO
L      FFCAL $ READS IN FORM FACTORS FROM FF DATA BLOCK
       CALL GBDATA (1, 4HBOTH)
L      GBCAL
       DIPNCH = 2HNO
                             (ORBIT DEFINITION)
ORBGEN INER, 0., 360., 10, 1, 1, 1
       CALL QODATA (0, 0, 4HTAPE, 2HNO, 0, 0, 0, 4HBOTH, 0)
L      QODATA $ WRITE OUT ABSORBED HEAT DATA TO BCDOU TAPE
END OF DATA
```

(d) Edit Operations Example - Restart from CMERG

Figure 3-6 (concl)

3.3 Model Data Blocks

3.3.1 Documentation Data

Experience has shown that thermal mathmatical models may have an extended useful life, especially if tape storage with convenient editing capability is available. The usual environment of sketchy and rarely updated documentation results in a waste of resources as these long-lived models are passed from analyst to analyst through project personnel changes.

As an aid in alleviating this problem, TRASYS users may document their efforts in an easily edited and easily accessed form in the documentation data block.

The documentation data block has the following format:

CC1	CC7	CC7 2

HEADER DOCUMENTATION DATA

Documentation Data Card 1
Documentation Data Card 2
. . . .
. . . .
. . . .
Documentation Data Card N

Documentation data cards have a data field from CC7 through 72 inclusive and have no restriction on their alphanumeric content. There is no practical limit on the number of documentation cards allowed.

The control field of documentation data cards is used for carriage control. The integer N appearing anywhere in CC1 through 6 of a documentation data card results in N lines being skipped before printout of that card. If less than N lines are available on a page, the card will begin a new page. If a new page is desired, the letter P is placed in CC1. A documentation data printout is available at the user's option. The flag DMPDOC appearing in the options block results in a printout of the documentation data prior to any preprocessor or processor operations.

3.3.2 Quantities and Array Data Blocks

3.3.2.1 Basic Concepts

The quantities and array data blocks have the primary function of providing the user with a convenient input point for any single variable and array data he plans to use during his execution. User constants are defined in the quantities data block and arrays in the array data block. A user variable or array may take any name not appearing in the program reserve name list or program control constant list (see Appendix A). Real, integer, or Hollerith data may be entered. Mode agreement is required for real and integer data names. Hollerith strings are limited to 6 characters in the quantities data block.

The pre-processor provides default values for all program control constants, so no control constant input is required in the quantities data. Further, all control constants can be redefined in the operations block. Thus, defining control constants at the quantities data block merely has the effect of redefining the default values. In general, this practice is not recommended because it is easy for the user to forget that he has a non-standard default value in his quantities data block when he is defining control constants in the operations data block.

3.3.2.2 Rules for Input

All quantities and array data are entered in the data field (Columns 7 through 72) of the cards following the appropriate header card. Specific rules for input are:

1) The general quantity data formats are:
   NAME = DV, (integer)
   ANAME = DV, (real)
   NAME (or ANAME) = DV where DV is a 1 to six character string. (Left justified, blank filled)

2) The general array data formats are:
   NAME = DV1, DV2 - - - DVN (integer)
   ANAME = DV1, DV2 - - - DVN (real)
   NAME (or ANAME) = *I AM A HOLLERITH ARRAY* (Hollerith)

3) Array data values may be operated as follows:
   NAME = DV1, REPEAT, DV2, N, DVN+2 (Repeats DV2 N times, continues with DVN+2)
   Real array repeat format is identical.

4) Variable names must consist of 3 to 6 alphanumeric characters, with an alphabetic character heading.

5) Any number of constant or array data values and names may be entered in Card Columns 7 through 72 inclusive. Input may continue to following cards with no data in the control field, provided the fields between commas are complete on each card.

6) Commas are assumed at the end of each card. Commas may be entered at the beginning or end of cards at the user's option. The read routine ignores these.

7) Cards may end but not begin with equal signs.

8) Empty fields (consecutive commas) are illegal.

9) Mode agreement between names and data values must be maintained.

3.3.2.3 Quantities and Array Data Accessing

Quantities and array data are placed in common and are thus accessible from any user or program-called execution routines. The following are the rules for accessing this data:

1) Quantities data are accessed by name only.
For example, with
    ANAM = DV,
in the quantities block, the statement
    VAL = ANAM
in any execution routine results in DV being stored under the name VAL. Mode must be preserved according to the rules of FORTRAN.

2) Array Data is accessed as illustrated by the following examples. Each example presumes that the array:
    ANAM = DV1, DV2 - - - DVN
appears in the array data block.

    A. SUBROUTINE CALLS
    The statement:
    CALL SUBX (ARG1, ARG2, ANAM ARG4 ---)
    in any execution routine will pass the entire ANAM array to subroutine SUBX.

    B. INTEGER COUNT
    The integer count for any array is accessed through the following function call:
    IC = IACT (ANAM)

    C. INDIVIDUAL DATA VALUES
    Individual data values are accessed under the usual rules of of FORTRAN. The statement:
    VAL = ANAM (6)
    results in DV6 being stored under the name VAL.

3) The array data block serves as the only means of reserving space for the operations data block. The following example illustrates this with:
    ANAMA = DV1, DV2, - - - DVN,
    XARRAY = REPEAT, 0., N
in the array data block, the statements:
    DO 1 I = 1, IC
1   XARRAY (I) = ANAMA (I)
in the operations data block will locate the ANAMA array in the first IC words of XARRAY.

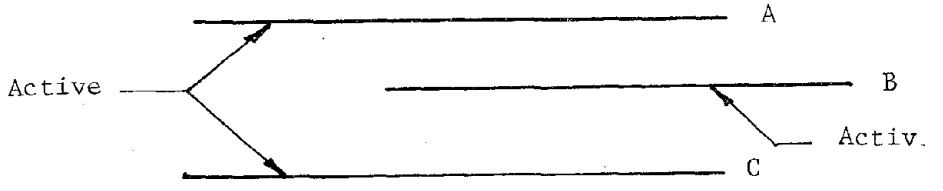
3.3.3 Surface Data

3.3.3.1 Basic Concepts

In its present state of development, TRASYS allows the user's geometric configuration to be made up of the following geometric shapes, or portions thereof:

(1) Rectangles
(2) Discs
(3) Polygons
(4) Right Circular Cylinders
(5) Cones
(6) Spheres
(7) Paraboloids
(8) Rectangular Parallelepipeds with 5 or 6 faces

The surface areas of these shapes are what TRASYS is concerned with. The volume within a sphere, for instance, has no bearing on the thermal radiation problem. Either or both sides of any surface can be defined as "active." Also, any surface can be defined as a "shadower" or "non shadower" depending on whether or not it is desired that it be considered in shadowing (blockage) calculations.



The active side concept is illustrated in the sketch. If form factors were computed in this geometry, F_{AC} and F_{BC} would exist because the active sides involved are in view of each other. Surface B, however, is ignored by surface A.

Again referring to the sketch, if surface B is defined as a shadower in form factor computations, it will affect the calculation of F_{AC}, reducing it accordingly. If not entered as a shadower, it would be totally "invisible" from surface A. Active side definition has no bearing on a surface's effect as a shadower. One further condition must exist for surface B to effect the value of F_{AC}, that is surfaces A and C must be flagged as "can be shaded" surfaces in form factor calculations.

A similar logic is used in computations of direct irradiation. Surfaces defined as shadowers may affect the direct irradiation computed depending on direction to the incident flux source.

Since all surfaces in nature can shade and be shaded, it may seem questionable to leave the shadowing definition up to the user. The reason for this is that significant amounts of computation time can be saved by flagging out surfaces that cannot enter into shadowing.

Any surface may be subdivided into nodal surfaces of equal or unequal size. In general, the nodal surfaces chosen should correspond with the isothermal nodes that appear in the user's thermal analyzer model. For various reasons, this may not be possible, so a convenient means for combining nodes is provided.

3.3.3.2 Coordinate System Definition

The surface data is associated with four different, right-handed cartesian coordinate systems:

Surface coordinate system
Intermediate coordinate system
Block coordinate system
Central coordinate system

There definitions are as follows:

Central Coordinate System (CCS) - The single coordinate system to which all vehicle surfaces must be related. This coordinate system is also used to orient the spacecraft relative to the sun, planet, or a star. This coordinate system is analogous to the body coordinate system used in trajectory tapes.

Block Coordinate System (BCS) - Any "block" of surfaces that will be moved relative to other surfaces during execution must be related to a named block coordinate system. Similarly, if it is desired to activate and/or deactivate a block of surfaces during the course of the problem, these surfaces are related to a separate block coordinate system.

Intermediate Coordinate System (ICS) - An intermediate coordinate system is used when it is convenient to relate a group of surfaces to a coordinate system distinct from any BCS or the CCS.

Surface Coordinate System (SCS) - Each surface is related to its own SCS in a manner that provides a convenient means of input. The surface must then be related to the CCS by defining the rotations and translations necessary to make the SCS and CCS coincide.

3.3.3.3 Coordinate System Hierarchy

Each surface and hence nodal surface defined in the surface data block may undergo three transformations, as follows, before processing begins:

SCS → ICS → BCS → CCS

where, for example the symbology SCS → ICS indicates a transform form SCS-defined 3-space to ICS defined 3-space. These transforms must be performed because all processing is done assuming surface definition in CCS-defined 3-space.

Depending on the complexity of each particular surface definition problem, the user may or may not concern himself with all the transforms. In the simplest case, the user defines a surface in terms of x, y, z coordinates in CCS 3-space. The program automatically generates an SCS for each surface and also generates the transforms necessary to describe the surface in CCS 3-space. In the most complex case, the user defines his surface in SCS 3-space, defines six rotation and translation variables for the SCS → ICS transform, defines six rotation and translation variables for the ICS → BSC transform, and finally six more variables for the BCS → CCS transform. For cases of intermediate complexity, for instance when an ICS is not needed, the ICS → BCS transform variables will default to zero and the user's SCS → ICS transform variables will, in reality define an SCS → BCS transform. Further, if neither an ICS or BCS is required, the SCS → ICS and ICS → BCS transforms will default to zero, and the user's SCS → ICS transform definition will, in reality, define an SCS → CCS transform.

3.3.3.4 Surface Data Input Philosophy

The user is provided with two distinct methods of defining his surface. He may define a surface relative to an SCS, then relate it to the remainder of the surfaces by defining the SCS-CCS translations and rotations; or he may locate the surface directly in relation to the CCS by entering the x, y, z coordinates of up to 15 points on his surface (point method). His choice of these methods depends on the particular surface being considered and its relationship to the remainder of the vehicle. In general, it is easy to define a surface relative to an SCS. This requires five numbers. It may or may not be convenient to determine the translation and rotation data (up to 6 numbers) needed for the SCS → ICS, SCS → BCS, or SCS → CCS relationship. When the computation of these rotation and translation parameters is laborious, the point method is usally a better choice. Except for polygons, this requires up to 5 point definitions per surface (15 numbers), but generally these points are on the surface involved and may be easily scaled from an engineering drawing.

3.3.3.5 Surface Data Variables

A total of 54 variable names are reserved for the purpose of surface data definition.

These variable names are defined in Table 3-III. Also tabulated are their default values, and the allowable range of each variable, where applicable. Figure 3-7 illustrates the relationship of the dimension surface data variables

Table 3-III Surface Data Input Detail

VARIABLE NAME	RANGE OR OPTIONS	DEFAULT VALUE	DESCRIPTION
GENERAL DATA:			
SURFN	1-99999	NONE	a. INTEGER ARRAY OF NODE NUMBERS ASSOCIATED WITH SURFACE b. INITIAL NODE NO. ON SURFACE
NNX	1-999	1	NO. OF NODES IN X DIRECTION
UNNX	N/A	NONE	X-DIMENSION ARRAY FOR UNEQUAL NODE BOUNDARIES
NNY	1-999	1	SAFE AS NNX EXCEPT Y-DIRECTION
UNNY	N/A	NONE	SAME AS UNNX EXCEPT Y-DIRECTION
NNZ	1-999	1	SAME AS NNX EXCEPT Z-DIRECTION
UNNZ	N/A	NONE	SAME AS UNNX EXCEPT Z-DIRECTION
NNAX	1-999	1	SAME AS NNX EXCEPT AX-DIRECTION
UNNAX	N/A	NONE	SAME AS UNNX EXCEPT AX-DIRECTION
NNR	1-999	1	SAME AS NNX EXCEPT R-DIRECTION
UNNR	N/A	NONE	SAME AS UNNX EXCEPT R-DIRECTION
TYPE	RECT, TRAP DISC, CYL CONE, SPHER PARAB, BOX 5 BOX6, POLY	NONE	SURFACE TYPE
IDUPSF	1-99999	NONE	NUMBER OF PREVIOUSLY INPUT SURFACE TO BE DUPLICATED
IMAGSF	1-99999	NONE	NUMBER OF PREVIOUSLY INPUT SURFACE TO BE IMAGED

Table 3-III (cont)

VARIABLE NAME	RANGE OR OPTIONS	DEFAULT VALUE	DESCRIPTION
GENERAL DATA:			
ACTIVE	TOP, BOTTOM BOTH (PLANAR SURFACES) IN, OUT, BOTH (SURFACES OF REVOLUTION)	NONE	ACTIVE SIDE DEFINITION TOP IS +Z FACE OF PLANAR SURFACES
BCSN	1-6 CHARACTER NAME	ALLBLK	BLOCK COORDINATE SYSTEM NAME - IDENTIFIES SURFACE WITH A BCS
COM	N/A	BLANKS	30 CHARACTERS OF COMMENT TO DESCRIBE SURFACE
SHADE	FF, DI, BOTH, NO, ONLY	BOTH FF*	SURFACE CAN SHADE FLAG FF: SHADES IN FORM FACTOR CALCULATIONS ONLY DI: SHADES IN DIRECT IRRADIATION CALCULATIONS ONLY BOTH: SHADES IN BOTH FF AND DI CALCULATIONS NO: SURFACE CANNOT SHADE ONLY: SURFACE IS A SHADOWER ONLY (FF AND DI)
BSHADE	FF, DI, BOTH NO	BOTH	SURFACE CAN BE SHADED FLAG FF: CAN BE SHADED IN FF CALCULATIONS ONLY DI: CAN BE SHADED IN DI CALCULATIONS ONLY BOTH: CAN BE SHADED IN BOTH FF AND DI CALCULATIONS NO: SURFACE CANNOT BE SHADED

Table 3-III (cont)

VARIABLE NAME	RANGE OR OPTIONS	DEFAULT VALUE	DESCRIPTION
DIMENSIONS DATA:			
AXMIN	$-270. \leq$ AXMIN $\leq 450.$	NONE	MIN. X-ANGLE SURFACES OF REVOLUTION
AXMAX	$-270. \leq$ AXMAX $\leq 450.$	NONE	MAX. X-ANGLE
ZMIN	N/A	NONE	MIN. DIMENSION-Z DIRECTION
ZMAX	N/A	NONE	MAX. DIMENSION-Z DIRECTION
RMIN	N/A	NONE	MINIMUM RADIUS - DISC SECTION
RMAX	N/A	NONE	MAXIMUM RADIUS - DISC SECTION
R	N/A	NONE	RADIAL DIMENSION
Z	N/A	NONE	Z DIMENSION
P1, P2 --- ETC.	P1 - P15	NONE	CARTESIAN POINT INPUT (GENERAL FORM: PN = XN, YN, ZN)
PROPERTIES DATA:			
ALPHA	$0. \leq$ ALPHA ≤ 1.0	NONE	ABSORPTIVITY-SOLAR
EMISS	$0. \leq$ EMISS ≤ 1.0	NONE	EMISSIVITY-IR
TRANI	$-1.0 \leq$ TRANI ≤ 1.0	0.0	TRANSMISSIVITY-IR
TRANS	$-1.0 \leq$ TRANS ≤ 1.0	0.0	TRANSMISSIVITY-SOLAR
SPRI	$0. \leq$ SPRI ≤ 1.0	0.0	SPECULAR REFLECTIVITY - IR
SPRS	$0. \leq$ SPRS ≤ 1.0	0.0	SPECULAR REFLECTIVITY - SØLAR

Table 3-III (cont)

VARIABLE NAME	RANGE OR OPTIONS	DEFAULT VALUE	DESCRIPTION
POSITION DATA:			
TX	N/A	0.0	TRANSLATION DISTANCE FROM ORIGIN OF CCS, BCS OR ICS TO ORIGIN OF SCS, MEASURED ALONG CCS, BCS OR ICS X-AXIS.
TY	N/A	0.0	SAME AS TX, EXCEPT ALONG Y-AXIS
TZ	N/A	0.0	SAME AS TX, EXCEPT ALONG Z-AXIS
ROTX	$-360. < ROTX \leq 360.$	0.0	ROTATION ANGLE TO ROTATE CCS, BCS OR ICS INTO SCS; ROTATES ABOUT CCS, BCS OR ICS X-AXIS, Y TOWARD Z POSITIVE.
ROTY	$-360. < ROTY \leq 360.$	0.0	SAME AS ROTX, EXCEPT ROTATES ABOUT Y, Z TOWARD X POSITIVE.
ROTZ	$-360. < ROTZ \leq 360.$	0.0	SAME AS ROTX, EXCEPT ROTATES ABOUT Z, X TOWARD Y POSITIVE.
ICS DEFINITION (I CARD) DATA:			
ICSN	1-99999	NONE	INTERMEDIATE COORDINATE SYSTEM NUMBER
TX	N/A	0.0	TRANSLATION DISTANCE FROM ORIGIN OF CCS OR BCS TO ORIGIN OF ICS, MEASURED ALONG CCS OR BCS X-AXIS.
TY	N/A	0.0	SAME AS TX, EXCEPT ALONG Y-AXIS.
TZ	N/A	0.0	SAME AS TX, EXCEPT ALONG Z-AXIS.

Table 3-III (concl)

VARIABLE NAME	RANGE OR OPTIONS	DEFAULT NAME	DESCRIPTION
ICS DEFINITION (I CARD) DATA:			
ROTX	$-360. < ROTX \leq 360.$	0.0	ROTATION ANGLE TO ROTATE CCS OR BCS INTO ICS; ROTATES ABOUT CCS OR BCS X-AXIS, Y TOWARD Z POSITIVE
ROTY	$-360. < ROTY \leq 360.$	0.0	SAME AS ROTX, EXCEPT ROTATES ABOUT Y, Z TOWARD X IS POSITIVE.
ROTZ	$-360. < ROTZ \leq 360.$	0.0	SAME AS ROTX, EXCEPT ROTATES ABOUT Z, X TOWARD Y POSITIVE.
R-CARD DATA:			
REFNO	1-99999	NONE	NUMBER OF REFLECTING PLANE SURFACE.
D-CARD DATA:			
DV	FLOATING POINT	1.0	LENGTH UNIT MULTIPLIER
N-CARD DATA:			
INC	INTEGER	0.0	SURFACE NUMBER CHANGE VALUE (SURFN = SURFN + INC).

*If the surface has a component of specular reflectance and the can-shade flag is either unspecified or set to "NO," the flag is reset to "FF." If the flag is set to "DI" it is reset to "BOTH." (Specular surfaces must be shadowers in the FF segment.)

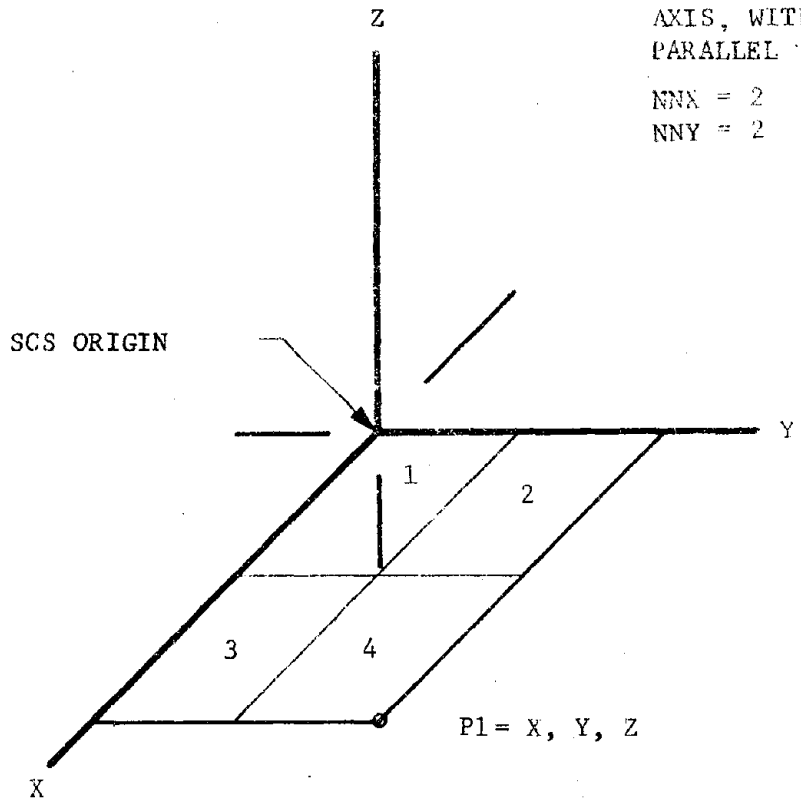
RECTANGLE

EXAMPLE: P1= 2.0, 3.0, 0.0

NOTE: ONE CORNER MUST BE ON THE Z AXIS, WITH RECTANGLE PARALLEL TO X-Y PLANE.

NNX = 2
NNY = 2

SCS METHOD



POINT METHOD

EXAMPLE: P1 = X1, Y1, Z1
P2 = X2, Y2, Z2
P3 = X3, Y3, Z3

NOTE: POINTS NUMBERED CCW AS VIEWED FROM ACTIVE SIDE WHEN ACTIVE IS FLAGGED "TOP"

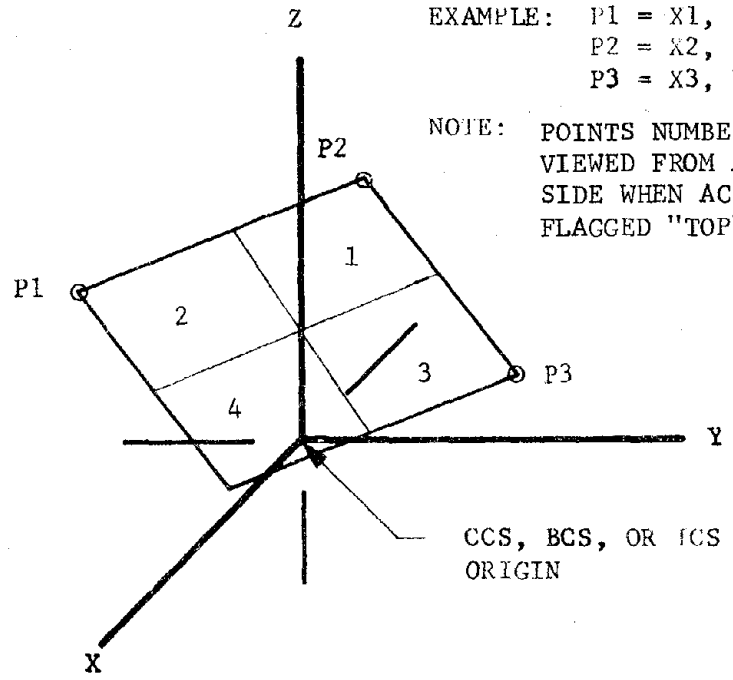
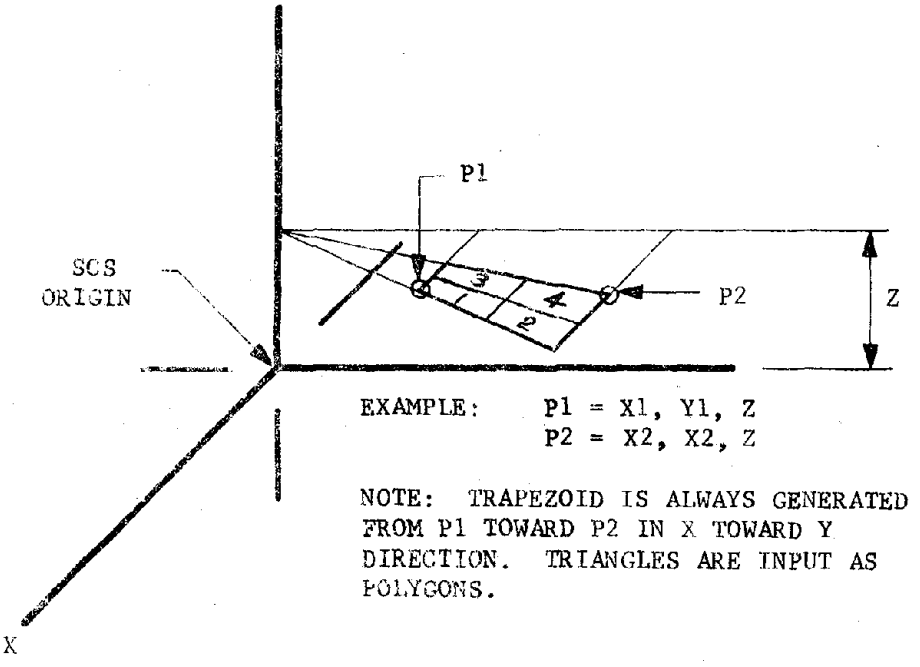


Figure 3-7 Surface Geometry Definition

TRAPEZOID

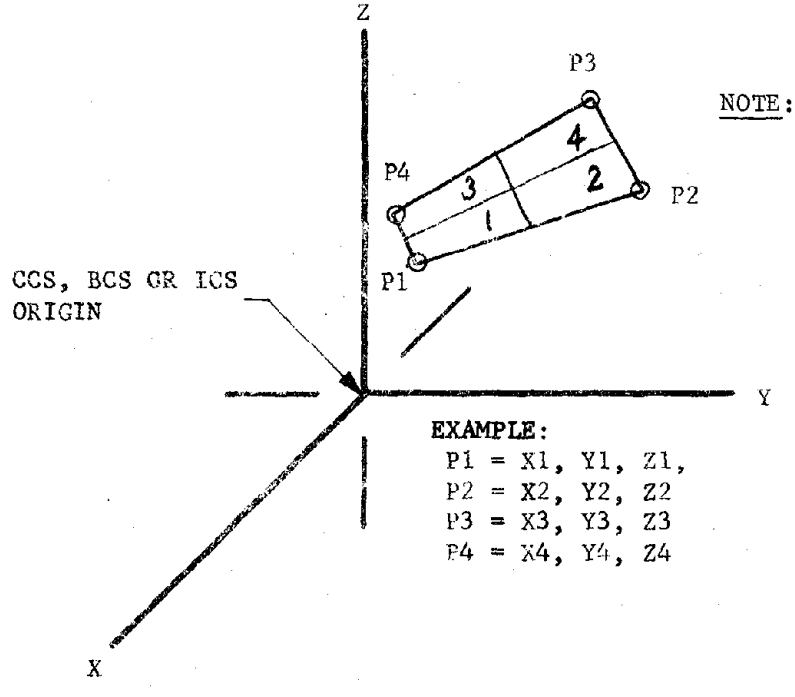
SCS METHOD



EXAMPLE: P1 = X1, Y1, Z
P2 = X2, X2, Z

NOTE: TRAPEZOID IS ALWAYS GENERATED FROM P1 TOWARD P2 IN X TOWARD Y DIRECTION. TRIANGLES ARE INPUT AS POLYGONS.

POINT METHOD



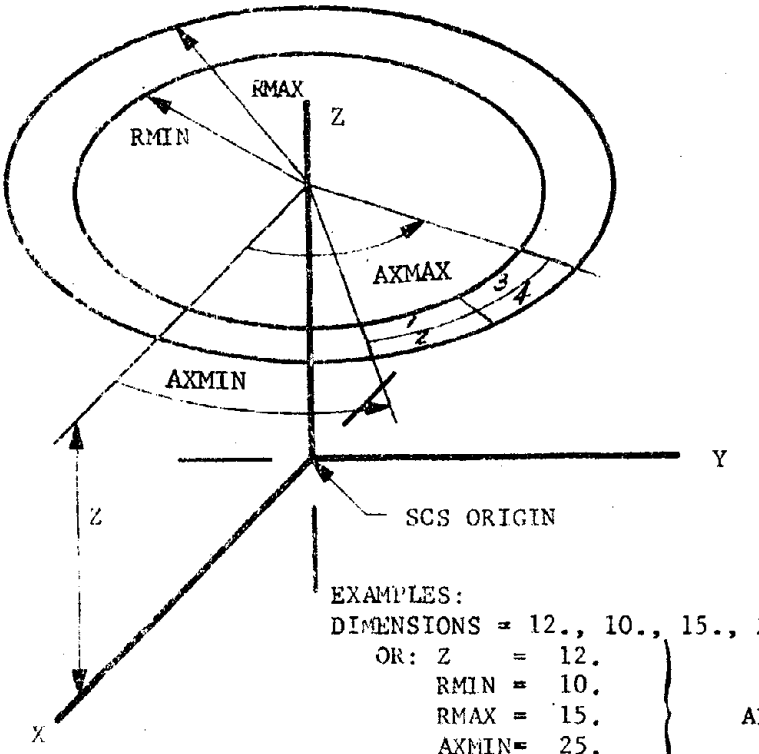
NOTE: POINTS ARE NUMBERED CCW AROUND FIGURE AS VIEWED FROM ACTIVE SIDE WHEN ACTIVE = TOP. P1-P4 MUST BE SHORTER THAN AND PARALLEL TO P2-P3.

EXAMPLE:
P1 = X1, Y1, Z1,
P2 = X2, Y2, Z2
P3 = X3, Y3, Z3
P4 = X4, Y4, Z4

Figure 3-7. (cont)

DISC

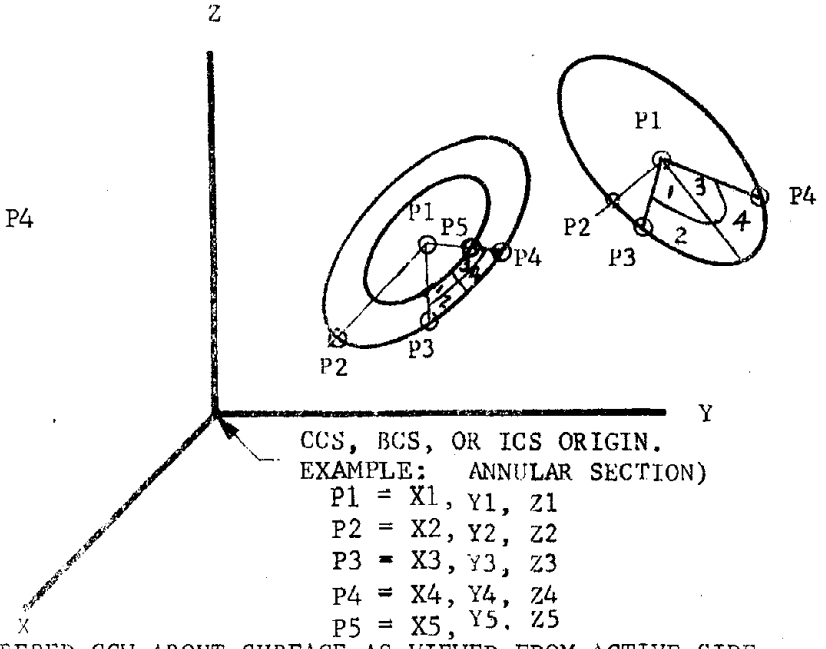
SCS
METHOD



EXAMPLES:
DIMENSIONS = 12., 10., 15., 25., 45., (STANDARD)
OR: Z = 12.
RMIN = 10.
RMAX = 15.
AXMIN= 25.
AXMAX= 45.
} ALTERNATE

EXAMPLE: (PIE-SECTION)
P1 = X1, Y1, Z1
P2 = X2, Y2, Z2
P3 = X3, Y3, Z3
P4 = X4, Y4, Z4
NOTE:
P1 = DISC CENTER, P2, P3, & P4
ENTERED CCW, AS SEEN FROM
ACTIVE SIDE

POINT
METHOD



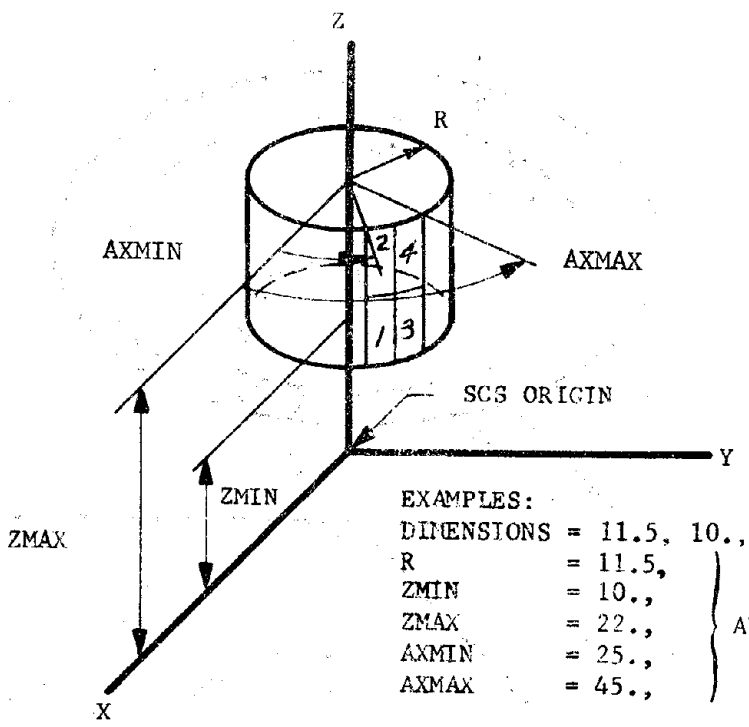
CCS, BCS, OR ICS ORIGIN.
EXAMPLE: ANNULAR SECTION)
P1 = X1, Y1, Z1
P2 = X2, Y2, Z2
P3 = X3, Y3, Z3
P4 = X4, Y4, Z4
P5 = X5, Y5, Z5

NOTE: P1, P3, P4, AND P5 NUMBERED CCW ABOUT SURFACE AS VIEWED FROM ACTIVE SIDE, WHEN ACTIVE = TOP. LINE P2-P1 MUST BE PERPENDICULAR TO LINE P4-P1 AND ALL OR PART OF THE ACTIVE SURFACE MUST LIE BETWEEN P2 AND P4.

Figure 3-7. (cont.)

CYLINDER

SCS METHOD



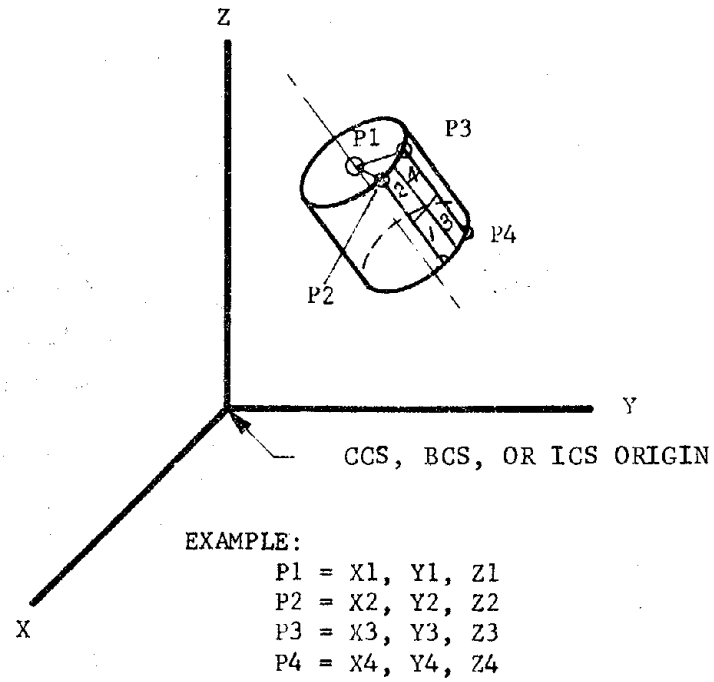
EXAMPLES:

DIMENSIONS = 11.5, 10., 22., 25., 45.

ALTERNATE:

R = 11.5,
ZMIN = 10.,
ZMAX = 22.,
AXMIN = 25.,
AXMAX = 45.,

POINT METHOD



EXAMPLE:

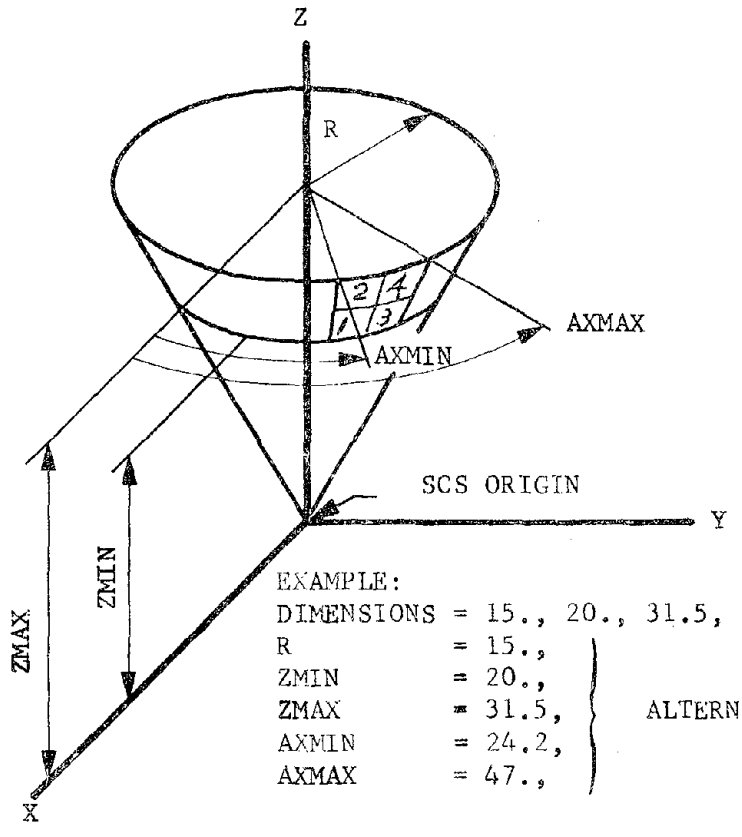
P1 = X1, Y1, Z1
P2 = X2, Y2, Z2
P3 = X3, Y3, Z3
P4 = X4, Y4, Z4

NOTE: SURFACE GENERATED FROM P2 TO P3 CCW ABOUT AXIS AS VIEWED FROM P1 END.

Figure 3-7. (cont)

CONE

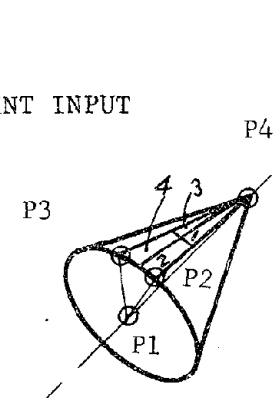
SCS METHOD



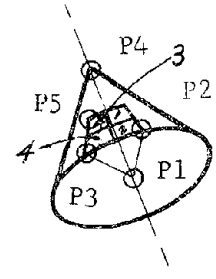
EXAMPLE:
DIMENSIONS = 15., 20., 31.5, 24.2, 47. (STD)
R = 15.,
ZMIN = 20.,
ZMAX = 31.5, } ALTERNATE
AXMIN = 24.2,
AXMAX = 47.,

POINT METHOD

FOUR POINT INPUT



FIVE POINT INPUT



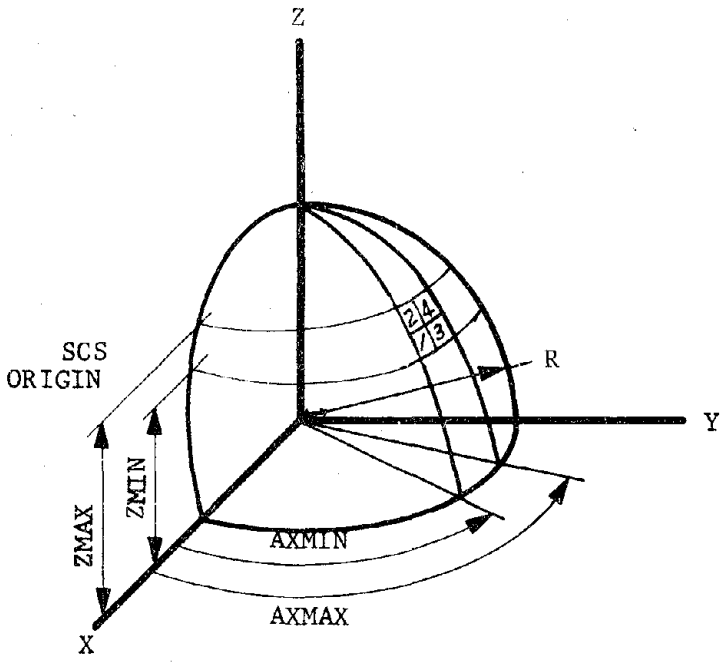
EXAMPLE:
P1 = X1, Y1, Z1
P2 = X2, Y2, Z2
P3 = X3, Y3, Z3
P4 = X4, Y4, Z4

NOTE: SURFACE GENERATED FROM P2 TO P3, CCW ABOUT AXIS AS VIEWED FROM P1 TOWARD APEX

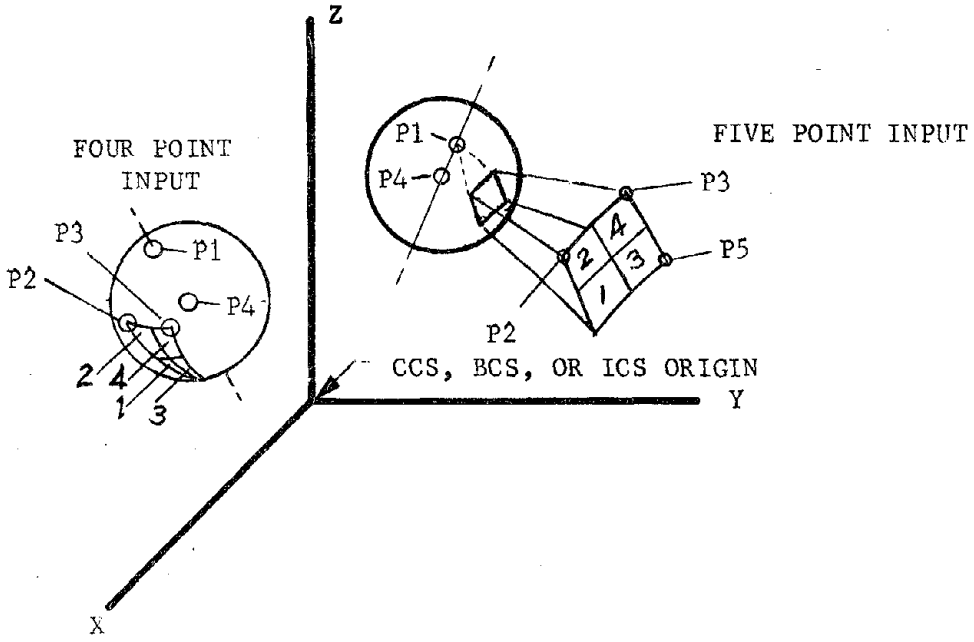
Figure 3-7. (cont)

SPHERE

SCS METHOD



POINT METHOD



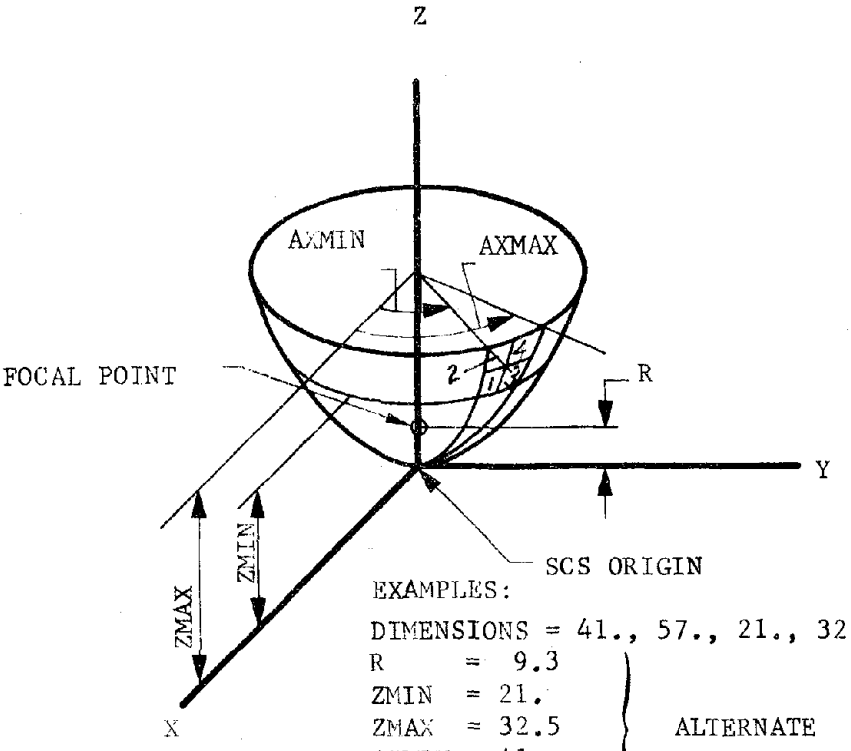
NOTES: P2 AND P3 LIE ON A "LATITUDE" LINE. SURFACE IS GENERATED FROM P2 TO P3, CCW AS VIEWED FROM P1 TOWARD P4, P3 AND P5 LIE ON A "LONGITUDE" LINE.

A COMPLETE SPHERE IS GENERATED FROM 3 POINTS: P1 - NORTH POLE, P2 - CENTER; P3 - POINT ON EQUATOR WHERE NODE GENERATION BEGINS.

Figure 3-7. (cont)

CIRCULAR PARABOLOID

SCS METHOD



EXAMPLES:

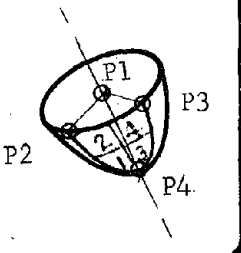
DIMENSIONS = 41., 57., 21., 32.5, 9.3 (STD.)

ALTERNATE

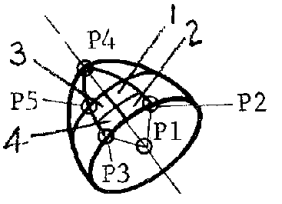
R = 9.3
ZMIN = 21.
ZMAX = 32.5
AXMIN = 41.
AXMAX = 57.

FOUR POINT INPUT

POINT METHOD



FIVE POINT INPUT



EXAMPLE:

P1 = X1, Y1, Z1 (APEX OF PARABOLOID).
P2 = X2, Y2, Z2
P3 = X3, Y3, Z3
P4 = X4, Y4, Z4
P5 = X5, Y5, Z5

NOTE: P1 AND P4 DEFINE AXIS OF REVOLUTION. SURFACE IS GENERATED FROM P2 TO P3 CCW AS VIEWED FROM P1 TOWARD P4

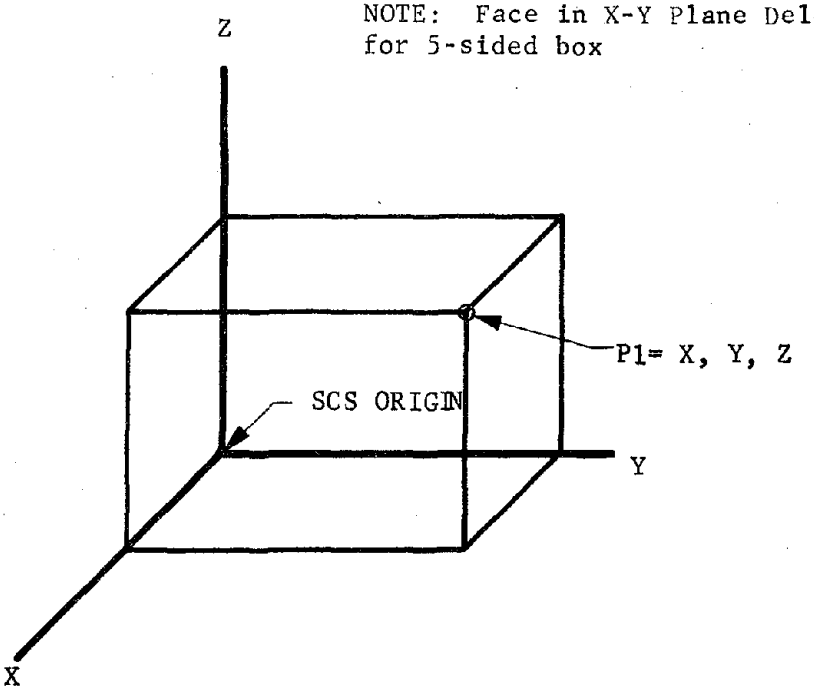
Figure 3-7. (cont)

FIVE AND SIX SIDED BOXES

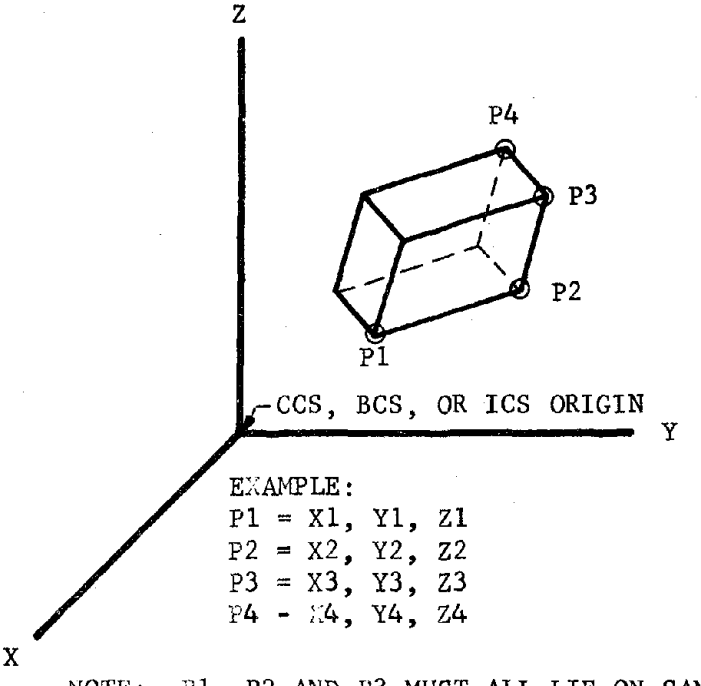
NOTE: ACTIVE = BOTH NOT ALLOWED

NOTE: Face in X-Y Plane Deleted for 5-sided box

SCS METHOD



POINT METHOD



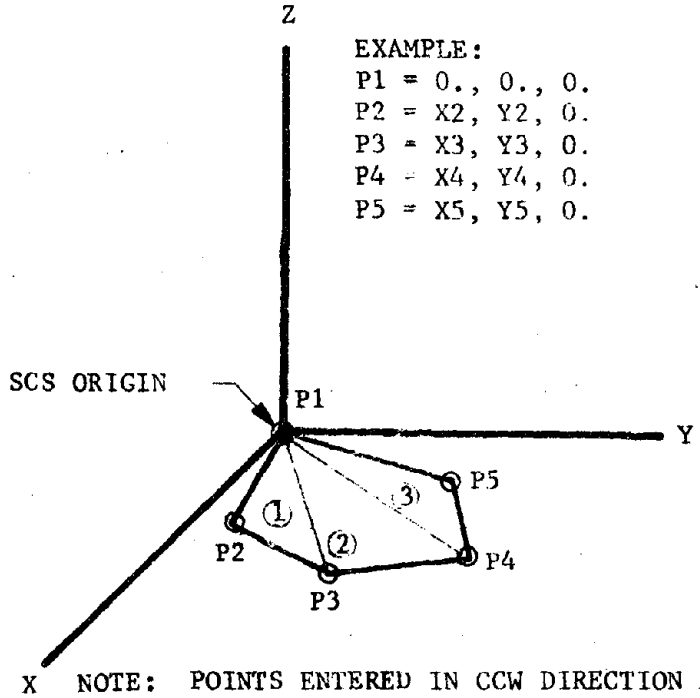
EXAMPLE:
P1 = X1, Y1, Z1
P2 = X2, Y2, Z2
P3 = X3, Y3, Z3
P4 - X4, Y4, Z4

NOTE: P1, P2 AND P3 MUST ALL LIE ON SAME FACE OF FIGURE, WITH P4 DIAGONALLY OPPOSITE P1. FACE CONTAINING P1, P2 and P3 DELETED FOR 5-SIDED BOX.

Figure 3-7. (cont)

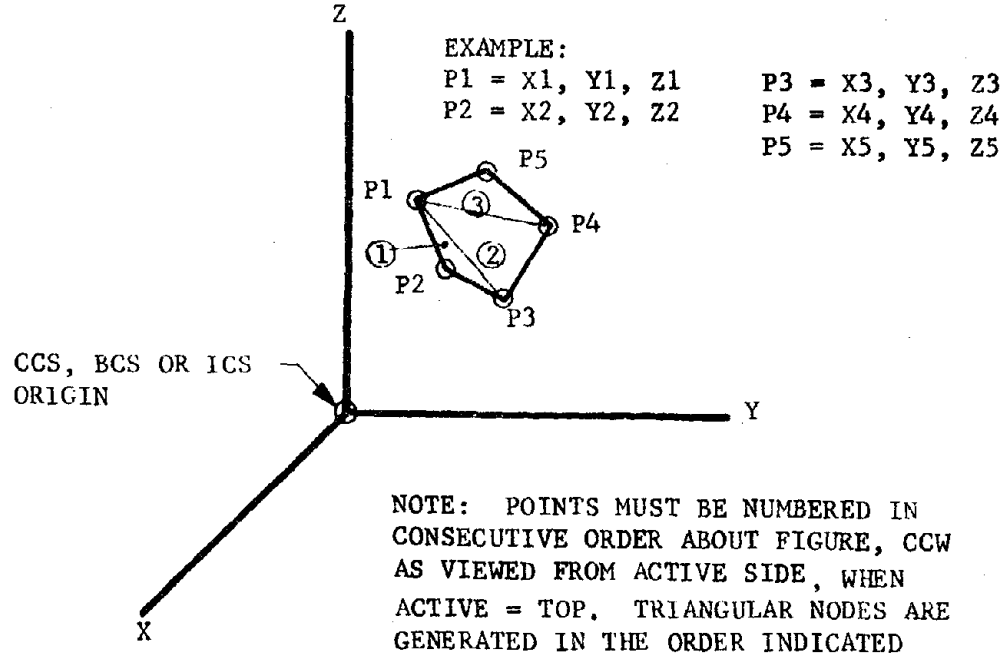
N-SIDED POLYGON

POINT
METHOD 1



NOTE: POINTS ENTERED IN CCW DIRECTION ABOUT FICURE AS VIEWED FROM ACTIVE SIDE WHEN ACTIVE = TOP. MAXIMUM N ALLOWED IS 15. TRIANGULAR NODES ARE GENERATED IN THE ORDER INDICATED BY THE CIRCLED NUMBERS.

POINT
METHOD 2



NOTE: POINTS MUST BE NUMBERED IN CONSECUTIVE ORDER ABOUT FIGURE, CCW AS VIEWED FROM ACTIVE SIDE, WHEN ACTIVE = TOP. TRIANGULAR NODES ARE GENERATED IN THE ORDER INDICATED BY THE CIRCLED NUMBERS.

Figure 3-7. (concl)

to each geometric figure. Both the surface coordinate system methods and point methods of input are shown. A careful study of Table 3-III and Figure 3-7 will pay dividends to the new TRASYS user.

The variables found in the surface data block can be grouped as follows:

- general data
- dimensional data
- properties data
- position data
- ICS definition data

A summary of the function of each data group follows:

General data - this "catch all" data group is used to define:

1) node identification numbers
2) surface type (disc, sphere, etc.)
3) active side information
4) shadowing information, and
5) nodal breakdown and dimension information.

Dimensional data - This data group is used to define the desired boundaries of the geometric surfaces and portions of same.

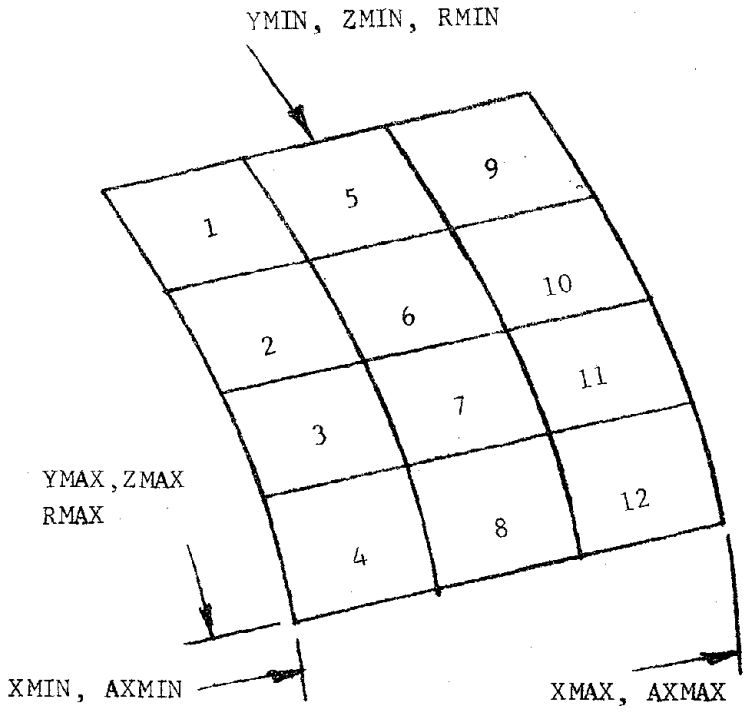
Properties data - This data group is used to define the optical properties of the surfaces. Properties allowed are diffuse solar absorptivity and transmissivity, diffuse infrared emissivity and transmissivity, specular solar reflectivity, and specular infrared reflectivity.

Position data - These data are the six rotation and translation variables necessary to locate an SCS relative to an ICS, BCS or CCS.

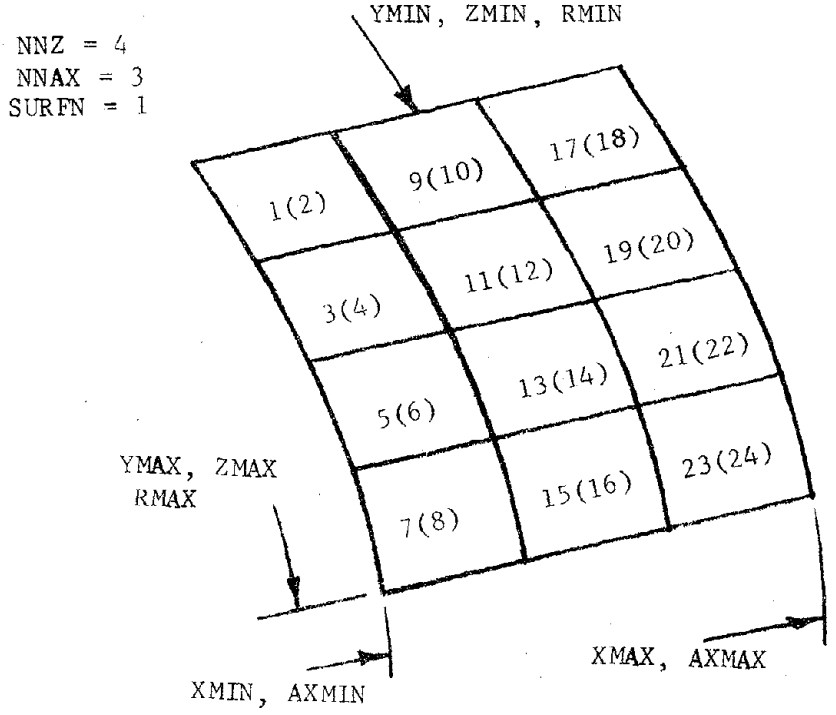
3.3.3.6 Nodal Surface Identification

When a surface is subdivided into nodal surfaces, the user has the option of numbering the nodal surface consecutively, beginning with the identification number he used for the surface, or arbitrarily, using a node number array. In either case, he must understand the scheme used by TRASYS to identify nodal surfaces. Figure 3-8 illustrates this process with examples of surfaces with single and dual active sides.

The user will no doubt quickly discover that Figure 3-8 shows node numbering schemes that are related to the SCS-referenced method but have no relation to the CCS-referenced point method. The number scheme functions with point input, however, because the first step in processing a point-defined surface is to provide it with an internally generated SCS. Once this is done, the node numbering scheme can proceed. It is necessary, therefore, for the user to understand how the internally generated surface coordinate system relates to his point input. This is illustrated for each surface type in Figure 3-9.

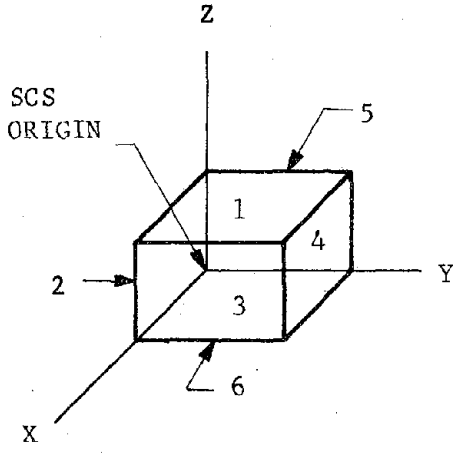


SINGLE ACTIVE SIDE. (VIEW FROM OUTSIDE OR TOP)

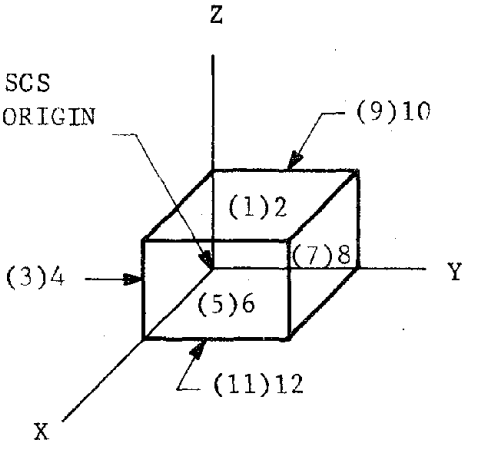


BOTH SIDES ACTIVE. ODD NUMBERED NODES ON INSIDE OR BOTTOM. EVEN NUMBERED NODES ON OUTSIDE OR TOP.

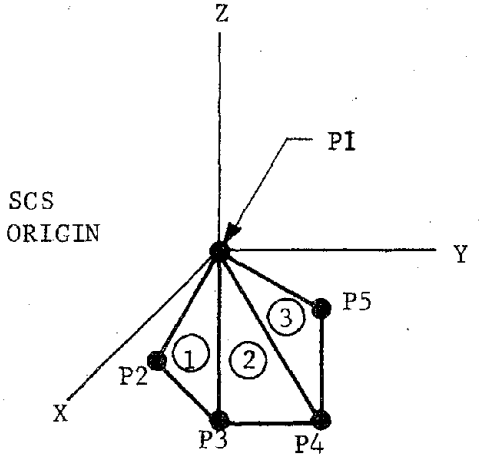
Figure 3-8 Node Generation Order



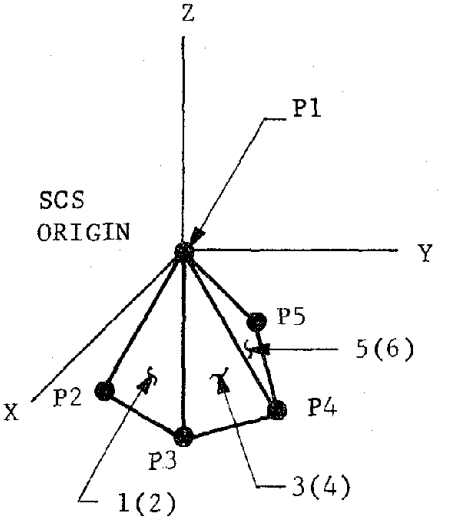
ACTIVE = INSIDE
OR OUTSIDE



ACTIVE = BOTH: ODD NUMBERED NODES INSIDE. EVEN NUMBERED NODES OUTSIDE.



ACTIVE = TOP
OR BOTTOM



ACTIVE = BOTH: EVEN NUMBERED NODES ON TOP (+Z) SIDE, ODD NUMBERED NODES ON BOTTOM

Figure 3-8. (concl)

An understanding of Figure 3-8 and 3-9 should enable the user to properly number his nodal surfaces when using point input to define the surface.

The user should realize that the generalized node breakdown schemes involving NNX, NNY, UNNX, UNNY, etc., do not pertain to the BOX and POLYGON surface types where a fixed node generation scheme exists. If a user desires to subdivide the faces of a box, the faces must be input as rectangles. Subdividing a polygon requires entering the individual triangles desired. The user may wonder at the capricious-looking node breakdown that results from his polygon input. This occurs because shadowing solutions exist for triangles, but not polygons. After processing, the polygon's triangles are combined for output as one node, but the user should be aware of this subdivision process in order to avoid duplication of node numbers.

3.3.3.7 Dimensional Units

Nodal areas are carried in data storage for direct irradiation and radiation conductor calculations. For this reason, surface data length inputs must be in feet, the standard TRASYS length unit. Convenient means of units control are provided by D-cards (Ref 3.3.3.12).

In regard to the surface data block, the user needs to remember that all the linear dimensions he uses in defining the surfaces of his model must be in feet (after D-card manipulations) and that all angular measurements must be in degrees (and decimal fractions of degrees) of arc.

3.3.3.8 Properties Data

In its present state of development, TRASYS is restricted to the assumptions that all surfaces are "grey", all surfaces emit diffusely, and all surfaces reflect with diffuse and specular components of reflectance.

$$\alpha_{IR} + \rho_{IR} = \rho^{s}_{IR} + \tau_{IR} = 1.0$$

$$\alpha_{s} + \rho_{s} + \rho_{s}{}^{s} + \tau_{s} = 1.0$$

where:

α = diffuse absorptivity

ρ = diffuse reflectivity component

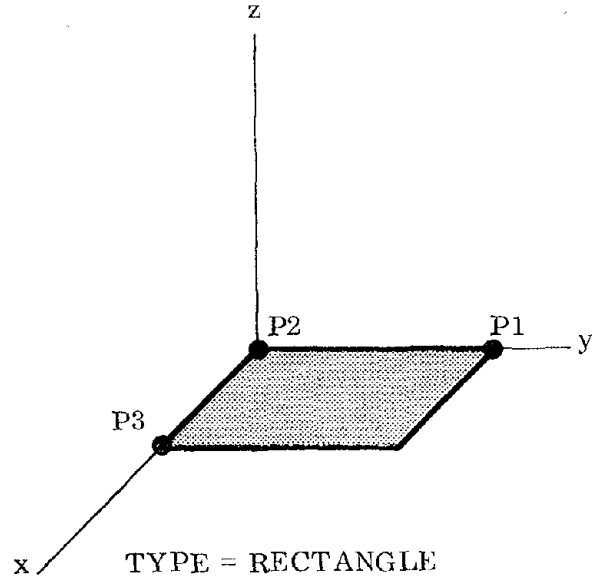
ρ^{s} = specular reflectivity component

τ = transmissivity

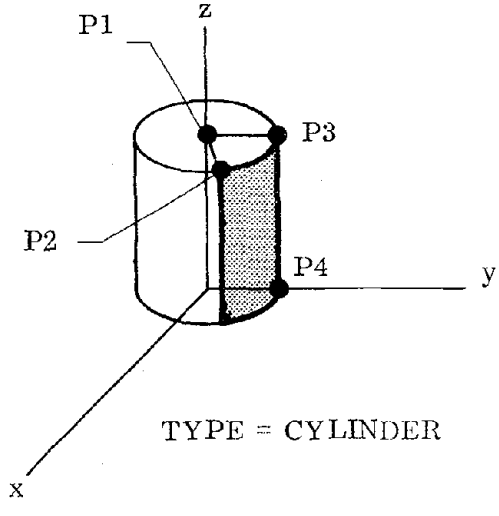
subscripts:

IR = infrared waveband

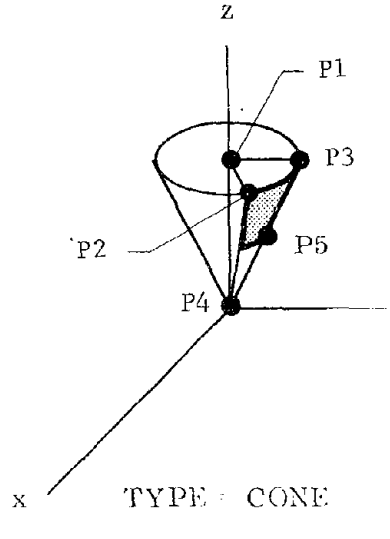
s = solar waveband



TYPE = RECTANGLE



TYPE = CYLINDER



TYPE = CONE

It might be observed that since semitransparent materials and specular surfaces are allowed, the form factors are not, in general, surface property independent but a function of the transmissivities and specular components of reflectivity.

Material transmissivity plays a part in form factor calculations where blockage by a semitransparent surface is involved. The assumption made for these calculations is that any element to element configuration factor with an intervening semitransparent surface is multiplied by a shadow factor equal to the value of the blocking surface's transmissivity. This is a reasonable approach for thin intervening bodies.

Since only surfaces, rather than bodies, are used in TRASYS calculations, only one face of the semitransparent body will "count" as a shadower. If two surfaces are input for one body, the square root of the transmissivity can be used as the shadow factor to avoid having the shadowed configuration factors erroneously multiplied by the square of the transmissivity. In this case, the user enters a negative transmissivity value. This is detected by the program and the absolute value of the square root of the transmissivity used as the shadow factor.

Specular reflectivity comes into play in the form factor calculations as a result of the imaging techniques used and the definition of an "image factor" as described in Appendix I.

It might be noted that the presence of semitransparent surfaces where $\tau_{IR} \neq \tau_s$ and/or the presence of specular surfaces where $\rho^s_{IR} \neq \rho^s_s$ results in separate form factor matrices for the infrared and the solar wavebands. Both of these matrices are carried in program data storage and are printed in the standard output.

3.3.3.9 Surface Data Format

3.3.3.9.1 Control Field Formats

Four different types of cards containing control field information are allowed in the surface data block. The card types are:

New Surface Card (S Card)
BCS Identifier Card (B Card)
ICS Definition Card (I Card)
Constant Definition Card (K Card)
Comment Card

S. Cards are used to signal the completion of the input for a surface and the beginning of a new surface. Their general format is:

CC1	CC7	CC7
		3
S	Any Surface Data	Card ID Information

Any data encountered beginning with an S card and ending with the card preceding the next S card is presumed to apply to a single surface and is defined as a surface description. If insufficient data to define a surface is found between two S cards, either default will be supplied or an error message results. If redundant data is entered an error message results.

B cards are used to identify surfaces with the desired block coordinate system. Their general format is:

CC1	CC7	CC7
		3
BCS	Block Coordinate System Name	Card ID

All surface descriptions encountered between two B cards will be keyed to the BCS name found in the leading B card. Any surfaces not preceded by a B card will be automatically keyed to a block coordinate system named ALLBLK. BCS ALLBLK has zero rotation and translation parameter values. That is, it coincides with the CCS.

I cards are used for definition of intermediate coordinate systems. Their general format is:

CC1	CC7	CC7
		3
I	Intermediate Coordinate System Data	Card ID

Continuation cards are allowed for ICS definition. In other words, all information encountered between an I card and another card containing information in CC1 (except for comment cards) is presumed to pertain to a single ICS.

NOTE: A general surface data deck structure rule is that all I cards must precede all S cards.

K cards are used for definition of user constants referred to in surface data. Their general format is:

CC1	CC7	CC7
		3
K	Constants Data	Card ID

Continuation cards are allowed for constants definition. All cards between a K card and the next card with data in CC1 (excepting comment cards) can be thought of as a data subblock that defines surface data constants.

NOTE: A general surface data deck structure rule is that all K cards must precede all S cards.

It should be noted that a basic difference exists between K-card constants and constants entered in the quantities data block. Unlike quantities data constants, K-card constants are used for surface data manipulation only, and are not available during processor execution.

Comment cards, with the following format:

CC1	CC7	CC7
		3
C	Comment Information	Card ID

may appear anywhere in the surface data block. Another means of entering comment information is to delimit a data field (CC7-72) with a $ and enter comment information to the right of it, as follows:

CC7	CC7
	3
Surface Data $ Comment Data	Card ID

This may tempt the user to place a $ in the data field of an otherwise blank card and enter a comment. This is illegal. It results in a blank data field and an error message.

3.3.3.9.2 Single Variable Input Format

Any single variable recognized as surface data may be entered in a card data field according to the general format:

CC7	CC7
	2
NAME1 = DV, NAME2 = DV, ---	

NAME1 and NAME2 may be any variable name defined by the surface data variables list (Table 3-I), plus in the case of K cards, the names may be as defined by the user, limited only by the mode and word length limit of 6 characters.

Single variable input is the only means available for defining the following list of surface data variables:

TYPE		NNX
ACTIVE		NNY
SHADE		NNAX
BSHADE		ICSN (in a surface description)
SPRI, SPRS	Note 1	SURFN
		IREFSF
		IDUPSF
		IMAGSF
		REFNO

Note: 1. Values of either or both of these variables greater than zero requires:

NNX = NNY = 1 (one node allowed per specular surface)
TYPE = RECT, DISC, TRAP, BOX5, BOX6, or POLY (specular surfaces must be planar)
SHADE = FF or BOTH (specular surfaces must be shadowers in FF segment. This flag is reset by the program to "FF" if unspecified or specified as "NO" and is reset to "BOTH" if specified as "DI.")

All other surface data variables may be defined in convenient "short form" array formats per subsections 3.3.3.9.3 through 3.3.3.9.

3.3.3.9.3 Intermediate Coordinate System Data Format

ICS data may be entered in array format as follows:

```
CC1          CC7                                                    CC7
                                                                      2
  I            ICSN, TX, TY, TZ, ROTX, ROTY, ROTZ
```

This array defines the translations and rotations necessary to transform a BCS (or CCS into the ICS. The three rotations, ROTX, ROTY, and ROTZ are performed in that order. If it is desired to alter the order of the rotations, the following hybrid format is used:

```
CC1          CC7                                                    CC7
                                                                      2
  I            ICSN, TX, TY, TZ, ROTY = DV,
               ROTZ = DV, ROTX = DV
```

for rotation first about the BCS Y-axis, second about the BCS Z-axis and third about BCS X-axis.

It should be noted that each ICSN value appears at least twice in the surface data block. Once in the I card defining the ICS, and again in each surface description where an SCS/ICS transform is desired.

3.3.3.9.4 Surface Identification Format

A single node surface is identified as follows, in single variable input format:

```
CC7                                                                 CC7
                                                                      2
  SURFN = DV (Integer)
```

If a surface is to be subdivided into several nodes, and they are not to be numbered consecutively, the node number array may be entered according to the format:

```
CC7                                                                 CC7
                                                                      2
  SURFN = DV1, DV2, ---DVN
```

for an N-node surface.

3.3.3.9.5 Properties Data Format

The diffuse properties data may be defined using the following format:

```
CC7                                                    CC7
                                                         2
  PROP = ALPHA, EMISS, TRANI, TRANS
```

If values for TRANI and TRANS are not encountered, they will default to zero.

Specular properties data must be input in the single variable format (see 3.3.3.9.2).

3.3.3.9.6 Dimensions Data Format

The dimensions data may be defined using the following format:

```
CC7                                                    CC7
                                                         2
  DIMEN = R, ZMIN, ZMAX, AXMIN, AXMAX
```

3.3.3.9.7 Point Data Format

The x, y, z coordinates of point data input are defined using the following format:

```
CC7                                                    CC7
                                                         2
  PN = XN, YN, ZN
```

N values up to 15 are recognized, depending on the surface type (Ref. Figure 3-4).

This is the only format allowed for point data. Single variable definitions are not allowed.

3.3.3.9.8 Position Data Format

The position data may be defined using the following format:

```
CC7                                                    CC7
                                                         2
  POSIT = TX, TY, TZ, ROTX, ROTY, ROTZ
```

This array defines the translations and rotations necessary to transform an ICS, BCS, or CCS into the SCS. The three rotations ROTX, ROTY, and ROTZ are performed in that order. If it is desired to alter the order of the rotations, the following format is used:

```
CC7                                                    CC7
                                                         2
  POSIT = TX, TY, TZ, ROTY = DV, ROTZ = DV, ROTX = DV
```

3.3.3.9.9 Comment Data Format

A Hollerith string of up to thirty characters may be entered with each surface description according to the following format:

CC7 CC7
2

COM = * Any Alphameric Data *

These comments will be passed to the processor and printed with the standard surface description output.

3.3.3.9.10 Node Boundary Dimensions

When it is desired to generate an unequal node breakdown on a surface, it is necessary to define the node boundaries using one or more of the UNNX, UNNY, UNNZ, UNNAX and UNNR arrays. Figure 3-10 illustrates this scheme for NNZ= 2, NNAX=3.

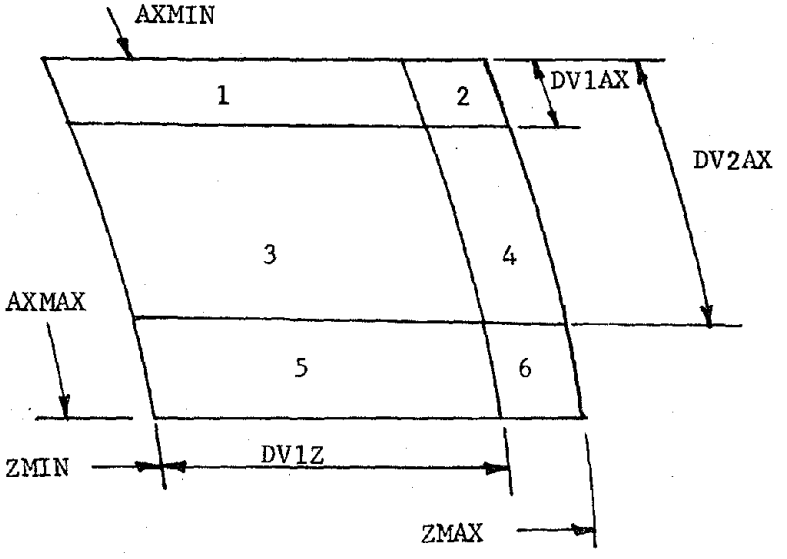


Figure 3-10 Examples of Unequal Node Boundaries

This example required the following unequal boundary arrays:

UNNAX = DV1AX, DV2AX
UNNZ = DV1Z

These arrays are entered in the surface data block according to the following format:

```
CC7                                                          CC7
                                                               2

  UNNAX = DV1AX, DV2AX
  UNNZ = DV1Z
```

The general format is:

```
CC7                                                          CC7
                                                               2

  UNNX = DV1, DV2, . . . DVN
```

where: N = NNX -1.

3.3.3.10 DUP Surfaces

The DUP option enables the user to create surfaces by duplicating previously input surfaces. The following restrictions and rules apply when using the DUP option:

a) Surfaces to be duplicated must appear in the surface data block before the surfaces that are to be created by duping.
b) Any or all of the surface description variables of the surface being duplicated can be changed.
c) If the surface to be duplicated was input by the point method and any changes are to be made in the points, all points must be input for the surface being created.
d) Generated surfaces such as boxes and polygons will be duped in their entirety. That is, the individual nodes generated by "box" or "polygon" cannot be duped.
e) Surfaces created by the DUP option may later be imaged (3.3.3.11).
f) The DUP surface and the surface DUPed must be defined with respect to the same ICS, BCS or CCS.
g) The surface to be duplicated is specified by setting the variable IDUPSF equal to the surface number.

3.3.3.10.1 DUP Option Example

A sample input deck using the DUP option is shown in Figure 3-11.

3.3.3.11 IMAGE Surfaces

The IMAGE option allows the user to create surfaces by imaging previously input surfaces in some specified reference plane. The following restrictions and rules apply when using the IMAGE option:

a) Surfaces to be imaged must appear in the surface data block before the surfaces that are to be created by imaging.
b) Reference planes (imaging planes) in which surfaces are to be imaged are special surfaces designated by R-Cards. These cards must appear in the same BCS sub-block as the surface(s) being imaged and the resultant image surface(s). Each of these planes is

```
HEADER OPTIONS DATA
TITLE  DUP OPTION SAMPLE PROBLEM
HEADER SURFACE DATA
S      SURFN          =10
       TYPE           =SPHERE
       R              =10.
       ZMIN           =-9.99
       ZMAX           = 9.99
       AXMIN          =0.
       AXMAX          =360.
       TX             =0.
       TY             =10.
       TZ             =20.
       ACTIVE         =OUT
       PROP           =0.2,0.9
       COM            =* SURFACE TO BE DUPED *
S      SURFN          =20
       IDUPSF         =10
       R              =20.
       ZMIN           =-19.99
       ZMAX           = 19.99
       TY             =-20.
       PROP           =0.5,0.8
       COM            =* DUPLICATE OF SURFACE 10 *
HEADER OPERATIONS DATA
STEP   1
       CALL BUILDC(ALLBLK)
L      NPLOT
END OF DATA
```

Figure 3-11 Sample Problem Using the DUP Option

assigned a unique identification number and is defined by specifying any three non-colinear points lying on its surface. These points are defined with respect to an ICS, a BCS or the CCS.

c) Generated surfaces such as boxes and polygons will be imaged in their entirety. That is, individual nodes generated by "box" or "polygon" cannot be imaged.

d) Surfaces created by imaging cannot be duped.

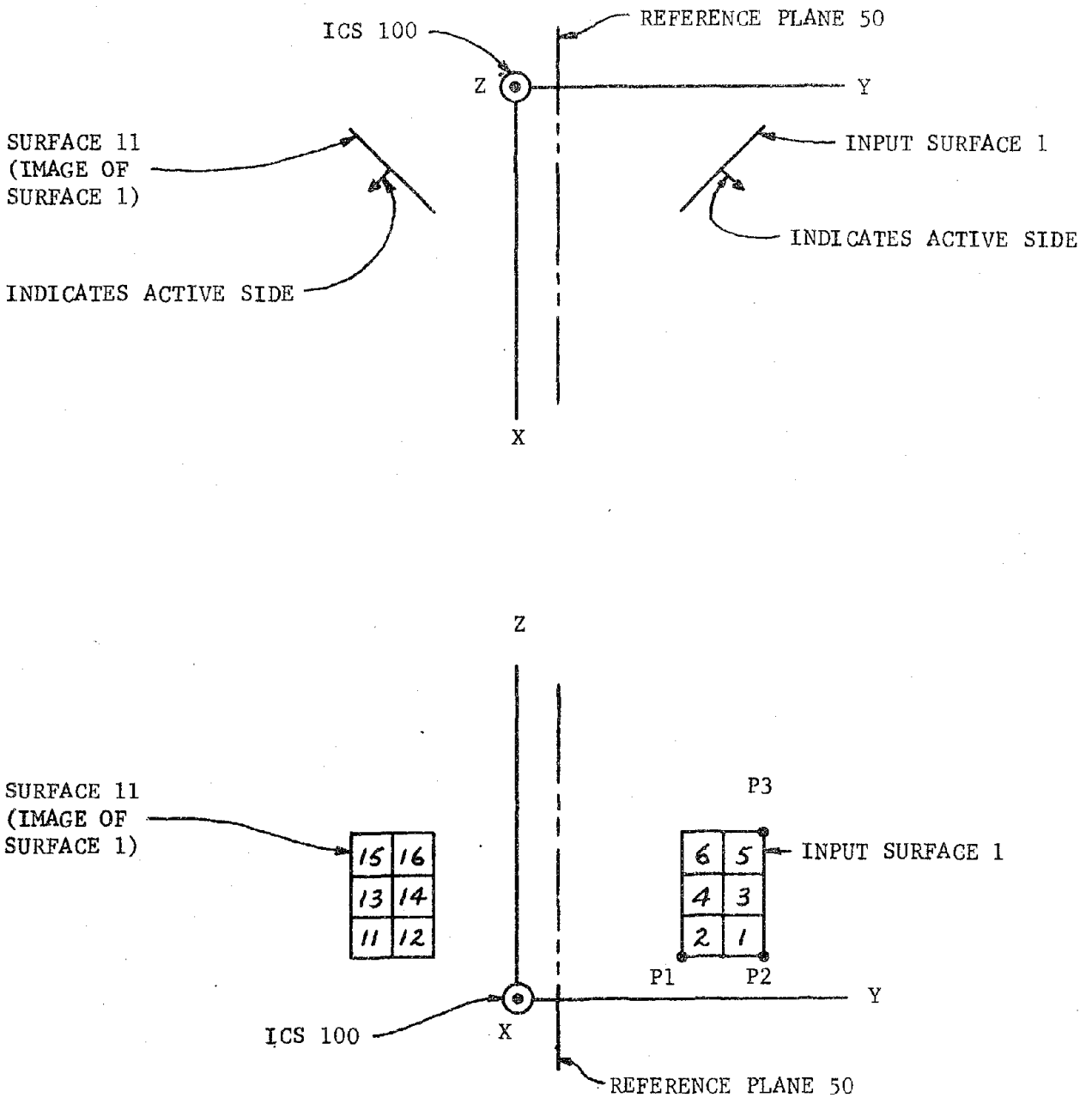
e) The image surface, the image plane and the surface imaged must all be defined with respect to the same ICS, BCS or the CCS.

f) The surface to be imaged is specified by setting the variable IMAGSF equal to the imaged surface number. The reference plane in which IMAGSF is to be imaged is specified by setting IREFSF equal to the reference plane number.

g) When a surface is imaged, the nodes are also imaged resulting in a reversed order of node numbering. The active side of the surface also follows image rules. Figure 3-12 illustrates these phenomena.

3.3.3.11.1 IMAGE Option Example

A sample input deck using the IMAGE option is shown in Figure 3-13.



NOTE: NODE X+10 IS THE IMAGE OF NODE X

Figure 3-12 Imaging of Nodes and Active Side (Image Option Sample Problem)

```
HEADER OPTIONS DATA
TITLE  IMAGE OPTION SAMPLE PROBLEM
HEADER SURFACE DATA
ICS    100,0.0,10.0,5.0,0.,80.0,90.0
S      SURFN       = 1
       TYPE        = RECT
       ACTIVE      = TOP
       NNX         = 3
       NNY         = 2
       PROP        = 0.5,0.9
       P1          = 3.0, 4.0, 1.0
       P2          = 1.0, 6.0, 1.0
       P3          = 1.0, 6.0, 4.0
       ICSN        = 100
       COM         = *SURFACE TO BE IMAGED*
S      SURFN       = 11
       IMAGSF      = 1
       IREFSF      = 50
       ICSN        = 100
       COM         = *IMAGE OF SURFACE 1*
R      REFNO       = 50
       P1          = 0.0, 1.0, 0.0
       P2          = 0.0, 1.0, 1.0
       P3          = 1.0, 1.0, 1.0
       ICSN        = 100
       COM         = *REFERENCE PLANE*
HEADER OPERATIONS DATA
STEP   1
       CALL BUILDC(ALLBLK)
L      NPLOT
END OF DATA
```

Figure 3-13 Sample Problem Using the Image Option

3.3.3.12 Linear Dimension Units Control

Since TRASYS computations cannot be made independent of dimensional units, it was necessary to choose a standard units system for compatibility between the various computation segments and subroutines. The TRASYS standard length unit is feet, which is oftentimes inconvenient when the user is working from engineering drawings in inches, or perhaps a metric unit. This problem has been eliminated by allowing for dimension change (D-cards) in the surface data input. These cards function as follows: when a D-card is encountered in the surface data, all linear dimensions in the surface (S-card) data following will be multiplied by the floating point data value on the D-card. This holds true until another D-card is encountered or until the end of the surface data block. All intermediate coordinate systems referenced by surfaces being modified by a D-card are also modified by the D-card. This means that the following rule must be observed carefully: the linear dimensions on any ICS referred to in a surface description must agree with the linear dimensions of the pertinent surface data, prior to modification by a D-card.

3.3.3.12.1 D-Card Formats

The D-Card format is:

CC1	CC7
D	DV (floating point)

3.3.3.12.2 D-Card Operations Example

A surface data block using D-Cards is shown in Figure 3-14

```
HEADER OPTIONS DATA
TITLE  CLASSES FOLLY
       MODEL=CLAS
       SHADO=TXXXX
HEADER SURFACE DATA
D      1./12.  $ FOLLOWING LINEAR DIMENSIONS ARE MULTIPLIED BY 1./12.
S      TYPE=CYL,
       SURFN=201,
       R=1.0,ZMIN=3.0,ZMAX=15.0,AXMIN=0.0,AxlAX=360.0,NNZ=1,NNAX=1,
       ACTIVE=OUT, ALPHA=0.3,EMISS=0.9,
       TY=5.0,
S      TYPE=SPHER,
       SURFN=301
       R=3.0,ZMIN=0.0,ZMAX=3.0,AXMIN=0.0,AXMAX=180.0,NNZ=1,NNAX=1,
       TZ=20.,
       ACTIVE=OUT,ALPHA=0.2,EMISS=0.9,
S      TYPE=CONE,
       SURFN=401,
       R=1.0, ZMIN=0.0,ZMAX=2.0,AXMIN=0.0,AXMAX=360.0,NNZ=1,NNAX=1,
       TZ=17.0,ROTY=180.,
       TY=5.0,
       ACTIVE=OUT,ALPHA=0.9,EMISS=0.9,
S      TYPE=CONE,
       SURFN=501,
       R=3.0,ZMIN=1.0,ZMAX=3.0,AXMIN=0.0,AXMAX=180.0,NNZ=1,NNAX=1,
       TZ=2.0,
       ACTIVE=OUT,ALPHA=0.9,EMISS=0.9,
D       1.   $ TERMINATES EFFECT OF PREVIOUS D-CARD
N       10   $ REMAINING NODE NUMBERS ARE INCREASED BY 10
S      TYPE=CONE,
       SURFN=701,
       R=1.0,ZMIN=1.0,ZMAX=2.0,AXMIN=0.0,AXMAX=360.0,NNZ=1,NNAX=1,
       TZ=1.0,TY=5.0,
       ACTIVE=OUT,ALPHA=0.9,EMISS=0.9,
S      TYPE=TRAP,
       SURFACE=901,
P      P1=0.707*4.0,0.707*5.0,4.0,
       P2=0.707*3.0,0.707*3.0,5.0,
       P3=0.707*3.0,0.707*3.0,8.0,
       P4=0.707*5.0,0.707*5.0,6.0,
       ACTIVE=BOTH,ALPHA=0.2,EMISS=0.9,
S      SURFN    =905,TYPE=POLY
       P1=-.707*5.,.707*5.,4.
       P2=-.707*3.,.707*3.,5.
       P3=0.707*3.0,0.707*3.0,8.0,
       P4=0.707*5.,.707*5.6.
       ACTIVE=BOTH,PROP=.2,.9
HEADER OPERATIONS DATA
STEP   1
       CALL BUILDO(ALLBLK)
L      NPLOT
END OF DATA
```

Figure 3-14 D-Card and N-Card Operations Example

3.3.3.13 Node Identification Number Control

The thermal analysis of large vehicles frequently involves combining several TRASYS models into one. The component models will generally have been generated independently, perhaps by different contractors, and node/surface number duplication in the various surface data blocks will be common. The laborious task of renumbering nodes to eliminate duplication is alleviated considerably by use of the N-Card option. When an N-Card is encountered in the surface data, all node and surface numbers in the surface (S-Card) data following will be changed accordingly to:

SURFN = SURFN + INC

where:

INC is an integer value found on the N-Card.

This holds true until another N-Card is encountered or until the end of the surface data block. Changing SURFN for a surface means that all node numbers associated with that surface are changed, whether automatically generated or input as an integer array.

The following N-Card restriction must be observed: the variable INC may take on any positive or negative integer value such that

$1 < SURFN + INC < 99999$

is true for all values of SURFN involved.

3.3.3.13.1 N-Card Formats

The N-Card format is:

CC1	CC7
N	DV (integer value for INC)

3.3.3.13.2 N-Card Operations Example

A surface data block using N-Cards is shown in Figure 3-14 of paragraph 3.3.3.12.

3.3.3.14 Shadower-Only Surfaces

In many radiation-dominated thermal analysis problems, there are surfaces which are so remote from the region of interest that they do not actively enter into the radiation network. These surfaces, however, block incoming radiation and the view to space for the nodes of interest. The use of shadower-only surfaces permits the user to account for this blockage without increasing the complexity of his problem in the region of interest.

The following rules apply in the use of shadower-only surfaces:

a) Shadower-only surfaces provide blockage in both the FF and the DI segments. They appear nowhere in the program output of the user's problem, however.
b) Form factors from node i to shadower-only surfaces are summed and added to F_{ii} to conserve energy (infers $T_{shadowers} = T_i$).
c) Because these surfaces are not active in the problem, neither the active side nor the surface optical properties need be input.
d) Shadower-only surfaces are specified by setting the shade flag; SHADE = ONLY.

e) Shadower-only surfaces must be added after all active surfaces. It is recommended that shadower-only surfaces be input in separate BCS(s) and that these BCS(s) be added last in the BUILDC-ADD sequence. (Reference Appendix D, PP D-2, D-3).

3.3.4 BCS Data

Each block coordinate system named in the surface data block must be defined in the BCS data block. This is done according to the following formats:

```
CC1          CC7                                                    CC7
                                                                      2
  B            BCSNAM, TX, TY, TZ, ROTX, ROTY, ROTZ
```

If the rotations are not to be performed in the standard x, y, z, order, the hybrid format:

```
CC1          CC7                                                    CC7
                                                                      2
  B            BCSNAM, TX, TY, TZ, ROTZ = DV,
               ROTX = DV, ROTY, = DV
```

may be used. Rotations in this example are performed first about Z, then X, then Y.

The variable names entered in the BCS data block are defined in Table 3-IV. Note that these definitions are almost identical to the ICS and position data of the surface data block, even to variable names. This creates no ambiguity, because the position and ICS variables are used in the preprocessor only, and unlike the BCS variables, are not addressable from the processor routines.

3.3.5 Form Factor Data

The form factor data block has two basic functions. It provides an entry point for form factor data that is known to the user in advance, and it provides for restart of form factor runs that were interrupted.

From the standpoint of TRASYS operations, information in the form factor data block is used by the preprocessor to define two arrays. First it defines the node identification number array, and second the form factor request matrix is filled. If a form factor data block is not encountered in the input stream, surface data information is used to define the node ID matrix and the form factor request matrix defaults everywhere to -1.0.

The form factor request matrix is a triangular matrix of the same form as a form factor matrix. It is used for detail direction of form factor computations, and finally for storage of the form factors. This is done as follows: prior to computing each form factor, the corresponding value in the request matrix is examined; if it is zero or greater than zero, it is presumed to be a valid form factor and is left in the matrix unchanged; if less than zero, the form factor is computed and stored in place of the negative number. Thus, if the request matrix is first initialized everywhere to -1.0, then the known form factors (including zeros where the surfaces are known to be "out of sight" of each other) are placed in the matrix, form factor computations can proceed with no waste motion.

Table 3-IV BCS Data Input Detail

VARIABLE NAME	RANGE OR OPTIONS	DEFAULT VALUE	DESCRIPTION
BCSN	ANY 6 CHARACTER NAME	NONE	BLOCK COORDINATE SYSTEM NAME
TX	N/A	0.0	TRANSLATION DISTANCE FROM ORIGIN OF CCS TO ORIGIN OF BCS, MEASURED ALONG CCS X-AXIS.
TY	N/A	0.0	SAME AS TX, EXCEPT ALONG Y-AXIS
TZ	N/A	0.0	SAME AS TX, EXCEPT ALONG Z-AXIS
ROTX	$-360. < ROTX \leq 360.$	0.0	ROTATION ANGLE TO ROTATE CCS INTO BCS; ROTATES ABOUT CCS X-AXIS, Y TOWARD Z POSITIVE.
ROTY	$-360. < ROTY \leq 360.$	0.0	SAME AS ROTX, EXCEPT ROTATES ABOUT Y, Z TOWARD X IS POSITIVE.
ROTZ	$-360. < ROTZ \leq 360.$	0.0	SAME AS ROTX, EXCEPT ROTATES ABOUT Z, X TOWARD Y IS POSITIVE.

The above discussion applies to a single problem, geometry. If more than one geometry exists in a given job, multiple request matrices are involved. The form factor data block provides for definition of as many request matrices as required.

At the user's option, the form factor segment will punch the cards, in form factor data format, that will define the request matrix necessary to continue form factor calculations from any point a run might have been interrupted. It is strongly recommended that the user request this punched data when his problem will involve a considerable amount of run time. The punched output will include a node identification matrix as well as the form factor values.

3.3.5.1 Variable Definitions

Form factor data block variables are defined in Table 3-V.

Table 3-V Form Factor Variable Definition

VARIABLE NAME	RANGE OR OPTIONS	DEFAULT VALUE	DESCRIPTION
INITL	ANY FLOATING POINT NUMBER	-1.0	VALUE INITIALLY STORED IN EACH REQUEST MATRIX LOCATION
STEPN	1-99 (INTEGER)	NONE	OPERATIONS BLOCK STEP NUMBER
NODEA	1-99999 (INTEGER ARRAY)	NONE	NODE IDENTIFICATION NUMBER ARRAY

3.3.5.2 Form Factor Data Formats

1) INITL is entered according to the following format:

CC1	CC7	CC7 2
INITL	DV (floating point)	

2) STEPN is entered according to the following format:

CC1	CC7	CC7 2
STEPN	DV (integer)	

3) The NODEA array is entered according to the following format:

CC1 CC7 CC7
2

NODEA NN1, NN2, . . . NNN, END

for an N-node matrix. The node numbers must be entered in the exact order that results from the surface data block and the order of BUILDC and ADD calls in the operations data block. For this reason, the chances of a user-written node number matrix being valid for a large problem are remote. When a node array is needed, use Subroutine FFNDP (Ref P3-68).

4) Single form factors are entered according to the following format:

CC7 CC7
2

NA, NB, DV
where: NA, NB and DV correspond respectively to I, J and the FA product in the expression:
$F_{I\ J}\ A_I = F_{JI}\ A_J = DV$

5) Multiple-repeated form factors involving a single node are entered according to the following format:

CC7 CC7
2

NA, NB, NC, DV

This will result in an FA product equal to DV being entered in the row of the request matrix corresponding to node NA, for columns corresponding to nodes NB through NC, inclusive.

6) An entire row of repeated form factors may be entered according to:

CC7 CC7
2

NA, ALL, DV

This will result in an FA product equal to DV being entered in the entire row of the request matrix that corresponds to node NA.

3.3.5.3 Form Factor Data Block Example

An example of a form factor data block is presented in Figure 3-15.

FORTRAN CODING FORM

		Punching Instructions		Page of
Program		Graphic	Card Form # *	Identification
Programmer	Date	Punch		73 80

C FOR COMMENT

STATEMENT NUMBER 1 5	Cont. 6	FORTRAN STATEMENT 7 10 15 20 25 30 35 40 45 50 55 60 65 70 72
HEADER		FORM FACTOR DATA
INITL		-1., $
STEPN		10 $
NODEA		10, 17, 20, 24, 32, END $
		10, 20, 0.25 $
		10, 32, 0.25 $
		20, 17, 0.10 $
		24, 32, 17, 20 $ EQUIVALENT FORM FACTOR
		30, ALL, 0.22

Figure 3-15 Form Factor Data Block Example

3.3.5.4 Equivalent Form Factors

Many radiation enclosures involve geometry that is symmetric in some manner and may, therefore, have many form factors that are exactly equivalent to other form factors because the node pairs involved are the same size and shape and "see" each other in the same way. The analyst can identify these situations, and if he can conveniently enter this information, a considerable amount of computer time may be saved in form factor computation. This capability has been provided, and the following sections describe the required form factor data block input.

3.3.5.4.1 Equivalence Data Formats

Form factor equivalence data records appear in the form factor data block in the following format:

```
CC7

  NNI, NNJ, NNK, NNL
```

All four variables are integer node numbers. The four numbers are interpreted to mean the form factor from NNI to NNJ is equal to the form factor from NNK to NNL. In common with the other form factor data, only one record of four words or less with the comma delineators may appear on a card (except for the node array).

3.3.5.4.2 Restrictions

The order in which the node numbers appear on the cards is controlled by the order node numbers appear in the node array. Form factors are computed from each node to the remaining nodes in the order the nodes appear in the node array. When a form factor is equivalent to another, the other form factor must have been previously computed so that it may be retrieved and stored. This means that the first occurrence in the node array, of either NNK or NNL must precede the first occurrence of either NNI or NNJ.

3.3.5.4.3 Punching a Node Array--Subroutine FFNDP

Writing a node array for a large complex problem is exceedingly prone to error, yet a node array must appear in the form factor data block even if it only contains equivalence cards. Subroutine FFNDP may be used to obtain a node array, punched in form factor data format. This may be done in a preliminary run, (when node plots are generated for instance) and the user will then have an error-free node array to use with his form factor equivalence data. The calling sequence is:

```
CC7

  CALL FFNDP
```

3.3.6 Shadow Factor Data

3.3.6.1 Basic Concepts

Shadow factors are fractional numbers that describe the amount of shadowing (blockage) encountered by collimated energy incident on a nodal surface. A shadow factor of one indicates no blockage, zero indicates 100 percent blockage. Blockage results from other parts of the spacecraft or from the surface itself, if nonplanar.

Shadow data consists of tables of shadow factors, one table per node. These are 171-point bivariate tables. When the direction to an energy source is specified, using clock and cone angles, the clock and cone angles are used as arguments in a double-linear interpolation that returns a shadow factor to be used in computing direct irradiation according to:

$$DI_{shadowed} = SF * DI_{nonshadowed}$$

The 171 points result from all combinations of 19 clock angles and nine cone angles, spaced as described in Appendix C.

The shadow factor data functions to provide a punched card entry point for shadow factor data that is known in advance, for restart of interrupted shadow factor generation runs, and to direct the updating of an existing shadow factor (SHADI) tape. The use of the shadow factor data block is illustrated by the following examples:

1) No shadow factor data block - if no shadow factor data block is encountered, it is assumed that no shadow data is known in advance and no shadow factor read (SHADI) tape is mounted. All shadow factor tables are computed and written on the SHADO (shadow factor output) tape as directed by operations block data. Figure 3-16a shows the pertinent options data block and operations data block input for this example.
2) Shadow factor data block input, SHADI tape to be updated - in this case, the shadow factor data block must contain at least one model name. This enables the program to find the proper set of tables on the SHADI tape. Shadow factor tables for selected nodes are updated either by direct replacement through TABLE cards, or by recomputation through RECOMP cards. In this case, the node number (NODEA) array is obtained from the SHADI tape and is not entered in the shadow factor data block. Figure 3-16b shows the pertinent options data block, shadow data block, and operations data block input for this example.
3) Restart of a run generating a SHADO tape - in this case, a partially complete SHADI tape is mounted. The information on this tape is passed directly to the SHADO tape and the remainder is computed. Figure 3-16c shows the pertinent options data block, shadow data block, and operations data block input for this example.

The following is some general information on the subject of shadow factor operation.

1) The SHADI and SHADO tapes have the same format (see Appendix C). SHADI is a read tape. SHADO is a write tape.
2) The model name on the SHADI tape is for reference in reading SHADI data and is not automatically passed over to the SHADO tape. The model name on the SHADO tape is defined only through an SFDATA call prior to SFCAL execution.
3) An SFCAL call is the only means of writing a SHADO tape. Thus, if an update operation consists only of replacing some tables on a SHADI tape with some data input on cards, an SFCAL call must still be made in the operations data block.
4) A shadow factor data block is mandatory whenever a SHADI tape is to be read. Thus, this block is required for restart of shadow factor generation, shadow factor table modification, and for using the shadow tape in computing fluxes. Except for table modification, the block will consist only of a header card and a card defining the model name.

```
HEADER OPTIONS DATA
       SHADO - TXXXX  $  TAPE NUMBER REMINDER
                    (OTHER DATA BLOCKS)
HEADER OPERATIONS DATA
STEP  1
      CALL BUILDC(ALLBLK)
      CALL SFDATA (0,4HMOD2)
L     SFCAL
END OF DATA
```

(a) Shadow Factor Operations, No -SHADI- Tape

```
HEADER OPTIONS DATA
       SHADI - TXXXX
       SHADO - TXXXX
               (OTHER DATA BLOCKS)
HEADER SHADOW DATA
MODEL    MOD3, SAVE  $  SAVE OPTION SAVES MOD3 FILE FILE ON -SHADO- TAPE
C          TOGETHER WITH MOD4
NODEA    101,102,103,104,105,110,120,130,140
RECOMP   104
RECOMP   130
TABLE    101,C2,REPEAT,1,.12,.5,.5,.4,.4,.3
             C9.REPEAT,.5,19
HEADER OPERATIONS DATA
STEP   1
       CALL BUILDC(ALLBLK)
       CALL SFDATA(4HMOD3,4HMOD4)
L      SFCAL
END OF DATA
```

(b) Shadow Factor Operations, Updating -SHADI- Tape

Figure 3-16. Shadow Factor Operations Detail

```
HEADER OPTIONS DATA
         SHADI - TXXXX
         SHADO - TXXXX
                  (OTHER DATA BLOCKS)
HEADER SHADOW DATA
MODEL MOD1
HEADER OPERATIONS DATA
STEP  1
      CALL BUILDC(ALLBLK)
      CALL SEDATA(4HMOD1,0)  $ -SHADO- TAPE FILE NAME DEFAULTS TO MOD1
L     SFCAL
END OF DATA
```

(c) Restart of -SHADO- Tape Generation Run

```
HEADER OPTIONS DATA
      SHADI - TXXXX
                  (OTHER DATA BLOCKS)
HEADER SHADOW DATA
MODEL  MOD12
HEADER OPERATIONS DATA
STEP  1
      CALL BUILDC(ALLBLK)
      CALL ADD(CMOD)
      CALL ORBIT2(3HMOD,0,,0.,0.,0.,0.,0.,0.)
      CALL DTDT2S(0,0.,120.,0.,180.,0.,120.*6080.,3HPUN)
C     COMPUTE DIRECT FLUXES WITH ANALYTICAL SHADOW COMPUTATIONS   JTATIONS
L     DICAL
                (ADDITIONAL ORBIT POINTS)
STEP  20
      CALL BUILDC(DMOD)
      CALL DIDT2S(0,10.,150.,120.,20.,1.2,75.*6080.,3HPUN)        JN)
      CALL SEDATA(5HMOD12,0)
L     SFCAL
C     COMPUTE DIRECT FLUXES USING SHADI TAPE
L     DICAL
              (ADDITIONAL ORBIT POINTS)
END OF DATA
```

(d) Direct Irradiation Calculations with Shadow Tapes

Figure 3-16. (concl)

3.3.6.2 Variable Definitions

Variable Name	Description	Default Value
MODEL[1]	Configuration name on SHADI tape Hollerith, up to 6 characters.	None
NODEA	Node number array	None

Notes:

1) This name defines the configuration name to look for on input tape SHADI. If not defined, it is assumed that no SHADI tape is present.

3.3.6.3 Shadow Data Formats

MODEL and NODEA are entered according to the following formats:

CC1	CC7
MODEL	CONF1 $ (any Hollerith name up to 6 characters)
NODEA	DV1, DV2, DV3, --- DVN $ (integer node numbers)

Instructions to recompute shadow factor tables for a specified list of nodes are entered as follows:

CC1	CC7
RECOMP	DV1 $ (integer node numbers)
RECOMP	DV2
.	.
.	.
.	.

Note that this input only applies when a SHADI tape is mounted.

A complete shadow factor table for one node is entered according to:

CC1	CC7
TABLE	NN, C1, DV1, DV2, --- DV19
	C2, DV1, DV2, --- DV19
	" " " "
	" " " "
	" " " "
	" " " "
	C9, DV1, DV2, --- DV19

The mnemonics C1 through C9 refer to the 9 cone angles in a shadow factor table. The 19 data values following are for the 19 clock angles in a shadow factor table. If a cone number mnemonic is omitted, the 19 data values associated with it default to zero (100 percent shadowed). If less than 19 data values are entered following a CX mnemonics, the data values encountered are stored consecutively beginning with Clock 1. For example, if

CX, DV1, DV2, DV3 --- DVN (N less than 19)

is encountered, the shadow factors for cone angle X, clock angles 1 through N will be DV1 through DVN. The shadow factors for clock angles N + 1 through 19 will default to zero.

Repeated data values may be entered using the repeat option for array data. For example, the card:

CC7

C6, REPEAT, 0.5, 12, REPEAT, 0., 7

will enter shadow factors of 0.5 for clock angles 1 through 12 and 0. for clock angles 13 through 19 in the cone angle 6 array.

The user is referred to the description of the shadow factor tape format (Appendix C) for an explanation of the way the clock and the cone angles relate to the energy source/vehicle orientations used in shadow factor generation.

3.3.6.4 Shadow Factor Operations Examples

Examples of shadow factor operations are shown in Figure 3-16.

3.3.7 Flux Data

3.3.7.1 Basic Concepts

The flux data block has two functions, it provides a punched card entry point for direct irradiation fluxes that are known to the user in advance, and it provides for restart of direct irradiation runs that are interrupted.

From the standpoint of TRASYS operations, information in the flux data block is used to define the flux data request matrix and a node identification array. If a flux data block is not encountered in the input stream, surface data information is used to define the node identification matrix and the flux request matrix is set everywhere to -1.0.

The flux request matrix is a 3 X NN array (where NN = number of nodes) that is used to direct flux calculations. In the direct irradiation processor segment, the three request matrix elements associated with each node are examined prior to any calculations. Any element less than zero results in a flux calculation. Zero elements and elements greater than zero are presumed to be valid flux values and are passed directly to output data storage. The three elements associated with each node correspond to solar, albedo and planetary **infrared** radiation, and are input in that order (Remember: Solar, Albedo, Planetary - SAP). The variables INITL, NODEA, and STEPN are used in the flux data block to perform the same functions as they do in the form factor data block (see Section 3.3.5.2).

At the user's option, the direct irradiation segment will punch BCD cards in flux data block format as computations proceed. Thus, in case of interruption, the run can be continued by entering the cards in the flux data block. It is strongly recommended that the user always use the punch option when generating direct fluxes for large problems. The punched output includes a node identification array as well as the flux values.

3.3.7.2 Variable Definitions

The flux data block contains three variables with names in common with the form factor data block. These variables are defined in Table 3-V.

3.3.7.3 Flux Data Formats

1) INITL, STEPN, and NODEA are entered using the same formats as the corresponding variables in the form factor data block (see Section 3.3.5.3).
2) Flux values may be entered in either of two formats. The quadruplet format is as follows:

```
CC7                                                    CC7
                                                         2
  NODID, DV1, DV2, DV3
  101, 232., 114., 99.*.317 $ example
```

Where:

NODID = node identification number
DV1 = incident solar flux
DV2 = incident albedo flux
DV3 = incident planetary infra-red flux

Restrictions:

One quadruplet only per card.
All four data values are required. No default logic applies.
Flux values must be in TRASYS standard units (Btu/hr-ft^2).

The single value format is as follows:

CC7 CC7
2

NODE = DV1, SUN = DV2, ALB = DV3
PLAN = DV4
or
DV1, S = DV2, A = DV3, P = DV4

Where:

DV1 = node identification number (integer)

DV2 = incident solar flux

DV3 = incident albedo flux

DV4 = planetary infrared flux

Restrictions:

All data values encountered between NODE values pertain to the preceding node number. Undefined data values default to INITL.

Standard punched card output from the direct flux calculation segment is in the single-value format.

3.3.7.4 Flux Data Block Example

An example of a flux data block is shown in Figure 3-17.

FORTRAN CODING FORM

Program	Punching Instructions: Graphic / Punch	Card Form #	Page of
Programmer / Date			Identification 73–80

```
HEADER FLUX DATA
INITL  -1. $
STEPN  10 $
NODEA  10, 17, 20, 24, 32, END $
       10, 2.979*144., 0., 0. $  SUN ORIENTED SURFACE
       17, 0., 25.74 , 14.8  $
       NODE = 20, SUN = 0., ALB = 25.74, PLAN = 14.8
```

Figure 3-17 Flux Data Block Example

3.3.8 Correspondence Data

3.3.8.1 Basic Concepts

The correspondence data block performs the function of providing the user with an input point for the node numbering data necessary to make his thermal radiation model correspond on a one-to-one basis with his thermal analyzer RC (Resistance Capacitance) model.

Using the information entered in this block, the radiation interchange and absorbed heat processor segments perform the necessary calculations to provide for energy conservation when the user desires to combine one or more TRASYS node surfaces into a single RC model node.

3.3.8.2 Variable Definitions

Correspondence data block variables are defined in Table 3-VI.

Table 3-VI Correspondence Data Variable Definition

VARIABLE NAME	RANGE OR OPTIONS	DEFAULT VALUE	DESCRIPTION
STEPN	1-99 (INTEGER)	NONE	OPERATIONS BLOCK STEP NUMBER
KOMB	N/A	NONE	HOLLERITH CONTROL FLAG FOR NODE COMBINE OPERATIONS

3.3.8.3 Correspondence Data Formats

1) STEPN is entered according to the following format:

CC1	CC7	CC7 2
STEPN	DV (integer)	

2) The control field flag, KOMB, is entered as follows:

CC1	CC7	CC7 2
KOMB		

3) Node correspondence data is entered according to the following format:

CC7

NODID = DV1, DV2, - - - - DVN

NODID is an RC model node number (one to five digits, integer) and DV1 through DVN are the TRASYS node numbers that will be combined into node NODID.

3.3.8.4 Correspondence Data Block Structure

The correspondence data found between a card defining a step number and the next step number definition card can be thought of as a correspondence data sub-block. One of these sub-blocks is required for each unique geometry of the user's problem, assuming node combine operations are required for each geometry. For example, if operations block steps 1 through 5 involve the same geometry, and steps 6 through 10 another, two correspondence data sub-blocks are required, one under STEPN = 1 and another under STEPN = 6. The correspondence data operations are performed in the order the data is encountered. Because of this, do not utilize a RC node number that is identical to a TRASYS node number that appears later in the correspondence data.

It should further be noted that correspondence data only applies to operations block steps involving RKCAL (radiation interchange) and QOCAL (absorbed heat output) program segment execution.

3.3.8.5 Correspondence Data Block Example

An example of a correspondence data block is shown in Figure 3-18.

FORTRAN CODING FORM

Program		Punching Instructions		Page of
Program		Graphic	Card Form #	Identification (73–80)
Programmer	Date	Punch		

C FOR COMMENT

STATEMENT NUMBER (1–5) | Cont. (6) | FORTRAN STATEMENT (7–72)

```
HEADER R CORRESPONDENCE DATA BLOCK
STEPN    10 $
KOMB
         50 = 10, 17, 32 = 40, 41, 42

         6 = 61, 62
```

Figure 3-18 Correspondence Data Block Example

3.3.9 Operations Data Block

3.3.9.1 Basic Concepts

The operations block can be thought of as a digital computer program coded in a somewhat modified FORTRAN language. The most powerful statements in the block are calls to processor library subroutines followed by "link" calls to primary processor program segments. Interspersed with these statements might be FORTRAN statements used to redefine any of the program variables in the reserve name list or control constant list, calls to user-supplied routines in the subroutines block, and any branching statements required for direction of problem solution logic. The operations data block is converted by the preprocessor to subroutine ODPROG. This routine serves as the driver for processor execution. In general, the conversion is a one-to-one passover of FORTRAN statements. The segment execution calls (L cards) however, result in the operating system dependent language necessary to define an overlay execution.

An operations block for a problem involving one or more geometry changes or more than one point in orbit consists of a series of modified FORTRAN programs known as steps. These steps are numbered, for reference in later operations, and to allow data from the form factor, flux, shadow, and correspondence data blocks to be properly transmitted to the processor. The steps are executed in the order encountered, regardless of the step number called out.

Some operations' block logic restrictions must be observed: each step must be serially executable, that is no branching from one step to another is allowed. Further, any DO loop containing a program segment execution (L card) in its field will not execute properly because the indices are lost when the segment is overlaid and removed from core. This, however, does not prevent coding the equivalent of a DO loop using idices located in common through being entered in the quantities data block. The user should keep in mind, too, that multiple executions of any program segments other than NPLOT, OPLOT, PLOT or SFGEN within a step will make later data retrieval impossible for that step, because provision for storage of only one set of each type of data is provided per step (Reference Figure 3-19). If the user is satisfied with the printed or punched output obtained during a particular segment's execution, this is no restriction. Statement numbers from 1 to 9999 may be used, and each statement number must be unique in the operations block. All program control constants and variables in common at execution of the operations block (subroutine ODPROG) may be found in Appendix A. This list is automatically extended to contain any constants and arrays entered in the quantities and array data block.

3.3.9.2 ORBGEN Option

Writing an operations data block for the calculation of direct irradiation and absorbed heats for an extensive series of points in orbit can be a tedious, repetitive job. To alleviate this, the ORBGEN option is available. When an ORBGEN card is encountered in the operations data block, a package of preprocessor routines use the data on the card to generate the operations data code necessary to compute direct irradiation, absorbed heats, and print a set of heat/rate vs time tables in a standard SINDA input format. These tables may be

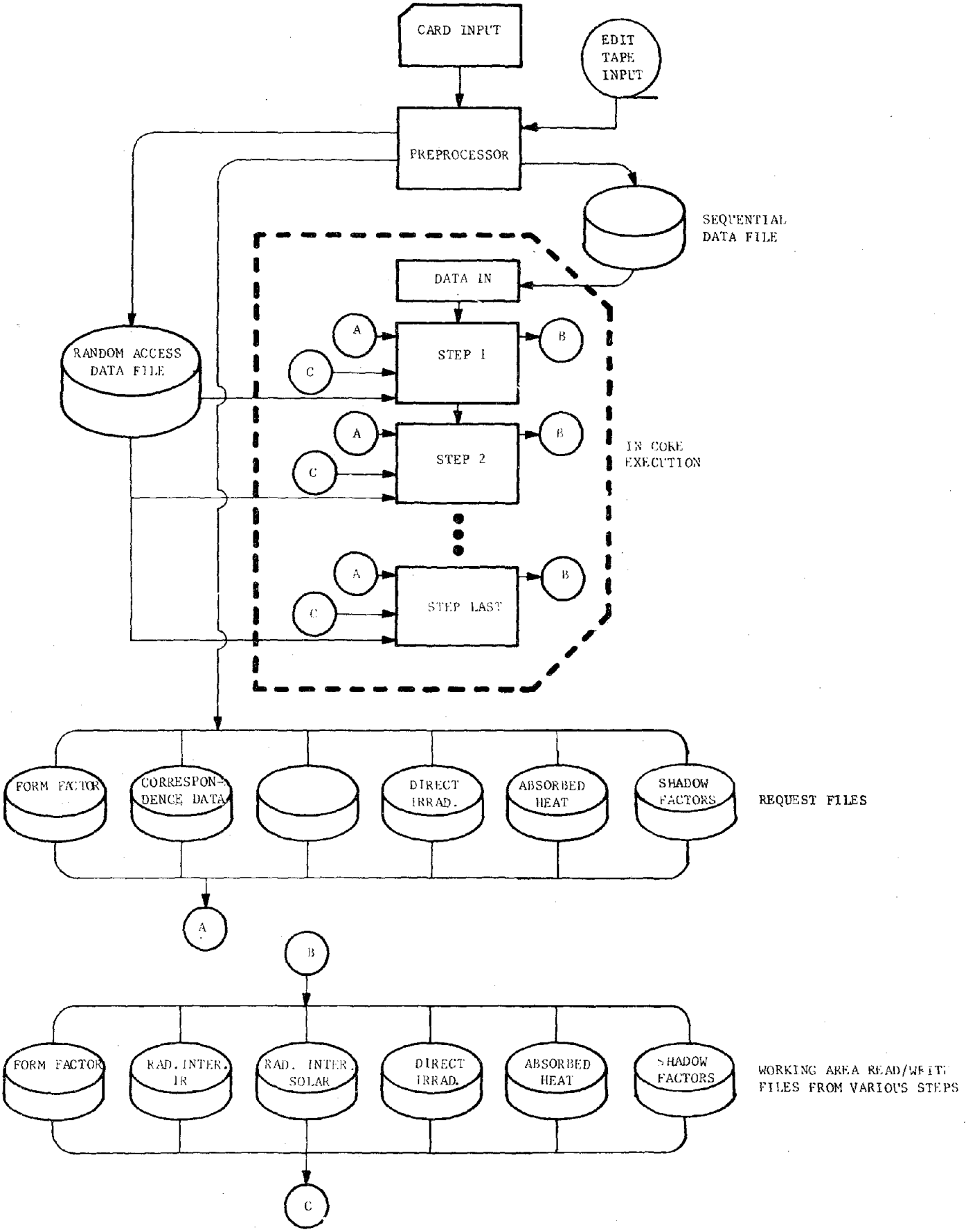


Figure 3-13 Program Data Storage Scheme

thought of as a default output that prevents loss of computed data. All flux and absorbed heat data computed using the generated code is stored in the usual manner and may be retrieved, manipulated, and output in any way the user desires.

ORBGEN cards are defined as follows:

Format

CC1	CC7
ORBGEN	TYPE, TRUANI, TRUANF, NPT, IAI, IAS, ICOR

Definitions

TYPE is a Hollerith variable defining the spacecraft orientation reference and pertinent orbit characteristics. Options are:

INER: Spacecraft is in planetary orbit, inertial (sun or star) oriented.

PLAN: Spacecraft is planet oriented.

CIRP: Spacecraft is planet oriented, in a circular orbit.

NOPL: Spacecraft is in a heliocentric orbit (no planet).

TRUANI is the true anomaly* at the first point in orbit $0 \leq$ TRUANI < 360.

TRUANF is the true anomaly at the final point in orbit. If TRUANF = TRUANI + 360, data for a complete orbit will be generated.

NPT is the number of equal true anomaly increments between the points for which fluxes and direct irradiation will be computed. If the planet shadow is not encountered by the orbit, NPT + 1 points will be computed. If the planet shadow is encountered, NPT + 5 points will be computed, thus describing the flux discontinuities at the planet shadow points.

IAI is the step number where an infrared grey-body factor matrix is in storage This step must precede or be the same step in which the ORBGEN card is entered.

IAS is the same as IAI, for a solar grey-body matrix.

ICOR is the step number reference for the pertinent correspondence table (entered as STEPN in the correspondence data block).

Restrictions

1) Prior to entering an ORBGEN card, the orbit must be defined through a call to ORBIT1 or ORBIT2.
2) Orientation must be defined through a call to ORIENT.
3) Spin must be defined, if applicable, through subroutine SPIN. If spin is not zero, INER and CIRP are not allowable for TYPE.
4) Problem geometry must be defined prior to any ORBGEN card.

---

*Reference Figure 4-7

5) If IAI and IAS are input as zero, the operations data generated is for fluxes only. No AQCAL calls will result, and no grey-body matrices are required.

6) Punch/tape flags and accuracy parameters must be defined through subroutine DIDT1 or DIDT1S prior to an ORBGEN card.

7) Because the user does not know the step numbers generated by the ORBGEN card, any calls to QODATA or PLDATA immediately following the ORBGEN card must use the ALL option.

3.3.9.3 Operations Block Formats

1) Step number cards are punched according to the following format:

CC1	CC7	CC7 2
STEP	DV (integer)	

Where DV is the integer step number, with the allowable range 1 to 9999.

2) Subroutine calls are made in the classic FORTRAN format, with the word CALL beginning in CC7. Calling sequences for each user-accessible processor routine can be found in Appendix D.

3) Computation segment (link) calls are made using the following format:

CC1	CC7	CC7 2
L	SEGNAM	

Where SEGNAM is the name of one of the program segments contained in the processor library. Appendix E describes the processor segments and their functions. Currently allowable options for SEGNAM are: NPLOT, OPLOT, SFCAL, FFCAL, DICAL, GBCAL, RKCAL, AQCAL, QOCAL, and PLOT.

3.3.9.4 Operations Block Examples

Operations block structure and function is illustrated by the listings of sample operations blocks in Figure 3-20.

Sample 1 of Figure 3-20 is a single step operations block that generates three node plots of a single geometry. Sample 2 is a two-step operations block that generates form factor matrices for two geometric configurations. Sample 3 is a 17-step operations block that generates direct irradiation data at 16 points in orbit, including the planet shadow in/out points. A geometry change at the shadow in/out points is involved. Sample 4 is a listing of the operations data block code generated by an ORBGEN card.

```
C       SAMPLE 1 -- OPERATIONS BLOCK FOR PLOT OPERATIONS


HEADER OPERATIONS DATA
STEP  1
C       BUILD GEOMETRY
      CALL BUILDC(BNAME1)
      CALL ADD(BNAME2)
      CALL ADD(BNAME3)
C       INITIALIZE FOR PLOT 1
      CALL NDATA(1,3HGEN,0,0,A20,1,3,2,0.,0.,37.)
C      INITIALIZE FOR PLOTS 2 AND 3
      CALL NDATAS(2,3H3-D,0)
      CALL NDATAS(3,1HX,0)
C       MAKE PLOTS 1,2 AND 3
L     NPLOT



C       SAMPLE 2 -- OPERATIONS BLOCK FOR FORM FACTOR OPERATIONS


HEADER OPERATIONS DATA
STEP  1
      CALL BUILDC(BNAME1)
      CALL ADD(BNAME2)
      CALL ADD(BNAME3)
C       SET FF CALCULATION PARAMETERS (PRINTS FF S, DOES NOT PUNCH)
      CALL FFDATA(0.,.2,0,0.,1.E-3,5HPRINT,0)
C       COMPUTE FORM FACTORS
L     FFCAL
STEP  2
C     MOVE BNAME2 SURFACES
      CALL CHGBLK(BNAME2,0.,0.,25.,1,2,3,0.,45.,0.)
      CALL BUILDC(BNAME1)
      CALL ADD(BNAME2)
C       CALCULATE FORM FACTORS WITH SAME PARAMETERS AS STEP 1
L     FFCAL
END OF DATA
```

Figure 3-20 Sample Operations Data Blocks

SAMPLE 3 --- OPERATIONS BLOCK FOR TWO GEOMETRY ABSORBED HEAT PROBLEM

```
HEADER OPERATIONS DATA
STEP  1
      CALL BUILDC(BNAME1)
      CALL ADD(BNAME2)
      CALL ADD(BNAME3)
C       SET FF CALCULATION PARAMETERS, PUNCH FFS
      CALL FFDATA(0,.2,0,0,1.E-3,0,3HPUN)
L     FFCAL
C       CALCULATE  GREY BODY FACTORS
      CALL GBDATA(0,4HBOTH)
L     GBCAL
C       SET RADK CALCULATION PARAMETERS, PUNCH RADKS
      CALL RKDATA(0,0,0,1000.5HSPACE,999,0,0,0)
C       COMPUTE RADKS
L     RKCAL
C       DEFINE ORBIT AND LOCATE SUN
      CALL ORBIT2(3HEAR,0.,90.,0,0,0,120.*6080.,0.)
C       ORIENT VEHICLE (CCS Z-AXIS TOWARD SUN)
      CALL ORIENT(3HSUN,1,2,3,0.,90.,0.)
C       SET DI COMPUTATION DATA
      CALL DIDT1S(0.,0,0.,3HPUN)
C       COMPUTE DIRECT IRRADIATION (DICOMP PARAMETERS DEFAULT TO COMPUTE ALL)
L     DICAL
C       SET ABSORBED HEAT CALCULATION PARAMETERS
      CALL AQDATA(1,1,0,0,0)
L     AQCAL
STEP  2
C        UPDATE TRUE ANOMALY, SET UP TO COMPUTE PLANET AND ALBEDO FLUXES
      TRUEAN       = 30.
      CALL DICOMP(1,0,0)
L     DICAL
L     AQCAL
STEP  3
      TRUEAN     =60.
      CALL DICOMP(1,0,0)
L     DICAL
L     AQCAL
STEP  4
      TRUEAN      =90.
      CALL DICOMP(1,0,0)
L     DICAL
L     AQCAL
STEP  5
C       SKIP OVER PLANET SHADOW
      TRUEAN       =270.
      CALL DICOMP(1,0,0)
L     DICAL
L     AQCAL
STEP  6
      TRUEAN        =300.
      CALL DICOMP(1,0,0)
L     DICAL
L     AQCAL
```


Figure 3-20 (cont'd)

```
STEP  8
C       STUFF TRUEAN = 0. VALUES (DUPLICATE POINT)
      CALL STFAQ(360.,0,1)
STEP  9
C       COMPUTE DATA AT SHADOW ENTRY POINT (DAYSIDE GEOMETRY)
      TRUEAN       =SHADIN - .1
      CALL DICOMP(1,0,0)
L     DICAL
L     AQCAL
STEP  10
C       COMPUTE DATA AT SHADOW OUT POINT (DAYSIDE GEOMETRY)
      TRUEAN       =SHAOUT + .1
      CALL DICOMP(1,0,0)
L     DICAL
L     AQCAL
C       PUNCH AQ,AQAVG VS. TIME TABLES - DAYSIDE
      CALL QODATA(3HALL,0,2HNO,3HPUN,0,0,0,4HBOTH,0)
L     QOCAL
STEP  11
C       BUILD DARKSIDE CONFIGURATION
      CALL BUILDC(BNAME1)
      CALL ADD(BNAME2)
      CALL ADD(BNAME4)
C       CALCULATE FFS (FFDATA PARAMETERS SET IN STEP 1)
L     FFCAL
C       CALCULATE GREY BODY MATRICES
      CALL GBDATA(0,4HBOTH)
L     GBCAL
C       SET RADK CALCULATION PARAMETERS, COMPUTE RADKS
      CALL RKDATA(0,0,0,1000,5HSPACE,999,0,0,0)
L     RKCAL
C       REORIENT TO PLANET
      CALL ORIENT(4HPLAN,1,2,3,0.,90.,0.)
      TRUEAN       =120.
L     DICAL
C       SET ABSORBED HEAT PARAMETERS, COMPUTE ABSORBED HEATS
      CALL AQDATA(11,11,0,0,0)
L     AQCAL
STEP  12
C       UPDATE TRUE ANOMALY, STUFF HEAT DATA FROM STEP 11 BECAUSE ORBIT
C       IS CIRCULAR, PLANET-ORIENTED
      CALL STFAQ(150.,0,11)
STEP  13
      CALL STFAQ(180.,0,11)
STEP  14
      CALL STFAQ(210.,0,11)
STEP  15
      CALL STFAQ(240.,0,11)
STEP  16
C       STUFF DATA FOR SHADOW ENTRY POINT (DARKSIDE CONFIGURATION)
      CALL STFAQ(SHADIN + .1,0,11)
STEP  17
C       STUFF DATA FOR SHADOW OUT POINT (DARKSIDE CONFIGURATION)
      CALL STFAQ(SHAOUT - .1,0,11)
C       PUNCH AQ,AQAVG VS TIME TABLES - DARKSIDE
      CALL QODATA(ISARY,0,2HNO,3HPUN,0,0,0,4HBOTH,0)
L     QOCAL
END OF DATA
```

Figure 3-20. (cont)

```
   SAMPLE 4 --- ORBIT GENERATION FROM AN ORBGEN CARD

  ORBIT GENERATION CARD FOLLOWS
ORBGEN   CIRC, 0.0, 360.0, 4, 1, 1, 2

C*********************  ORBIT GENERATION STARTS HERE  **********************
STEP  10000                                                                 *
       TRUEAN        =       0.                                             *
       TRUANF        =     360.000                                          *
       TRUANI        =       0.                                             *
       IAI           =       1                                              *
       IAS           =       1                                              *
       PLTYPE        = 6HPLSAVE                                             *
       CALL DICOMP(0.0.0)                                                   *
L       DICAL                                                               *
       NSREF         = 10000                                                *
       PLTYPE        = 6HPLREAD                                             *
       CALL AQDATA(IAI.IAS.0.0.0)                                           *
L       AQCAL                                                               *
STEP  10001                                                                 *
       CALL STEAQ (TRUANF.0.0.10000)                                        *
STEP  10002                                                                 *
       TRUEAN        =      90.000                                          *
       CALL DICOMP(0.0.10000)                                               *
L       DICAL                                                               *
       CALL AQDATA(IAI.IAS.0.0.0)                                           *
L       AQCAL                                                               *
STEP  10003                                                                 *
       TRUEAN        =     180.000                                          *
       CALL DICOMP(0.0.10000)                                               *
L       DICAL                                                               *
       CALL AQDATA(IAI.IAS.0.0.0)                                           *
L       AQCAL                                                               *
STEP  10004                                                                 *
       TRUEAN        =     270.000                                          *
       CALL DICOMP(0.0.10000)                                               *
L       DICAL                                                               *
       CALL AQDATA(IAI.IAS.0.0.0)                                           *
L       AQCAL                                                               *
STEP  10005                                                                 *
       IF(SHADIN.LT.0.)                  GO TO 90400                       *
       TRUEAN        = SHADIN-0.1                                           *
       IF(TRUEAN.LT.TRUANI.OR.                                              *
     1    TRUEAN.GT.TRUANF)              GO TO  90000
       CALL DICOMP(0.4HZERO.10000)                                          *
I       DICAL                                                               *
       CALL AQDATA(IAI.IAS.0.0.0)                                           *
L       AQCAL                                                               *
90000  CONTINUE                                                             *
STEP  10006                                                                 *
       TRUEAN        = SHADIN+0.1                                           *
       IF(TRUEAN.LT.TRUANI.OR.                                              *
     1    TRUEAN.GT.TRUANF)              GO TO  90100
       CALL DICOMP(0.0.10000)                                               *
L       DICAL                                                               *
       CALL AQDATA(IAI.IAS.0.0.0)                                           *
L       AQCAL                                                               *
90100  CONTINUE                                                             *
```

Figure 3-20. (cont)

```
STEP  10007                                                              *
       TRUEAN       = SHAOUT+0.1                                         *
       IF(TRUEAN.LT.TRUANI.OR.                                           *
     1    TRUEAN.GT.TRUANF)              GO TO  90200
       CALL DICOMP(0,4HZERO,10000)                                       *
L       DTCAL                                                            *
       CALL AODATA(IAI,IAS,0,0,0)                                        *
L       AOCAL                                                            *
90200  CONTINUE                                                          *
STEP  10008                                                              *
       TRUEAN       = SHAOUT-0.1                                         *
       IF(TRUEAN.LT.TRUANI.OR.                                           *
     1    TRUEAN.GT.TRUANF)              GO TO  90300
       CALL DICOMP(0,0,10000)                                            *
L       DTCAL                                                            *
       CALL AODATA(IAI,IAS,0,0,0)                                        *
L       AOCAL                                                            *
90300  CONTINUE                                                          *
90400  CONTINUE                                                          *
       CALL OODATA(3HALL,0,0,0,0,0,0,0,     2)                          *
L      OOCAL                                                             *
C                                                                        *
C**********************  ORBIT GENERATION ENDS HERE  **********************
```

Figure 3-20 (concl)

3.3.10 Subroutine Data Block

3.2.10.1 Basic Concepts

The subroutine data block is a collection of FORTRAN language subroutines supplied by the user in order to extend or modify TRASYS capabilities for the problem at hand. These subroutines may be either user-addressable (from the operations block) or program-addressable, from the various computation segments.

Unless the user is creating what amounts to a major rewrite of a computation segment, the program subroutines in his subroutine block will bear the same name as processor library subroutines. The effect of this name duplication is that the user-supplied routine in the subroutine data block is compiled in lieu of the processor library subroutine prior to execution. Removal of such a routine reactivates the like-named library routine.

Two deviations from FORTRAN language are defined for the subroutine data block. L-cards are used to identify subroutines with particular processor segments, and COMMON cards are used to automatically supply program common to the subroutines.

3.3.10.2 Subroutine Block Formats

Subroutine data block format is illustrated in Figure 3-21. The segment names on the L-cards are strictly order-dependent. The L-cards with their associated subroutines need not all be present, but they must be encountered in the order shown. Subroutines in the leading sub-block, with no L-card, are addressable only from the operations data block.

COMMON cards are optional. When used, they serve to insert all labeled and blank common lists, associated with the segment named on the preceding L-card, into the subroutine. Appendix A-3 defines the variable names in common for subroutine ODPROG and the various segments.

Note that COMMON cards preclude beginning a subroutine block comment card with the word COMMON.

3.3.11 End of Data Card

The input deck must conclude with an END OF DATA card punched according to the format:

```
CC1          CC7

  END OF DATA
```

FORTRAN CODING FORM

	Punching Instructions		Page of
F		Card Form # *	Identification 73 80
Programmer / Date	Punch		

C FOR COMMENT

STATEMENT NUMBER 1 5	Cont 6	FORTRAN STATEMENT 7 10 15 20 25 30 35 40 45 50 55 60 65 70 72
HEADER		SUBROUTINE DATA
R		VALUE
		SUBROUTINE VALUE(X,Y,Z)
		Z =SQRT(X**2+Y**2)
		RETURN
		END
L		FFCAL
R		FFROW
		SUBROUTINE FFROW
COMMON		
		CALL FFRSUM
		IF(SUM(IN).GT.2.) GO TO 100
		WRITE(NFF)NODE(IN),(FFVALS(I),I=IN,NNOD),(FFVALI(I),I=IN,NNOD)
		RETURN
100		WRITE(NOUT,1)IN
1		FORMAT(24H FORM FACTOR SUM FOR ROW,I6,34H IS GREATER THAN 2., ABO
	1	RTING RUN)
		CALL ABT
		END
L		DICAL
R		DIPREP

Figure 2-91 Subroutine Data Block Example

FORTRAN CODING FORM

Program		Punching Instructions		Page of
Programmer	Date	Graphic / Punch	Card Form # *	Identification 73–80

C FOR COMMENT

STATEMENT NUMBER (1–5) | Cont. (6) | FORTRAN STATEMENT (7–72)

```
      DO 100 I=31,NNOD
      QDP(I)    =0.
      QDR(I)    =0.
100   CONTINUE
      RETURN
      END
R     DIPRES
      SUBROUTINE DIPRES
COMMON
      DO 100 I=31,NNOD
      QDS(I)    =0.
100   CONTINUE
      RETURN
      END
L     GBCAL
R     NAME1
      (SUBROUTINE NAME1)
L     RKCAL
R     NAME2
      (SUBROUTINE NAME2)
L     AQCAL
```

Figure 3-21 (cont)

FORTRAN CODING FORM

	Punching Instructions		Page of
Program	Graphic	Card Form # *	Identification
Programmer / Date	Punch		73 80

C FOR COMMENT

STATEMENT NUMBER 1 5	Cont 6	FORTRAN STATEMENT 7 10 15 20 25 30 35 40 45 50 55 60 65 70 72
		R NAME3
		(SUBROUTINE NAME3)
		L QOCAL
		R NAME4
		(SUBROUTINE NAME4)

Figure 3-21 (concl)

4. USER CALLED ROUTINES

4.1 Basic Concepts

User called routines may be defined as those subroutines and computation segments callable from the operations block. Unless one or more subroutines of this type are entered in the user's subroutine data block, the user callable subroutines contained in the processor library comprise the entire list of user callable subroutines. Segments cannot be entered in the input stream.

4.2 Processor Library

The user callable processor library routines are listed below. In general, they are grouped according to their association with each of the processor computation segments. Page references for the subroutine descriptions in Appendix D are included.

4.2.1 Library Listing of Subroutines

	Name	Page	Name	Page
General Subroutines	BUILDC	D-2	ADD	D-3
	CHGBLK	D-4	FFNDP	D-35
	NDUPCK	D-37	TAPELS	D-36
Plot Package Subroutines	NDATA	D-5	ODATA	D-6
	NDATAS	D-5	ODATAS	D-6
	PLDATA	D-21		
Form Factor Subroutines	FFDATA	D-8	ADSURF	D-34
Direct Irradiation Subroutines	ORBIT1	D-9	ORBIT2	D-10
	DIDT1	D-11	DIDT1S	D-11
	DIDT2	D-12	DIDT2S	D-12
	SPIN	D-13	ORIENT	D-14
	DICOMP	D-23	DITTP	D-24
	DITTPS	D-24		

	Name	Page	Name	Page
Radiation Interchange Subroutines	GBDATA	D-15	RKDATA	D-16
	GBAPRX	D-31	RCDATA	D-32
Absorbed Heat Subroutines	AQDATA	D-17	STFAQ	D-18
Absorbed Heat Output Subroutine	QODATA	D-19		
Shadow Factor Subroutine	SFDATA	D-20		
Data Modification Subroutines	MODAR	D-26	MODPR	D-27
	MODTR	D-28	MODPRS	D-29
	MODSHD	D-30		

4.2.2 Library Listing of Processor Segments

Plot Package Segments	NPLOT	E-2	OPLOT	E-2
	PLOT	E-2		
Form Factor Segment	FFCAL	E-3		
Direct Irradiation Segment	DICAL	E-4		
Shadow Factor Generator Segment	SFCAL	E-5		
Radiation Interchange Segments	RKCAL	E-6	GBCAL	E-6
	RCCAL	E-6		
Absorbed Heat Segment	AQCAL	E-7		
Absorbed Heat Output Segment	QOCAL	E-8		

4.3 Subroutine Descriptions

4.3.1 Basic Concepts

The user-callable subroutines in the processor library fall into two functional groups. The most numerous group consists of subroutines used to define the program variables and set the logic flags that are required before a computational segment can be linked into the processor and executed. Variable definition in this manner, as opposed to definition from the data blocks, achieves two important goals. First, in any complex problem many segment calls are made, necessitating frequent redefinition of program variables. Under these conditions, data block input would be redundant. Second, the subroutine calls, in classic FORTRAN format, form natural groups of input variables, and as the calls are input serially in the user's operations block, he is provided a highly visible presentation of the variable definitions existing at each stage of his execution. Thus, if the user carefully proofreads a listing of his operations block, his logic and variable definitions should be error-free. If his geometry inputs have been verified using the plot package, the user may proceed with some confidence to consume a large block of computer time. It is the hope of the TRASYS designers that these features will ease somewhat the all too prevalent garbage in-garbage out syndrome.

The second group of processor library subroutines perform data handling tasks that eliminate redundant calculations. For example, direct irradiation may be required at 15 orbit points for a sun-oriented spacecraft. Armed with the knowledge that his solar flux is everywhere constant (outside the planet shadow), the user may compute the solar flux in Step 1, then use the STFAQ routine to retrieve the data from Step 1 and place it in data storage for any of the other 14 steps he desires.

Appendix D is composed of summary descriptions of each user subroutine. Definitions of each variable in the calling sequences are given, together with their default values, where applicable. The additional material necessary to use the subroutines is presented in the remainder of this section. After achieving a working knowledge, the user should find Appendix D sufficient for his quick-reference needs.

4.3.2 General Subroutines

Subroutines BUILDC and ADD are used to choose, from the various blocks of surfaces in the surface input data, what

blocks are to be assembled to create the problem geometry. Also, the relative spatial positions of the surfaces and nodes in the active blocks may be changed using subroutine CHGBLK.

4.3.2.1 Subroutine BUILDC

Calling Sequence: CALL BUILDC (BCSNAM)

This subroutine begins the process of assembling the geometry desired from the blocks of surfaces in the surface data block. BCSNAM is any block coordinate system name found in the surface data block. If no BCS is named in the surface data block, CALL BUILDC (ALLBLK) will define a geometry consisting of the entire surface data block. This call must be made for geometry definitions or after geometry redefinition via subroutine CHGBLK. A BUILDC call voids any previous BUILDC and ADD calls.

4.3.2.2 Subroutine ADD

Calling sequence: CALL ADD (BCSNAM)

This subroutine adds another block of surfaces to the geometry defined by previous BUILDC and ADD calls. One BUILDC call must precede any ADD call in the operations block.

4.3.2.3 Subroutine CHGBLK

Calling sequence: CALL CHGBLK (BCSNAM, TX, TY, TZ, IROTX, IROTY, IROTZ, ROTX, ROTY, ROTZ)

This subroutine is used to spatially relocate blocks of surfaces by redefining a block coordinate system's location and orientation in central coordinate system 3-space. BCSNAM is the Hollerith name identifying the block coordinate system being changed. Its spelling must be exactly as called out in the BCS data block.

The arguments TX, TY, TZ, ROTX, ROTY, ROTZ are the translation and rotation parameters necessary to transform the central coordinate system into the block coordinate system in its new position. These parameters are further discussed in paragraphs 3.3.3 and 3.3.4.

The arguments IROTX, IROTY and IROTZ control the order in which the rotations ROTX, ROTY and ROTZ are performed. They may take on the integer values 1, 2, or 3. For example, for

IROTX = 3, IROTY = 1, and IROTZ = 2, the rotations will be performed in the order ROTY first, ROTZ second, and ROTX third. If zero is passed for all three arguments, the default values IROTX = 1, IROTY = 2, and IROTZ = 3 will result and the rotations will be performed ROTX first, ROTY second, and ROTZ third.

4.3.3 Form Factor Subroutine

The subroutine FFDATA is used to set the variables and control constants required before executing the FFCAL computation segment. An FFDATA call prior to all FFCAL executions is not mandatory because each FFDATA argument assumes a default value (ref Appendix D). The variables defined by an FFDATA call will hold for any subsequent FFCAL executions in the operations block.

4.3.3.1 Subroutine FFDATA

Calling sequence: CALL FFDATA (FFACC, FFACCS, FFNOSH, FFRATL, FFMIN, FFPRNT, FFPNCH)

FFACC is the variable that provides user control of the node surface elemental breakdown used for double integration form factor calculations. In general, the accuracy of a form factor calculation is proportional to the ratio of each elemental area divided by the square of the distance between each elemental area pair involved. Thus, if the element count for each node were chosen so that a given value of this ratio were never exceeded, then the error of each form factor calculation would similarly be limited. The form factor segment logic provides for this, and FFACC is the upper limit allowed for the area - distance squared ratio. The default value used, (FFACC = .1) provides form factor accuracy of approximately 3 percent for parallel flat plates. Background information for this accuracy relationship can be found in Appendix B. The user is cautioned that his problem run time is tied directly to his nodal element count, and indiscriminate reduction in the value of FFACC can be costly. The recommended approach to accuracy improvement is to selectively re-compute suspect form factors using a reduced FFACC value.

When node pairs are situated such that the interelement distances vary a great deal, it is sometimes necessary to temporarily subdivide node pairs in order to obtain sufficient accuracy. The parameter FFRATL controls this. The number of elements is dictated by a weighted average distance and compared to FFRATL. If this value exceeds FFRATL, the node pair is subdivided. When

accuracy problems are encountered with node pairs having large interelement distance variation (nodes with congruent edges, for instance), the recommended procedure is to enter an FFRATL value lower than the default value (FFRATL = 15.), and selectively recompute the form factors. No change in FFACC should be required for this operation. Additional descriptive material on this technique can be found in Appendix B.

The elemental breakdown of node pairs also influences form factor accuracy when shadowing by intervening surfaces is involved. For large magnitude form factors, the node pair element breakdown required to satisfy shadowing considerations is computed and used if it exceeds that dictated by separation distance (see Appendix B). The element count dictated by shadowing is inversely proportional to the parameter FFACCS. The default value for FFACCS (FFACCS = .1) was chosen based on experience. If the user knows that one or more of his significant form factors will be heavily influenced by shadowing, the recommended procedure is to selectively compute such form factors using a reduced value of FFACCS.

The parameter FFMIN is used to reduce radiation interchange factor compute time by inserting zeros in the form factor matrix in place of very small values. The default value, FFMIN = 1. - E-6 is sufficiently small that it does not materially affect the energy balance of the problem.

The remaining FFDATA parameters, FFNOSH, FFPRNT and FFPNCH are used to override shadowing computations and to provide print and punch output options. The user is referred to Appendix D for instruction in their use.

4.3.4 Plot Package Subroutines

4.3.4.1 Subroutines NDATA, NDATAS

Calling sequences:

```
CALL NDATA (NV, VU, SCL, SELN, TIT, IROTX,
IROTY, IROTZ, ROTX, ROTY, ROTZ)

CALL NDATAS (NV, VU, SCL)
```

These calls are used to define plot parameters prior to executing the NPLOT program segment. A call to one of these routines prior to an NPLOT execution is not mandatory, because all arguments have default values (ref Appendix D). The variables defined by NDATA or NDATAS calls will hold for any subsequent NPLOT execution in the operations block.

The NV parameter allows the user to make up to 6 NDATA or NDATAS calls, thereby defining up to 6 plot operations, before executing NPLOT. One NPLOT execution will execute all the plot operations defined.

VU defines the type of plot desired. The options are 3H3-D, 1HX, 1HY, 1HZ, 3HALL, and 3HGEN. 3-D results in a 3-dimensional pictorial plot. X, Y, and Z produce orthographic projections of the geometry as seen from the X, Y, and Z axes of the CCS, respectively. ALL results in four frames, 3-D, X, Y, and Z. GEN is a general 3-D plot, where the user has control of the orientation of the CCS axes relative to his point of view.

SCL is the plot scale factor, defined by:

SCL = length on plot frame/length of surface where lengths of surfaces are as defined in the surface data block.

The user should keep in mind that his hardcopy plot frames are probably about 17.8 cm (7 inches) square.

SELN is the name of the array that contains a list of the integer node identification numbers the user desires to plot selectively. The selective node number array is entered in the array data block.

TIT is the name of the Hollerith array containing any title the user desires on his plot frame. Up to 66 characters are allowed.

IROTX, IROTY, IROTZ, ROTX, ROTY, and ROTZ are the group of six parameters defining point of view from which the user will "see" his problem geometry in a general view. For ROTX, ROTY, and ROTZ identically zero, the central coordinate system appears in plots as shown in Figure 4-1.

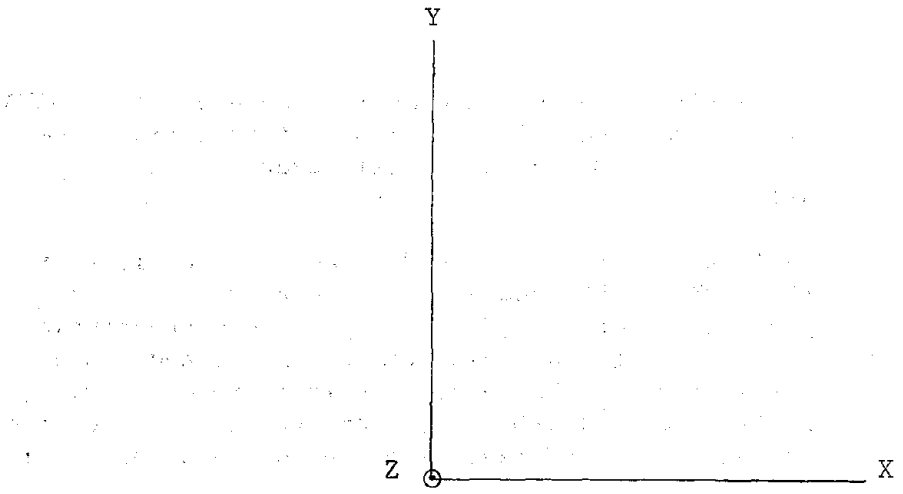


Figure 4-1 Node Plot Coordinate System Reference

Using Figure 4-1 as the reference position, the user may arbitrarily relocate the axes by defining ROTX, ROTY and ROTZ to relocate the reference system so that it coincides with the desired systems location. The order of the rotations is defined using IROTX, IROTY, and IROTZ.

4.3.4.2 Subroutines ODATA, ODATAS

Calling sequences: CALL ODATA (NV, VU, SCL, SCLR, RPLN, TRUEAN, TIMEST, TIME, SELN, TIT, IROTX, IROTY, IROTZ, ROTX, ROTY, ROTZ)

CALL ODATAS (NV, VU, SCL, SCLR, RPLN, TRUEAN, TIMEST, TIME)

These subroutines are functionally analogous to NDATA and NDATAS in relation to execution of the orbit plotter segment, OPLOT.

NV is defined and functions identically with the similarly named parameter in NDATA. VU defines the plot type. Options are 3H3-D, 4HBETA, 5HCIGMA, 3HSUN, 3HALL, and 3HGEN. 3H3-D results in a 3-dimensional pictorial plot of the planet and spacecraft. 4HBETA results in an edge-on view of the orbit plane, with the BETA angle shown true. The 5HCIGMA view places the orbit plane in the plot frame, as seen from north of the celestial equator. 3HALL produces four frames, 3-D, CIGMA, BETA, and SUN. GEN results in a plot with the orbit coordinate system axes rotated according to user definition. SCL relates the user's geometry dimensions to plot scale according to:

SFAC = SCL/OPMAX (NNS)

where

SFAC is the absolute surface scale factor (same as SCL in NDATA),

OPMAX (NNS) is the maximum extension of any surface point from the CCS origin (user surface data units).

The user should generally enter a value of SCL equal to about 1/2 the desired planet radius in inches of plot frame.

SCLR is the distance from the planet center where the user desires to see his CCS origin (in inches on plot frame).

RPLAN is the planet radius as plotted in inches. The planet radius default value used is 3.56 cm. (1.4 inches). The default values for SCL and SCLR are related to RPLAN according to the relationships:

SCLR = 8.*RPLAN/7.

SCL = (3.15 - SCLR)/2.

The user may note that his spacecraft's altitude, as it appears in the plots, is not related in any way to actual orbit altitude. This is because the primary reason for orbit plots is for visualization of orientation. Orbit radius is, however, available in common as the variable RTHET in the operations block. Therefore, if the user cares to consider the scaling involved, he may write operations block logic to relate SCLR to actual orbit radius.

TRUEAN is the true anomaly at the orbit point being defined for plotting (degrees from periapsis passage).

TIME is the time at which the orbit point plot is desired, in hours (required only if TRUEAN is not defined).

TIMEST is time of periapsis passage, in hours (required only if TRUEAN is not defined).

The remaining ODATA arguments, SELN, TIT, IROTX, IROTY, IROTZ, ROTX, ROTY, and ROTZ, are exactly analogous to the identically named NDATA arguments (ref Section 4.3.4.1). Figure 4-2 depicts the orbit coordinate system as plotted for ROTX = ROTY = ROTZ = 0.

4.3.4.3 Subroutine PLDATA

Calling sequence: CALL PLDATA (IPLUNT, IPLSN, IPLNA, PLCRUF, PLLABX, PLLABY, PLTIT1, PLTIT2, PLXMPF, PLYMPF)

This subroutine is used to define parameters necessary to execute the output data plotter segment PLOT. Refer to Appendix D, p D-21 for argument definitions.

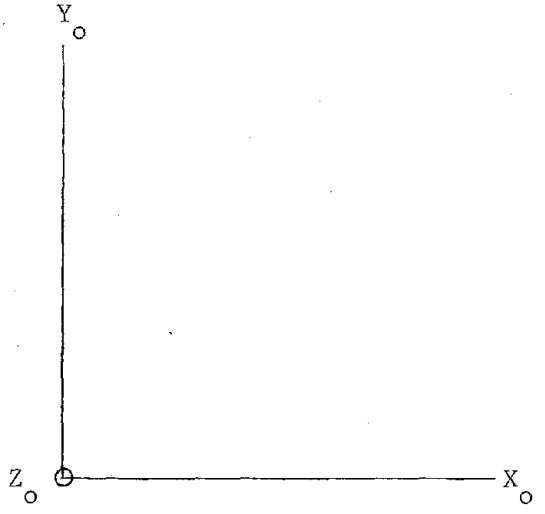


Figure 4-2 Orbit Plot Coordinate System Reference

4.3.5 Direct Irradiation Subroutines

The direct irradiation subroutines are used to spatially locate the spacecraft relative to energy sources. The various calls give the user the option of locating his spacecraft using classical orbit parameters, with a modified sun-referenced set of orbit parameters, with look angles, or with trajectory tape parameters. Subroutines are also available for defining spacecraft orientation and spin rate.

4.3.5.1 Subroutine ORBIT1

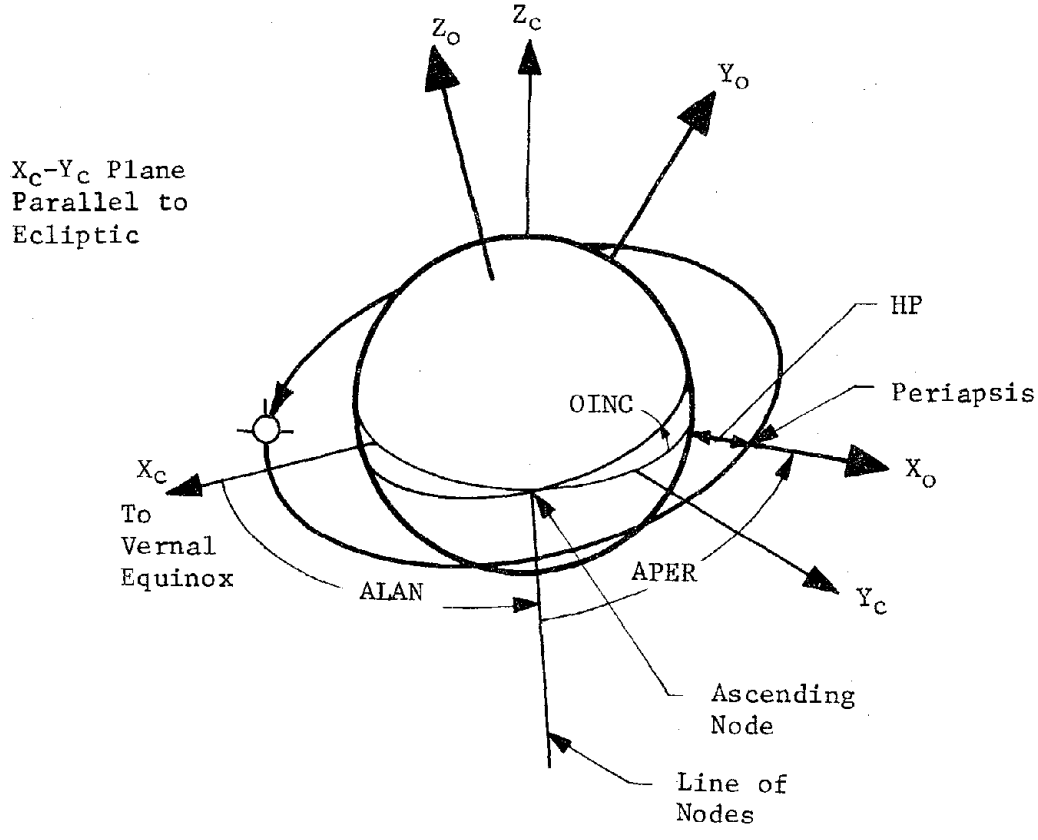
Calling sequence: CALL ORBIT1 (PNAME, ALAN, APER, OINC, TIMEST, HP, HA, SUNRA, SUNDEC, STRRA, STRDEC)

or:

CALL ORBIT1 (PNAME, ALAN, APER, OINC, TIMEST, HP, ECC, SUNRA, SUNDEC, STRRA, STRDEC)

This subroutine defines an orbit using classic orbit parameters and locates the sun in the same celestial coordinate system referenced to the Vernal Equinox (reference Figures 4-3 and 4-4).

In the case of a star-oriented spacecraft, the star is located in the celestial coordinate system, in the same manner as the sun.



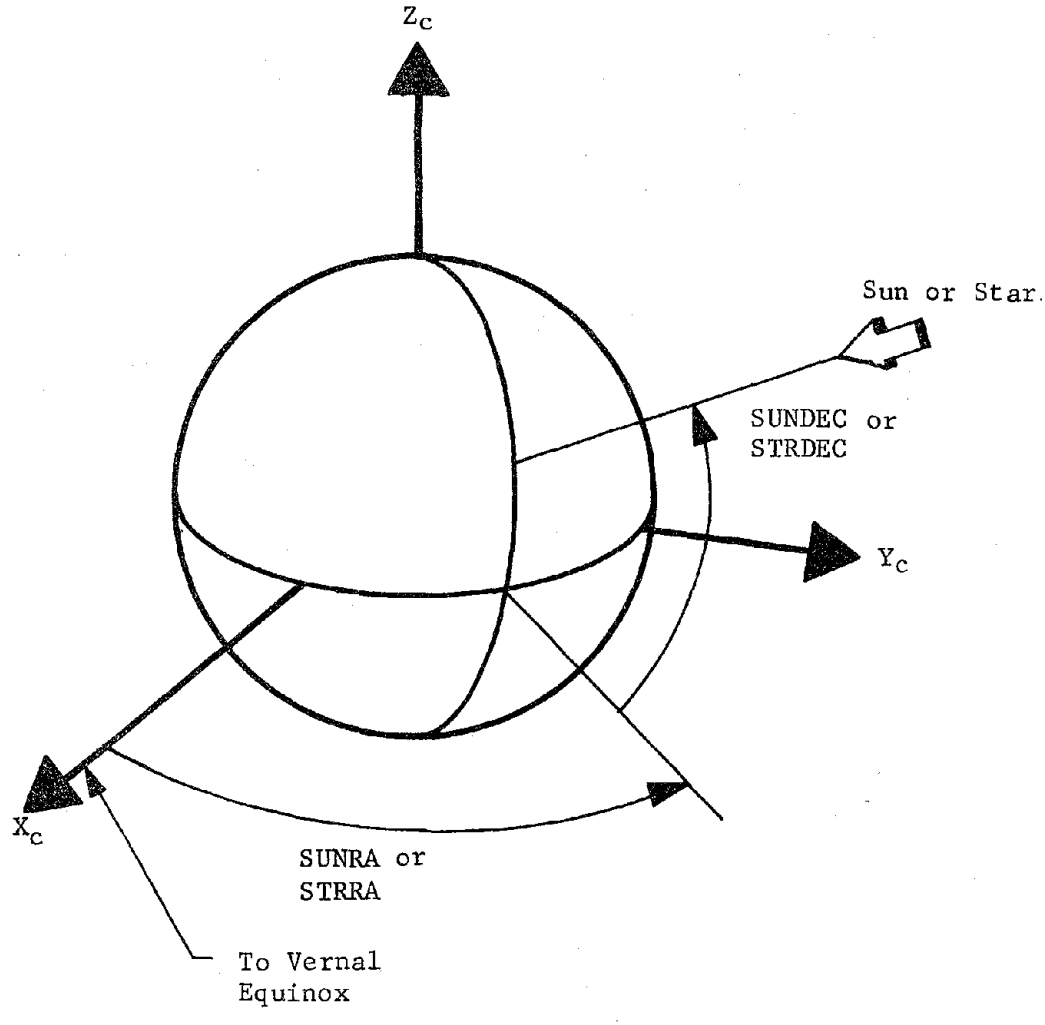
ALAN - Longitude of ascending node measured from X_c axis to line of nodes; positive toward Y_c

APER - Argument of perifocus. measured in orbit X_o-Y_o plane in direction of S/C motion from ascending node to periapsis

HP - Altitude at periapsis

OINC - Orbit inclination (angle between X_c-Y_c plane and orbit plane as seen from ascending node; $0. \leq OINC \leq 180^o$ ($OINC > 90^o$ for retrograde orbits)

Figure 4-3 Orbit Definition in a Celestial Coordinate System



SUNRA, STRRA - Right ascension of sun/star; measured in X_c-Y_c plane from X_c axis; positive toward Y_c axis; $0. \leq RA \leq 360$

SUNDEC, STRDEC - Declination of sun/star; positive from X_c-Y_c plane toward Z_c; $-90. \leq DEC \leq 90$

Figure 4-4 Sun and Star Locations in Celestial Coordinate System

PNAME is a Hollerith name used as a flag to direct the definition of the planet-dependent parameters. The allowable PNAME options are: 3HMER, 3HVEN, 3HEAR, 3HMOO, 3HMAR, 3HJUP, 3HSAT, 3HNEP, 3HURA, and 3HSUN. These names serve to define the following variables:

PRAD - planet radius

SOL - solar constant at the average planet-sun distance

PALB - planet albedo value (surface solar reflectance)

WDS - infrared emissive power at planet surface, dark side

WSS - infrared emissive power at planet surface, subsolar point

GRAV - acceleration of gravity at planet surface

The values obtained from the different planet name arguments are tabulated in Table 4-1. Note that the values tabulated are in metric units. The values stored in core, however, are in the TRASYS base units system, that is, length in feet, time in hours, energy in British thermal units. If the user desires to manipulate these quantities using his own operations block FORTRAN code, he would expect them to be in the ft - hour - Btu units.

Note that only Mercury and the Earth's moon are treated as bodies with nonuniform surface temperatures. This is correct for airless, slow-rotating planets. For these two bodies, the emissive power is considered everywhere constant on the dark side. On the sunlit side, the emissive power reduces from the subsolar value to the darkside value at the terminator, according to a cosine law.

The user is cautioned that his results using PNAME = 3HMER may be extremely misleading. This planet's eccentric orbit, plus its nearness to the sun, results in a solar constant variation of from approximately 6 to over 10 Earth "suns" during the Mercury year. Corresponding variations in the subsolar emissive power occur. The recommended procedure for PNAME = 3HMER is for the user to properly define SOL and WSS according

Table 4-I Stored Planet Property Values

Planet	Planet Radius (km)[a]	(ft)	Albedo	Solar Constant at Mean Planet Distance (w/m^2)[b]	(B/Ft^2-hr)	Darkside Emissive Power (w/m^2)[c]	(B/ft^2-hr)	Subsolar Emissive Power (w/m^2)[c]	(B/ft^2-hr)	(m/s^2)[d]	(ft/s^2)
Mercury	2485.	(8.153 E06)	0.058	8920.	(2830)	0.	(0.)	8402.	(2666.)	3.513	(11.49)
Venus	6199.	(20.34 E06)	0.76	2570.	(815.5)	154.2	(48.93)	154.2	(48.93)	8.462	(24.68)
Earth	6370.	(20.90 E06)	0.30	1352.	(429)	236.6	(75.08)	236.6	(75.08)	9.844	(32.20)
Moon	1738.	(5.702 E06)	0.047	1352.	(429)	6.5	(2.060)	1288.	(408.7)	1.622	(5.306)
Mars	3314.	(10.87 E06)	0.148	577.3	(183.2)	123.0	(39.03)	123.0	(39.03)	3.921	(12.83)
Jupiter	69885.	(229.3 E06)	0.51	49.6	(15.74)	6.1	(1.936)	6.1	(1.936)	26.04	(85.18)
Saturn	57515.	(188.7 E06)	0.50	14.7	(4.66)	1.8	(.5711)	1.8	(.5711)	11.17	(36.54)
Uranus	25482.	(83.61 E06)	0.66	3.65	(1.16)	.31	(.0983)	.31	(.0983)	11.52	(37.68)
Neptune	24850.	(81.53 E06)	0.62	1.48	(.47)	.14	(.0444)	.14	(.0444)	8.977	(29.36)
Sun	698500.	(2291. E06)		--		6.262×10^7		6.262×10^7		273.8	(895.6)

[a] Values stored in program are in ft.

[b] Referenced to 1352 w/m^2 (429 Btu/hr-ft^2) at 1 AU. Values stored in program are in Btu/ft^2-hr.

[c] Values stored in program are in Btu/hr-ft^2.

[d] Values stored in program are in ft/s^2.

to his knowledge of the planet-sun distance. This is done using two FORTRAN statements immediately following his ORBIT1 call. This same technique is available, of course, whenever the user desires to change WDS, WSS, or SOL to values other than the built-in nominals. The user's values will hold until another ORBIT call is encountered.

ALAN, APER, OINC, HP, and HA are the longitude of the ascending node, argument of perifocus, orbit inclination, and periapsis and apoapsis altitudes. These are the five parameters necessary to define an orbit in the celestial coordinate system. The alternate ORBIT1 call allows the input of eccentricity (ECC) in lieu of apoapsis altitude. TIMEST is the time of periapsis passage, in hours.

The angular measurement arguments are in decimal degrees of arc. The altitudes must be specified in feet. Note that FORTRAN allows arithmetic operations within argument lists; thus the following ORBIT1 call might be used where HP is known to be 150 nautical miles:

```
CALL ORBIT1 (3HEAR, 32., 90., 22.5., 0., 6080.
*150., .94, -41., 18., 0., 0.)
```

SUNRA and SUNDEC are the right ascension and declination of the sun, respectively, input in decimal degrees of arc (See Figure 4-4).

STRRA and STRDEC are the right ascension and declination, respectively, of a star for a star-oriented mission (see Figure 4-4). Zero or dummy arguments are passed for non-star-oriented missions.

For heliocentric orbits, (PNAME = 3HSUN) ALAN, APER, OINC, SUNRA, and SUNDEC have no meaning and are passed as zero or dummy arguments.

4.3.5,2 Subroutine ORBIT2

Calling sequences:

```
CALL ORBIT2 (PNAME, CIGMA, BETA, CIGMAS,
BETAS, TIMEST, HP, HA)
```

or:

```
CALL ORBIT2 (PNAME, CIGMA, BETA, CIGMAS,
BETAS, TIMEST, HP, ECC)
```

This subroutine defines an orbit using sun-referenced parameters in the orbit coordinate system. The orbit coordinate system has its X and Y axes in the orbit plane and its Z axis completes a right-handed set. The spacecraft travels from X towards Y.

PNAME functions identically with ORBIT1.

CIGMA and BETA locate the solar vector in the orbit coordinate system (see Figure 4-5). CIGMAS and BETAS locate a star in the orbit coordinate system (see Figure 4-5). Again, these arguments are zero or dummy for non-star-oriented missions.

For heliocentric orbits, (PNAME = 3HSUN) ORBIT2 is not applicable.

4.3.5.3 Subroutine ORIENT

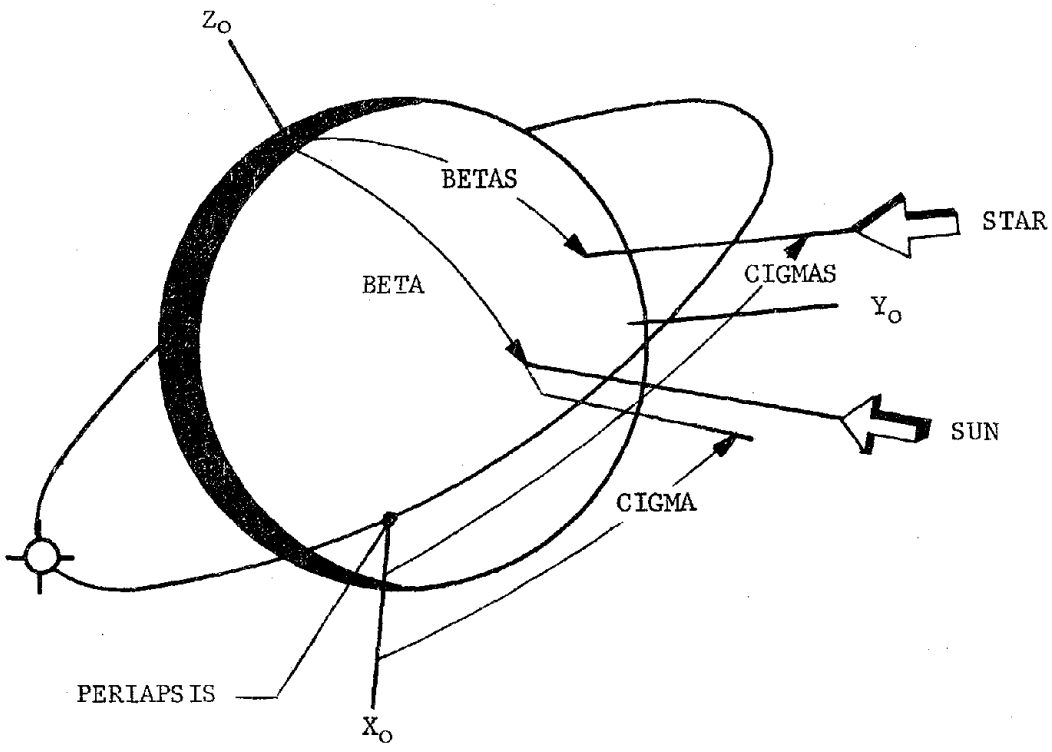
Calling sequence: CALL ORIENT (TYPE, IROTX, IROTY, IROTZ, ROTX, ROTY, ROTZ)

This subroutine is used to define the spacecraft orientation relative to space-environment heat sources. Orientation is accomplished by relating the spacecraft central coordinate system to a vehicle coordinate system (VCS) that remains fixed, relative to a heat source or a star reference.

TYPE is a Hollerith name used as a flag to define orientation of the VCS. Allowable options for TYPE are 4HPLAN, 3HSUN, 4HSTAR, or 4HTAPE. Figure 4-6 depicts the VCS relationship to the heat sources, star reference, and orbit coordinate system for the PLAN, SUN, and STAR options. The X_V-axis points to the planet, sun, or star and the Z_V-axis is in the same half-space as the Z_O-axis. The Y_V-axis lies in the orbital plane and completes the right-handed set. The TAPE option allows orientation to be defined from a trajectory tape.

An ambiguity exists when the sun or star vector is parallel to the Z_O-axis. In this case, the Y_V-axis is defined to be in the direction to the velocity vector.

IROTX, IROTY, IROTZ, ROTX, ROTY, and ROTZ are the rotation parameters necessary to locate the spacecraft CCS relative to the VCS and, hence, the heat source(s). ROTX is the rotation angle to rotate the VCS into the CCS; it rotates about X_V-axis, Y_V toward Z_V positive. ROTY is the same as ROTX, except about the Y_V-axis; ROTZ is the same as ROTX, except about the Z_V-axis, X_V toward Y_V positive.



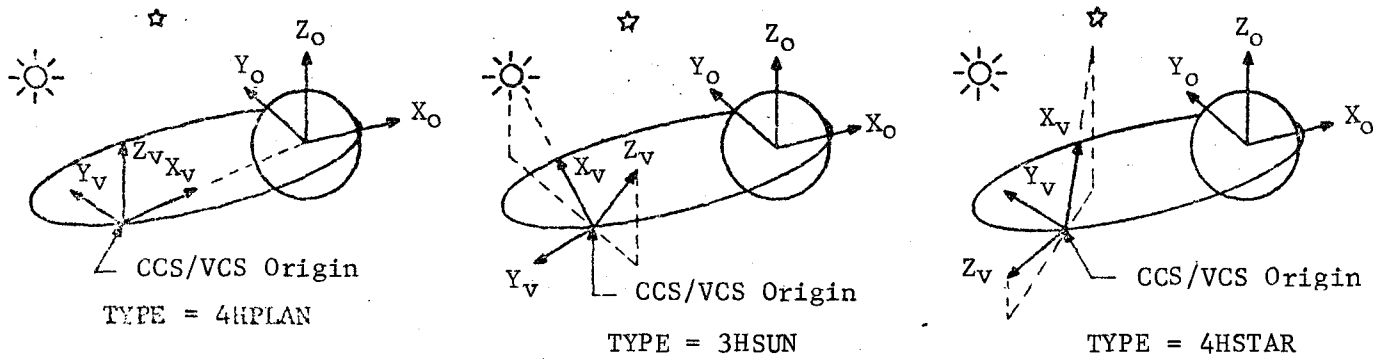
CIGMA - Angle from X_O axis to sun vector projection in X_O - Y_O plane. Measured CCW as seen from Z_O axis (in direction of S/C motion).
$0. \leq$ CIGMA ≤ 360

CIGMAS - Same as CIGMA except to star vector projection

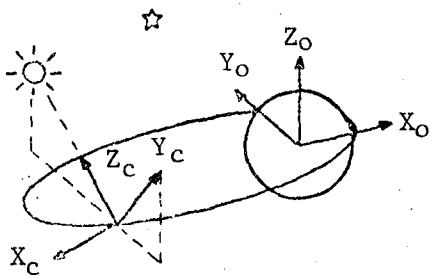
BETA - Angle from Z_O axis to sun vector
$0 \leq$ BETA ≤ 180

BETAS - Same as BETA, except to star vector

Figure 4-5 Orbit Definition in Orbit Coordinate System



Orientation Example
(CCS Z-axis locked to sun):
TYPE = 3HSUN



IROTX = 1
IROTY = 2
IROTZ = 3 — Rotates VCS into CCS, -270° about Y axis
ROTX = 0.
ROTY = -270.
ROTZ = 90° — Rotates 90° about Z axis

Figure 4-6 Vehicle Orientation with Subroutine ORIENT

IROTX, IROTY, and IROTZ control the order in which the rotations are performed. Integers 1, 2, and 3 are the allowed options. For example, IROTX = 1, IROTZ = 2, and IROTY = 3 results in rotation first about X_v, then about Z_v, and then about Y_v.

4.3.5.4 Subroutines DIDT1 and DIDT1S

Calling sequences: CALL DIDT1 (DINOSH, DIACC, DIACCS, TRUEAN, NSPFF, TIMEPR, DIPNCH)

or:

CALL DIDT1S (TRUEAN, NSPFF, TIMEPR, DIPNCH)

These subroutines allow the user to define the form factor and shadowing accuracy parameters used in his direct irradiation calculations. Additionally, these routines can be used to update the spacecraft position in orbit by defining true anomaly (reference Figure 4-7). True anomaly can be defined directly or by defining a current time.

DINOSH is a shadow/no shadow flag for direct irradiation calculations. DINOSH = 4HSHAD retains shadowing calculations. DINOSH = 4HNOSH bypasses shadowing calculations.

DIACC is the element selection accuracy factor for node planet form factor calculations. Its function is similar to FFACC in form factor calculations and its default value is 0.25. (ref para 4.3.3.1 and Appendix B).

DIACCS is the element selection accuracy factor for shadowing calculations (not applicable when DINOSH = 4HNOSH). Its function is similar to FFACCS in form factor calculations and its default value is 0.1. (ref para 4.3.3.1 and Appendix B).

TRUEAN is the true anomaly of the spacecraft measured in decimal degrees of arc from periapsis passage in direction of spacecraft motion.

TIMEPR is used to define true anomaly in terms of time; TIMEPR is current time, in hours. If TIMEPR is defined, TRUEAN in decimal degrees is returned to the operations block in common. If TRUEAN is defined, current time is returned to the operations block under the variable name TIMEPR.

NSPFF specifies a step number within which a planetary flux calculation was made. If the FORTRAN statement: PLSAVE =

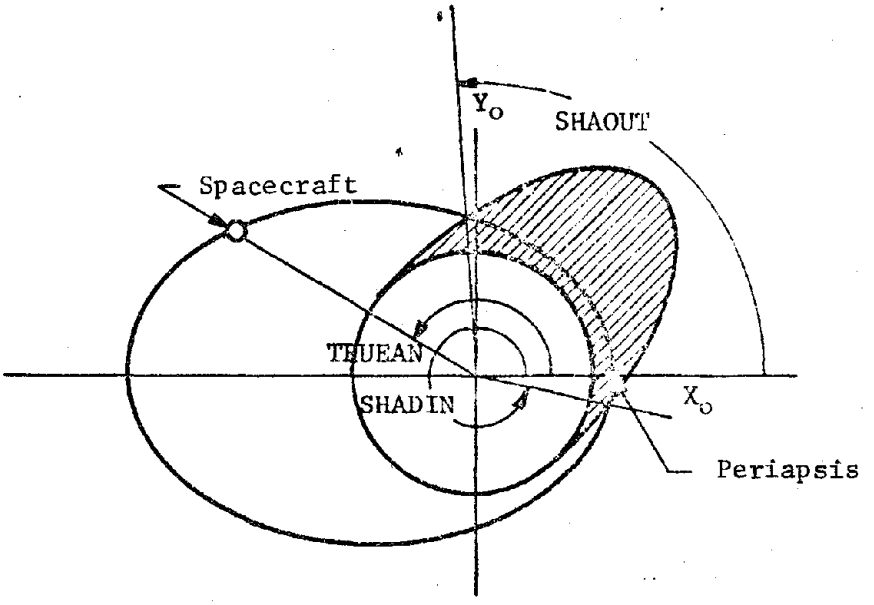


Figure 4-7 Definition of True Anomaly and Shadow Entry/Exit Points

4HSAVE appears prior to this calculation, the form factor matrix from the spacecraft to planet will be stored out of core under step NSPFF. If this form factor matrix is valid for additional orbit points (i.e. circular orbit, planet oriented), the FORTRAN statement PLSAVE = 4HREAD is made in the first step subsequent to NSPFF. This will result in the planet form factor calculations being bypassed for the subsequent steps, with a corresponding saving in computer time.

Subroutine DIDT1S is a short form version of DIDT1 to be used when the user does not desire to specify his accuracy and shadow/no shadow flag. See Appendix D for DIDT1 and DIDT1S argument default values.

DIPNCH is the flag for punching direct irradiation data in the flux data block format. Options are 3HPUN, 2HNO, and 4HTAPE.

4.3.5.5 Subroutines DIDT2 and DIDT2S

Calling sequences: CALL DIDT2 (DINOSH, DIACC, DIACCS, NSPFF, SUNCL, SUNCO, PLCL, PLCO, TIMEPR, ALT, DIPNCH)

or:

CALL DIDT2S (NSPFF, SUNCL, SUNCO, PLCL, PLCO, TIMEPR, ALT, DIPNCH)

These subroutines are identical in function to DIDT1 and DIDT1S in that they define the shadowing and accuracy parameters to be used in the subsequent direct flux segment execution, as well as furnish the parameters necessary to define the spacecraft's spatial relation with the sun and planet heat sources.

The arguments DINOSH, DIACC, DIACCS, and NSPFF are exactly as discussed in paragraph 4.3.5.4 and tabulated in Appendix D.

SUNCL, SUNCO, PLCL, and PLCO are the clock and cone angles needed to define the direction of the sun and planet position vectors in vehicle coordinate system 3-space. Figure 4-8 shows how these parameters are defined. Their input units are decimal degrees of arc.

ALT is the spacecraft altitude, above the planet, this argument must be input in feet.

It should be noted that a DIDT2 call is not sufficient to define all the variables needed for a direct irradiation segment execution. In general, an ORBIT1 or ORBIT2 call must be made, or the variables PRAD, SOL, PALB, WDS, and WSS (ref para 4.3.5.1) must be defined individually in operations block FORTRAN statements. The call to ORBIT1 or ORBIT2 need only define PNAME. The remaining arguments may be dummys.

The user should also be aware that spacecraft spin, as defined by subroutine SPIN, is not applicable when DIDT2 or DIDT2S

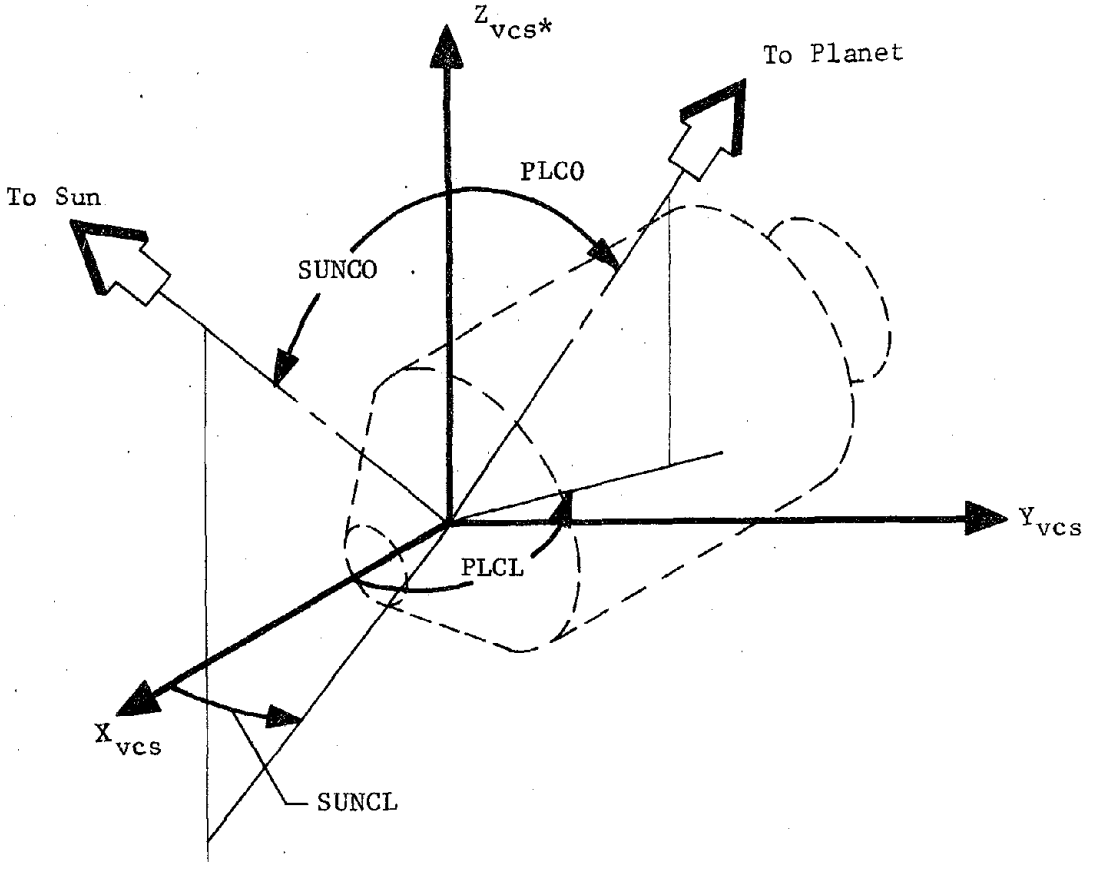


Figure 4-8 Spacecraft Orientation with Subroutine DIDT2

are called, since DIDT2 and DIDT2S directly define spacecraft orientation as well as position in space.

4.3.5.6 Subroutine SPIN

Calling sequence: CALL SPIN (CLOCK, CONE, RATE, TRUANS, SPNTM)

---

*If subroutine ORIENT is not called prior to DIDT2 or DIDT2S, the vcs and ccs coincide. This is the recommended mode of use. Star is not allowed as an orient type when using DIDT2 or DIDT2S.

This subroutine is used to define spacecraft spin. The arguments CLOCK and CONE define the spin axis with reference to the central coordinate system. RATE defines the spacecraft spin rate about the spin axis in degrees per hour. Figure 4-9 illustrates the clock and cone angles, and the algebraic sign convention used with RATE.

The time spin begins is defined through TRUANS or SPNTM. If the user knows the time his spin begins, he specifies it directly as SPNTM. If he knows the true anomaly where it begins he specifies TRUANS and passes zero for SPNTM.

Spacecraft spin computations are done on the basis of the following: the spacecraft is assumed to be in the orientation defined by the last call to subroutine ORIENT at SPNTM. At any subsequent points in time, the spacecraft is reoriented, presuming a constant spin-rate, about the SPIN-defined spin axis, over the time elapsed since SPNTM.

4.3.5.7 Subroutine DICOMP

Calling sequence: CALL DICOMP (ISOLFL, IALBFL, IPLAFL)

This subroutine allows the user to define the logic used in a subsequent DICAL execution. The choice of computing, stuffing from another step, or zeroing out individual solar, albedo, and planetary fluxes is available. See Appendix D, p D-23, for argument definitions.

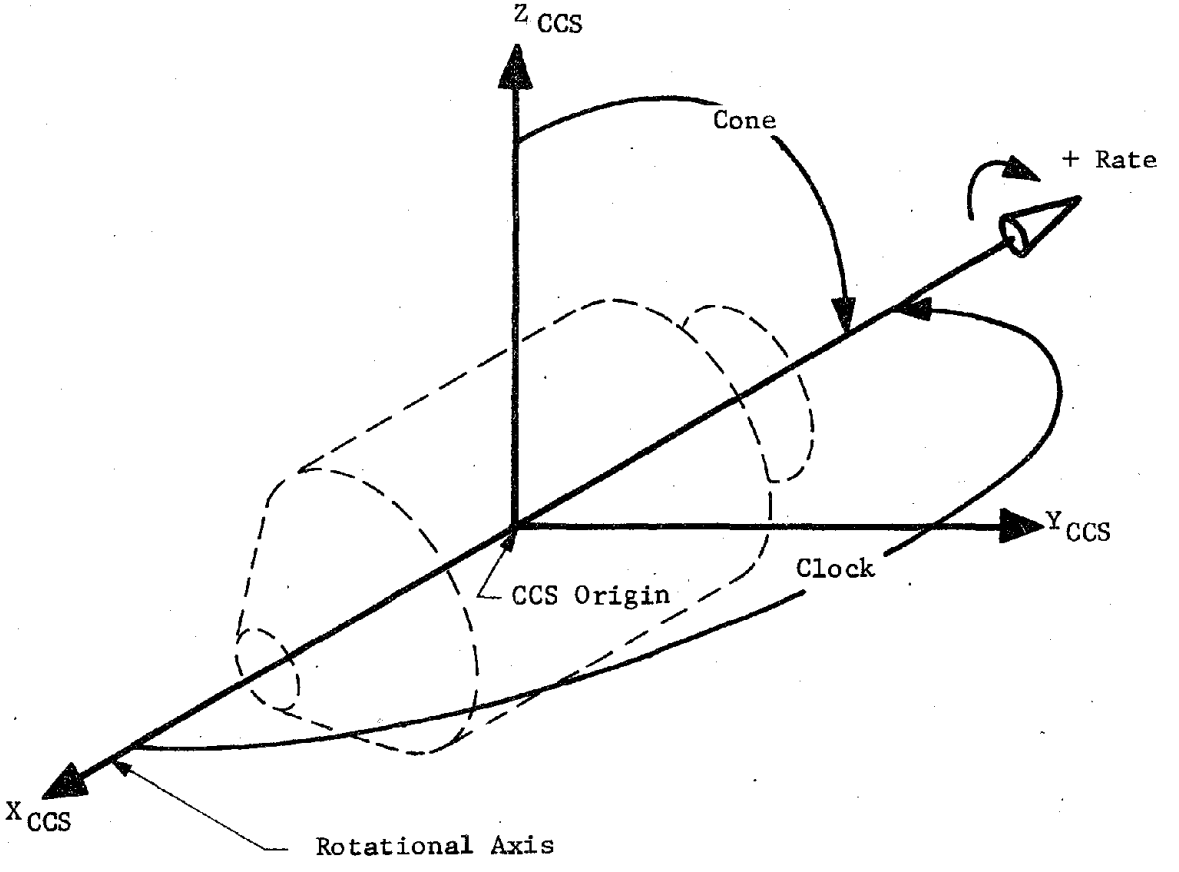
4.3.5.8 Subroutines DITTP and DITTPS

Calling sequence: CALL DITTP (TIME, ITYPE, PLANAM, IDWDN, FIDEN, NTIM, NTYPE, NCLPL, NCOPL, NCLS, NCOS, NRAD, NWOR, ALTMF, IFLS, DIPNCH)

or: CALL DITTPS (TIME, ITYPE)

These subroutines allow the user to define his mission by reading trajectory tapes of the attitude timeline variety. The pertinent data are read from the tape and placed in storage for use by the DICAL segment through an internal call to DIDT2.

Subroutine DITTP allows the user to define the trajectory tape format, identify the proper file on multifile tapes, define the attitude parameters for his first compute point, and position the tape for reading subsequent points. Subsequent



Note: Spin axis coordinates illustrated for the case clock = 180, cone = 90

Figure 4-9 Spacecraft Spin Definition

points are read using DITTPS, which presumes that the tape is previously positioned to the correct file, and a call to DITTPS results in repeated reads of trajectory tape records until a time value equal to TIME is encountered. If ITYPE is defined (as an integer data value), repeated reads are made until TIME is encountered, then reading continues until a special event identifier

equal to ITYPE is encountered. Note that this tape reading method precludes calling for a time value less than the time argument used in a previous DITTP or DITTPS call.

Figure 4-10 is an operations data block segment that generates direct irradiation for three time points, using a trajectory tape. In Step 2, the trajectory tape is positioned to a file named ZLV1 and the data are read at TIME = 10.0 hours, which is not a special event point. The planet involved is Earth, flux output is punched, and the spacecraft altitude data on the tape are in nautical miles (ALTMF = 6080.). Trajectory tape format information is as follows:

a) Tape records are 58 words long (NWOR = 58).
b) Tape is for 1 body (IBOD = 0).
c) File identification is found in word 1 of tape records (IDWDN = 1),
d) Time is found in word 3 of tape records (NTIM = 3).
e) Special event identifier is found in word 5 of tape records (NTYPE = 5).
f) Planet center-to-spacecraft distance is found in word 13 of tape records (NRAD = 13).
g) Clock angle-to-planet vector is found in word 9, (NCLPL = 9).
h) Cone angle-to-planet vector is found in word 10, (NCOPL = 10).
i) Clock angle-to-sun vector is found in word 11, (NCLS = 11).
j) Cone angle-to-sun vector is found in word 12, (NCOS = 12).

Step 2 reads trajectory tape information at time = 10.5 hours, which is not a special event. Step 3 reads trajectory tape information at a special event of type 2, which occurs just subsequent to 11.0 hours.

4.3.6 Radiation Interchange Subroutines

4.3.6.1 Subroutine GBDATA

Calling sequence: CALL GBDATA (IGBSFF, GBWBND)

This subroutine defines the parameters necessary prior to executing the GBCAL segment to obtain a grey-body factor matrix.

```
HEADER OPERATIONS DATA
STEP  1
      CALL BUILDC(ALLBLK)
L     FFCAL
      CALL GBDATA(0,4HBOTH)
L     GBCAL
STEP  2
      CALL DITTP(10,0,0,3HEAR,1,4HZLV1,3,5,9,10,11,12,13,58,6080,,0,
     13HPUN)
L     DTCAL
STEP  3
      CALL DITTPS(10,5,0)
L     DTCAL
STEP  4
      CALL DITTPS(11,0,2)
L     DTCAL
```

Figure 4-10 Trajectory Tape Operations Example

Argument IGBSFF identifies the previously executed step number that computed or otherwise defined the form factor matrix corresponding to the grey-body factor matrix desired. Prior to the GBDATA call, calls to BUILDC and ADD must be in effect that define the geometry as it was defined in Step IGBSFF.

Argument GBWBND (Options: 3HSOL, 2HIR, 4HBOTH) defines the energy waveband--solar, infrared, or both--that will be used in grey-body factor calculation.

4.3.6.2 Subroutine RKDATA

Calling sequence: CALL RKDATA (IRKNGB, RKPNCH, RKMIN, IRKCN, RKSP, IRKNSP, SIGMA, RKAMPF, RKTAPE)

This subroutine defines the parameters necessary prior to executing the RKCAL program segment to obtain radiation conductors (RADKs) in thermal analyzer format.

Argument IRKNGB identifies the previously executed step number in which grey-body factor matrix corresponding to the desired radiation conductors was computed. Prior to the RKDATA call, calls to BUILDC and ADD must be in effect that define the problem geometry as it was in Step IRKNGB.

Argument RKPNCH (Options: 3HPUN, 2HNO) is the punch/no punch flag for radiation conductors on BCD card format.

Argument RKMIN defines the lower limit of the radiation conductor values that will be punched or put on BCD tape. RKMIN is defined as follows:

for a valid radiation conductor,

$$f_{ij} / \epsilon_i > \text{FMIN}$$

where

f_{ij} is the grey-body factor from node i to j,
ϵ_i is the infrared emittance of node i.

Argument IRKCN is the initial radiation conductor identification number. The radiation conductors are numbered consecutively from IRKCN.

Arguments RKSP and IRKNSP provide the information to define radiation conductors to space for problems that do not form a complete enclosure. RKSP (Options: 5HSPACE, 2HNO) is the flag for calculation of radiation conductors to space. When RKSP = 5HSPACE, radiation conductors to space for node i are computed according to:

$$f_{\text{i-space}} = \epsilon_i - \sum_{j}^{N} f_{ij}$$

for an N-node problem. IRKNSP is the user-defined identification number for his space node.

SIGMA and RKAMPF are available for the user to obtain unit agreement between his radiation model and thermal analyzer model. SIGMA is the Stefan-Boltzmann constant that will appear on the radiation conductor and RKAMPF is an arbitrary multiplication factor available to change from the TRASYS standard area units (square feet) to the area unit the user desires. If RKAMPF is 1.0, the area unit associated with SIGMA must be square feet.

Argument RKTAPE (Options: 4HTAPE, 2HNO) allows the user to write his radiation conductors to his BCD RADK tape (thermal analyzer format).

All RKDATA arguments have default values (see Appendix D) so that an RKDATA call before an RKCAL execution is not mandatory.

4.3.7 Absorbed Heat Subroutines

4.3.7.1 Subroutine AQDATA

Calling sequence: CALL AQDATA (IAQGBI, IAQGBS, RSOLAR, RALB, RPLAN)

This subroutine defines the parameters necessary prior to executing the AQCAL program segment to compute absorbed heats.

All arguments are step number references to the current or previously executed steps that have resulted in valid direct irradiation and grey-body data being placed in out-of-core storage.

IAQGBI is the step number reference for an infrared grey-body matrix.

IAQGBS is the step number reference for a solar grey-body matrix.

RSOLAR is a solar-heat rate multiplying factor (defaults to 1.0).

RALB is an albedo-heat-rate multiplying factor (defaults to 1.0).

RPLAN is a planetary-heat-rate multiplying factor (defaults to 1.0).

Step number arguments will default to the current step number, so that an AQDATA call is not required if all necessary data are in storage under the current step number. Prior to an AQCAL execution, the BUILDC and ADD calls in effect must agree exactly with those in effect when each of the five referenced steps were executed.

4.3.7.2 Subroutine STFAQ

Calling sequence: CALL STFAQ (TRUEAN, TIMEPR, NSTP)

This subroutine stuffs values of absorbed heat and/or direct flux computed in a previously executed step into out-of-core storage for the current step. It also stores time for the current step, defined either directly or from true anomaly.

The argument NSTP is the step number from which the desired absorbed heat values will be obtained.

The geometry, as defined by BUILDC and ADD calls, in effect at the time any STFAQ call is made must agree exactly with that in effect when step NSTP was executed.

4.3.7.3 Subroutine QODATA

Calling sequence: CALL QODATA (NSARRY, NTMARY, QOTAPE, QOPNCH, QOAMPF, QOFMPF, QOTMPF, QOTYPE, IQOCOR)

This subroutine defines the parameters necessary to allow absorbed heat data in thermal analyzer format to be generated in a subsequent QOCAL execution.

Argument NSARRY is the name of an array containing the previously executed step numbers where the desired absorbed-heat data can be found in storage. Unless NSARRY = 3HALL, this array must be entered in the array data block and must be dimensioned to agree exactly with the number of valid step numbers it contains. The user is referred to Figure 3-13 for examples of ways this array may be defined. If the 3HALL option is used, all absorbed heat data computed since the last call to BUILDC will be output.

Argument NTMARY is the thermal analyzer array number the user desires for his time array when Q vs time tables are being generated. The Q arrays generated will be numbered consecutively from NTMARY+1.

Arguments QOTAPE and QOPNCH are flags to control the form of Q table output. Options are 3HPUN, 2HNO for punch/no punch control, and 4HTAPE, 2HNO for write/no write to BCD tape.

Arguments QOAMPF, QOFMPF and QOTMPF are the multiplying factors for area, energy, and time, respectively. The default values of 1.0 result in time in hours, area in square feet, and energy in Btu/hr.

Argument QOTYPE controls the type of output obtained. 3HTAB results in Q vs time tables; 2HAV results in an average Q for the time period defined by NSARRY.

Argument IQOCOR defines the correspondence data block step number to be used. This argument defaults to the current step number.

4.3.8 Shadow Factor Subroutine

4.3.8.1 Subroutine SFDATA

Calling sequence: CALL SFDATA (NAMEI, NAMEO)

This subroutine is used to label files on shadow factor input (SHADI) and shadow factor output (SHADO) tapes. This call is mandatory if a SHADI tape is used so that the correct file is selected. SHADO tape files are identified with the Hollerith name NAMEO.

4.3.9 Data Modification Routines

A series of routines are available that enable the user to change certain types of data from the operations data block. This provides a convenient way to perform many types of parametric studies without the necessity of making multiple runs and error-prone changes to the surface data.

The series of routines allows the following node properties to be changed:

a) Area;

b) Diffuse infrared emissivity and/or solar absorptivity;

c) Specular infrared and/or solar reflectivity;

d) Infrared and/or solar transmissivities;

e) SHADE/BSHADE flags.

Calling sequences are designed so that the properties may be changed for one or all active nodes with one call. The use and function of these routines is explained in the following sections.

4.3.9.1 Subroutine MODAR

Calling sequence: CALL MODAR (ND, AR)

This subroutine changes the area of a designated node or the area of all currently active nodes by use of a multiplier.

Argument Name	Description	Options	Default
ND	Node Number Designator	a) Any Active Node Number (Integer).	None
		b) 3HALL	
AR	Desired Value for Area	a) Floating-Point Data Value	None
		b) Area Multiplier[1] (3HALL Option Only)	

Note: 1. When ND = 3HALL, all active node areas are modified according to AREA = AREA*AR.

Restriction: Call not valid prior to geometry definition through calls to BUILDC and ADD.

4.3.9.2 Subroutine MODPR

Calling sequence: CALL MODPR (ND, ALPHA, EMISS)

This subroutine modifies the diffuse infrared emissivity and/or the diffuse solar absorptivity of a designated node.

Argument Name	Description	Options	Default
ND	Node Number Designator	Any Active Node Number	None
ALPHA	Diffuse Solar Absorptivity	a) $0. \leq DV \leq 1.$	None
		b) $DV < 0.$	
EMISS	Diffuse IR Emissivity	a) $0. \leq DV \leq 1.$	None
		b) $DV < 0.$	

Note: 1. If ALPHA < 0. or EMISS < 0., current values are not changed.

Restriction: Call not valid prior to geometry definition through calls to BUILDC and ADD.

4.3.9.3 Subroutine MODTR

Calling sequence: CALL MODTR (ISR, TRANS, TRANI)

This subroutine modifies the solar and/or intrared transmissivity of a designated surface.

Argument Name	Description	Options	Default
ISR	Surface Number Designator	Any Active Surface Number	None
TRANS	Solar Transmissivity	a) $0. \leq DV \leq 1.$	None
		b) $DV < 0.$[1]	
TRANI	IR Transmissivity	a) $0. \leq DV \leq 1.$	None
		b) $DV < 0.$[1]	

Note: 1. If TRANI < 0. or TRANS < 0., current values are not changed.

2. Transmissivity changes affect the entire surface.

Restriction: Call not valid prior to geometry definition through calls to BUILDC and ADD.

4.3.9.4 Subroutine MODPRS

Calling sequence: CALL MODPRS (ND, SPRS, SPRI)

This subroutine modifies the solar and/or infrared specular reflectivity of a designated node.

Argument Name	Description	Options	Default
ND	Node Number Designator	Any Active Node Number	None
SPRS	Specular Reflectivity, Solar	a) $0. \leq DV \leq 1.$	None
		b) $DV < 0.$[1]	
SPRI	Specular Reflectivity, Infrared	a) $0. \leq DV \leq 1.$	None
		b) $DV < 0.$[1]	

Notes: 1. If SPRI < 0. or SPRS < 0., current values are not changed.

Restrictions: 1. This call is applicable only to nodes defined as specular reflectors in the surface data block.

2. Call not valid prior to geometry definition through calls to BUILDC and ADD.

4.3.9.5 Subroutine MODSHD

Calling sequence: CALL MODSHD (ISR, SHADE, BSHADE)

This subroutine modifies the SHADE/BSHADE flags for a designated surface.

Argument Name	Description	Option	Default
ISR	Surface Number Designator	Any Active Surface Number	None
SHADE	Can Shade Flag	FF, DI, BOTH, NO, 0[1]	None
BSHADE	Can Be Shaded Flag	FF, DI, BOTH, NO, 0[1]	None

Note: 1. If SHADE or BSHADE data values are zero, their values are not changed.

2. Shade flag changes affect the entire surface.

Restrictions: 1. Call not valid prior to geometry definition through calls to BUILDC and ADD.

2. Call not applicable to shadower-only surfaces.

4.3.10 Approximate Radiant Interchange Factors

A routine is available in the processor library that computes diffuse-grey-body interchange factors according to the first-order approximation:

$$\mathcal{F}_{ij} = \text{PROPI}*\text{PROPJ}*F_{ij}$$

where

PROPI and PROPJ are the diffuse surface properties, solar absorptivity or infrared emissivity; $\mathcal{F}_{ij}$ is the approximate radiant interchange factor between surfaces i and j; F_{ij} is the form factor.

This equation is, of course, exact for a black enclosure (PROPI = PROPJ = 1 for all i, j) and gives a reasonable approximation if all surface properties are 0.8 or greater.

A call to subroutine GBAPRX in lieu of executing the GBCAL segment will generate the approximate grey-body factor data and store them in the same manner as GBCAL. There is a significant saving in computer time when the problem size is 200 nodes or greater.

4.3.10.1 Subroutine GBAPRX

Calling sequence: CALL GBAPRX (IGBSFF, GBWBND)

This subroutine calculates grey-body radiant interchange factors using an approximate relationship and stores the results in data storage.

Argument Name	Description	Options	Default
IGBSFF	Step Number for Form Factors	Integer	Current Step No.
GBWBND	Waveband Definition Name	2HIR, 3HSOL 4HBOTH	4HBOTH

Note: Input zero for default action.

Restriction: None.

4.3.11 Radiation Condenser Segment

The radiation condenser segment provides the user with two methods of radiation model simplification.

The first of these methods, which is referred to as the Multiple Enclosure Simplification Shield (MESS) technique, allows a complex radiation enclosure to be modularized into discrete subenclosures by the assignment of imaginary interface shield nodes. Each of these smaller enclosures can be analyzed independently of the others, resulting in more efficient use of computers and manpower.

The second method, referred to as the Effective Radiation Node (ERN) technique, is used to reduce the number of radiation couplings required to thermally model an enclosure by replacing small conductors from each node with a single conductor coupled to the enclosure ERN.

The techniques and their application are described in more detail in Appendix F.

4.3.11.1 Subroutine RCDATA

Calling sequence: CALL RCDATA (IRCNGB, RCPNCH, RCMIN, IRCCN, RCSP, IRCNSP, SIGMA, RCAMPF, RCTAPE, RFRAC, NERN, IPRIME, ISECND)

This is a user-called subroutine that defines the parameters used in RCCAL for the condensation and output of radiation conductors (RADKS).

Variable Descriptions and Default Values

Variable	Description	Default Value
IRCNGB	Step Number Reference for Infrared Grey-Body Factors	Assumes Current Step
RCPNCH	Punch/No Punch Flag. Options: 3HPUN, 2HNO	3HPUN
RCMIN	Minimum Value of F/ε That Will Result in a Valid RADK	0.0001
IRCCN	Initial Radiation Conductor Number	1
RCSP	Flag for Calculation of RADKS to Space. Options: 5HSPACE, 2 HNO	2HNO
IRCNSP	Space Node Number	32767
SIGMA	Stefan-Boltzmann Constant	1.713E-9
RCAMPF	Area Multiplying Factor	1.0
RCTAPE	Flag to Write RADKS to BCDOU Tape. Options: 4HTAPE, 2HNO	2HNO
RFRAC	Significant Radiation Fraction: Ref Appendix F, equation 6.	None
NERN	Effective Radiation Node (ERN) Number	None
IPRIME	Array Name for Array of Primary MESS Node Numbers and Special Node Numbers	None
ISECND	Array Name for Array of Secondary MESS Node Numbers	None

Restrictions:

1. RCDATA must be called prior to RCCAL execution because all of the variables are not defaulted.

2. Current model geometry must agree with that of step IRCNGB.

3. IPRIME and ISECND arrays must be input in the array data block to specify MESS node pairs and special nodes. IPRIME contains a list of all primary MESS nodes and all special nodes in that order. ISECND contains a list of all secondary MESS nodes in a one-to-one correspondence with the primary MESS nodes in IPRIME.

4.3.12 Adiabatic "Closure" Surfaces

It is sometimes desirable to conserve energy in a thermal radiation problem that does not constitute a complete enclosure. The usual means of doing this is to complete the enclosure with an adiabatic reflector surface. This can be accomplished by entering a rudimentary closure surface in the surface data and using subroutine ADSURF to add the closure surface to the form factor matrix after form factors have been computed for the real surfaces in the problem.

4.3.12.1 Subroutine ADSURF

Calling Sequence: CALL ADSURF (BCSN, FFSN)

BCSN is the block coordinate system name under which the closure surface appears in the surface data. FFSN is the previously executed step number under which form factors were computed.

When ADSURF is called, the form factor matrix is read from step FFSN, and form factors from each node to the closure surface are computed by subtracting the form-factor row sums from 1.0. This new row of form factors is added to the form factor matrix and the resulting matrix is stored under the current step. Two form factor matrices are now available to the user for use in generating radiation interchange data.

5. PROCESSOR SEGMENTS

5.1 Pictorial Plot Segments

5.1.1 Node Plotter

Calling sequence: L NPLOT

This segment provides the user with 3-dimensional and/or orthographic projection pictorial plots of his problem geometry. Its primary use is to verify surface data input prior to proceeding with computations of radiation interchange or absorbed heat data. Examples of its output can be found in Appendix H.

This segment has no provision for user intervention in the form of program called subroutines that the user may modify. Control is provided through the NDATA or NDATAS subroutines.

5.1.2 Orbit Plotter

Calling sequence: L OPLOT

This segment provides the user with a pictorial representation of his spacecraft in relation to the body it orbits and the sun.

The planet and its shadow are depicted, together with a pictorial view of the spacecraft in orbit. The standard output enables the user to verify his orbit in relation to the sun, and spacecraft orientation relative to the sun, planet, or star. Examples of its output can be found in Appendix H.

This segment has no provision for user intervention beyond that provided by subroutines ODATA and ODATAS.

5.1.3 Data Plotter

Calling sequence: L PLOT

This segment provides the capability to plot any computed or input data as x versus y plots. The segment automatically writes a binary plot data unit (disc or drum) for producing plots of incident or absorbed heat rates or fluxes as a function of time (Ref subroutine PLDATA, Appendix D-21).

The segment also provides a completely general plot capability if the user inputs operations data block FORTRAN to prepare the plot data unit prior to executing the PLOT segment. This type of plot operation is illustrated by the plot unit format described below.

The plot segment flow diagram is shown in Figure 5-1.

5.1.4 Binary Plot Unit Format

Write format: NAME, N, (DATA (I), I = 1, N)

where:
- NAME: type of record
- N : number of words in record
- DATA: array of data

Record 1, TYPE = FRAME, N = 6

word	1	5HFRAME
	2	4
	3	XMIN
	4	XMAX
	5	YMIN
	6	YMAX

Record 2, TYPE = LABELX, N = 7

word	1	6HLABELX
	2	MAXIMUM OF 5, 30 characters, maximum, 6/per word
	3	LABEL ARRAY (1)
	4	(2)
	5	(3)
	6	(4)
	7	(5)

Record 3, TYPE = LABELY, N = 7 (max)

word	1	6HLABELY
	2	N (MAXIMUM OF 7, 30 characters, maximum, 6/per word)
	3	LABEL ARRAY (1)
	4	(2)
	5	(3)
	6	(4)
	7	(5)

Record 4, TYPE = NODENO, N = 3

word	1	6HNODENO
	2	1
	3	INTEGER NODE #

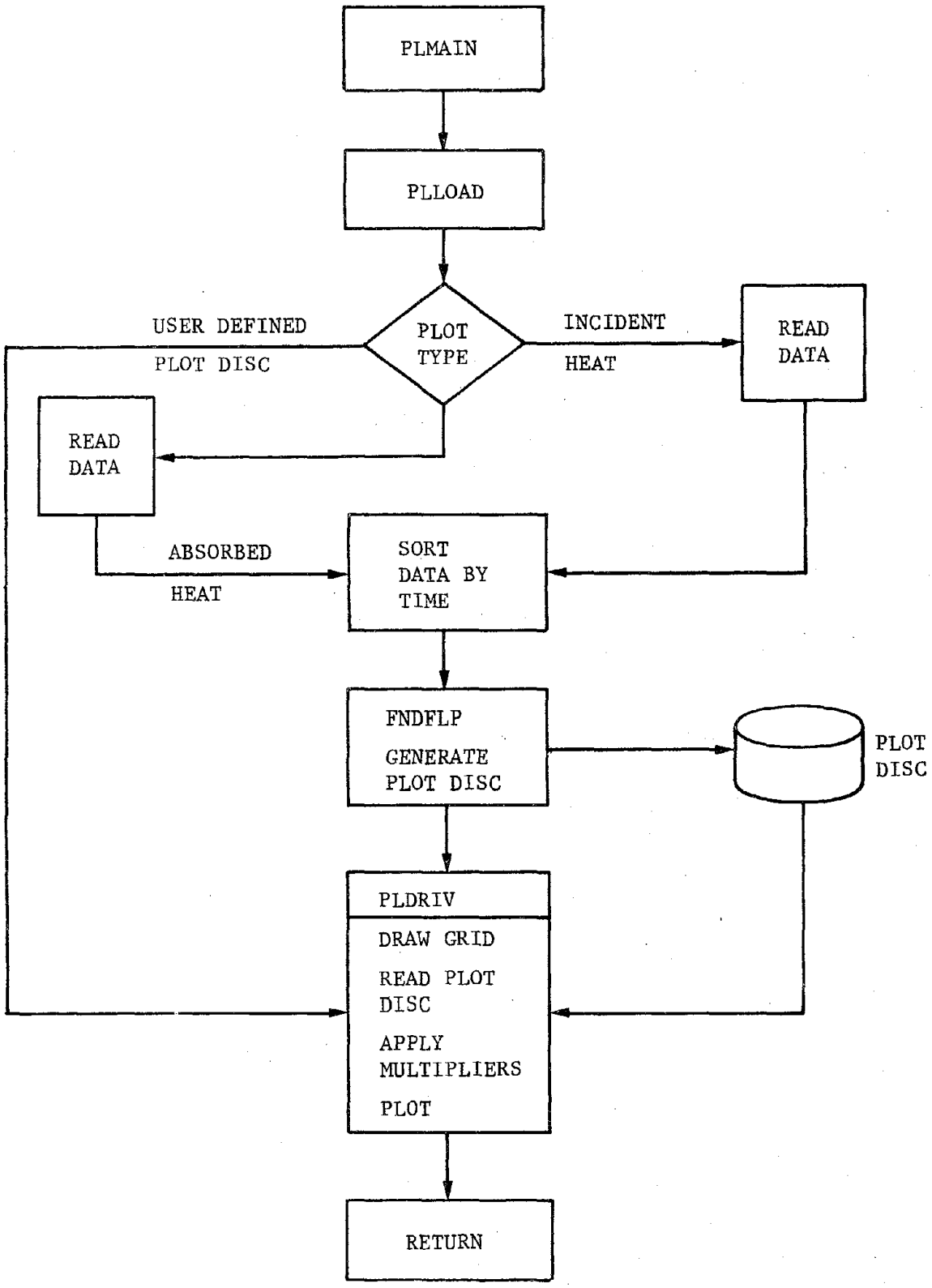


Figure 5-1 PLOT Segment Flow Diagram

Record 5, TYPE = TITLE 1, N = 12 (max)

word	1	6HTITLE 1
	2	N(MAXIMUM OF 12, 60 characters, maximum, 6/per word)
	3	TITLE ARRAY 1
	4	2
	.	.
	.	.
	12	10

Record 6, TYPE = TITLE 2, N = 14 (max)

word	1	6HTITLE 2
	2	N(MAXIMUM OF 14, 72 characters, maximum, 6/per word)
	3	TITLE ARRAY 1
	4	2
	.	.
	.	.
	14	12

Record 7, TYPE = INDEP, N = user supplied

word	1	5HINDEP
	2	N ($1 \leq N \leq 1000$)
	3	DATA (1)
	.	
	.	
	N+2	DATA (N)

Record 8, TYPE = DEPEND

word	1	6HDEPEND
	2	N ($1 \leq N \leq 1000$)

Note: a) TYPE DEPEND can occur as many times on a file as desired for multiple plots on a frame.

b) Any records but FRAME, INDEP and DEPEND types may be omitted.

5.2 Form Factor Segment

Calling sequence: L FFCAL

This segment computes form factor matrices for any geometric enclosure using a numerical integration method. Internode blockage is accounted for with differing solar and IR transmissivities and, because semitransparent and specular surfaces are

allowed, two form factor matrices are computed: FFS for the solar waveband and FFI for the infrared waveband.

Provision for user intervention via the subroutine data block is available through three program-called subroutines: (1) prior to computation of each form factor through subroutine FFPRE, (2) at the completion of a row of form factors through subroutine FFROW, and (3) at the completion of the entire matrix through subroutine FFEND. The logic flow of the FFCAL segment is shown in Figure 5-2.

FFCAL provides printed output, as well as punched-in form factor data block format, at the user's option. Examples of FFCAL output are shown in Appendix H.

5.3 Shadow Factor Segment

Calling sequence: L SFCAL

This segment computes shadow factor tables for each node of a spacecraft configuration to be used in direct irradiation calculations when analytic shadow effect calculations are not desirable.

Primary output for this segment is a tape containing shadow factor tables for each node. This tape contains shadow data packed according to a format as presented in Appendix C. The shadow factor tables are also output in printed form. An example of the printed output resulting from an SFCAL direction can be found in Appendix H. Shadow factors for both the solar and infrared wavebands are computed, because semitransparent shadowing surfaces are allowed.

No user intervention is provided for this segment. Prior to an SFCAL call, the user may set a flag to obtain punched shadow factor data through the statement SFPNCH = 3HPUN. Punched output obtained will be in shadow factor data block format.

5.4 Radiation Interchange Segments

Calling sequences: L GBCAL
L RKCAL

Segment GBCAL computes a matrix of diffuse grey-body radiation interchange factors and places them in out-of-core storage for later use in computing absorbed heat or radiation conductors. Solutions for either the solar or infrared wave bands may be requested. No user intervention provisions are made beyond that of subroutine GBDATA.

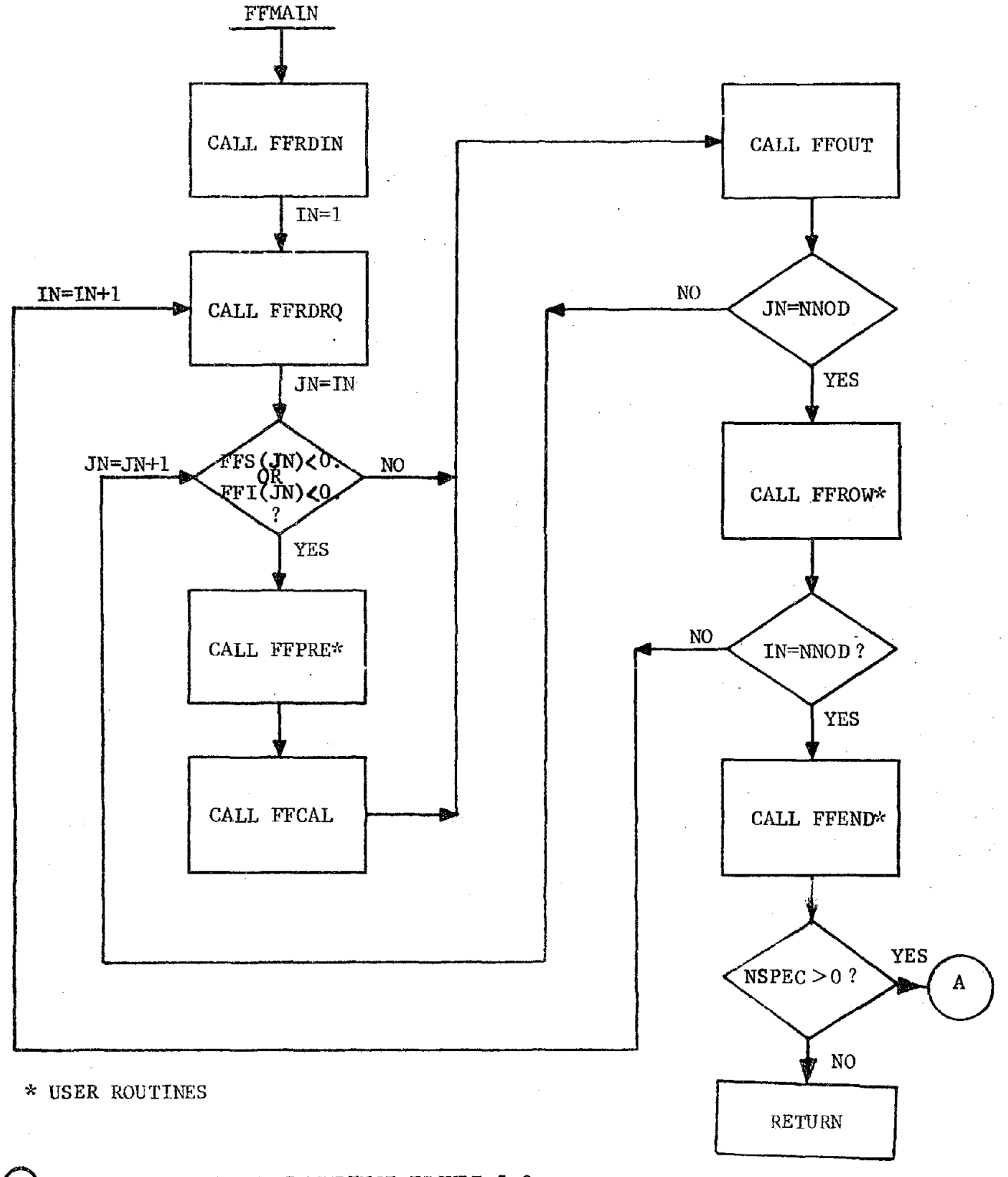


Figure 5-2 Segment FFCAL Flow Diagram

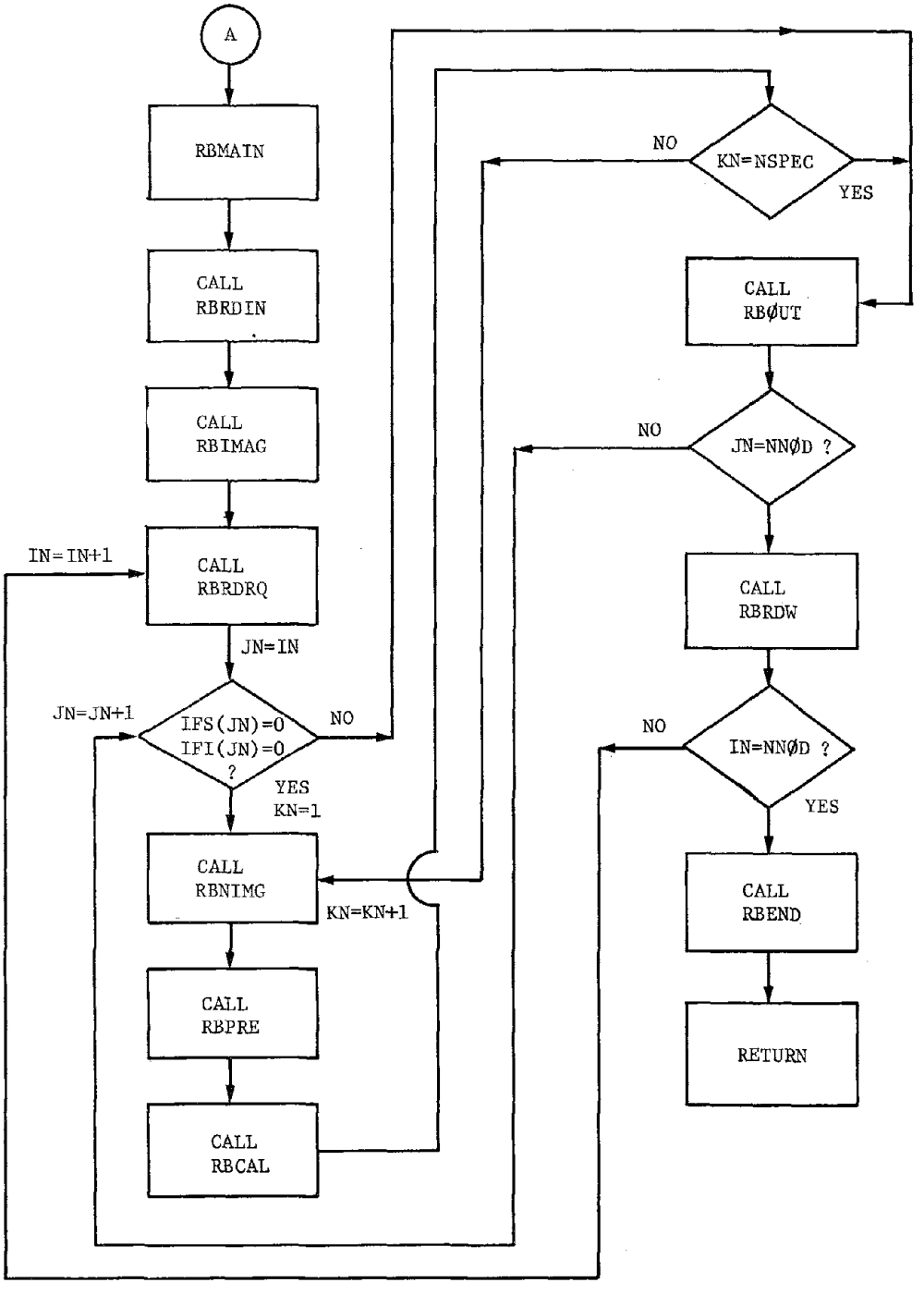


Figure 5-8 Segment RBCAL Flow Diagram

Segment RKCAL computes radiation conductors for thermal analyzer models and provides output in punched card or BCD tape form. A printout of the card/tape record images is also provided. Three program-called user routines are used to provide user intervention through his subroutines block. Subroutine RKPRE provides for any special initialization desired before computations begin. Subroutine RKPNCH performs the actual punch/tape write operations, and the user may obtain data in any thermal analyzer program format by altering format statements in this routine. User routine RKEND provides for user intervention prior to return to operations block control. Figure 5-4 shows segment RKCAL logic flow.

An example of RKCAL output can be found in Appendix H.

5.5 Direct Irradiation Segment

Calling sequence: L DICAL

This segment computes the thermal radiation directly incident on external spacecraft surfaces due to the presence of the sun or a nearby planet. Three components are computed: direct solar, reflected solar from the planetary surface (albedo) and infrared planetary emission. Shadowing effects due to internode blockage are accounted for. Normally, shadowing is computed analytically; however, at the user's option shadowing may be computed by the use of shadow data provided on a shadow factor tape. This requires that the shadow factor tape be mounted and the flag SFTAPE be defined through a FORTRAN statement in the operations block prior to the DICAL call. For example, if the statement:

```
SFTAPE = CONF1
```

appears prior to a DICAL call, the analytic shadowing calculations will be bypassed. Further, a shadow factor tape file with the name CONF1 will be used for shadow calculations. If no file with this name is found, the run will abort.

Four program called subroutines are provided for user intervention through his subroutine data block. Subroutine DIPRES provides for special initialization prior to solar flux calculations Similarly, DIPREP is called prior to planetary/albedo flux calculations. Subroutine DIENDS and DIENDP provide for user intervention subsequent to solar and planetary/albedo calculations, respectively. Figure 5-5 depicts the logic flow of segment DICAL.

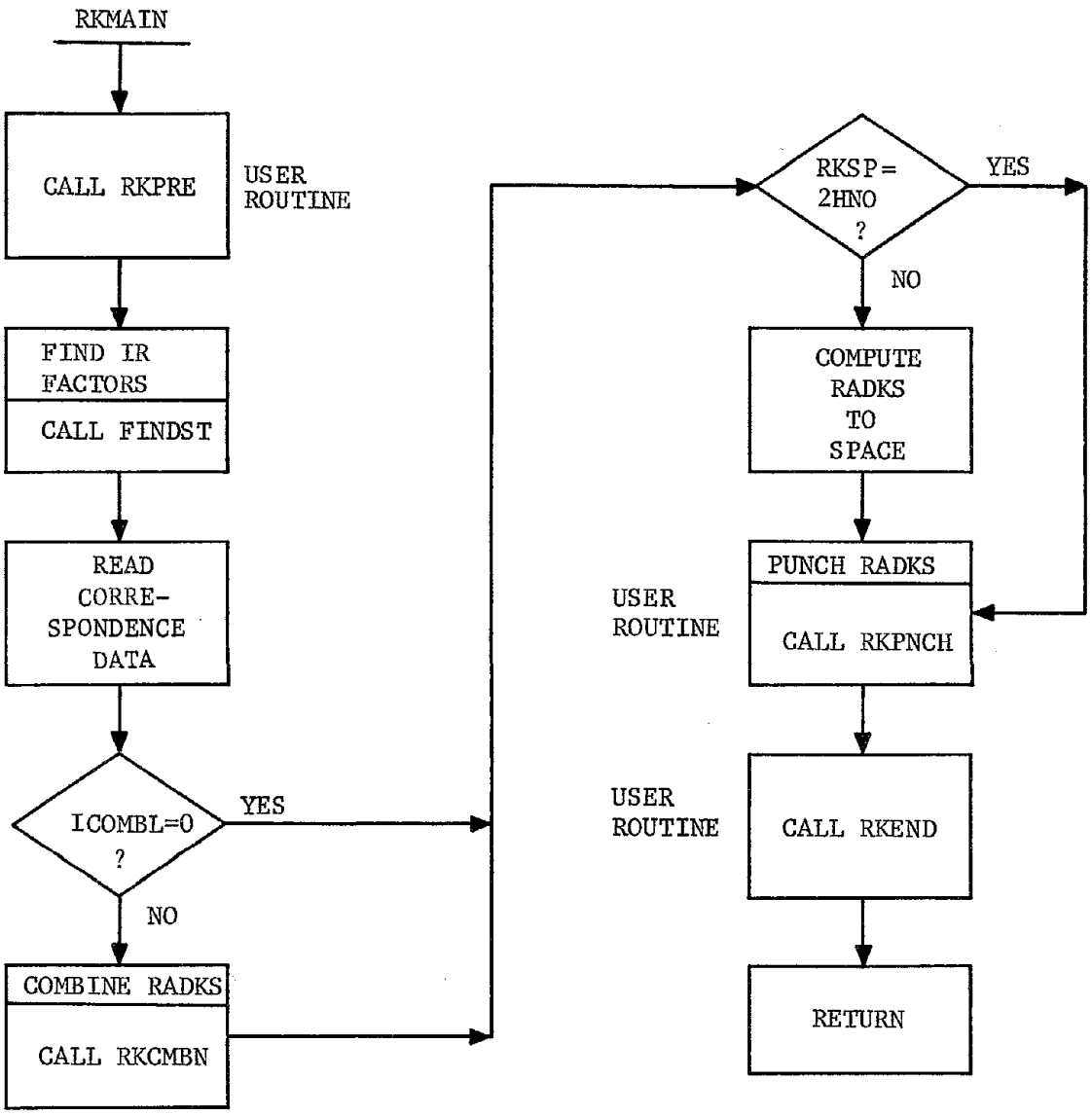


Figure 5-4 Segment RKCAL Flow Diagram

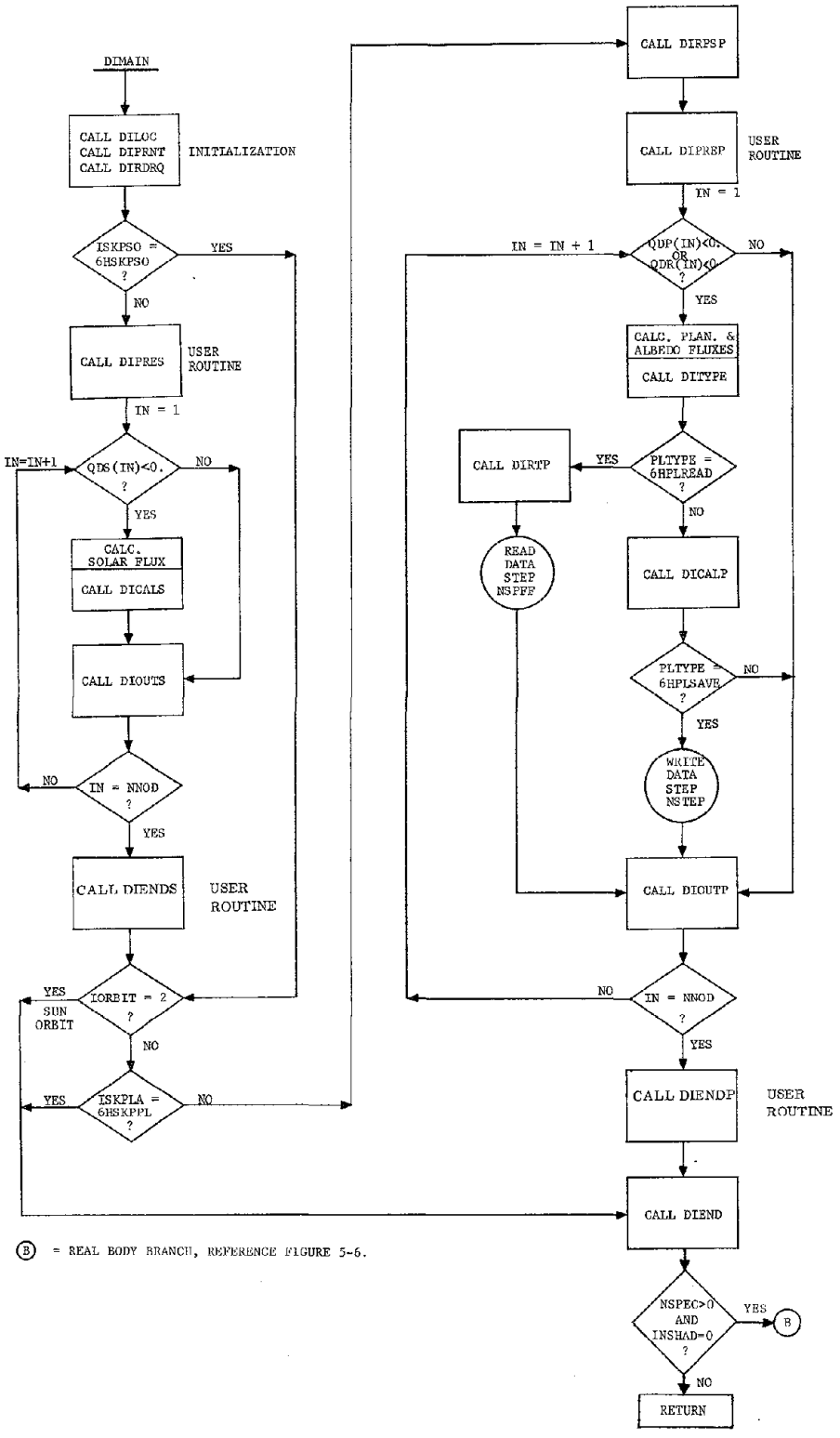


Figure 5-5 Segment DICAL Flow Diagram

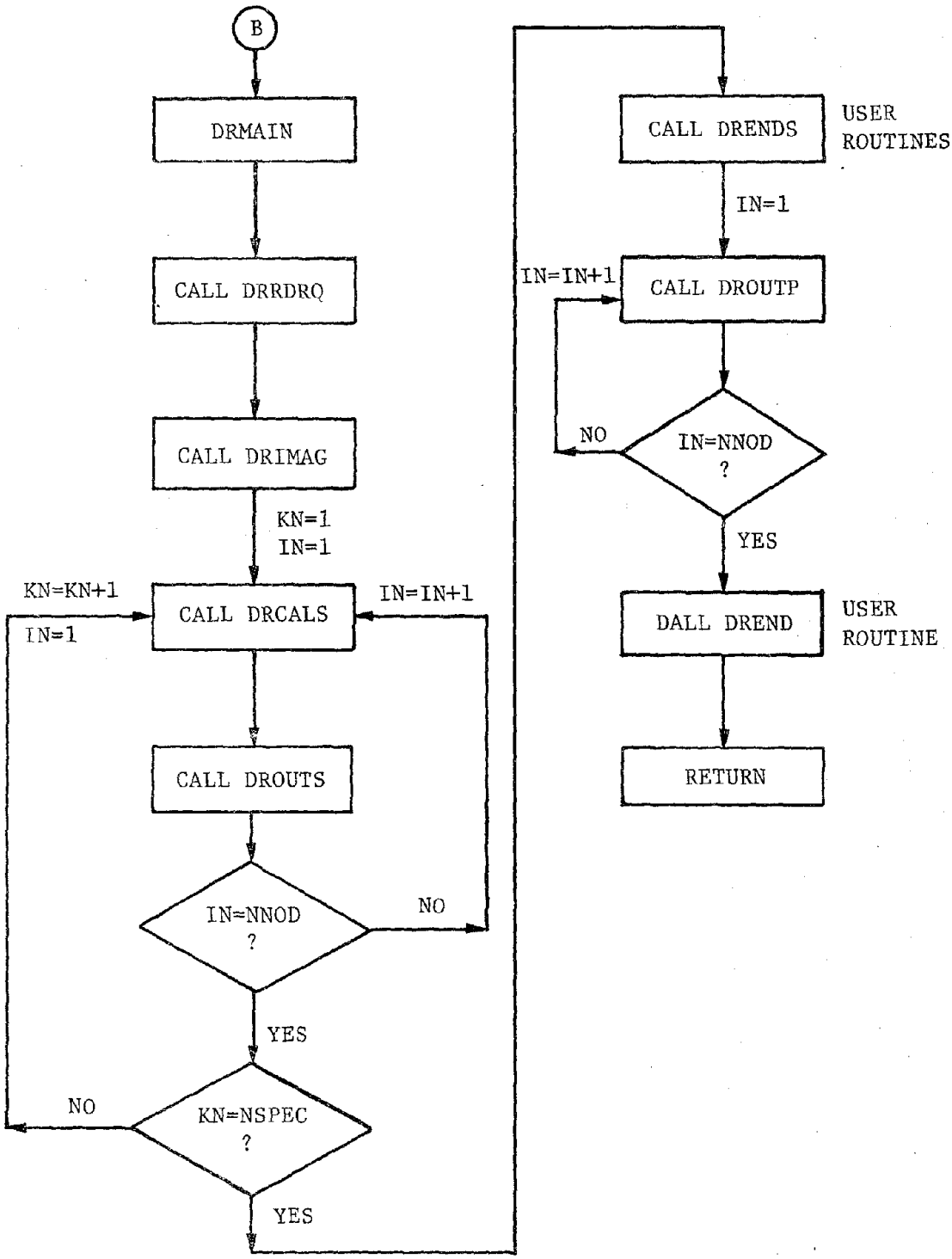


Figure 5-6 Segment DRCAL Flow Diagram

DICAL output is placed in out-of-core storage for later use in absorbed flux calculations. In addition, direct irradiation data may be punched at the user's option. This data is in the flux data block format. A printout of the direct irradiation data is provided also. An example of DICAL output can be found in Appendix H.

5.6 Absorbed Heat Segment

Calling sequence: L AQCAL

This segment utilizes direct irradiation and radiant interchange data in out-of-core storage as input. From this, it computes absorbed heat values for each external spacecraft node. Internode reflections are accounted for in both the solar and infrared wavebands.

No user intervention through the subroutine data block is provided.

Segment AQCAL output is placed in out-of-core storage. Printed output is also provided. An example of AQCAL output can be found in Appendix H.

5.7 Absorbed Heat Output Segment

Calling sequence: L QOCAL

This segment utilizes absorbed heat data in out-of-core storage to provide heat source tables in thermal analyzer format. At the user's option, heat versus time tables or orbital average heat data are provided for each external node.

Output is provided on punched cards or BCD tape. Q versus time data is in thermal analyzer array data format, with a singlet time array and a corresponding singlet Q array for each node. Also punched are thermal analyzer interpolation subroutine cards for each node. Orbital average data is punched in source data block format.

Standard output is in SINDA thermal analyzer format. Subroutine DA1lMDA is used for the interpolation subroutine. Output for other thermal analyzers may be obtained by altering the format statements in subroutine QOSAVE and entering the altered version in the subroutines data block. Card image printout of QOCAL output is provided. An example can be found in Appendix H.

Segment QOCAL logic flow is shown in Figure 5-7.

5.8 Radiation Condenser Segment

Calling sequence: L RCCAL

Segment RCCAL computes radiation conductors, simplifies and condenses these conductors using the ERN and MESS techniques, and provides output in punched card and/or BCD tape form. A printout of the card/tape record images as well as the original (uncondensed) RADKS is also provided. Three program-called user routines are used to provide user intervention through his subroutines block. Subroutine RCPRE provides for any special initialization desired before computations begin. Subroutine RCPNCH performs the actual punch and tape write operations, and the user may obtain data in any thermal analyzer program format by altering format statements in this routine. User routine RCEND provides for user intervention prior to return to operations block control. Figure 5-8 shows segment RCCAL logic flow. RCCAL theory is presented in Appendix F.

5.8.1 Sample Problem Using ERN/MESS Technique

The optics housing of the High Altitude Observatory (HAO) solar telescope, which is mounted on the Skylab Apollo Telescope Mount, is shown in Figure 5-9. Both the original enclosure and the modularized enclosure are shown along with the ERNs and the MESS nodes. Figure 5-10 shows the nodal breakdown for the enclosure.

TRASYS input for subenclosure 1 (see Figure 5-9) is shown in Figure 5-11. The TRASYS-RCCAL segment output is shown in Appendix H.

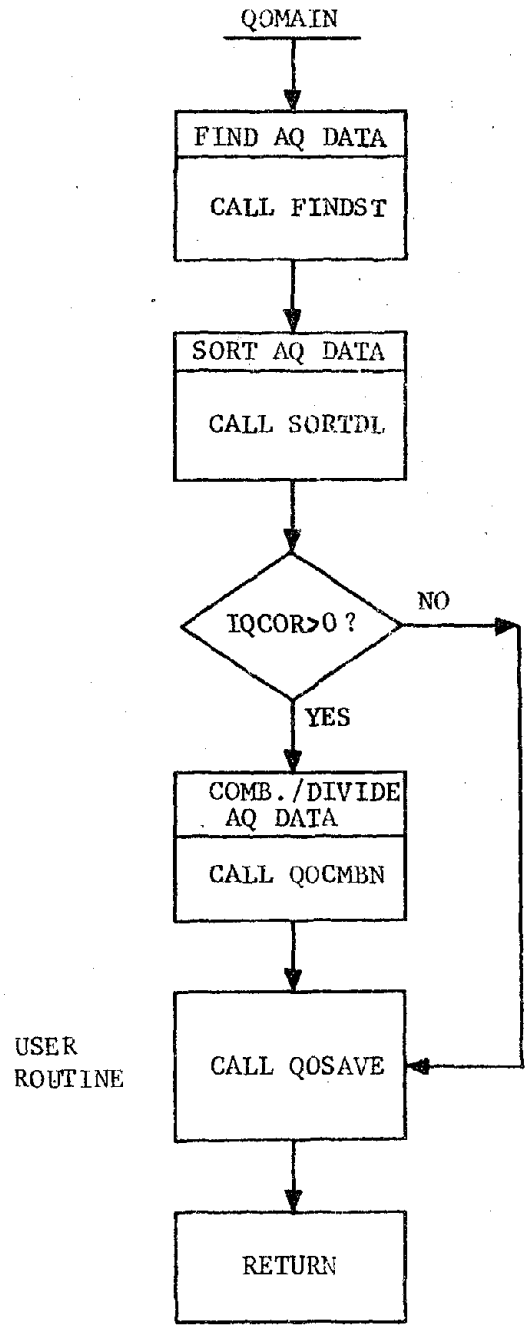


Figure 5-7 Segment QOCAL Flow Diagram

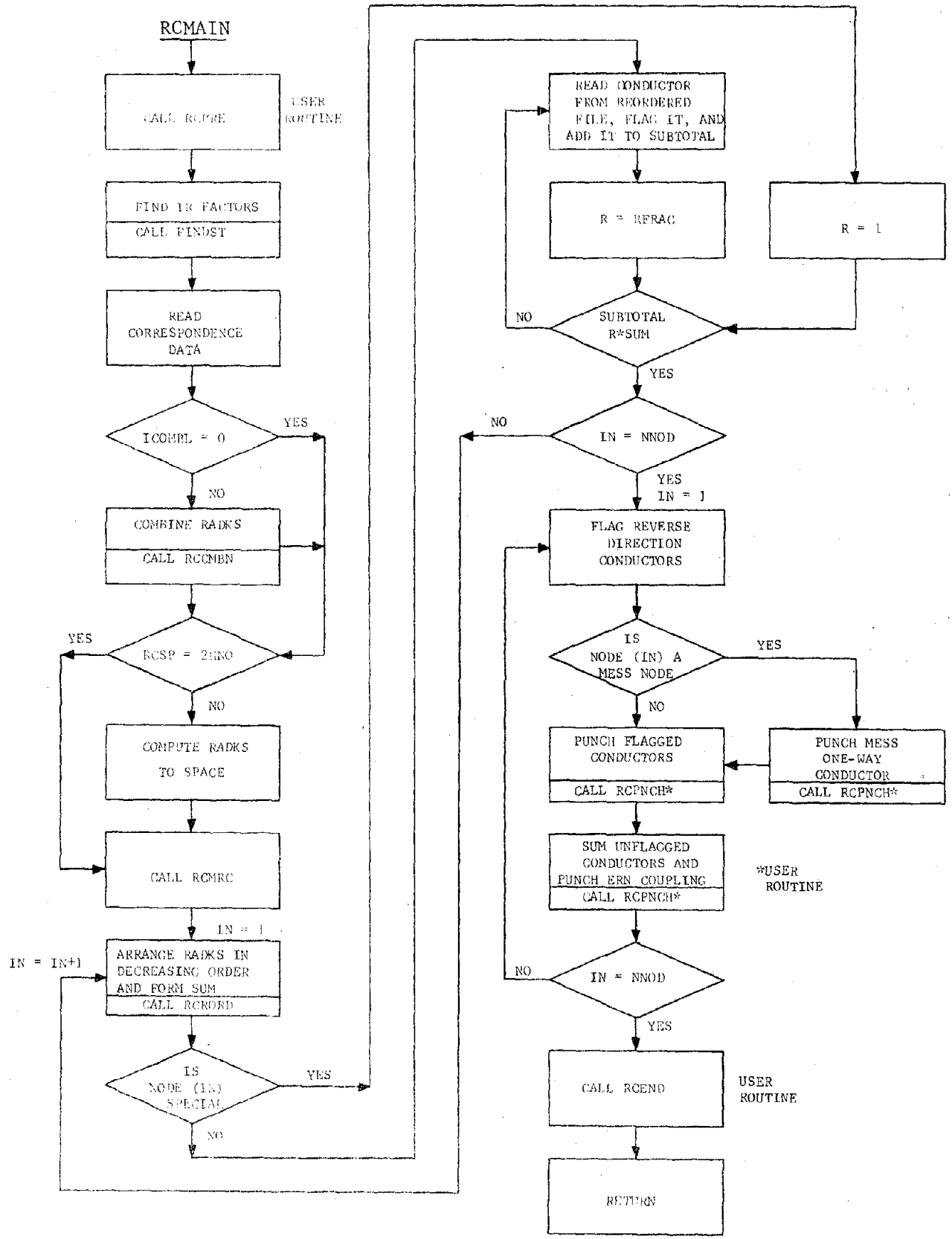


Figure 5-8 Segment RCCAL Flow Diagram

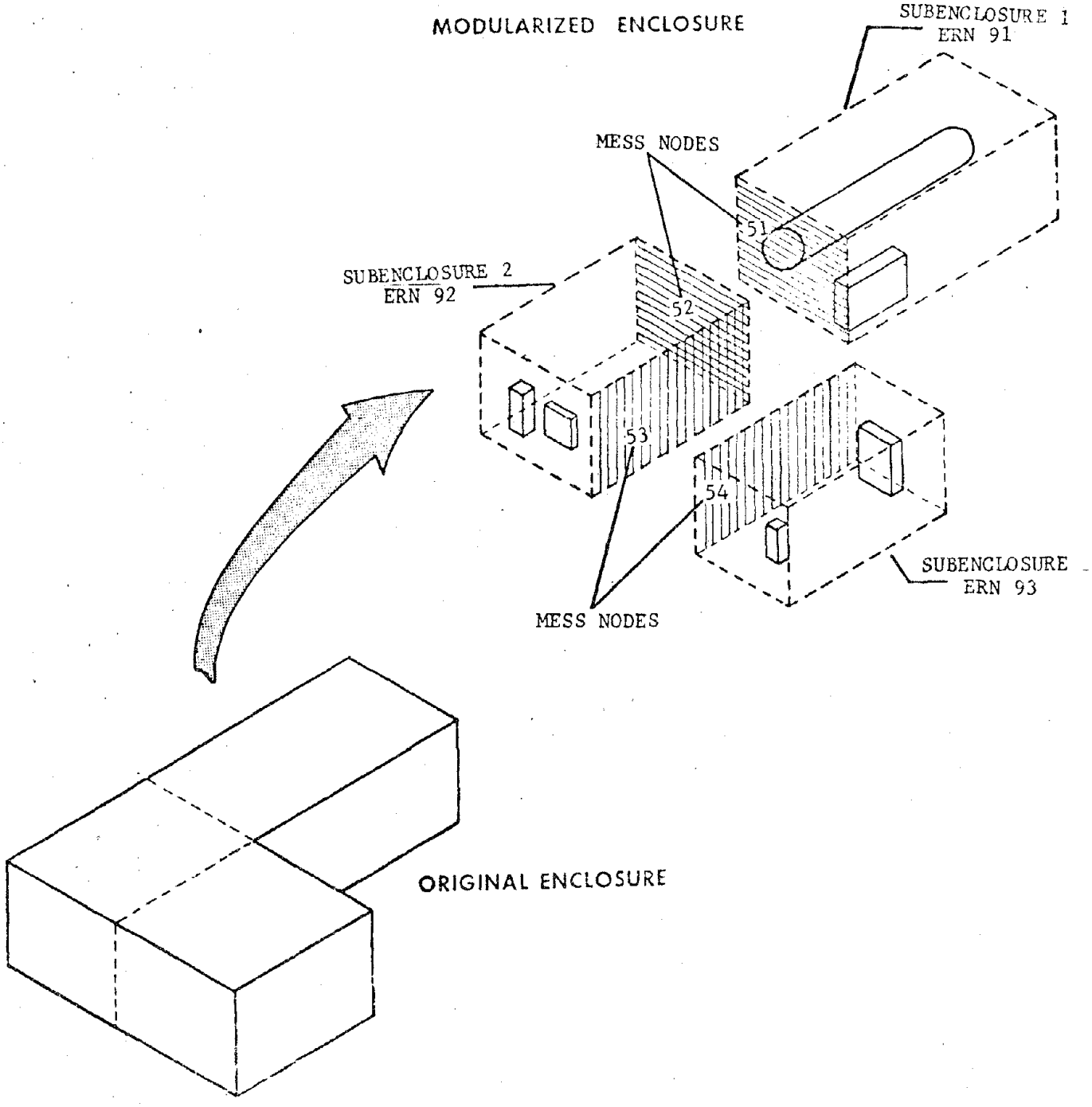


Figure 5-9 HAO Experiment Optics Housing Modularized Enclosures

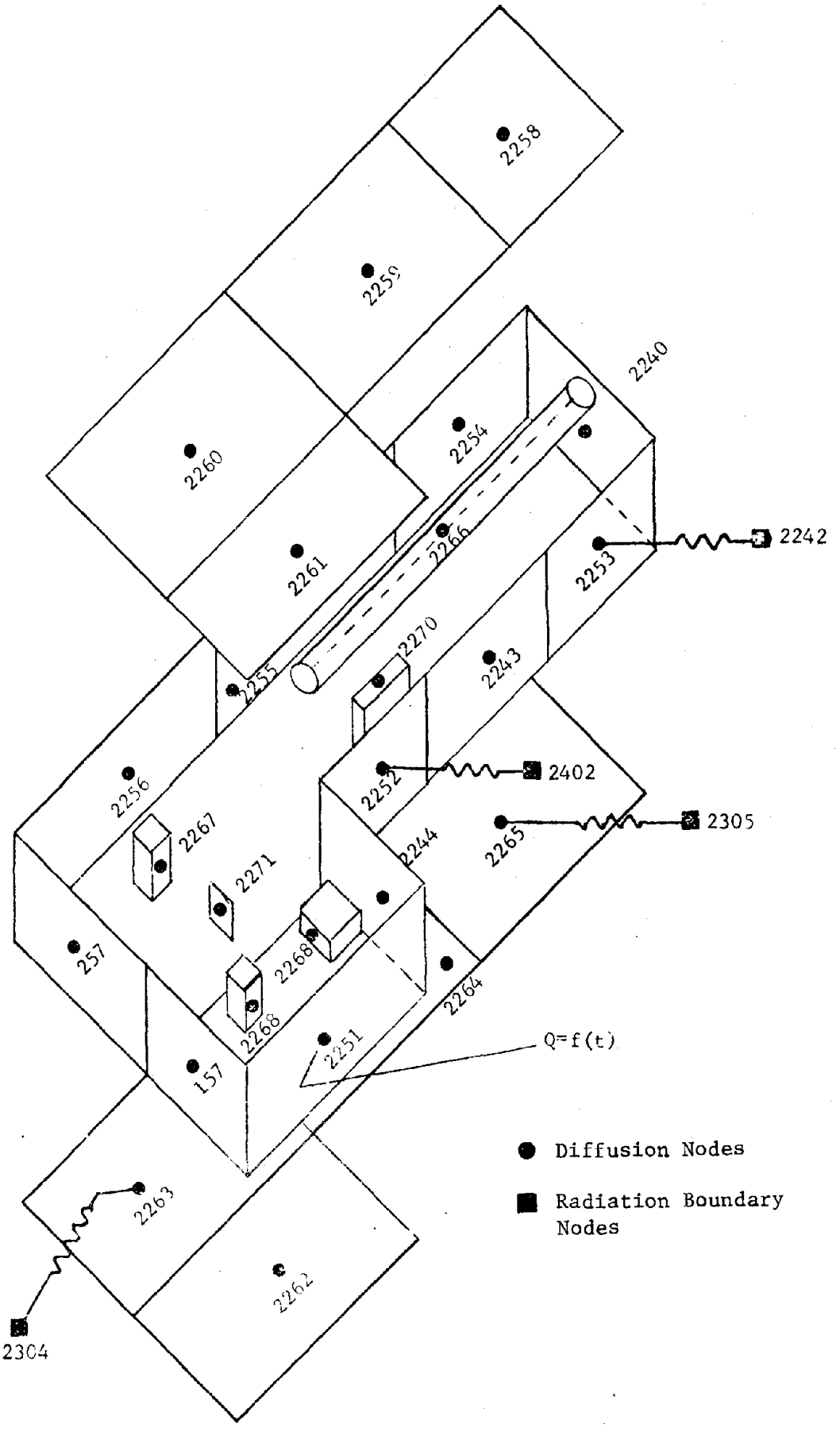


Figure 5-10 Apollo Telescope Mount HAO Experiment Optics Housing Sample Problem

```
HEADER OPTIONS DATA
TITLE  RADIATION CONDENSER SAMPLE PROBLEM
       MODEL          =HAO
HEADER ARRAY DATA
       IPPIME         =51
       ISECND         =52
HEADER SURFACE DATA
S      SUREN          =1
       TYPE           =RECT
       ACTIVE         =TOP
       SHADE          =FF
       BSHADE         =FF
       P1             =10.,9.,0.
       PROP           =0.1,0.1
S      SUREN          =2
       TYPE           =RECT
       ACTIVE         =TOP
       SHADE          =FF
       BSHADE         =FF
       P1             =10.,77.92879,0.
       NNY            =15
       UNNY           =1.3125,2.2875,3.2625,4.1725,11.351,18.5575,25.5045,
                     28.6545,31.6295,37.6495,45.5588,54.4125,63.2318,72.0955
       PROP           =0.9,0.9
HEADER FORM FACTOR DATA
STEPN  1
INITL   0.0
NODES  1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,END *
       1,  2,  0.037798 *  90. *
       1,  3,  0.005768 *  90. *
       1,  4,  0.006289 *  90. *
       1,  6,  0.104575 *  90. *
       1,  7,  0.118291 *  90. *
       1,  8,  0.060140 *  90. *
       1,  9,  0.043028 *  90. *
       1, 10,  0.032649 *  90. *
       1, 11,  0.027951 *  90. *
       1, 12,  0.119425 *  90. *
       1, 13,  0.161426 *  90. *
       1, 14,  0.105845 *  90. *
       1, 15,  0.119568 *  90. *
       1, 16,  0.053506 *  90. *
       2,  6,  0.160944 * 13.125 *
       2,  7,  0.040225 * 13.125 *
       2, 11,  0.043852 * 13.125 *
       2, 12,  0.216057 * 13.125 *
       2, 13,  0.020062 * 13.125 *
       2, 14,  0.115054 * 13.125 *
       2, 15,  0.025691 * 13.125 *
       2, 16,  0.007440 * 13.125 *
       3,  6,  0.021628 *  9.75 *
       3,  8,  0.280861 *  9.75 *
       3, 11,  0.300885 *  9.75 *
       3, 12,  0.174021 *  9.75 *
       3, 14,  0.081130 *  9.75 *
       4,  6,  0.004979 *  9.75 *
       4,  7,  0.049930 *  9.75 *
       4,  8,  0.117956 *  9.75 *
       4,  9,  0.050425 *  9.75 *
       4, 10,  0.010519 *  9.75 *
       4, 12,  0.183472 *  9.75 *
```


Figure 5-11 RCCAL Sample Problem Input (cont)

```
 4,  13,   0.133590 *   9.75 $
 4,  14,   0.024245 *   9.75 $
 4,  15,   0.134895 *   9.75 $
 4,  16,   0.085010 *   9.75 $
 5,   6,   0.028614 *   9.10 $
 5,   7,   0.008978 *   9.10 $
 5,   8,   0.239230 *   9.10 $
 5,   9,   0.003409 *   9.10 $
 5,  10,   0.000480 *   9.10 $
 5,  11,   0.029433 *   9.10 $
 5,  14,   0.548228 *   9.10 $
 5,  15,   0.063264 *   9.10 $
 5,  16,   0.004413 *   9.10 $
 6,   8,   0.111438 *  71.785 $
 6,   9,   0.024525 *  71.785 $
 6,  10,   0.009124 *  71.785 $
 6,  11,   0.166134 *  71.785 $
 6,  12,   0.254095 *  71.785 $
 6,  13,   0.029230 *  71.785 $
 6,  14,   0.280099 *  71.785 $
 6,  15,   0.034727 *  71.785 $
 6,  16,   0.013945 *  71.785 $
 7,   8,   0.044039 *  72.065 $
 7,   9,   0.063644 *  72.065 $
 7,  10,   0.054888 *  72.065 $
 7,  11,   0.013817 *  72.065 $
 7,  12,   0.026117 *  72.065 $
 7,  13,   0.263905 *  72.065 $
 7,  14,   0.034592 *  72.065 $
 7,  15,   0.280429 *  72.065 $
 7,  16,   0.160389 *  72.065 $
 8,  11,   0.154383 *  69.470 $
 8,  12,   0.176175 *  69.470 $
 8,  13,   0.051308 *  69.470 $
 8,  14,   0.223080 *  69.470 $
 8,  15,   0.054380 *  69.470 $
 8,  16,   0.014259 *  69.470 $
 9,  11,   0.009565 *  31.500 $
 9,  12,   0.029146 *  31.500 $
 9,  13,   0.271163 *  31.500 $
 9,  14,   0.038298 *  31.500 $
 9,  15,   0.275746 *  31.500 $
 9,  16,   0.085091 *  31.500 $
10,  11,   0.006027 *  29.750 $
10,  12,   0.008478 *  29.750 $
10,  13,   0.247228 *  29.750 $
10,  14,   0.010768 *  29.750 $
10,  15,   0.251456 *  29.750 $
10,  16,   0.291733 *  29.750 $
11,  12,   0.220815 *  60.200 $
11,  13,   0.010974 *  60.200 $
11,  14,   0.257296 *  60.200 $
11,  15,   0.017966 *  60.200 $
11,  16,   0.028121 *  60.200 $
12,  14,   0.154333 *  79.093 $
12,  15,   0.044610 *  79.093 $
12,  16,   0.009757 *  79.093 $
13,  14,   0.052702 *  88.537 $
13,  15,   0.198592 *  88.537 $
13,  16,   0.158787 *  88.537 $
14,  16,   0.013765 *  88.193 $
15,  16,   0.170217 *  88.537 $
```

Figure 5-10 RCCAL Sample Problem Input (concluded)

```
HEADER CORRESPONDENCE DATA
STEPN  1
KOMB
        2265          =1
        2270          =2,3,4,5
        2255          =6
        2254          =7
        2252          =8
        2243          =9
        2253          =10
        51            =11
        2264          =12
        2265          =13
        2259          =14
        2258          =15
        2240          =16
HEADER OPERATIONS DATA
STEP    1
        CALL BUILDC(ALLBLK)
        CALL FFDATA(0,0,0,0,0,0,3HRUN)
L       FFCAL
        CALL GBDATA(1,2HTR)
L       GBCAL
        CALL RCDATA(1,3HRUN,0,1000,0,0,1.713E-9,1./144.,2HNO,0,91,
     1  IPRIME,ISEDNO)
L       RCCAL
END OF PROBLEM
```

Figure 5-11 RCCAL Sample Problem Input

APPENDIX A

RESERVED WORD and SEGMENT COMMON LISTS

RESERVED WORD LIST
(COMMON FOR OPERATIONS DATA BLOCK)

```
      COMMON /CCONST/     DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGRSFF  , GRWRND  , RRAMPF  , IRKCN   , RKMIN   , IRKNGR
     .          , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQQARY  , IQQCOR  , QQAMPF  , QQFMPF  , QQTMPF  , QQRMPF
     .          , QQTYPE  , IQQTAR  , QQTAPE  , QQPNCH  , IQQTME  , NBCDSK
     .          , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSET    , NSFQ    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /TAPE  /     NDI       , NDIR      , NFF       , NFFR
     1                  , NGBIR     , NGBIRR    , NGBSO     , NGBSOR
     2                  , NDUT      , NRAN      , NSCR1     , NSCR2
     3                  , NSCR3     , NTQ       , NTQR      , NPLS
     4                  , NPLSR     , NSQNTL    , NPUN      , NBCDDU
     5                  , NSHADD    , NTRAJ     , NRSO      , NRTO
     6                  , NUSER1    , NUSER2    , KBCDDU    , KSHADD
     7                  , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE /     TITLE (11)            , NTITLE(11)
     1                  , MODELN    , DTE       , TMF       , IPAGE
     2                  , LINE      , IMSTEP
      COMMON /PLOT  /     NPNNP(5)              , NPTIT(9)
     1                  , NPVU(6)               , ZNPROT(6,6)
     2                  , ZNPSCL(6)             , IOPNNP(6)
     3                  , IOPTIT(6)             , IOPNVU(6)
     4                  , OPROT(6,6)            , OPSCL(6)
     5                  , OPSCLP(6)             , IOPNV
     6                  , OPRPLN(6)             , OPTRUE(6)
     7                  , OPTIMP(6)             , OPTIMS(6)
      COMMON /INDX  /     INDX  ( 1019)
      COMMON /INDXS /     INDXS (    1)
      COMMON /INDXN /     INDXN (    1)
      COMMON /DIMS  /     DIMS(3,    1)
      COMMON /OSTR  /     OSTR  (5,     1)
      COMMON /IFS   /     IFS   (    1)
      COMMON /IKS   /     IKS   (    1)
      COMMON /PR    /     PR    (2,     1)
      COMMON /PSH   /     PSH   (4,     1)
      COMMON /TSTR  /     TSTR  (3,3,     1)
      COMMON /ALPH  /     ALPH  (    1)
      COMMON /AREA  /     AREA  (    1)
      COMMON /EMISS /     EMISS (    1)
      COMMON /SRIR  /     SRIR  (    1)
      COMMON /SRSO  /     SRSO  (    1)
      COMMON /TRIR  /     TRIR  (    1)
      COMMON /TRSO  /     TRSO  (    1)
      COMMON /NODE  /     NODE  (    1)
      COMMON /ODTEMP/     ODTEMP(    1)
```

RESERVED WORD LIST (CONT.)

```
      COMMON /ORBIT /     ALAN      , ASUN      , PSD       , OMP
     1                  , ECC       , IORNT     , IORBIT    , PERIOD
     2                  , WSS       , PALB      , PRAD      , RSUN
     3                  , CIGMAS    , BETAS     , APER      , WDS
     4                  , WSUN      , TIMEST    , TIMEPR    , TRUEAN
     5                  , SOL       , ISKPSO    , GRAV      , OINC
     6                  , HA        , HP        , SUNRA     , STRRA
     7                  , STRDEC    , SUNDEC    , CIGMA     , BETA
     8                  , RTHET     , ORNT(3,3)             , SPINT(3,3)
     9                  , ICALFL    , NSPFF     , CLOCK     , CONE
     0                  , RATE      , ROTX      , ROTY      , ROTZ
     1                  , IROTX     , IROTY     , IROTZ     , PNAME
     2                  , ISET      , PLTYPE    , INSHAD    , SHADIN
     3                  , SHAOUT    , SUNCL     , SUNCO     , PLCL
     4                  , PLCO      , TIMSP
      COMMON /DSTORE/      IDSTR (12,3)
      COMMON /ISTPDR/      ISTPDR(     1)
      COMMON /NSPEC /      NSPEC
      COMMON /PLOTTR/      IPLUNT    , PLCRVF    , PLXMPF    , PLYMPF
     1                  , IPLNA     , IPLSN     , PLLABX(5)
     2                  , PLLABY(5)             , PLTIT1(10)
     3                  , PLTIT2(12)
      COMMON /DIRCT /      DIRCT(    10)
```

COMMON BLOCKS FOR AQCAL

```
      COMMON /CONST/     DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGBSFF  , GRWBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGB
     .          , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQOARY  , IQOCOR  , QOAMPF  , QOFMPF  , QOTMPF  , QORMPF
     .          , QOTYPE  , IQOTAB  , QOTAPE  , QOPNCH  , IQOTME  , NBCOSK
     .          , RSOLAB  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /TAPE   /     NDI       , NDIR      , NFF       , NFFR
     1                   , NGBIR     , NGBIRP    , NGBSO     , NGBSOR
     2                   , NOUT      , NRAN      , NSCR1     , NSCR2
     3                   , NSCR3     , NTQ       , NTQR      , NPLS
     4                   , NPLSR     , NSQNTL    , NPUN      , NBCDOU
     5                   , NSHADO    , NTRAJ     , NRSO      , NRTO
     6                   , NUSER1    , NUSER2    , KBCDOU    , KSHADO
     7                   , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE  /     TITLE (11)            , NTITLE(11)
     1                   , MODELN    , DTE       , TME       , IPAGE
     2                   , LINE      , IHSTEP
      COMMON /PLOT   /     NPNNP(6)              , NPTIT(9)
     1                   , NPVU(6)               , ZNPROT(6,6)
     2                   , ZNPSCL(6)             , IOPNNP(6)
     3                   , IOPTIT(6)             , IOPNVU(6)
     4                   , OPROT(6,6)            , OPSCL(6)
     5                   , OPSCLR(6)             , IOPNV
     6                   , OPRPLN(5)             , OPTRUE(6)
     7                   , OPTIMP(6)             , OPTIMS(6)
      COMMON /INDX   /     INDX   ( 1019)
      COMMON /INDXS  /     INDXS  (    1)
      COMMON /INDXN  /     INDXN  (    1)
      COMMON /DIMS   /     DIMS(3,     1)
      COMMON /OSTR   /     OSTR   (5,     1)
      COMMON /IFS    /     IFS    (    1)
      COMMON /IKS    /     IKS    (    1)
      COMMON /PR     /     PR     (2,    1)
      COMMON /PSH    /     PSH    (4,    1)
      COMMON /TSTR   /     TSTR   (3,3,     1)
      COMMON /ALPH   /     ALPH   (    1)
      COMMON /AREA   /     AREA   (    1)
      COMMON /FMISS  /     FMISS  (    1)
      COMMON /SRIR   /     SRIR   (    1)
```

COMMON BLOCKS FOR AQCAL (CONT.)

```
      COMMON /SRSO  /     SRSO  (     1)
      COMMON /IRIR  /     IRIR  (     1)
      COMMON /TRSO  /     TRSO  (     1)
      COMMON /NODE  /     NODE  (     1)
      COMMON /ODTEMP/     ODTEMP(     1)
      COMMON /ORBIT /     ALAN      , ASUN      , PSD       , DWP
     1                  , ECC       , IORNT     , IORBIT    , PERIOD
     2                  , WSS       , PALB      , PRAD      , RSUN
     3                  , CIGMAS    , BETAS     , APER      , WDS
     4                  , WSUN      , TIMEST    , TIMEPR    , TRUEAN
     5                  , SOL       , ISKPSO    , GRAV      , OINC
     6                  , HA        , HP        , SUNRA     , STRRA
     7                  , STRDEC    , SUNDEC    , CIGMA     , BETA
     8                  , RTHET     , ORNT(3,3)             , SPINT(3,3)
     9                  , ICALFL    , NSPFF     , CLOCK     , CONE
     0                  , RATE      , ROTX      , ROTY      , ROTZ
     1                  , IROTX     , IROTY     , IROTZ     , PNAME
     2                  , ISFT      , PLTYPE    , INSHAD    , SHADIN
     3                  , SHAOUT    , SUNCL     , SUNCO     , PLCL
     4                  , PLCO      , TIMSP
      COMMON /DSTORE/     IDSTR (12,3)
      COMMON /ISTPDR/     ISTPDR(     1)
      COMMON /NSPEC /     NSPEC
      COMMON /PLOTTR/     IPLUNT    , PLCRVF    , PLXMPF    , PLYMPF
     1                  , IPLNA     , IPLSN     , PLLABX(5)
     2                  , PLLABY(5)             , PLTIT1(10)
     3                  , PLTIT2(12)
      COMMON /AQQDS /     QDS   (     1)
      COMMON /AQQDR /     QDR   (     1)
      COMMON /AQQDP /     QDP   (     1)
      COMMON /QAS   /     QAS   (     1)
      COMMON /QAR   /     QAR   (     1)
      COMMON /QAP   /     QAP   (     1)
      COMMON /GBSO  /     GBSO  (     1)
      COMMON /GBIR  /     GBIR  (     1)
      COMMON /AQTEMP/     AQTEMP(     1)
```

COMMON BLOCKS FOR DICAL

```
      COMMON /CCONST/    DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGRSFF  , GRWBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGB
     .          , IRKNSP  , RKPNCH  , RKSF    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQOARY  , IQOCOR  , QOAMPF  , QOFMPF  , QOTMPF  , QORMPF
     .          , QOTYPE  , IQOTAB  , QOTAPE  , QOPNCH  , IQOTME  , NBCDSK
     .          , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /TAPE   /     NDI       , NDIR      , NFF       , NFFR
     1                   , NGBIR     , NGBIRR    , NGBSO     , NGBSOR
     2                   , NOUT      , NRAN      , NSCR1     , NSCR2
     3                   , NSCR3     , NTQ       , NTQR      , NPLS
     4                   , NPLSR     , NSQNTL    , NPUN      , NBCDOU
     5                   , NSHADO    , NTRAJ     , NRSO      , NRTO
     6                   , NUSER1    , NUSER2    , KBCDOU    , KSHADO
     7                   , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE  /     TITLE (11)            , NTITLE(11)
     1                   , MODELN    , DTE       , TME       , IPAGE
     2                   , LINE      , IHSTEP
      COMMON /PLOT   /     NPNNP(6)              , NPTIT(9)
     1                   , NPVU(6)               , ZNPROT(6,6)
     2                   , ZNPSCL(6)             , IOPNNP(6)
     3                   , IOPTIT(6)             , IOPNVU(6)
     4                   , OPROT(6,6)            , OPSCL(6)
     5                   , OPSCLR(6)             , IOPNV
     6                   , OPRPLN(6)             , OPTRUE(6)
     7                   , OPTIMP(6)             , OPTIMS(6)
      COMMON /INDX   /     INDX   ( 1019)
      COMMON /INDXS  /     INDXS  (    1)
      COMMON /INDXN  /     INDXN  (    1)
      COMMON /DIMS   /     DIMS(3,     1)
      COMMON /DSTR   /     DSTR   (5,     1)
      COMMON /IFS    /     IFS    (    1)
      COMMON /IKS    /     IKS    (    1)
      COMMON /PR     /     PR     (2,     1)
      COMMON /PSH    /     PSH    (4,     1)
      COMMON /TSTR   /     TSTR   (3,3,     1)
      COMMON /ALPH   /     ALPH   (    1)
      COMMON /AREA   /     AREA   (    1)
      COMMON /EMISS  /     EMISS  (    1)
      COMMON /SRIR   /     SRIR   (    1)
      COMMON /SRSO   /     SRSO   (    1)
      COMMON /TRIR   /     TRIR   (    1)
      COMMON /TRSO   /     TRSO   (    1)
```

COMMON BLOCKS FOR DICAL (CONT.)

```
      COMMON /NODE   /     NODE   (     1)
      COMMON /ODTEMP/     ODTEMP(     1)
      COMMON /ORBIT  /     ALAN     , ASUN     , PSD      , DWP
     1                   , ECC      , IORNT    , IORBIT   , PERIOD
     2                   , WSS      , PALB     , PRAD     , RSUN
     3                   , CIGMAS   , BETAS    , APER     , WDS
     4                   , WSUN     , TIMEST   , TIMEPR   , TRUEAN
     5                   , SOL      , ISKPSO   , GRAV     , OINC
     6                   , HA       , HP       , SUNRA    , STRRA
     7                   , STRDEC   , SUNDEC   , CIGMA    , BETA
     8                   , RTHET    , ORNT(3,3)           , SPINT(3,3)
     9                   , ICALFL   , NSPFF    , CLOCK    , CONE
     0                   , RATE     , ROTX     , ROTY     , ROTZ
     1                   , IROTX    , IROTY    , IROTZ    , PNAME
     2                   , ISFT     , PLTYPE   , INSHAD   , SHADIN
     3                   , SHADOT   , SUNCL    , SUNCO    , PLCL
     4                   , PLCO     , TIMSP
      COMMON /DSTORE/     IDSTR (12,3)
      COMMON /ISTPDP/     ISTPDR(     1)
      COMMON /NSPEC  /    NSPEC
      COMMON /PLOTTR/     IPLUNT   , PLCRVF   , PLXMPF   , PLYMPF
     1                   , IPLNA    , IPLSN    , PLLABX(5)
     2                   , PLLABY(5)           , PLTIT1(10)
     3                   , PLTIT2(12)
      COMMON /ISHAD  /    ISHAD(     1)
      COMMON /QDP    /    QDP(     1)
      COMMON /QDR    /    QDR(     1)
      COMMON /QDS    /    QDS(     1)
```

COMMON BLOCKS FOR FFCAL

```
      COMMON /CCONST/     DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGBSFF  , GBWBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGB
     .          , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQOARY  , IQOCOR  , QOAMPF  , QOFMPF  , QOTMPF  , QORMPF
     .          , QOTYPE  , IQOTAB  , QOTAPE  , QOPNCH  , IQOTME  , NBCDSK
     .          , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /TAPE  /     NDI       , NDIR    , NFF     , NFFR
     1                  , NGBIR     , NGBIRR  , NGBSO   , NGBSOR
     2                  , NOUT      , NRAN    , NSCR1   , NSCR2
     3                  , NSCR3     , NTQ     , NTQR    , NPLS
     4                  , NPLSR     , NSQNTL  , NPUN    , NBCDOU
     5                  , NSHADO    , NTRAJ   , NRSO    , NRTO
     6                  , NUSER1    , NUSER2  , KBCDOU  , KSHADO
     7                  , KTRAJ     , KRSO    , KRTO
      COMMON /TITLE /     TITLE (11)          , NTITLE(11)
     1                  , MODELN    , DTE     , TME     , IPAGE
     2                  , LINE      , IHSTEP
      COMMON /PLOT  /     NPNNP(6)            , NPTIT(9)
     1                  , NPVU(6)           , ZNPROT(6,6)
     2                  , ZNPSCL(6)         , IOPNNP(6)
     3                  , IOPTIT(6)         , IOPNVU(6)
     4                  , OPROT(6,6)        , OPSCL(6)
     5                  , OPSCLR(6)         , IOPNV
     6                  , OPRPLN(6)         , OPTRUE(6)
     7                  , OPTIMP(6)         , OPTIMS(6)
      COMMON /INDX  /     INDX   ( 1019)
      COMMON /INDXS /     INDXS  (    1)
      COMMON /INDXN /     INDXN  (    1)
      COMMON /DIMS  /     DIMS(3,     1)
      COMMON /DSTR  /     DSTR   (5,     1)
      COMMON /IFS   /     IFS    (    1)
      COMMON /IKS   /     IKS    (    1)
      COMMON /PR    /     PR     (2,    1)
      COMMON /PSH   /     PSH    (4,    1)
      COMMON /TSTR  /     TSTR   (3,3,     1)
      COMMON /ALPH  /     ALPH   (    1)
      COMMON /AREA  /     AREA   (    1)
      COMMON /EMISS /     EMISS  (    1)
      COMMON /SRIR  /     SRIR   (    1)
      COMMON /SRSO  /     SRSO   (    1)
      COMMON /TRIR  /     TRIR   (    1)
      COMMON /TRSO  /     TRSO   (    1)
      COMMON /NODE  /     NODE   (    1)
      COMMON /QQTEMP/     QQTEMP (    1)
```

COMMON BLOCKS FOR FFCAL (CONT.)

```
      COMMON /ORBIT /     ALAN     , ASUN     , PSD      , DWP
     1                  , ECC      , IORNT    , IORBIT   , PERIOD
     2                  , WSS      , PALB     , PRAD     , RSUN
     3                  , CIGMAS   , BETAS    , APER     , WDS
     4                  , WSUN     , TIMEST   , TIMEPP   , TRUEAN
     5                  , SOL      , ISKPSO   , GRAV     , OINC
     6                  , HA       , HP       , SUNRA    , STRRA
     7                  , STRDEC   , SUNDEC   , CIGMA    , BETA
     8                  , RTHET    , ORNT(3,3)           , SPINT(3,3)
     9                  , ICALFL   , NSPFF    , CLOCK    , CONE
     0                  , RATE     , ROTX     , ROTY     , ROTZ
     1                  , IROTX    , IROTY    , IROTZ    , PNAME
     2                  , ISFT     , PLTYPE   , INSHAD   , SHADIN
     3                  , SHAOUT   , SUNCL    , SUNCO    , PLCL
     4                  , PLCO     , TIMSP
      COMMON /DSTORE/      IDSTR (12,3)
      COMMON /ISTPDR/      ISTPDR(     1)
      COMMON /NSPEC /      NSPEC
      COMMON /PLOTTR/      IPLUNT   , PLCRVF   , PLXMPF   , PLYMPF
     1                  , IPLNA    , IPLSN    , PLLABX(5)
     2                  , PLLABY(5)           , PLTIT1(10)
     3                  , PLTIT2(12)
      COMMON /FFVALI/      FFVALI(     1)
      COMMON /FFVALS/      FFVALS(     1)
      COMMON /FFSHDC/      ISHAD(     1)
      COMMON /FFSUMC/      SUM(     1)
```

COMMON BLOCKS FOR GBCAL

```
      COMMON /CCONST/    DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGBSFF  , GBWBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGR
     .          , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQOARY  , IQOCOR  , QOAMPF  , QOFMPF  , QOTMPF  , QORMPF
     .          , QOTYPE  , IQOTAB  , QOTAPE  , QOPNCH  , IQOTME  , NBCDSK
     .          , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /IAPE   /     NDI       , NDIR      , NFF       , NFFR
     1                   , NGBIR     , NGBIRR    , NGBSO     , NGBSOR
     2                   , NOUT      , NRAN      , NSCR1     , NSCR2
     3                   , NSCR3     , NTQ       , NTQR      , NPLS
     4                   , NPLSR     , NSQNTL    , NPUN      , NBCDOU
     5                   , NSHADO    , NTRAJ     , NRSO      , NRTO
     6                   , NUSER1    , NUSER2    , KBCDOU    , KSHADO
     7                   , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE  /     TITLE (11)            , NTITLE(11)
     1                   , MODELN    , DTE       , TME       , IPAGE
     2                   , LINE      , IHSTEP
      COMMON /PLOT   /     NPNNP(6)              , NPTIT(9)
     1                   , NPVU(6)               , ZNPROT(6,6)
     2                   , ZNPSCL(6)             , IOPNNP(6)
     3                   , IOPTIT(6)             , IOPNVU(6)
     4                   , OPROT(6,6)            , OPSCL(6)
     5                   , OPSCLR(6)             , IOPNV
     6                   , OPRPLN(6)             , OPTRUE(6)
     7                   , OPTIMP(6)             , OPTIMS(6)
      COMMON /INDX   /     INDX   ( 1019)
      COMMON /INDXS  /     INDXS  (    1)
      COMMON /INDXN  /     INDXN  (    1)
      COMMON /DIMS   /     DIMS(3,     1)
      COMMON /DSTR   /     DSTR   (5,     1)
      COMMON /IFS    /     IFS    (    1)
      COMMON /IKS    /     IKS    (    1)
      COMMON /PR     /     PR     (2,     1)
      COMMON /PSH    /     PSH    (4,     1)
      COMMON /TSTR   /     TSTR   (3,3,     1)
      COMMON /ALPH   /     ALPH   (    1)
      COMMON /AREA   /     AREA   (    1)
      COMMON /EMISS  /     EMISS  (    1)
      COMMON /SRIR   /     SRIR   (    1)
      COMMON /SRSO   /     SRSO   (    1)
      COMMON /TRIR   /     TRIR   (    1)
      COMMON /TRSO   /     TRSO   (    1)
      COMMON /NODE   /     NODE   (    1)
      COMMON /ODTEMP/      ODTEMP(    1)
```

COMMON BLOCKS FOR GBCAL (CONT.)

```
      COMMON /ORBIT /     ALAN      , ASUN      , PSI       , DWP
     1                  , ECC       , IORNT     , IORBIT    , PERIOD
     2                  , RSS       , RALP      , RRAD      , RSUN
     3                  , CIGMAS    , BETAS     , APER      , WDS
     4                  , WSUN      , TIMEST    , TIMEPR    , TRUEAN
     5                  , SOL       , ISKPSO    , GRAV      , OINC
     6                  , RA        , RP        , SUNRA     , STRRA
     7                  , STRDEC    , SUNDEC    , CIGMA     , BETA
     8                  , RTHET     , ORNT(3,3)             , SPINT(3,3)
     9                  , ICALFL    , NSPEF     , CLOCK     , CONE
     0                  , RATE      , ROTX      , ROTY      , ROTZ
     1                  , IROTA     , IROTY     , IROTZ     , PNAME
     2                  , ISFI      , PLTYPE    , INSHAD    , SHADIN
     3                  , SHAOUT    , SUNCL     , SUNCO     , PLCL
     4                  , PLCO      , TIMSP
      COMMON /DSTORE/     DSTR (12,3)
      COMMON /ISTPOR/     ISTPOR(     1)
      COMMON /NSPEC /     NSPEC
      COMMON /PLOTTR/     IPLUNT    , PLCRVF    , PLXMPF    , PLYMPF
     1                  , IPLNA     , IPLSN     , PLLABX(5)
     2                  , PLLABY(5)             , PLTIT1(10)
     3                  , PLTIT2(12)
      COMMON /RA    /     RA    (     1)
      COMMON /SPACE /     SPACE (     1)
      COMMON /XSPACE/     XSPACE(     1)
```

COMMON BLOCKS FOR NPLOT

```
      COMMON /CONST/    DTR     , RTD     , PI      , MAXBC   , NN
     .           , NS      , NNOD    , NSURF   , JOVL    , NSTEP   , NSSTEP
     .           , NBLKLN  , DTACC   , DTACCS  , DTNOSH  , DTPNCH  , FFACC
     .           , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .           , IGPSFF  , GAMMND  , PRAMPF  , IRKCN   , RKMIN   , IRKNGB
     .           , IRKNSP  , RKPNCH  , FRSP    , RKTAPE  , SIGMA   , ITRC10
     .           , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .           , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .           , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .           , IQDARY  , IQDCOR  , QDAMPF  , QDFMPF  , QDTMPF  , QDRMPF
     .           , QDTYPE  , IQDTAB  , QDTAPE  , QDPNCH  , IQDTME  , NBCDSK
     .           , RSOLAR  , RALB    , FPLAN   , RFRAC   , IMESS   , ISPND
     .           , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .           , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANT
     .           , NSTSOL  , NSTPL
      COMMON /TAPE   /      NDI       , NDIR      , NFF       , NFFR
     1                    , NGBIR     , NGBIRR    , NGBSO     , NGBSOR
     2                    , NOUT      , NRAM      , NSCR1     , NSCR2
     3                    , NSCR3     , NTQ       , NTQR      , NPLS
     4                    , NPLSR     , NSQNTL    , NPUN      , NBCDOU
     5                    , NSHADO    , NTRAJ     , NRSO      , NRTO
     6                    , NUSER1    , NUSER2    , KBCDOU    , KSHADO
     7                    , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE  /      TITLE (11)            , NTITLE(11)
     1                    , MODELN    , DTE       , TME       , IPAGE
     2                    , LINE      , IPSTEP
      COMMON /PLOT   /      NPNNP(6)              , NPTIT(9)
     1                    , NPVU(6)             , ZNPROT(6,6)
     2                    , ZNPSCL(6)           , IOPNNP(6)
     3                    , IOPTIT(6)           , IOPNVU(6)
     4                    , OPROT(6,6)          , OPSCL(6)
     5                    , OPSCLR(6)           , IOPNV
     6                    , OPRPLN(6)           , OPTRUE(6)
     7                    , OPTIMP(6)           , OPTIMS(6)
      COMMON /INDX   /      INDX   ( 1019)
      COMMON /INDXS  /      INDXS  (     1)
      COMMON /INDXN  /      INDXN  (     1)
      COMMON /DIMS   /      DIMS(3,      1)
      COMMON /DSTR   /      DSTR   (5,       1)
      COMMON /IFS    /      IFS    (     1)
      COMMON /IKS    /      IKS    (     1)
      COMMON /PR     /      PR     (2,      1)
      COMMON /PSH    /      PSH    (4,      1)
      COMMON /TSTR   /      TSTR   (3,3,       1)
      COMMON /ALPH   /      ALPH   (     1)
      COMMON /AREA   /      AREA   (     1)
      COMMON /EMISS  /      EMISS  (     1)
      COMMON /SRIR   /      SRIR   (     1)
      COMMON /SRSO   /      SRSO   (     1)
      COMMON /TRIR   /      TRIR   (     1)
      COMMON /TRSO   /      TRSO   (     1)
      COMMON /NODE   /      NODE   (     1)
      COMMON /QDTEMP/       QDTEMP(      1)
```

COMMON BLOCKS FOR NPLOT (CONT.)

```
      COMMON /ORBIT /     ALAN     , ASUN     , PSI      , DNP
     1                  , ECC      , IORNT    , IORBIT   , PERIOD
     2                  , WSS      , PALB     , PRAD     , RSUN
     3                  , CIGMAS   , BETAS    , APER     , WDS
     4                  , WSUN     , TIMEST   , TIMEPR   , TRUEAN
     5                  , SOL      , ISKPSO   , GRAV     , OINC
     6                  , RA       , RP       , SUNRA    , STRRA
     7                  , STRDEC   , SUNDEC   , CIGMA    , BETA
     8                  , RTHET    , URNT(3,3)           , SPINT(3,3)
     9                  , ICALFL   , NSPEF    , CLOCK    , CONE
     0                  , RATE     , ROTX     , ROTY     , ROTZ
     1                  , IROTX    , IROTY    , IROTZ    , PNAME
     2                  , ISET     , PLTYPE   , INSHAD   , SHADIN
     3                  , SHADUT   , SUNCL    , SUNCO    , PLCL
     4                  , PLCO     , TIMSP
      COMMON /DSTOPF/     IDSTR (12,3)
      COMMON /ISTPOR/     ISTPOR(    1)
      COMMON /NSPEC /     NSPEC
      COMMON /PLOTTR/     IPLUNT   , PLCRVF   , PLXMPF   , PLYMPF
     1                  , IPLNA    , IPLSN    , PLLABX(5)
     2                  , PLLABY(5)           , PLTIT1(10)
     3                  , PLTIT2(10)
      COMMON /MNP   /     MNP(    1)
```

COMMON BLOCKS FOR OPLOT

```
      COMMON /CCONST/    DTK     , RTD     , PI      , MAXBC   , NN
     .          , NS     , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN , DIACC   , DIACCS , DINOSH , DIPNCH , FFACC
     .          , FFACCS , FFMIN   , FFRATL , FFNOSH , FFPNCH , FFPRNT
     .          , IGBSFF , GBWBND , RKAMPF , IRKCN   , RKMIN   , IRKNGB
     .          , IRKNSP , RKPNCH , RKSP   , RKTAPE , SIGMA   , ITRC10
     .          , ITRC20 , ITRC30 , ITRC40 , ITRC50 , ITRC60 , ITRC70
     .          , ITRC80 , ITRC90 , ITRCA0 , ITRCB0 , ITRCC0 , ITRCD0
     .          , ITRALL , IAQGBI , IAQGBS , IAQSDS , IAQSDA , IAQSDP
     .          , IQOARY , IQOCOR , QOAMPF , QOFMPF , QOTMPF , QORMPF
     .          , QOTYPE , IQOTAR , QOTAPE , QOPNCH , IQOTME , NBCOSK
     .          , RSOLAR , RALB   , RPLAN  , RFRAC  , IMESS  , ISPND
     .          , NERN   , NMESS  , NSPND  , NSFT   , NSFO   , ISOLFL
     .          , IALBFL , IPLAFL , IAI    , IAS    , TRUANF , TRUANI
     .          , NSTSOL , NSTPL
      COMMON /TAPE  /    NDI     , NDIR    , NFFI    , NFFR
     1          , NGBIR  , NGBIRR  , NGBSO   , NGBSOR
     2          , NOUT   , NRAN    , NSCR1   , NSCR2
     3          , NSCR3  , NTQ     , NTQR    , NPLS
     4          , NPLSR  , NSQNTL  , NPUN    , NBCDOU
     5          , NSHADO , NTRAJ   , NRSO    , NRTO
     6          , NUSER1 , NUSER2  , KBCDOU  , KSHADO
     7          , KTRAJ  , KRSO    , KRTO
      COMMON /TITLE /    TITLE (11)          , NTITLE(11)
     1          , MODELN , DTE     , TME     , IPAGE
     2          , LINE   , IHSTEP
      COMMON /PLOT  /    NPNNP(6)            , NPTIT(9)
     1          , NPVU(6)           , ZNPROT(6,6)
     2          , ZNPSCL(6)         , IOPNNP(6)
     3          , IOPTIT(6)         , IOPNVU(6)
     4          , OPROT(6,6)        , OPSCL(6)
     5          , OPSCLR(6)         , IOPNV
     6          , OPRPLN(6)         , OPTRUE(6)
     7          , OPTIMP(6)         , OPTIMS(6)
      COMMON /INDX  /    INDX  ( 1019)
      COMMON /INDXS /    INDXS (    1)
      COMMON /INDXN /    INDXN (    1)
      COMMON /DIMS  /    DIMS(3,    1)
      COMMON /DSTR  /    DSTR  (5,     1)
      COMMON /IFS   /    IFS   (    1)
      COMMON /IKS   /    IKS   (    1)
      COMMON /PR    /    PR    (2,    1)
      COMMON /PSH   /    PSH   (4,    1)
      COMMON /ISTR  /    ISTR  (3,3,     1)
      COMMON /ALPH  /    ALPH  (    1)
      COMMON /AREA  /    AREA  (    1)
      COMMON /EMISS /    EMISS (    1)
      COMMON /SRIR  /    SRIR  (    1)
      COMMON /SRSO  /    SRSO  (    1)
      COMMON /TRIR  /    TRIR  (    1)
      COMMON /TRSO  /    TRSO  (    1)
      COMMON /NODE  /    NODE  (    1)
      COMMON /ODTEMP/    ODTEMP(    1)
```

COMMON BLOCKS FOR OPLOT (CONT.)

```
      COMMON /ORBIT /     ALAN     , ASUN     , PSD      , DWP
     1                  , ECC      , IORNT    , IORBIT   , PERIOD
     2                  , WSS      , PALB     , PRAD     , RSUN
     3                  , CIGMAS   , BETAS    , APER     , WDS
     4                  , WSUN     , TIMEST   , TIMEPR   , TRUEAN
     5                  , SOL      , ISKPSO   , GRAV     , OINC
     6                  , HA       , HP       , SUNRA    , STRRA
     7                  , STRDEC   , SUNDEC   , CIGMA    , BETA
     8                  , RTHET    , ORNT(3,3)           , SPINT(3,3)
     9                  , ICALFL   , NSPFF    , CLOCK    , CONE
     0                  , RATE     , ROTX     , ROTY     , ROTZ
     1                  , IROTX    , IROTY    , IROTZ    , PNAME
     2                  , ISFT     , PLTYPE   , INSHAD   , SHADIN
     3                  , SHAOUT   , SUNCL    , SUNCO    , PLCL
     4                  , PLCO     , TIMSP
      COMMON /DSTORE/     IDSTR (12,3)
      COMMON /ISTPDR/     ISTPDR(     1)
      COMMON /NSPEC /     NSPEC
      COMMON /PLOTTR/     IPLUNT   , PLCRVF   , PLXMPF   , PLYMPF
     1                  , IPLNA    , IPLSN    , PLLABX(5)
     2                  , PLLABY(5)           , PLTIT1(10)
     3                  , PLTIT2(12)
      COMMON /MSP    /    MSP(     1)
      COMMON /JSURF /     JSURF(     1)
```

COMMON BLOCKS FOR PLOT

```
      COMMON /CCONST/    DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGBSFF  , GBWBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGB
     .          , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQOARY  , IQOCUR  , QOAMPF  , QOFMPF  , QOTMPF  , QORMPF
     .          , QOTYPE  , IQOTAB  , QOTAPE  , QOPNCH  , IQOTME  , NBCDSK
     .          , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /TAPE  /      NDI       , NDIR      , NFF       , NFFR
     1                   , NGBIR     , NGBIRR    , NGBSO     , NGBSOR
     2                   , NOUT      , NRAN      , NSCR1     , NSCR2
     3                   , NSCR3     , NTQ       , NTQR      , NPLS
     4                   , NPLSR     , NSQNTL    , NPUN      , NBCDOU
     5                   , NSHADO    , NTRAJ     , NRSO      , NRTO
     6                   , NUSER1    , NUSER2    , KBCDOU    , KSHADO
     7                   , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE /      TITLE (11)            , NTITLE(11)
     1                   , MODELN    , DTE       , TME       , IPAGE
     2                   , LINE      , IHSTEP
      COMMON /PLOT  /      NPNNP(6)              , NPTIT(9)
     1                   , NPVU(6)               , ZNPROT(6,6)
     2                   , ZNPSCL(6)             , IOPNNP(6)
     3                   , IOPTIT(6)             , IOPNVU(6)
     4                   , OPROT(6,6)            , OPSCL(6)
     5                   , OPSCLR(6)             , IOPNV
     6                   , OPRPLN(6)             , OPTRUE(6)
     7                   , OPTIMP(6)             , OPTIMS(6)
      COMMON /INDX  /      INDX  ( 1019)
      COMMON /INDXS /      INDXS (    1)
      COMMON /INDXN /      INDXN (    1)
      COMMON /DIMS  /      DIMS(3,    1)
      COMMON /DSTR  /      DSTR  (5,     1)
      COMMON /IFS   /      IFS   (    1)
      COMMON /IKS   /      IKS   (    1)
      COMMON /PR    /      PR    (2,     1)
      COMMON /PSH   /      PSH   (4,     1)
      COMMON /TSTR  /      TSTR  (3,3,     1)
      COMMON /ALPH  /      ALPH  (    1)
      COMMON /AREA  /      AREA  (    1)
      COMMON /EMISS /      EMISS (    1)
      COMMON /SRIR  /      SRIR  (    1)
      COMMON /SRSO  /      SRSO  (    1)
```

COMMON BLOCKS FOR PLOT (CONT.)

```
      COMMON /TRIR  /     TRIR  (     1)
      COMMON /TRSO  /     TRSO  (     1)
      COMMON /NODE  /     NODE  (     1)
      COMMON /ODTEMP/     ODTEMP(     1)
      COMMON /ORBIT /     ALAN      , ASUN      , PSD       , DWP
     1                  , ECC       , IORNT     , IORBIT    , PERIOD
     2                  , WSS       , PALB      , PRAD      , RSUN
     3                  , CIGMAS    , BETAS     , APER      , WDS
     4                  , WSUN      , TIMEST    , TIMEPR    , TRUEAN
     5                  , SOL       , ISKPSO    , GRAV      , OINC
     6                  , HA        , HP        , SUNRA     , STRRA
     7                  , STRDEC    , SUNDEC    , CIGMA     , BETA
     8                  , RTHET     , ORNT(3,3) ,           , SPINT(3,3)
     9                  , ICALFL    , NSPFF     , CLOCK     , CONE
     0                  , RATE      , ROTX      , ROTY      , ROTZ
     1                  , IROTX     , IROTY     , IROTZ     , PNAME
     2                  , ISFT      , PLTYPE    , INSHAD    , SHADIN
     3                  , SHAOUT    , SUNCL     , SUNCO     , PLCL
     4                  , PLCO      , TIMSP
      COMMON /DSTORE/     IDSTR (12,3)
      COMMON /ISTPDR/     ISTPDR(     1)
      COMMON /NSPEC /     NSPEC
      COMMON /PLOTTR/     IPLUNT    , PLCRVF    , PLXMPF    , PLYMPF
     1                  , IPLNA     , IPLSN     , PLLABX(5)
     2                  , PLLABY(5)             , PLTIT1(10)
     3                  , PLTIT2(12)
```

COMMON BLOCKS FOR QOCAL

```
      COMMON /CCONST/    DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGBSFF  , GBWBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGB
     .          , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQQARY  , TOOCOR  , QQAMPF  , QQFMPF  , QQTMPF  , QQRMPF
     .          , QQTYPE  , IQQTAB  , QQTAPE  , QQPNCH  , IQQTME  , NBCDSK
     .          , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /TAPE   /    NDI       , NDIR      , NFF       , NFFR
     1                  , NGBIR     , NGBIRR    , NGBSO     , NGBSOR
     2                  , NOUT      , NRAN      , NSCR1     , NSCR2
     3                  , NSCR3     , NTQ       , NTQR      , NPLS
     4                  , NPLSR     , NSQNTL    , NPUN      , NBCDOU
     5                  , NSHADO    , NTRAJ     , NRSO      , NRTO
     6                  , NUSER1    , NUSER2    , KBCDOU    , KSHADO
     7                  , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE  /    TITLE (11)            , NTITLE(11)
     1                  , MODELN    , DTE       , TME       , IPAGE
     2                  , LINE      , IHSTEP
      COMMON /PLOT   /    NPNNP(6)              , NPTIT(9)
     1                  , NPVU(6)               , ZNPROT(6,6)
     2                  , ZNPSCL(6)             , IOPNNP(6)
     3                  , IOPTIT(6)             , IOPNVU(6)
     4                  , OPROT(6,6)            , OPSCL(6)
     5                  , OPSCLR(6)             , IOPNV
     6                  , OPRPLN(6)             , OPTRUE(6)
     7                  , OPTIMP(6)             , OPTIMS(6)
      COMMON /INDX   /    INDX   ( 1019)
      COMMON /INDXS  /    INDXS  (    1)
      COMMON /INDXN  /    INDXN  (    1)
      COMMON /DIMS   /    DIMS(3,     1)
      COMMON /DSTR   /    DSTR   (5,     1)
      COMMON /IFS    /    IFS    (    1)
      COMMON /IKS    /    IKS    (    1)
      COMMON /PR     /    PR     (2,     1)
      COMMON /PSH    /    PSH    (4,     1)
      COMMON /TSTR   /    TSTR   (3,3,     1)
      COMMON /ALPH   /    ALPH   (    1)
      COMMON /AREA   /    AREA   (    1)
      COMMON /EMISS  /    EMISS  (    1)
      COMMON /SRIR   /    SRIR   (    1)
      COMMON /SRSO   /    SRSO   (    1)
```

COMMON BLOCKS FOR QOCAL (CONT.)

```
      COMMON /TRIR  /     TRIR  (     1)
      COMMON /TRSO  /     IRSO  (     1)
      COMMON /NODE  /     NODE  (     1)
      COMMON /ODTEMP/     ODTEMP(     1)
      COMMON /ORBIT /     ALAN      , ASUN      , PSO       , DWP
     1                  , ECC       , IORNT     , IORBIT    , PERIOD
     2                  , ASS       , PALR      , PRAD      , RSUN
     3                  , CIGMAS    , BETAS     , APER      , WDS
     4                  , WSUN      , TIMEST    , TIMEPR    , TRUEAN
     5                  , SOL       , ISKPSO    , GRAV      , OINC
     6                  , HA        , HP        , SUNRA     , STRRA
     7                  , STRDEC    , SUNDEC    , CIGMA     , BETA
     8                  , RIHET     , OPNT(3,3)             , SPINT(3,3)
     9                  , ICALFL    , NSPEF     , CLOCK     , CONE
     0                  , RATE      , ROTX      , ROTY      , ROTZ
     1                  , IROTX     , IROTY     , IROTZ     , PNAME
     2                  , ISFT      , PLTYPE    , INSHAD    , SHADIN
     3                  , SHADUT    , SUNCL     , SUNCO     , PLCL
     4                  , PLCO      , TIMSP
      COMMON /DSTORE/     IDSTR (12,3)
      COMMON /ISTPDP/     ISTPDR(     1)
      COMMON /NSPEC /     NSPEC
      COMMON /PLOTTR/     IPLUNT    , PLCRVF    , PLAMPF    , PLYMPF
     1                  , IPLNA     , IPLSN     , PLLABX(5)
     2                  , PLLABY(5)             , PLTIT1(10)
     3                  , PLTIT2(12)
      COMMON /QONODT/     NODET (     1)
      COMMON /QAVERG/     QAVERG(     1)
      COMMON /QOCMB /     ICOMB (   100)
      COMMON /QOFRST/     IFIRST(   100)
      COMMON /AREAT /     AREAT (     1)
```

COMMON BLOCKS FOR RCCAL

```
      COMMON /CCONST/    DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NHOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGRSFF  , GRNBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGR
     .          , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGGI  , IAQGRS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQQARY  , IQQCOR  , QQAMPF  , QQFMPF  , QQTMPF  , QQRMPF
     .          , QQTYPE  , IQQTAB  , QQTAPE  , QQPNCH  , IQQTME  , NBCDSK
     .          , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSET    , NSEQ    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /TAPE   /    NDI       , NDIR    , NFF     , NFFR
     1                  , NGBIR     , NGBIRR  , NGBSQ   , NGBSQR
     2                  , NOUT      , NRAN    , NSCR1   , NSCR2
     3                  , NSCR3     , NTQ     , NTQR    , NPLS
     4                  , NPLSR     , NSQNTL  , NPUN    , NBCDQQ
     5                  , NSHADQ    , NTRAJ   , NRSQ    , NRTQ
     6                  , NUSER1    , NUSER2  , KBCDQQ  , KSHADQ
     7                  , KTRAJ     , KRSQ    , KRTQ
      COMMON /TITLE  /    TITLE (11)          , NTITLE(11)
     1                  , MODELN    , DTE     , TME     , IPAGE
     2                  , LINE      , IHSTEP
      COMMON /PLOT   /    NPNNP(6)            , NPTIT(9)
     1                  , NPVU(6)            , ZNPROT(6,6)
     2                  , ZNPSCL(6)          , IOPNNP(6)
     3                  , IOPTIT(6)          , IOPNVU(6)
     4                  , OPROT(6,6)         , OPSCL(6)
     5                  , OPSCLR(6)          , IOPNV
     6                  , OPRPLN(6)          , OPTRUE(6)
     7                  , OPTIMP(6)          , OPTIMS(6)
      COMMON /INDX   /    INDX   ( 1019)
      COMMON /INDXS  /    INDXS  (    1)
      COMMON /INDXN  /    INDXN  (    1)
      COMMON /DIMS   /    DIMS(3,     1)
      COMMON /DSTR   /    DSTR   (5,     1)
      COMMON /IFS    /    IFS    (    1)
      COMMON /IKS    /    IKS    (    1)
      COMMON /PR     /    PR     (2,     1)
      COMMON /PSH    /    PSH    (4,     1)
      COMMON /TSTR   /    TSTR   (3,3,     1)
      COMMON /ALPH   /    ALPH   (    1)
      COMMON /AREA   /    AREA   (    1)
      COMMON /EMISS  /    EMISS  (    1)
      COMMON /SRIR   /    SRIR   (    1)
      COMMON /SRSQ   /    SRSQ   (    1)
```

COMMON BLOCKS FOR RCCAL (CONT.)

```
      COMMON /TRIR  /     TRIR  (     1)
      COMMON /TRSO  /     TRSO  (     1)
      COMMON /NODE  /     NODE  (     1)
      COMMON /ODTEMP/     ODTEMP(     1)
      COMMON /ORBIT /     ALAN      , ASUN      , PSD       , DWP
     1                  , ECC       , IORNT     , IORBIT    , PERIOD
     2                  , NSS       , PALB      , PRAD      , RSUN
     3                  , SIGMAS    , BETAS     , APER      , WDS
     4                  , WSUN      , TIMEST    , TIMEPR    , TRUEAN
     5                  , SOL       , ISKPSO    , GRAV      , OINC
     6                  , HA        , HP        , SUNRA     , STRRA
     7                  , STRDEC    , SUNDEC    , SIGMA     , BETA
     8                  , RTHET     , ORNT(3,3)             , SPINT(3,3)
     9                  , ICALFL    , NSPEF     , CLOCK     , CONE
     0                  , RATE      , ROTX      , ROTY      , ROTZ
     1                  , IROTX     , IROTY     , IROTZ     , PNAME
     2                  , ISFT      , PLTYPE    , INSHAD    , SHADIN
     3                  , SHAOUT    , SUNCL     , SUNCO     , PLCL
     4                  , PLCO      , TIMSP
      COMMON /DSTORE/     IDSTR (12,3)
      COMMON /ISTPDR/     ISTPDR(     1)
      COMMON /NSPEC /     NSPEC
      COMMON /PLOTTR/     IPLUNT    , PLCRVF    , PLXMPF    , PLYMPF
     1                  , IPLNA     , IPLSN     , PLLABX(5)
     2                  , PLLABY(5)             , PLTIT1(10)
     3                  , PLTIT2(12)
      COMMON /ISPN  /     ISPN  (   100)
      COMMON /MSND  /     MSND  (   100)
      COMMON /NDS   /     NDS   (     1)
      COMMON /SFS   /     SF    (     1)
      COMMON /SPCND /     SPACND(     1)
```

COMMON BLOCKS FOR RKCAL

```
      COMMON /CCONST/    DTR     , RTD     , PI      , MAXBC   , NN
     .          , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .          , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .          , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .          , IGBSFF  , GBWBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGB
     .          , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .          , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .          , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .          , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .          , IQOARY  , IQOCOR  , QOAMPF  , QOFMPF  , QOTMPF  , QORMPF
     .          , QOTYPE  , IQOTAB  , QOTAPE  , QOPNCH  , IQOTME  , NBCDSK
     .          , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .          , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .          , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .          , NSTSOL  , NSTPL
      COMMON /TAPE   /     NDI       , NDIR      , NFF       , NFFR
     1                   , NGBIR     , NGBIRR    , NGBSO     , NGBSOR
     2                   , NOUT      , NRAN      , NSCR1     , NSCR2
     3                   , NSCR3     , NTQ       , NTQR      , NPLS
     4                   , NPLSR     , NSQNTL    , NPUN      , NBCDOU
     5                   , NSHADO    , NTRAJ     , NRSO      , NRTO
     6                   , NUSER1    , NUSER2    , KBCDOU    , KSHADO
     7                   , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE  /     TITLE (11)            , NTITLE(11)
     1                   , MODELN    , DTE       , TME       , IPAGE
     2                   , LINE      , IHSTEP
      COMMON /PLOT   /     NPNNP(6)              , NPTIT(9)
     1                   , NPVU(6)               , ZNPROT(6,6)
     2                   , ZNPSCL(6)             , IOPNNP(6)
     3                   , IOPTIT(6)             , IOPNVU(6)
     4                   , OPROT(6,6)            , OPSCL(6)
     5                   , OPSCLR(6)             , IOPNV
     6                   , OPRPLN(6)             , OPTRUE(6)
     7                   , OPTIMP(6)             , OPTIMS(6)
      COMMON /INDX   /     INDX   ( 1019)
      COMMON /INDXS  /     INDXS  (    1)
      COMMON /INDXN  /     INDXN  (    1)
      COMMON /DIMS   /     DIMS(3,     1)
      COMMON /DSTR   /     DSTR   (5,     1)
      COMMON /IFS    /     IFS    (    1)
      COMMON /IKS    /     IKS    (    1)
      COMMON /PR     /     PR     (2,     1)
      COMMON /PSH    /     PSH    (4,     1)
      COMMON /TSTR   /     TSTR   (3,3,     1)
      COMMON /ALPH   /     ALPH   (    1)
      COMMON /AREA   /     AREA   (    1)
      COMMON /EMISS  /     EMISS  (    1)
      COMMON /SRIR   /     SRIR   (    1)
      COMMON /SRSO   /     SRSO   (    1)
      COMMON /TRIR   /     TRIR   (    1)
      COMMON /TRSO   /     TRSO   (    1)
      COMMON /NODE   /     NODE   (    1)
      COMMON /ODTEMP/      ODTEMP(    1)
```

COMMON BLOCKS FOR RKCAL (CONT.)

```
      COMMON /ORBIT /     ALAN     , ASUN     , PSD      , DWP
     1                  , ECC      , IORNT    , IORBIT   , PERIOD
     2                  , WSS      , PALB     , PRAD     , RSUN
     3                  , CIGMAS   , BETAS    , APER     , WDS
     4                  , WSUN     , TIMEST   , TIMEPR   , TRUEAN
     5                  , SOL      , ISKPSO   , GRAV     , OINC
     6                  , HA       , HP       , SUNRA    , STRRA
     7                  , STRDEC   , SUNDEC   , CIGMA    , BETA
     8                  , RTHET    , ORNT(3,3)           , SPINT(3,3)
     9                  , ICALFL   , NSPFF    , CLOCK    , CONE
     0                  , RATE     , ROTX     , ROTY     , ROTZ
     1                  , IROTX    , IROTY    , IROTZ    , PNAME
     2                  , ISET     , PLTYPE   , INSHAD   , SHADIN
     3                  , SHADUT   , SUNCL    , SUNCO    , PLCL
     4                  , PLCO     , TIMSP
      COMMON /DSTORE/     IDSTR (12,3)
      COMMON /ISTPDR/     ISTPDR(    1)
      COMMON /NSPEC /     NSPEC
      COMMON /PLOTTR/     IPLUNT   , PLCRVF   , PLXMPF   , PLYMPF
     1                  , IPLNA    , IPLSN    , PLLABX(5)
     2                  , PLLABY(5)           , PLTIT1(10)
     3                  , PLTIT2(12)
      COMMON /RKCMB /     ICOMB (  100)
      COMMON /RKERST/     IERST(  100)
      COMMON /RKNODT/     NODET (    1)
      COMMON /SF    /     SF    (    1)
      COMMON /SPACNO/     SPACNO(    1)
```

COMMON BLOCKS FOR SFCAL

```
      COMMON /SFSHDC/     ISHAD(     1)
      COMMON /SFQDP /     QDP(     1)
      COMMON /SFQDR /     QDR(     1)
      COMMON /SFQDS /     QDS(     1)
      COMMON /CCONST/     DTR     , PTD     , PI      , MAXBC   , NN
     .           , NS      , NNOD    , NSURF   , IOVL    , NSTEP   , NSSTEP
     .           , NBLKLN  , DIACC   , DIACCS  , DINOSH  , DIPNCH  , FFACC
     .           , FFACCS  , FFMIN   , FFRATL  , FFNOSH  , FFPNCH  , FFPRNT
     .           , IGBSFF  , GBWBND  , RKAMPF  , IRKCN   , RKMIN   , IRKNGB
     .           , IRKNSP  , RKPNCH  , RKSP    , RKTAPE  , SIGMA   , ITRC10
     .           , ITRC20  , ITRC30  , ITRC40  , ITRC50  , ITRC60  , ITRC70
     .           , ITRC80  , ITRC90  , ITRCA0  , ITRCB0  , ITRCC0  , ITRCD0
     .           , ITRALL  , IAQGBI  , IAQGBS  , IAQSDS  , IAQSDA  , IAQSDP
     .           , IQOARY  , IQOCOR  , QOAMPF  , QOFMPF  , QOTMPF  , QORMPF
     .           , QOTYPE  , IQOTAB  , QOTAPE  , QOPNCH  , IQOTME  , NBCDSK
     .           , RSOLAR  , RALB    , RPLAN   , RFRAC   , IMESS   , ISPND
     .           , NERN    , NMESS   , NSPND   , NSFT    , NSFO    , ISOLFL
     .           , IALBFL  , IPLAFL  , IAI     , IAS     , TRUANF  , TRUANI
     .           , NSTSOL  , NSTPL
      COMMON /TAPE  /     NDI       , NDIR      , NFF       , NFFR
     1                  , NGBIR     , NGBIRR    , NGBSO     , NGBSOR
     2                  , NOUT      , NRAN      , NSCR1     , NSCR2
     3                  , NSCR3     , NTQ       , NTQR      , NPLS
     4                  , NPLSR     , NSQNTL    , NPUN      , NBCDOU
     5                  , NSHADO    , NTRAJ     , NRSO      , NRTO
     6                  , NUSER1    , NUSER2    , KBCDOU    , KSHADO
     7                  , KTRAJ     , KRSO      , KRTO
      COMMON /TITLE /     TITLE (11)              , NTITLE(11)
     1                  , MODELN    , DTE       , TME       , IPAGE
     2                  , LINE      , IHSTEP
      COMMON /PLOT  /     NPNNP(6)                , NPTIT(9)
     1                  , NPVU(6)               , ZNPROT(6,6)
     2                  , ZNPSCL(6)             , IOPNNP(6)
     3                  , IOPTIT(6)             , IOPNVU(6)
     4                  , OPROT(6,6)            , OPSCL(6)
     5                  , OPSCLR(6)             , IOPNV
     6                  , OPRPLN(6)             , OPTRUE(6)
     7                  , OPTIMP(6)             , OPTIMS(6)
      COMMON /INDX  /     INDX   ( 1019)
      COMMON /INDXS /     INDXS (     1)
      COMMON /INDXN /     INDXN (     1)
      COMMON /DIMS  /     DIMS(3,     1)
      COMMON /DSTR  /     DSTR   (5,     1)
      COMMON /IFS   /     IFS    (     1)
      COMMON /IKS   /     IKS    (     1)
      COMMON /PR    /     PR     (2,     1)
      COMMON /PSH   /     PSH    (4,     1)
      COMMON /TSTR  /     TSTR   (3,3,     1)
      COMMON /ALPH  /     ALPH   (     1)
      COMMON /AREA  /     AREA   (     1)
      COMMON /EMISS /     EMISS  (     1)
```

COMMON BLOCKS FOR SFCAL (CONT.)

```
      COMMON /SRIR  /     SRIR  (     1)
      COMMON /SRSO  /     SRSO  (     1)
      COMMON /TRIR  /     TRIR  (     1)
      COMMON /TRSO  /     TRSO  (     1)
      COMMON /NODE  /     NODE  (     1)
      COMMON /ODTEMP/     ODTEMP(     1)
      COMMON /ORBIT /     ALAN      , ASUN      , PSU       , DWP
     1                  , ECC       , IORNT     , IORBIT    , PERIOD
     2                  , WSS       , PALB      , PRAD      , RSUN
     3                  , CIGMAS    , BETAS     , APER      , WDS
     4                  , WSUN      , TIMEST    , TIMEPR    , TRUEAN
     5                  , SOL       , ISKPSO    , GRAV      , OINC
     6                  , HA        , HP        , SUNRA     , STRRA
     7                  , STRDEC    , SUNDEC    , CIGMA     , BETA
     8                  , RTHET     , ORNT(3,3)             , SPINT(3,3)
     9                  , ICALFL    , NSPFF     , CLOCK     , CONE
     0                  , RATE      , ROTX      , ROTY      , ROTZ
     1                  , IROTX     , IROTY     , IROTZ     , PNAME
     2                  , ISFT      , PLTYPE    , INSHAD    , SHADIN
     3                  , SHAOUT    , SUNCL     , SUNCO     , PLCL
     4                  , PLCO      , TIMSP
      COMMON /IDSTORE/     IDSTR (12,3)
      COMMON /ISTPOR/     ISTPOR(     1)
      COMMON /NSPEC /     NSPEC
      COMMON /SFSHDC/     ISHAD(     1)
      COMMON /SFQDP /     QDP(     1)
      COMMON /SFQOR /     QOR(     1)
      COMMON /SFQOS /     QOS(     1)
```

APPENDIX B

FORM FACTOR CALCULATION ACCURACY

A. ELEMENTAL GRID VARIATIONS

The form factor for two finite areas, A_I and A_J (Fig. B-1), is defined as

$$F_{IJ} = \frac{1}{A_I} \int_{A_I} \int_{A_J} \frac{\cos \theta_i \cos \theta_j}{\pi r_{ij}^2} \, dA_J \, dA_I. \qquad [B\text{-}1]$$

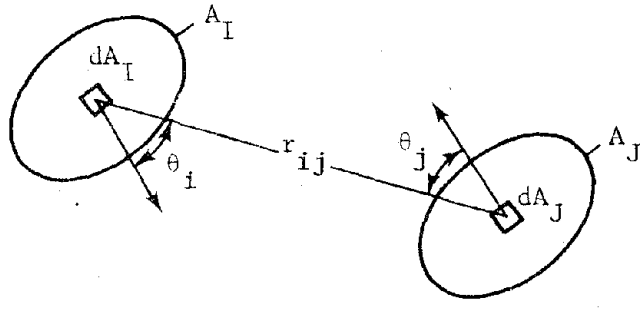


Figure B-1 Determination of Form Factors

A finite-difference approximation of Equation [B-1] is

$$F_{IJ} = \frac{1}{A_I} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\cos \theta_i \cos \theta_j}{\pi r_{ij}^2} A_j \, A_i. \qquad [B\text{-}2]$$

Equation [B-2] approaches an exact representation of Equation [B-1] as the size of the elemental areas, A_i and A_j, approach zero. For identical, parallel, directly opposed rectangles, the empirical relationship of elemental area size-to-separation distance versus form factor error shown in Figure B-2 is obtained.

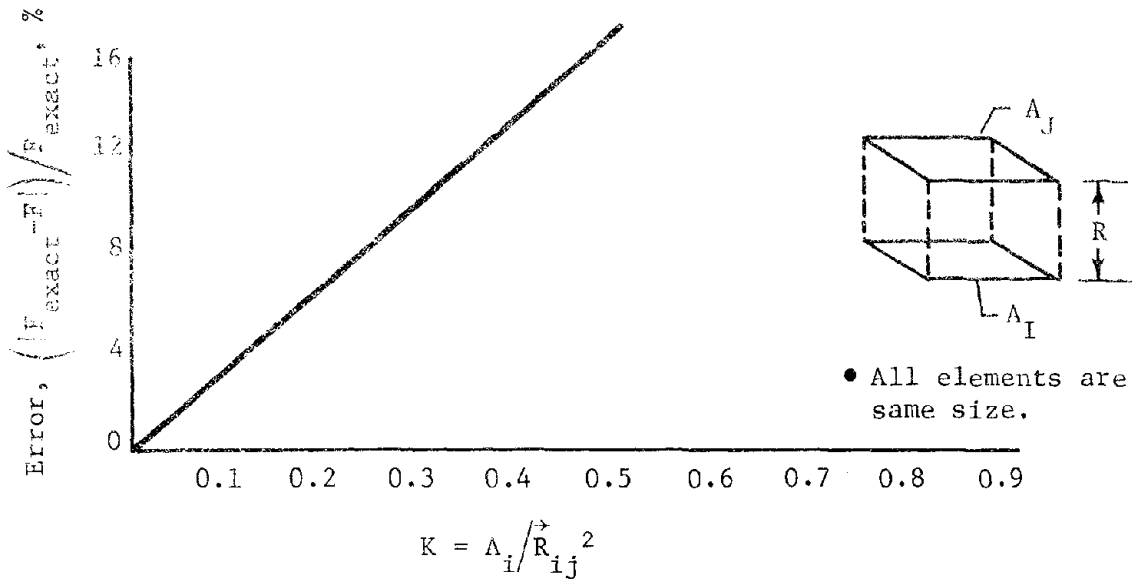


Figure B-2 Error Characteristics for Identical, Parallel, Directly Opposed Rectangles

For this case,

$$A_i = K\, r_{ij}^{\,2}, \qquad [B\text{-}3]$$

where K is a proportionality constant.

A more general form of Equation [B-2], considering that

$$dF_{i-j} = \frac{\cos\theta_i \cos\theta_j}{\pi r_{ij}^{\,2}}\, dA_j \qquad [B\text{-}4]$$

or

$$\frac{dA_j}{dF_{i-j}} = \frac{\pi r_{ij}^{\,2}}{\cos\theta_i \cos\theta_j}, \qquad [B\text{-}5]$$

is

$$A_i \approx \frac{K\, r_{ij}^{\,2}}{\cos\theta_i \cos\theta_j}. \qquad [B\text{-}6]$$

MTRAP version 1.0, LOHARP, and TRASYS use Equation B-2, modified with a shadowing constant, to compute form factors. The total number (i.e., size) and distribution of elemental areas is left to the user to define in MTRAP version 1.0 and LOHARP. The number of elements to be selected is defined by the closest node a given node "sees." The selection of elements, however, usually ends up being somewhat arbitrary or simply a matter of economics; i.e., the finer the grid, the more machine time required. In reality, the selection of elemental areas is an independent problem for each form factor.

The basic assumption for reasonably accurate form factors is, from Equation [B-6], that the elemental area size is small compared to the separation distance between two elements.

The TRASYS program uses a technique using Equation [B-6] to automatically select the element grid sizes of each node pair consistent with a user-defined accuracy parameter, K. If all elemental areas on each of two nodes were the same size and had the same separation distance, r_{ij}, the apparent number of elements on a node to satisfy the accuracy value K would be (from Equation [B-6]

$$N_I = \frac{A_I}{A_i} = \frac{A_I}{K} \frac{\cos \theta_i \cos \theta_j}{r_{ij}^2} . \qquad [B-7]$$

Since each element pair on the two nodes may have a different separation distance, a different apparent number of equal-sized elements will be required.

The approach used in the TRASYS program is a simple arithmetic average of element contributions, i.e.,

$$N_{opt_I} = \frac{A_I}{K m_i m_j} \sum_{i=1}^{m_i} \sum_{j=1}^{m_j} \frac{\cos \theta_i \cos \theta_j}{r_{ij}^2}, \qquad [B-8]$$

where m_i and m_j are the initial number of elements arbitrarily chosen for nodes I and J.

The initial number of elements is chosen just large enough for a representative sample. A similar optimum number of elements for node J can be defined.

The total number of elements defined by Equation [B-8] is distributed uniformly over the node using a criterion that attempts to make the elements square. The arithmetic average technique assumes that the mean separation distance between nodes is large compared with the variation of separation distance over the two nodes. A check is made to see if this assumption is violated. The maximum number of elements defined by any element pair on the two nodes (Equation [B-7]) is compared with the arithmetic average value (Equation [B-8]). If the ratio of N_{max}/N_{opt} is greater than $N_{critical}$, the two nodes are temporarily subdivided into subnodes. $N_{critical}$ is an input value defined by the user. The numbers of subnodes used are proportional to $\left(N_{max}/N_{opt}\right)_I$ and $\left(N_{max}/N_{opt}\right)_J$. The optimum grid elements are computed independently for each subnode using a separation-distance, weighted-average criterion, rather than an arithmetic one. The form factors resulting from the subnode pairs are then combined using form-factor algebra. Thus, elemental grids vary for each form factor and may be nonuniform over a node as required to satisfy input accuracy requirements.

B. NODAL PRELIMINARY SHADOWING CHECKS

Shadowing checks between elemental areas account for considerable machine time. Machine time could be saved if unnecessary checks were eliminated. The usual procedure in MTRAP version 1.0 is to process all the surfaces until either the form factor contribution is reduced to zero by shadowing surfaces, or all surfaces identified as shadowers have been investigated. The function of the nodal shadowing checks is to eliminate from the element-to-element shadowing checks all surfaces that cannot cause shadowing on any portion of the two nodes under consideration. The technique used in the TRASYS program is a significant modification of the technique used in LOHARP.

The nodal shadowing checks consist of constructing a sphere around each node for which form factors are being evaluated and for each shadowing surface. The radii of the spheres are such that the node or surface is completely enclosed. A test cylinder or cone frustum, depending on the relative sizes of the two spheres in question, is constructed as shown in Figure B-3. For the cylinder, the radius is equal to the larger of the two spheres. The cylinder or cone frustum's axial coordinate is a vector between the centers of the two spheres plus the sum of the two sphere radii. Next, the shadowing surfaces' enclosing spheres are checked to determine whether they intersect the test cylinder or cone frustum. Only surfaces whose sphere intersects the test cylinder or cone frustum will be considered in the actual element-to-element shadowing checks for these two nodes.

This technique of preliminary shadowing checks allows identification of any surface that shades the nodes in question. However, other marginal ones will also be identified.

In the detailed element-to-element shadowing checks, an element pair is either completely shadowed or not at all. The accuracy, then, of representing the shadow is proportional to the total number of elements on both nodes. The number of elements on the shadowing surface(s) is of no consideration. In the TRASYS program it is assumed that accurate shadowing is required only for large-magnitude form factors. If the preliminary shadowing checks identify shadowing surfaces for the form factor in question, the number of elements defined to represent the shadow is

$$N_{s_I} = F_{IJ} \, B / K_s$$

$$N_{s_J} = F_{JI} \, B / K_s , \qquad [B-9]$$

where

K_s is an input shadowing accuracy factor, and

B is a proportionality constant determined by trial and error.

The number of elements used for any given node for form factors is taken as the maximum of that defined in Equation [B-8] or Equation [B-9].

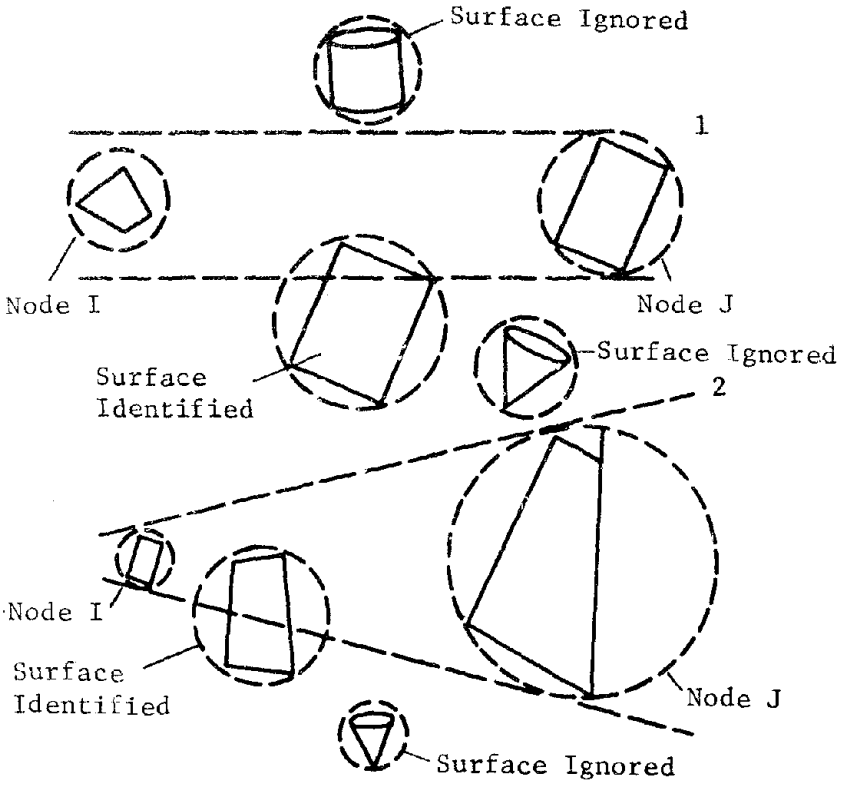


Figure B-3 Nodal Preliminary Shadowing Techniques

APPENDIX C

SHADOW FACTOR TAPE FORMAT

The Shadow Factor (SHADI) tape is a multifile binary tape. Each file contains a 171-element, bivariate, shadow-factor table for each node in a configuration. The files are identified by configuration (model) name and step number. The step number is that under which the tables were originally generated. The date each file was generated is also stored.

The first record in each file has the following format:

Word 1 - Configuration name. (Any Hollerith name of up to 6 characters, left-justified, blank-filled)

Word 2 - Date generated. (Hollerith, month, day, year.)

Word 3 - Number of nodes in configuration, NN (integer)

Words 4 thru NN + 3 - node number array. (one integer node number per word)

The second record of each file has the following format:

Word 1 contains 9 shadow factors, for cone angles 1 through 9 at clock angle 1. These shadow factors are for the first node in the node array.

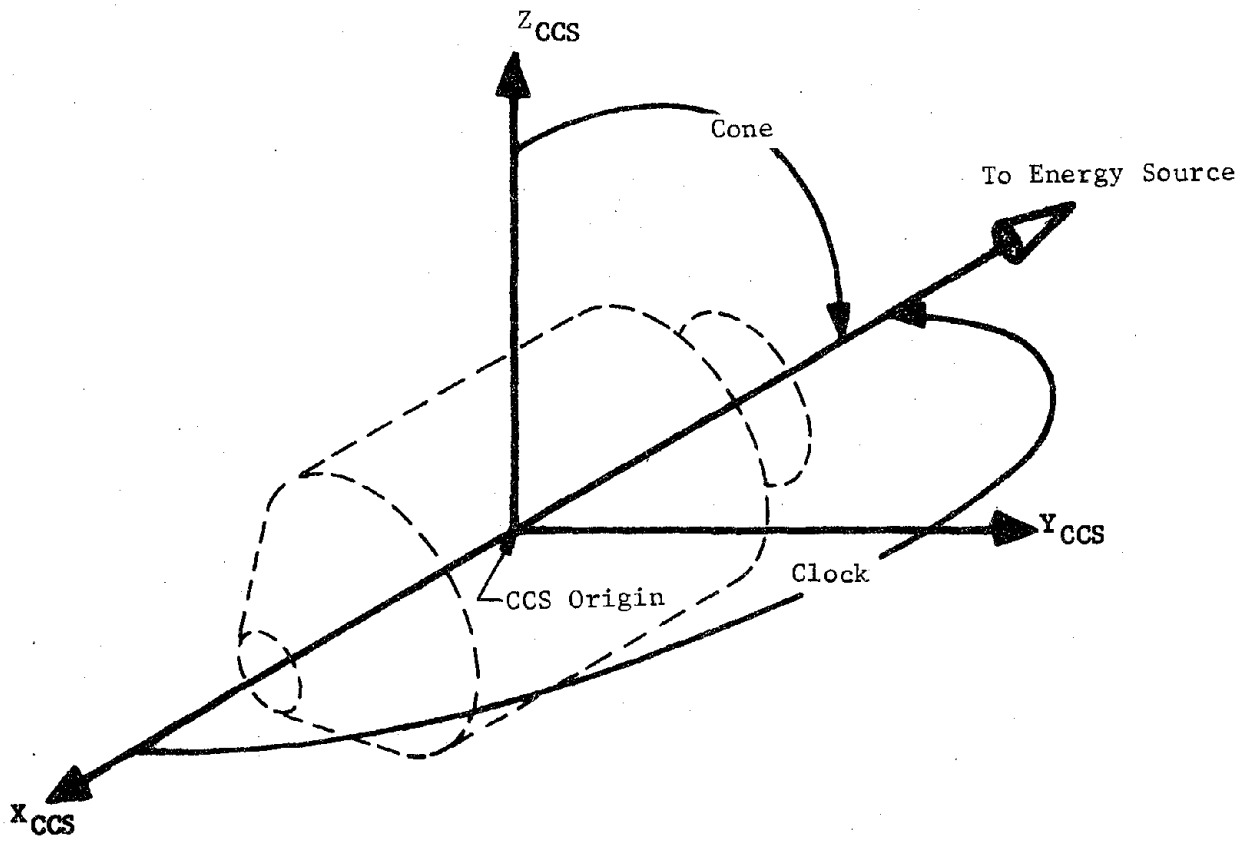
Words 2 through 19 each contain 9 shadow factors, for cone angles 1 through 9, at each clock angle, 2 through 19. Word 19 completes a shadow-factor table for node 1.

Words 20 through 190 are in 19-word groups identical in structure to words 1 through 19. Word 190 completes the shadow-factor table for the tenth node in the node array and the second record.

The third through last record of each file contains the shadow-factor tables for groups of ten nodes, in the order encountered in the node array. These records have the same format as record 2.

The tape is terminated by a double EOF. 100 configurations (files) per tape are allowed.

Clock angles 1 through 19 range from 0° to 360° in 20° increments about the central coordinate system z-axis. (See Figure AC-1) The 0° and 360° points are repeated to avoid wrap-around interpolation. Cone angles 1 through 9 are the inverse cosines of -1.0, -0.75, -0.5, -0.25, 0., 0.25, 0.5, 0.75, and 1.0, respectively. The shadow factors for nine cone angles are packed into one 36-bit word, providing 4 bits per shadow factor.



Shadow table point illustrated for
Cone 5 (90^{o}), Clock 10 (180^{o}).

Figure C-1 Energy Source Direction for Shadow Factor Tables

APPENDIX D

SUBROUTINE DESCRIPTIONS

SUBROUTINE NAME: BUILDC

PURPOSE:

This subroutine is used to define as problem geometry all nodes and surfaces identified with a Block Coordinate System. BUILDC is the first call to define a new configuration.

VARIABLE NAME:

BCSNAM is a Hollerith block coordinate system name consisting of up to 6 characters.

RESTRICTIONS:

Must be called prior to any Subroutine ADD calls within a step. BCSNAM must be ALLBLK, or a block coordinate systems name as defined in surface data.

NOTE: If BCSNAM = ALLBLK, all surfaces in the surface data block become problem geometry.

CALLING SEQUENCE:

```
CALL BUILDC (BCSNAM)
```

SUBROUTINE NAME: ADD

PURPOSE:

This subroutine adds to the problem geometry all nodes/surfaces contained in BCSNAM

VARIABLE NAME:

BCSNAM is a Hollerith Block Coordinate System name of up to 6 characters.

RESTRICTIONS:

Call valid only after previous calls to BUILDC or ADD within current step. BCSNAM must be a block coordinate system name as defined in surface data.

CALLING SEQUENCE:

CALL ADD (BCSNAM)

SUBROUTINE NAME: CHGBLK

PURPOSE:

This subroutine allows user to change block coordinate system parameters where:

BCSNAM - block coordinate system to be changed
TX - translation along CCS X-axis
TY - translation along CCS Y-axis
TZ - translation along CCS Z-axis
IROTX - order X rotation is to be performed (1,2,3)
IROTY - order Y rotation is to be performed (1,2,3)
IROTZ - order Z rotation is to be performed (1,2,3)
ROTX - angle of rotation about CCS X-axis
ROTY - angle of rotation about CCS Y-axis
ROTZ - angle of rotation about CCS Z-axis

RESTRICTIONS:

TX, TY, TZ, ROTX, ROTY, ROTZ must be floating-point numbers.
IROTX, IROTY, IROTZ must be integers, 1, 2, or 3

CALLING SEQUENCE:

CALL CHGBLK (BCSNAM, TX, TY, TZ, IROTX, IROTY, IROTZ, ROTX, ROTY, ROTZ)

SUBROUTINE NAMES: NDATA, NDATAS

PURPOSE:

These subroutines may be called prior to a call to the node plotter segment to define optional views and miscellaneous parameters where:

Parameter	Description	Options*	Default
NV	View number	1-6	1
VU	View	3HALL, 3H3-D 1HX, 1HY, 1HZ 3HGEN	3HALL
SCL	Scale	Floating-point no.	Automatic scale
SELN	Name of array containing identification numbers of nodes to be selectively plotted	Array name	Plot all nodes
TIT	Array name of plot title	Array name (array length 66 characters max.)	Uses job title
IROTX, IROTY, IROTZ	Order of rotations (for VU = 3HGEN)	1,2,3 (any order)	1,2,3
ROTX, ROTY, ROTZ	View rotations (for VU = 3HGEN)	Real no.	0.0, 0.0, 0.0

* Input zero for default action

NOTE: The NV parameter allows the user to define up to 6 plot operations that will be executed with one NPLOT call. Later in execution, (after a geometry change, for instance), he can execute the same 6 operations or change one or more by reference to the appropriate NV before his NPLOT call.

RESTRICTIONS:

None

CALLING SEQUENCE:

```
CALL NDATA (NV, VU, SCL, SELN, TIT, IROTX, IROTY, IROTZ, ROTX,
ROTY, ROTZ)

CALL NDATAS (NV, VU, SCL)
```

SUBROUTINE NAMES: ODATA, ODATAS

PURPOSE:

These subroutines may be called prior to a call to the orbit plotter to define optional views and miscellaneous parameters where:

Parameter	Description	Options*	Default
NV	View number	1-6	1
VU	View	3HALL, 3H3-D, 4HBETA, 5HCIGMA, 3HSUN, 3HGEN	3HALL
SCL	Spacecraft size measured from CCS origin in plot frame dimensions	Real no.	Computed automatically
SCLR	Orbit radius in plot frame dimensions	Real no.	Computed automatically
RPLN	Planet radius in plot frame dimension	Real no.	1.4 inches
TRUEAN	True anomaly	Real no.	None
TIMEST	Time of periapsis passage	Real no.	None
TIME	Present time	Real no.	
SELN	Name of array containing surface numbers to be selectively plotted	Array name (array length 66 characters, max.)	Plots all surfaces defines as shadowers
TIT	Array name of plot title	Array name	Uses job title
IROTX, IROTY, IROTZ	Order of rotations (for view = 3HGEN)	1,2,3 (any order)	1,2,3
ROTX, ROTY, ROTZ	View rotations (for view = 3HGEN)	Real no.	0.0, 0.0, 0.0

* Input zero for default action

The NV parameter allows the user to define up to 6 plot operations that will be executed with one OPLOT call. Later in execution (after a geometry change, for instance), he can execute the same 6 operations or change one or more by reference to the appropriate NV before his OPLOT call.

SUBROUTINES ODATA, ODATAS (CONT'D)

RESTRICTIONS:

Calls valid only after orbit has been defined.

CALLING SEQUENCE:

CALL ODATA (NV, VU, SCL, SCLR, RPLN, TRUEAN, TIMEST, TIME, SELN, TIT, IROTX, IROTY, IROTZ, ROTX, ROTY, ROTZ)

CALL ODATAS (NV, VU, SCL, SCLR, RPLN, TRUEAN, TIMEST, TIME)

SUBROUTINE NAME: FFDATA

PURPOSE:

This subroutine will define parameters used in FFCAL if other than default values are used.

DEFINITIONS:

Variable Name	Default Values
FFACC - orientation accuracy factor	0.05
FFACCS - shadowing accuracy factor	0.1
FFNOSH - shadowing override flag (4HNOSH, 4HSHAD)	4HSHAD
FFRATL - distance/area ratio factor	15.0
FFMIN - eliminate small form factors	1.E-6
FFPRNT - flag to print form factors (5HPRINT, 2HNO)	5HPRINT
FFPNCH - flag to punch form factors (3HPUN, 4HTAPE*, 2HNO)	3HPUN

RESTRICTIONS:

None

NOTES: Example: CALL FFDATA (0., 0., 4HNOSH, 0, 1.E-3,5HPRINT, 0)
Results in no shadowing computations, form factors below 0.001 ignored, form factors printed, default values used elsewhere.
If value passed is zero, default value assumed.

CALLING SEQUENCE:

CALL FFDATA (FFACC, FFACCS, FFNOSH, FFRATL, FFMIN, FFPRNT, FFPNCH)

*Writes to tape RTO.

SUBROUTINE NAME: ORBIT1

PURPOSE:

This subroutine defines spacecraft orbits in terms of classic orbital mechanics parameters and a celestial coordinate system.

DEFINITIONS:

Variable Names			Default Values
PNAME	-	name of orbit-centered body	None
ALAN	-	longitude of ascending node	None
APER	-	argument of perifocus	None
OINC	-	orbit inclination	None
TIMEST	-	time of periapsis passage, hours	0.0
HP	-	altitude at periapsis	None
HA	-	altitude at apoapsis	None
ECC	-	orbit eccentricity	None
SUNRA	-	right ascension of sun	None
SUNDEC	-	declination of sun	None
STRRA	-	right ascension of star	None
STRDEC	-	declination of star	None

RESTRICTIONS:

None

CALLING SEQUENCE:

CALL ORBIT1 (PNAME, ALAN, APER, OINC, TIMEST, HP, HA, SUNRA, SUNDEC, STRRA, STRDEC)

CALL ORBIT1 (PLANAM, ALAN, APER, OINC, TIMEST, HP, ECC, SUNRA, SUNDEC, STRRA, STRDEC)

NOTES:

PNAME options are as follows: 3HMER, 3HVEN, 3HEAR, 3HMOO, 3HMAR, 3HJUP, 3HSAT, 3HNEP, 3HURA, and 3HSUN. These names are used to key the following program variables:

- WDS - darkside infrared emissive power at planet surface
- PALB - planet albedo value (solar reflectance)
- PRAD - planet radius
- WSS - infrared emissive power at sbusolar point
- SOL - solar "constant" at planet-sun distance
- GRAV - planet gravitational constant at surface

Sixth argument is tested for magnitude. If ≤ 1.0, ECC is assumed. If > 1.0, HA assumed.

Execution of this subroutine defines the planetary shadow entry and exit points (ref Figure 4-7).

SUBROUTINE NAME: ORBIT2

PURPOSE:

This subroutine defines spacecraft orbits and sun/star position-orbit relationship in the orbit coordinate system.

DEFINITIONS:

Variable Names			Default Values
PNAME	-	name of orbit-centered body	None
CIGMA	-	clock angle - X_o axis to solar vector projection	None
BETA	-	cone angle - Z_o axis to solar vector	None
CIGMAS	-	clock angle - X_o axis to star vector projection	None
BETAS	-	cone angle - Z_o axis to star vector projection	None
TIMEST	-	time of periapsis passage, hours	0.0
HP	-	altitude of periapsis	None
HA	-	altitude of apoapsis	None
ECC	-	orbit eccentricity	None

RESTRICTIONS:

None

CALLING SEQUENCE:

CALL ORBIT2 (PNAME, CIGMA, BETA, CIGMAS, BETAS, TIMEST, HP, HA)

CALL ORBIT2 (PNAME, CIGMA, BETA, CIGMAS, BETAS, TIMEST, HP, ECC)

NOTES: See subroutine ORBIT1. This call not applicable to heliocentric orbits.

SUBROUTINE NAMES: DIDT1, DIDT1S

PURPOSE:

Calls to define direct irradiation shadowing and accuracy parameters and to compute heat source position vectors from true anomaly or time.

DEFINITIONS:

Variable Names		Default Values
DINOSH	- shadow/no shadow flag (Options: 4HNOSH, 4HSHAD)	4HSHAD (shadow calculations not bypassed)
DIACC	- element selection accuracy factor for node/planet form factors	0.25
DIACCS	- element selection accuracy factor for shadowing calculations	0.10
TRUEAN	- true anomaly	None
NSPFF	- step number reference to obtain node-planet form factors if desired	0 (new form factors computed)
TIMEPR	- time	None
DIPNCH	- flux punch flag (Options: 3HPUN, 4HTAPE*, 2HNO)	3HPUN

RESTRICTIONS:

Either TRUEAN or TIMEPR must be defined in call.

CALLING SEQUENCE:

CALL DIDT1 (DINOSH, DIACC, DIACCS, TRUEAN, NSPFF, TIMEPR, DIPNCH)

CALL DIDT1S (TRUEAN, NSPFF, TIMEPR, DIPNCH)

*Writes to tape RTO.

SUBROUTINE NAMES: DIDT2, DIDT2S

PURPOSE:

Calls to define direct irradiation shadowing and accuracy parameters and to compute heat source position vectors from look angles.

DEFINITIONS:

DINOSH, DIACC, DIACCS, NSPFF, DIPNCH } Reference DIDT1

SUNCL, SUNCO - look angles to sun (clock, cone)
PLCL, PLCO - look angles to planet (clock, cone)
TIMEPR - present time
ALT - spacecraft altitude

RESTRICTIONS:

These calls must be preceded by a call to ORBIT2 in order to define orbit-centered body.

CALLING SEQUENCE:

```
CALL DIDT2 (DINOSH, DIACC, DIACCS, NSPFF, SUNCL, SUNCO
PLCL, PLCO, TIMEPR, ALT, DIPNCH)

CALL DIDT2S (NSPFF, SUNCL, SUNCO, PLCL, PLCO, TIMEPR, ALT, DIPNCH)
```

SUBROUTINE NAME: SPIN

PURPOSE:

Subroutine to define spacecraft spin rate, spin axis, and time of beginning of spin.

DEFINITIONS:

Variable Names	Default Values
CLOCK - clock angle - CCS x-axis to spin axis projection	0.
CONE - cone angle - CSS z-axis to spin axis	0.
RATE - spin rotation rate, degrees/hour (positive clockwise as viewed along spin axis from origin)	0.
TRUANS - true anomaly where spin begins	0.
SPNTM - time corresponding to TRUANS	0.

RESTRICTIONS:

Must be called subsequent to orbit definition through subroutines ORBIT1 or ORBIT2.

NOTES:

a. The time at which spin begins may be defined either directly through SPNTM or through TRUANS. If SPNTM = 0, SPNTM is computed from TRUANS.

b. Spinning may be stopped only by a call to subroutine SPIN with RATE = 0. and spin stop time or true anomaly defined.

CALLING SEQUENCE:

CALL SPIN (CLOCK, CONE, RATE TRUANS, SPNTM)

SUBROUTINE NAME: ORIENT

PURPOSE:

To define spacecraft orientation relative to orbital heat sources.

DEFINITIONS:

Variable Names	Default Values
TYPE - orientation type	None
IROTX - order of rotation about x-axis	1
IROTY - order of rotation about y-axis	2
IROTZ - order of rotation about z-axis	3
ROTX - rotation about VCS x-axis to rotate VCS into CSS	0.
ROTY - rotation about VCS y-axis to rotate VCS into CSS	0.
ROTZ - rotation about VCS z-axis to rotate VCS into CSS	0.

RESTRICTIONS:

Not recommended for use with DIDT2, DIDT2S.

CALLING SEQUENCE:

CALL ORIENT (TYPE, IROTX, IROTY, IROTZ, ROTX, ROTY, ROTZ)

NOTES: TYPE options are as follows: 4HPLAN, 3HSUN, 4HSTAR, 4HTAPE.
Individual default values obtained by passing zero.

SUBROUTINE NAME: GBDATA

PURPOSE:

Defines parameters used by segment GBCAL in computing a grey-body factor matrix.

Variable Names	Default Value
IGBSFF - step no. reference for form factors	Assumes current step
GBWBND - waveband definition name (2HIR, 3HSOL, 4HBOTH)	None

RESTRICTIONS:

Current step geometry definition must match definition of FF step reference.

CALLING SEQUENCE:

CALL GBDATA (IGBSFF, GBWBND)

SUBROUTINE NAME: RKDATA

PURPOSE:

This subroutine defines parameters used in RKCAL for output of radiation conductors (RADKS).

Variable Names	Default Value
IRKNGB - step number reference for infrared grey-body factors.	Assumes current step
RKPNCH - punch/no punch flag. Options: 3HPUN, 2HNO	3HPUN
RKMIN - minimum value of $\mathcal{F}/\epsilon$ that will result in a valid RADK	0.0001
IRKCN - initial radiation conductor number	1
RKSP - flag for calculation of RADKS to space. Options: 5HSPACE, 2HNO	2HNO
IRKNSP - space node number	32767
SIGMA - Stefan-Boltzmann constant	1.713E-9
RKAMPF - area multiplying factor	1.0
RKTAPE - flag to write RADKS to BCD tape. Options: 4HTAPE, 2HNO	2HNO

NOTES: If not called prior to RKCAL execution, default values will be assumed. Individual default values obtained by passing zero arguments.

RESTRICTIONS:

Current geometry must agree with that of step IRKNGB.

CALLING SEQUENCE:

CALL RKDATA (IRKNGB, RKPNCH, RKMIN, IRKCN, RKSP, IRKNSP, SIGMA, RKAMPF, RKTAPE)

SUBROUTINE NAME: AQDATA

PURPOSE:

This subroutine defines parameters used in AQCAL for calculation of absorbed heats. Direct fluxes for AQ calculations are obtained from current step data storage.

Variable Names	Default Values
IAQGBS - step no. reference for solar grey-body matrix	Last previous step that
IAQGBI - step no. reference for IR grey-body matrix	computed grey-body factors.
RSOLAR - multiplying factor for solar absorbed heat	1.0
RALB - multiplying factor for albedo absorbed heat	1.0
RPLAN - multiplying factor for planetary absorbed heat	1.0

RESTRICTIONS:

Must be called subsequent to a DICAL execution within same step.

NOTES: If not called prior to a AQCAL execution (within same step), default values assumed. Individual default values obtained by passing zero arguments.

CALLING SEQUENCE:

CALL AQDATA (IAQGBI, IAQGBS, RSOLAR, RALB, RPLAN)

SUBROUTINE NAME: STFAQ

PURPOSE:

This subroutine stuffs known values of direct flux and absorbed heat from a previously executed step into current step data storage. It also defines time of current step, either directly or from true anomaly.

VARIABLE NAMES:	Default Value
TRUEAN - true anomaly, degrees	None
TIMEPR - current time, hours	None
NSTP - step number reference for known DI and AQ values	None

RESTRICTIONS:

Current geometry must agree with that of NSTP.

CALLING SEQUENCE:

```
CALL STFAQ (TRUEAN, TIMEPR, NSTP)
```

NOTE:

If TRUEAN.GT.0., time is computed from TRUEAN; otherwise TIMEPR is passed directly to current step data storage.

SUBROUTINE NAME: QODATA

PURPOSE:

This subroutine used to define the absorbed heat output format to be obtained from the subsequent QOCAL execution.

DEFINITIONS:

Variable Names		Default Value
NSARRY	- array of step numbers where absorbed Q data is stored. Options: array name, 3HALL	3HALL
NTMARY	- thermal analyzer time array number (Q arrays numbered consecutively from NTMARY + 1)	1
QOTAPE	- BCD tape output flag. Options: 4HTAPE, 2HNO	2HNO
QOPNCH	- punch output flag. Options: 3HPUN, 2HNO	3HPUN*
QOAMPF	- area multiplication factor	1.0
QOFMPF	- energy multiplication factor	1.0
QOTMPF	- time multiplication factor	1.0
QOTYPE	- type of output flag. Options: 3HTAB for Q vs time tables, 2HAV for orbital average Q data, 4HBOTH for both	4HBOTH
IQOCOR	- step number of correspondence table	Current step

RESTRICTIONS:

Current geometry definition must agree with geometry of all steps in NSARRY.

NOTES:

Sort is made to obtain monotonically increasing time array. Trapezoidal-rule average made for orbital average heat tables.

QOAMPF applies only to areas on thermal analyzer subroutine call output.
QOFMPF applies only to heat flux array output.
QOTMPF applies to time array output and to value of period on subroutine call output.

CALLING SEQUENCE:

CALL QODATA (NSARRY, NTMARY, QOTAPE, QOPNCH, QOAMPF, QOFMPF, QOTMPF, QOTYPE, IQOCOR)

*Defaults to 2HNO at JSC/Univac.

SUBROUTINE NAME: SFDATA

PURPOSE:

This subroutine is used to define configuration names on shadow-factor input (SHADI) and output (SHADO) tapes.

DEFINITIONS:

Variable Names	Default
NAMEI - name of configuration identifying desired file on SHADI tape (Hollerith)	None
NAMEO - configuration name stored on SHADO tape file being generated (Hollerith)	a. STEPN[1] b. NAMEI[2]

RESTRICTIONS:

None

NOTES:

1. If SFDATA is not called prior to an SFCAL execution, NAMEO defaults to the current step number.

2. A call with NAMEO undefined; i.e., CALL SFDATA(NAME,0), equates NAMEO and NAMEI.

CALLING SEQUENCE:

CALL SFDATA (NAMEI, NAMEO)

SUBROUTINE NAME: PLDATA

PURPOSE:

This subroutine defines parameters necessary to execute the data plotter.

DEFINITIONS:

Variable Name	Description	Options	Default
IPLUNT	Plot data flag (a composite Hollerith word)	Letter 1: A - Absorbed I - Incident Letter 2: F - Fluxes R - Rates Letters 3, 4, 5, & 6 (as required) S - Solar A - Albedo P - Planetary T - Total ALL - All	None
IPLSN	Identifies steps to be plotted	A. 3HALL B. Array of step numbers	3HALL
IPLNA	Identifies nodes to be plotted	A. 3HALL B. Array of node numbers	3HALL
PLCRVF	Flag for curve-fitting	3HYES, 2HNO	3HYES
PLLABX	Plot label X	Array name (array length 28 characters max.)	Blanks
PLLABY	Plot label Y	Array name (array length 28 characters max.)	Blanks
PLTIT1	Plot label title line 1	Array name (array length 58 characters max.)	Blanks
PLTIT2	Plot label title line 2	Array name (array length 70 characters max.)	Blanks
PLXMPF	X-axis multiplying factor	Real no.	1.0
PLYMPF	Y-axis multiplying factor	Real no.	1.0

RESTRICTIONS: None

NOTE:

a. Examples of IPLUNT:

3HARP; plots absorbed rates, planetary
5HIFALL; plots all incident fluxes
5HAFSAP; plots solar, albedo and planetary absorbed fluxes

b. Any set of dependent- and independent-variable data pairs may be plotted if IPLUNT = 1 and the data are written to disc/drum unit 1 in advance (reference Section 5.1.3)

CALLING SEQUENCE:

CALL PLDATA (IPLUNT, IPLSN, IPLNA, PLCRUF, PLLABX, PLLABY, PLTIT1, PLTIT2, PLXMPF, PLYMPF)

SUBROUTINE NAME: DICOMP

PURPOSE:

This subroutine is used to define logic used in subsequent DICAL execution.

DEFINITIONS:

Variable Names	Options	Default
ISOLFL - solar flux	a. 4HZERO - zeros out solar flux for all nodes b. 0 (integer) - results in computation of solar fluxes c. STEPN (integer step number) - stuffs solar fluxes from STEPN into current step storage	0 (compute)
IALBFL - albedo flux compute/stuff flag	Same as for ISOLFL	0
IPLAFL - planetary flux compute/stuff flag	Same as for ISOLFL	0

RESTRICTIONS:

None

NOTES:

1. Compute/stuff flags are overridden by the planet shadow. Nonzero solar or albedo fluxes will never be stuffed into storage for a point within the planet shadow.

2. Failure to call DICOMP prior to a DICAL execution results in default values for all three flags.

CALLING SEQUENCE:

CALL DICOMP (ISOLFL, IALBFL, IPLAFL)

SUBROUTINE NAMES: DITTP, DITTPS

PURPOSE:

These subroutines read data from a trajectory tape and define spacecraft/heat source parameters through subroutine DIDT2. DITTP is called initially in order to define planetary parameters and position the tape for subsequent time points. DITTPS is used to update time and attitude/position data.

DEFINITIONS:

Variable Names			Options
TIME	-	mission time	Real no.
ITYPE	-	identifier for special event record	Integer
PLANAM	-	name of orbit-centered planet (if applicable) (ref ORBIT1)	Hollerith
IDWDN	-	number of word FIDEN in identification record	Integer
FIDEN	-	file identification word	Hollerith
NTIM	-	number of time word in information record	Integer
NTYPE	-	number of word ITYPE in information record	Integer
NCLPL	-	number of word containing clock angle-to-planet vector	Integer
NCOPL	-	number of word containing cone angle-to-planet vector	Integer
NCLS	-	number of word containing clock angle-to-sun vector	Integer
NCOS	-	number of word containing cone angle-to-sun vector	Integer
NRAD	-	number of word containing planet center-to-spacecraft distance	Integer
NWOR	-	number of words in tape record	Integer
ALTMF	-	multiplying factor to convert units of NRAD word to feet	Real no.

Variable Names		Options
IBOD -	one-body/two-body flag 0 - One-body tape 1 - Two-body tape, use body 1 2 - Two-body tape, use body 2	integer
DIPNCH -	Punch/no punch flag for orbital flux output	Hollerith

RESTRICTIONS:

a. Calls to DITTPS to update time and type can be made only after a call to DITTP is in effect.

b. The TIME argument in DITTPS calls must be greater than any previously defined TIME argument until the tape is repositioned through a call to DITTP.

CALLING SEQUENCES:

CALL DITTP (TIME, ITYPE, PLANAM, IDWDN, FIDEN, NTIM, NTYPE, NCLPL, NCOPL, NCLS, NCOS, NRAD, NWOR, ALTMF, IBOD, DIPINCH)

CALL DITTPS (TIME, ITYPE)

SUBROUTINE NAME: MODAR

PURPOSE:

This subroutine changes the area of a designated node, or changes the area of all currently active nodes by use of a multiplier.

Argument Name	Description	Options	Default
ND	Node number designator	a. Any active node number (integer)	None
		b. 3HALL	
AR	Desired value for area	a. Floating-point data value	None
		b. Area multiplier[1] (3HALL option only)	

NOTE:

1. When ND = 3HALL, all active node areas are modified according to: AREA = AREA*AR.

RESTRICTION:

Call not valid prior to geometry definition through calls to BUILDC and ADD.

CALLING SEQUENCE:

```
CALL MODAR (ND, AR)
```

SUBROUTINE NAME: MODPR

PURPOSE:

This subroutine modifies the diffuse infrared emissivity and/or the diffuse solar absorptivity of a designated node.

Argument Name	Description	Options	Default
ND	Node number designator	Any active node number	None
ALPHA	Diffuse solar absorptivity	a. $0. \leq DV \leq 1.$ b. $DV < 0.$	None
EMISS	Diffuse IR emissivity	a. $0. \leq DV \leq 1.$ b. $DV < 0.$	None

NOTE:

1. If ALPHA$<$0. or EMISS$<$0., current values are not changed.

RESTRICTION:

Call not valid prior to geometry definition through calls to BUILDC and ADD.

CALLING SEQUENCE:

```
CALL MODPR (ND, ALPHA, EMISS)
```

SUBROUTINE NAME: MODTR

PURPOSE:

This subroutine modifies the solar and/or infrared transmissivity of a designated surface.

Argument Name	Description	Options	Default
ISR	Surface number designator	Any active surface number	None
TRANS	Solar transmissivity	a. $0. \leq DV \leq 1.$	None
		b. $DV < 0.$[1]	
TRANI	IR transmissivity	a. $0. \leq DV \leq 1.$	None
		b. $DV < 0.$[1]	

NOTES:

1. If TRANI$<0.$ or TRANS$<0.$, current values are not changed.

2. Transmissivity changes affect entire surface.

RESTRICTION:

Call not valid prior to geometry definition through calls to BUILDC and ADD.

CALLING SEQUENCE:

CALL MODTR (ISR, TRANS, TRANI)

SUBROUTINE NAME: MODPRS

PURPOSE:

This subroutine modifies the solar and/or infrared specular reflectivity of a designated node.

Argument Name	Description	Options	Default
ND	Node number designator	Any active node number	None
SPRS	Specular reflectivity, solar	a. $0. \leq DV \leq 1.$ b. $DV < 0.$[1]	None
SPRI	Specular reflectivity, infrared	a. $0. \leq DV \leq 1.$ b. $DV < 0.$[1]	None

NOTES:

1. If SPRI$<$0. or SPRS$<$0., current values are not changed.

RESTRICTIONS:

1. This call applicable only to nodes defined as specular reflectors in the surface data block.
2. Call not valid prior to geometry definition through calls to BUILDC and ADD.

CALLING SEQUENCE:

```
CALL MODPRS (ND, SPRS, SPRI)
```

SUBROUTINE NAME: MODSHD

PURPOSE:

This subroutine modifies the SHADE/BSHADE flags for a designated surface.

Argument Name	Description	Options	Default
ISR	Surface number designator	Any active surface number	None
SHADE	"Can shade" flag	FF, DI, BOTH, NO, 0[1]	None
BSHADE	"Can be shaded" flag	FF, DI, BOTH, NO, 0[1]	None

NOTES:

1. If SHADE or BSHADE data values are zero, their values are not changed.

2. Shade flag changes affect entire surface.

RESTRICTIONS:

Call not valid prior to geometry definition through calls to BUILDC and ADD.

Call not applicable to shadower-only surfaces.

CALLING SEQUENCE:

CALL MODSHD (ISR, SHADE, BSHADE)

SUBROUTINE NAME: GBAPRX

PURPOSE:

This subroutine calculates grey-body radiant interchange factors using an approximate relationship and stores the results in data storage.

Argument Name	Description	Options	Default
IGBSFF	Step number for form factors	Integer	Current step no.
GBWBND	Waveband definition name	2HIR, 3HSOL 4HBOTH	4HBOTH

NOTE:

Input zero for default action.

RESTRICTIONS:

None.

CALLING SEQUENCE:

CALL GBAPRX (IGBSFF, GBWBND)

SUBROUTINE NAME: RCDATA

PURPOSE:

This is a user-called subroutine that defines the parameters used in RCCAL for the condensation and output of radiation conductors (RADKS).

VARIABLE DESCRIPTIONS AND DEFAULT VALUES:

Variable		Description	Default Value
IRCNGB	-	Step number reference for infrared grey-body factors	Assumes current step
RCPNCH	-	Punch/no punch flag. Options: 3HPUN, 2HNO	3HPUN
RCMIN	-	Minimum value of $\mathscr{F}/\varepsilon$ that will result in a valid RADK	0.0001
IRCCN	-	Initial radiation conductor number	1
RCSP	-	Flag for calculation of RADKS to space. Options: 5HSPACE, 2HNO	2HNO
IRCNSP	-	Space node number	32767
SIGMA	-	Stefan-Boltzmann constant	1.713E-9
RCAMPF	-	Area multiplying factor	1.0
RCTAPE	-	Flag to write RADKS to BCD tape. Options: 4HTAPE, 2HNO	2HNO
RFRAC	-	Significant radiation fraction: radiation conductors of a node to be left intact divided by the sum of the node conductors	None
NERN	-	Effective radiation node (ERN) number	None
IPRIME	-	Array name for array of primary MESS node numbers and special node numbers	None
ISECND	-	Array name for array of secondary MESS node numbers	None

RESTRICTIONS:

RCDATA must be called prior to RCCAL execution since all of the variables are not defaulted.

Current model geometry must agree with that of step IRCNGB.

IPRIME and ISECND arrays must be input in the array data block to specify MESS node pairs and special nodes. IPRIME contains a list of all primary MESS nodes and all special nodes in that order. ISECND contains a list of all secondary MESS nodes in IPRIME.

CALLING SEQUENCE:

CALL RCDATA (IRCNGB, RCPNCH, RCMIN, IRCCN, RCSP, IRCNSP, SIGMA, RCAMPF, RCTAPE, RFRAC, NERN, IPRIME, ISECND)

SUBROUTINE NAME: ADSURF

PURPOSE:

This subroutine functions to add an adiabatic "closure" surface to the problem geometry and adds the pertinent form factors to the form factor matrix.

Parameter	Description
BCSN	Name of a block coordinate system containing the "closure" surface
FFSN	Step number under which desired form factor matrix is stored

RESTRICTIONS:

Block coordinate system BCSN must appear in the surface data block with one and only one surface. This surface may be of any type and dimension, but must be completely defined, including the surface properties desired for the "closure" surface.

Subroutine ADSURF may not be called within or prior to step FFSN.

The geometry in effect when ADSURF is called must include that in effect for step FFSN, plus BCSN.

CALLING SEQUENCE:

```
CALL ADSURF (BCSN, FFSN)
```

SUBROUTINE NAME: FFNDP

PURPOSE:

This subroutine is used to obtain a node number array, punched on cards in format used in form factor and flux data block.

RESTRICTIONS:

None

CALLING SEQUENCE:

CALL FFNDP

SUBROUTINE NAME: TAPELS

PURPOSE:

This subroutine is used to obtain a listing of data on the BCDOU tape.

VARIABLE NAME:

N is the number of files to be listed.

RESTRICTION:

Call valid only after BCD output data has been written to BCDOU tape from RKCAL, RCCAL, and/or QOCAL executions.

NOTE:

If N is greater than the number of files on BCDOU and the user has not put an end of file on BCDOU from the operations data block, the run will abort.

CALLING SEQUENCE:

```
CALL TAPELS (N)
```

SUBROUTINE NAME: NDUPCK

PURPOSE:

This subroutine is used to determine if any node number duplication exists in the currently effective problem geometry.

RESTRICTIONS:

None

NOTE:

Should be called just after the BUILDC and ADD series that defines a geometry with doubtful node numbers. If duplication is encountered, problem will abort with appropriate message(s).

CALLING SEQUENCE:

CALL NDUPCK

APPENDIX E

SEGMENT DESCRIPTIONS

SEGMENT NAME: NPLOT

PURPOSE:

This segment generates pictorial plots of nodal surfaces.

RESTRICTIONS:

None

CALLING SEQUENCE: L NPLOT

SEGMENT NAME: OPLOT

PURPOSE:

This segment generates pictorial plots of the spacecraft in orbit.

RESTRICTIONS:

None

CALLING SEQUENCE: L OPLOT

SEGMENT NAME: PLOT

PURPOSE:

This segment generates function vs time plots of absorbed and incident heat rates and fluxes. When used in conjunction with operations block FORTRAN that writes data to a plot data unit, this segment provides general x vs y plot capability.

RESTRICTIONS:

Reference Subroutine PLDATA

CALLING SEQUENCE: L PLOT

SEGMENT NAME: FFCAL

PURPOSE:

This segment calculates all form factors for the active configuration defined by previous calls to BUILDC and ADD.

CALLING SEQUENCE: L FFCAL

SEGMENT NAME: DICAL

PURPOSE:

This segment computes solar, planetary, and albedo irradiation incident on spacecraft nodes.

RESTRICTIONS:

Call valid only after previous calls have been made to define spacecraft geometry, location in space, characteristics and distances of heat source bodies, and computation accuracy parameters.

CALLING SEQUENCE: L DICAL

SEGMENT NAME: SFCAL

PURPOSE:

a. Segment computes analytically and stores on tape tables of internode blockage (shadow) factors for use in direct irradiation calculations.

b. When a complete shadow factor tape supplied, segment is executed in order to pass shadow tables into program storage and initialize DICAL to compute irradiation using shadow tables.

RESTRICTIONS:

None

CALLING SEQUENCE: L SFCAL

SEGMENT NAME: RKCAL

PURPOSE:

This segment computes radiation conductor values and punches (at user's option) output data in thermal analyzer format. Output card images are printed.

RESTRICTIONS:

Call valid after spacecraft geometry is defined and matching form factor matrix is computed.

CALLING SEQUENCE: L RKCAL

SEGMENT NAME: GBCAL

PURPOSE:

Segment computes and stores grey-body factor matrix.

RESTRICTIONS:

Same as RKCAL

CALLING SEQUENCE: L GBCAL

SEGMENT NAME: RCCAL

PURPOSE:

This segment computes radiation conductors, simplifies and condenses these conductors using the ERN and MSS techniques, and provides output in punched card and/or BCD tape form.

RESTRICTIONS:

Same as RKCAL

CALLING SEQUENCE: L RCCAL

SEGMENT NAME: AQCAL

PURPOSE:

This segment computes absorbed heat rates in two wavebands, accounting for diffuse reflection.

RESTRICTIONS:

Appropriate direct irradiation, grey-body factors, and surface properties must be in system storage.

CALLING SEQUENCE: L AQCAL

SEGMENT NAME: QOCAL

PURPOSE:

This segment accesses absorbed flux data and generates orbital average and absorbed flux vs time arrays. Arrays are output in thermal analyzer format on cards or BCD tape.

RESTRICTIONS:

None

CALLING SEQUENCE: L QOCAL

APPENDIX F

RADIATION CONDENSER SEGMENT THEORY.

A. BASIC CONCEPTS

The Multiple Enclosure Simplification Shield (MESS) technique and the Effective Radiation Node (ERN) technique are independent and can be discussed separately. Consider an N-node radiative enclosure that forms a section of a complex thermal model. The temperature of node i is a function of thermal radiation coupling and the applied heat load, Q_i (assume that heat loads resulting from conduction and convection are included in Q_i). The steady-state temperature of node i is then given by

$$T_i = \left[\left(\sum_{j=1}^{N} \sigma A_i F^*_{ij} T_j^4 + Q_i \right) \Big/ \sum_{j=1}^{N} \sigma A_i F^*_{ij} \right]^{1/4} \qquad (1)$$

B. ERN TECHNIQUE

In applying the ERN technique, the enclosure radiation conductors for the ith node are divided into P_i primary and $N-P_i$ secondary couplings. The summation term in the numerator of Equation (1) can then be written as follows:

$$\sum_{j=1}^{N} \sigma A_i F_{ij} T_j^4 = \sum_{k=1}^{P_i} \sigma A_i F_{ik} T_k^4 + \sum_{\ell=P_i+1}^{N} A_i F_{i\ell} T_\ell^4 \qquad (2)$$

The number of radiation conductors can be reduced by arranging the conductors in decreasing order of the conductor value ($A_i F_{ij}$) and replacing the secondary coupling summation of Equation (2) with a single conductor coupled to an ERN. That is,

---

*In Appendix F, the letter F shall denote the grey-body factor, $\mathcal{F}$.

$$\sum_{\ell=P_i+1}^{N} \sigma A_i F_{i\ell} T_\ell^4 = \left[\sum_{\ell=P_i+1}^{N} \sigma A_i F_{i\ell}\right] T_{ERN}^4 \qquad (3)$$

The ERN temperature is calculated by the thermal analyzer program as a steady-state node temperature based on a fourth-power, conductor-weighted average of the enclosure node temperatures using the secondary conductors.

$$T_{ERN} = \left[\sum_{i=1}^{N} \sum_{\ell=P_i+1}^{N} \sigma A_i F_{i\ell} T_i^4 \Big/ \sum_{i=1}^{N} \sum_{\ell=P_i+1}^{N} \sigma A_i F_{i\ell}\right]^{1/4} \qquad (4)$$

Using the relationships of Equations (2) and (3), the approximate ith node temperature can be written from Equation (1) as a function of the ERN temperature.

$$T_i' = \left\{\left[\sum_{k=1}^{P_i} \sigma A_i F_{ik} T_k^4 + \left(\sum_{\ell=P_{i+1}}^{N} \sigma A_i F_{i\ell}\right) T_{ERN}^4 + Q_i\right] \Big/ \sum_{j=1}^{N} \sigma A_i F_{ij}\right\}^{1/4} \qquad (5)$$

C. APPLICATION OF THE ERN TECHNIQUE

The significant radiation fraction defined by the relationship

$$RFRAC = \sum_{k=1}^{P_i} \sigma A_i F_{ik} \Big/ \sum_{j=1}^{N} \sigma A_i F_{ij} = \sum_{k=1}^{P_i} F_{ik} \Big/ \epsilon_i \qquad (6)$$

is specified by the user. The number of primary conductors, P_i, is determined by summing conductor values for a given node until the sum is greater than the fraction RFRAC of the sum of all conductors to the node. That is,

$$\sum_{k=1}^{P_i} \sigma A_i F_{ik} > \text{RFRAC}^* \sum_{j=1}^{N} \sigma A_i F_{ij} \qquad (7)$$

All primary and reverse direction conductors are flagged to be used intact. The secondary conductors for each node are summed to determine the conductor value for the node-to-ERN coupling.

Since the error in the approximate temperature is a function of the enclosure temperature band, the ERN technique results can be improved if nodes that deviate significantly from the average temperature of the enclosure are not coupled to the ERN. These analyst-defined nodes are referred to as special nodes.

The percentage reduction in enclosure conductors and subsequent network error as a result of applying the ERN technique are controlled by the analyst's selection of an RFRAC value consistent with the known accuracy of problem parameters (enclosure geometry, surface optical properties, etc).

Experience has shown that the greatest percentage reduction in conductors results for enclosures with more than 75 nodes, significant shadowing, and low-emittance surfaces. An RFRAC value of 0.7 has been found to result in a significant reduction in conductors with acceptable error for typical radiation enclosures.

D. MESS TECHNIQUE

The MESS technique provides the analyst with a means of dividing a radiation enclosure into an arbitrary number of subenclosures.

MESS node pairs are defined by the analyst at the interface between subenclosures as two planar surfaces with the property of absorbing and emitting all energy incident upon them (black surfaces). Consider an N-node subenclosure, n, as shown in Figure F-1, where the subscripts r and r' refer to the MESS node pair of the nth and jth subenclosures, respectively. Temperatures in n are affected by $T_{MESS_{r'}}$, which represents the average thermal effect of the j subenclosure nodes on the nodes of n. The primary conductors of Equation (2) include conductors to MESS nodes. For a general subenclosure, n, with R_n interface MESS nodes, the primary radiation coupling summation for node i is

$$\sum_{k=1}^{P_i} \sigma A_i F_{ik} T_k^4 = \sum_{r=1}^{R_n} \sigma A_i F_{ir} T_{MESS_{r'}}^4 + \sum_{\ell=R_n+1}^{P_i} \sigma A_i F_{i\ell} T_\ell^4 \qquad (8)$$

An energy balance on MESS node r' gives

$$T_{MESS_{r'}} = \left[\left(\sum_{\substack{m=1 \\ m \neq r'}}^{R_j} \sigma A_{r'} F_{r'm} T_{MESS_{m'}}^4 + \sum_{k=R_j+1}^{P_{r'}} \sigma A_{r'} F_{r'k} T_k^4 + \sigma A_{r'} F_{r'r} T_{MESS_r}^4 \right) \Big/ \left(\sum_{\substack{m=1 \\ m \neq r'}}^{R_j} \sigma A_{r'} F_{r'm} + \sum_{k=R_j+1}^{P_{r'}} \sigma A_{r'} F_{r'k} + \sigma A_{r'} F_{r'r} \right) \right]^{1/4} \qquad (9)$$

$F_{r'r}$ represents the reflections between n and j due to nonblack subenclosure surfaces and is obtained from the radiation interchange matrix for each subenclosure.

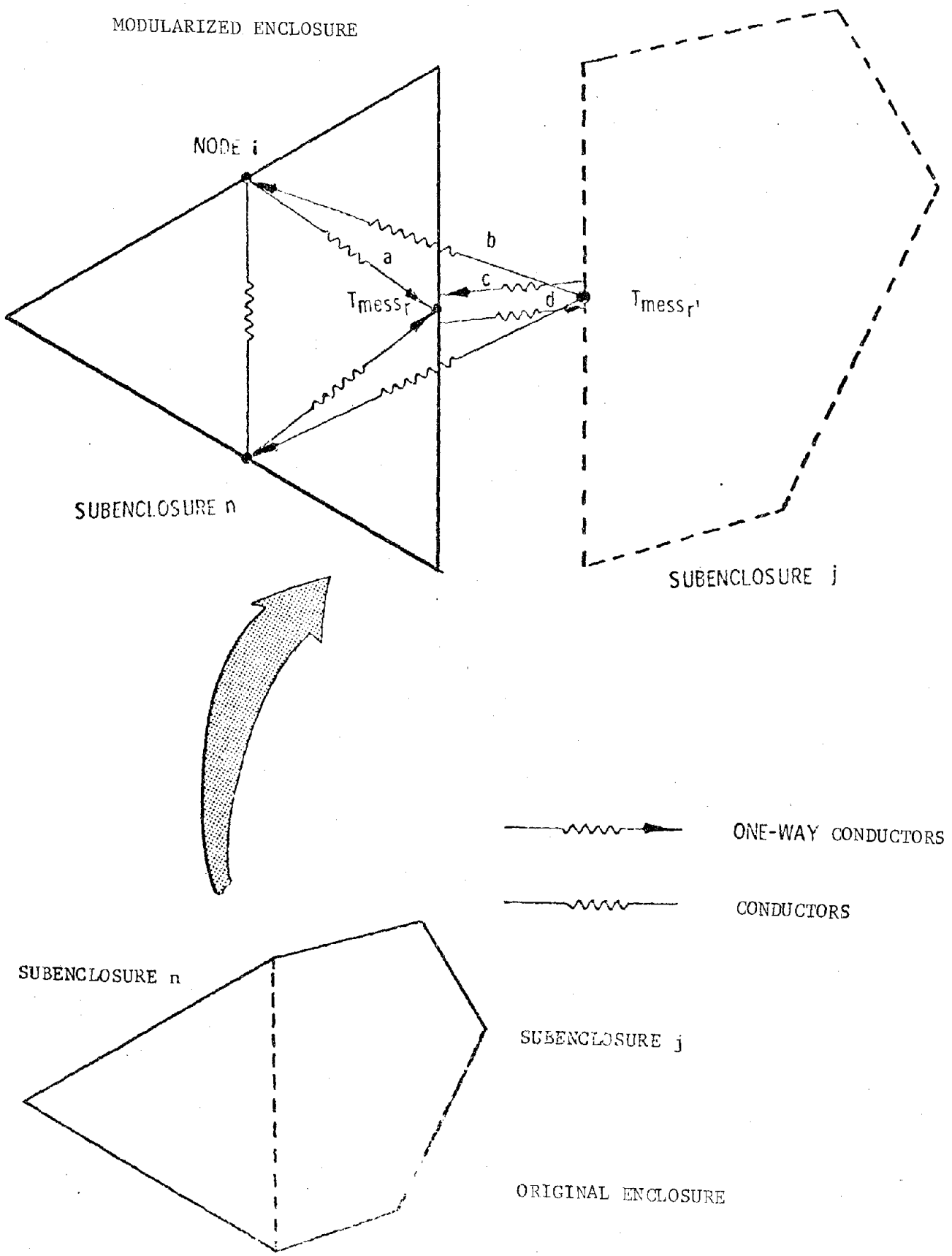


Figure F-1 MESS Technique One-Way Conductors

The approximate temperature of the ith node is obtained from Equation (5), using Equation (8), as

$$T_i' = \left\{ \left[\sum_{r=1}^{R_n} \sigma A_i F_{ir} T_{MESS_{r'}}^4 + \sum_{\ell=R_n+1}^{P_i} \sigma A_i F_{i\ell} T_\ell^4 + \left(\sum_{h=P_i+1}^{N} \sigma A_i F_{ih} \right) T_{ERN}^4 + Q_i \right] \Big/ \left(\sum_{r=1}^{R_n} \sigma A_i F_{ir} + \sum_{\ell=R_n+1}^{P_i} \sigma A_i F_{i\ell} + \sum_{h=P_i+1}^{N} \sigma A_i F_{ih} \right) \right\}^{\frac{1}{4}} \quad (10)$$

The error in T_i' is a complex function of the percentage of ERN secondary conductors, temperature band of the subenclosure nodes, and the number of subenclosures. In a variety of problems studied, the error has been found to be negligible.

E. APPLICATION OF THE MESS TECHNIQUE

Generation of MESS one-way conductors from the subenclosure radiant interchange matrix requires that the analyst specify the interface MESS node pairs. As node conductors are generated, MESS nodes are flagged and appropriate one-way conductors are generated for use in the thermal analyzer program.

The location of MESS node pairs in an enclosure is influenced by:

a. the number of subenclosure surfaces;

b. geometric considerations;

c. expected thermal gradients;

d. the number of analysts available to work on the enclosure.

Optimum reduction in form factors and conductors occurs in large enclosures divided such that the subenclosures contain approximately equal numbers of nodes. For enclosures divided into two approximately equal subenclosures, up to 50% reduction in the number of form factors and conductors can be expected.

F. ERN/MESS APPLICATION

The ERN and MESS techniques can be applied separately or simultaneously as the particular problem dictates. When they are applied simultaneously, an ERN is defined for each subenclosure and the MESS nodes are considered to be special nodes; that is, MESS nodes are not coupled to the ERN.

G. SUMMARY

The ERN/MESS technique reduces the number of form factors and radiation conductors necessary for enclosure radiation analysis and extends the analysis to include enclosures of arbitrary complexity. The use of the ERN/MESS technique can result in significant savings in time, both for the analyst and the computer.

APPENDIX G

TAPE NAME DESIGNATIONS

TAPE NAME	AVAILABILITY**	DESCRIPTION
NOUT	PP/P	Print Output File
DI	PP	Preprocessor Data Input File
RIO	PP	Random Access Data File
EDITI	PP	EDITI Tape
EDITO	PP	EDITO Tape
CMERG	PP	CMERGE Tape
EMERG	PP	EMERGE Tape
RSTRTI	PP	Permanent Restart Input Tape*
RSTRTO	PP/P	Permanent Restart Output Tape*
PNCH	PP	Punch Output File
SC1	PP	Scratch File
SC2	PP	Scratch File
SC3	PP	Scratch File
CMPL	PP	Operations Data Compile File
SQNTL	PP/P	Sequential Data File
DIR	PP/P	Direct Irradiation Restart File
FFR	PP/P	Form Factor Restart File
GBIRR	PP/P	Correspondence Data Input File
DLSR	PP/P	Planet Form Factor Save File
SHADI	PP	Shadow Factor Read Tape
SHADO	PP/P	Shadow Factor Write Tape
RTI	PP	Temporary Restart Input Tape*
RTO	PP/P	Temporary Restart Output Tape*
BCDOU	PP/P	BCD Data Output Tape

TRAJ	PP/P	Trajectory Tape
USER1	P	User File
USER2	P	User File
GBIRR	PP/P	Correspondence Data File
TQR	PP/P	Total Heat Rates, Restart File*
FF	PP/P	Form Factor Data Storage File
GBIR	P	Infrared Grey-Body Storage File
GBSO	P	Solar Grey-Body Storage File
PLS	P	Planetary Form Factor Save File
TQ	P	Total Heat Rates Storage
SCR1	P	Scratch File
SCR2	P	Scratch File
SCR3	P	Scratch File
RAN	P	Random I/O Data File (Equivalent to NRIO)
PUN	P	Punch Output File
DI	P.	Direct Irradiation Data Storage File

**PP: Preprocessor; P: Processor

*Not Currently Used.

APPENDIX I

DEVELOPMENT OF EQUATIONS FOR DIFFUSE-PLUS-SPECULAR RADIATION ANALYSIS

I. ANALYTICAL DEVELOPMENT

The assumptions and groundrules and the development of the mathematical equations for diffuse-plus-specular radiation analysis are presented in this section.

1. Assumptions and Groundrules

The following assumptions and groundrules were used in the analytical development presented herein for diffuse-plus-specular radiation analysis techniques.

a. All surfaces are considered to be semi-gray (accounts for absorption and reflection, but no emission in the ultraviolet portion of the spectrum; accounts for absorption and reflection as well as emission in the infrared portion of the spectrum).

b. Equations are developed for use in analyzing radiation enclosures consisting of diffuse, specular, and/or diffuse-plus-specular surfaces using an imaging technique.

c. All surfaces are considered to emit diffusely and to reflect with diffuse and specular components such that the relationship

$$\epsilon_i + \rho_i^d + \rho_i^s + \tau_i = 1.$$

is satisfied.

d. All surfaces with specular components of reflectance are restricted to planar surfaces to simplify imaging.

e. Only first-order images are considered (that is, no images of images or images in images are generated).

2. Development of Equations

The development of equations for radiation interchange factors follows the same procedure for both the infrared and ultraviolet portions of the spectrum with the only differences being in notation. Therefore, only those equations applicable to the infrared portion of the spectrum are developed here.

Consider a radiation enclosure consisting of N surfaces. The net heat flux from any one of these surfaces can be represented by

$$q_{i,net} = \sigma \sum_{j=1}^{N} \mathcal{J}_{ij} \left(T_i^4 - T_j^4\right) \qquad (1)$$

where $\mathcal{J}_{ij}$ is the radiation interchange factor that couples surface i to surface j.

The method of approach that is applied here in the development of radiation interchange factor ($\mathcal{J}_{ij}$) equations is an extension of the method set forth by Gebhart for purely diffuse enclosures (references 1, 2 and 3. The special utility in this formulation is that it yields coefficients which represent the fraction of energy emitted by a surface that is absorbed by another surface after reaching the absorbing surface by all possible paths.

Considering first-order images only, the general equation for the Gebhart-type absorption factors for a diffuse-plus-specular enclosure can be written

$$\beta_{ij} = \epsilon_j F_{ij} + \epsilon_j \sum_{k=1}^{N} \rho_k^s F_{ij(k)} + \sum_{m=1}^{N} \rho_m^d F_{im} \beta_{mj} + \sum_{k=1}^{N} \sum_{m=1}^{N} \rho_m^d \rho_k^s F_{im(k)} \beta_{mj}; \quad i=1, 2,\ldots,N; \; j=1,2,\ldots N \qquad (2)$$

By means of a term by term examination, equation (2) can be interpreted as follows:

The fraction of the energy leaving surface i that is finally absorbed by surface j equals the sum of

a. the energy that goes directly from surface i to surface j and is absorbed,

b. the energy that goes from surface i to surface j by all possible first-order specular reflections and is absorbed,

c. that fraction of the energy that goes directly from surface i to each of the surfaces in the enclosure, finally arrives at surface j by all possible paths, and is absorbed, and

d. that fraction of the energy that goes from surface i to each of the surfaces in the enclosure by all possible first-order specular reflections, thence to surface j by all possible paths, and is absorbed.

Rearranging the terms in equation (2) yields

$$\beta_{ij} = \epsilon_j \left[F_{ij} + \sum_{k=1}^{N} \rho_k^s \; F_{ij(k)}\right] + \sum_{m=1}^{N} \rho_m^d \left[F_{im} + \sum_{k=1}^{N} \rho_k^s \; F_{im(k)}\right] \beta_{mj} \; ;$$

$$i = 1,2,\ldots, N \quad , \quad j = 1,2,\ldots,N \qquad (3)$$

Equation (3) can be further simplified by defining an "image factor" (ϕ_{ij}) as that fraction of the energy that leaves surface i and arrives at surface j both directly and by all possible first-order specular reflections, such that

$$\phi_{ij} = F_{ij} + \sum_{k=1}^{N} \rho_k^s \; F_{ij(k)} \; ;$$

$$i = 1,2,\ldots,N \quad ; \; j = 1,2,\ldots,N \qquad (4)$$

Substitution of equation (4) into equation (3) yields

$$\beta_{ij} = \epsilon_j \phi_{ij} + \sum_{m=1}^{N} \rho_m^d \; \phi_{im} \beta_{mj} \qquad (5)$$

Rearrangement of the terms in equation (5) yields

$$\sum_{m=1}^{N} (\delta_{im} - \rho_m^d \phi_{im}) \beta_{mj} = \epsilon_j \phi_{ij} \qquad (6)$$

Equation (6) can be represented in matrix form as

$$[D_{ij}] [\beta_{ij}] = [\epsilon_j \phi_{ij}] \qquad (7)$$

where D is an N X N coefficient matrix with general element

$$D_{ij} = \delta_{ij} - \rho_j^d \phi_{ij} \qquad (8)$$

The systems of equations represented by (7) can be solved by matrix inversion to obtain the absorption factors (β_{ij})

$$\beta_{ij} = \sum_{m=1}^{N} D_{im}^{-1} \; \epsilon_j \phi_{mj} \qquad (9)$$

The radiation interchange factors ($\mathcal{F}_{ij}$) are related to the absorption factors (see reference 1) by the expression

$$\mathcal{F}_{ij} = \epsilon_i \beta_{ij} \tag{10}$$

and, using the usual arguments for the conservation of energy, the reciprocity relation for the N^2 values of $\mathcal{F}_{ij}$ is

$$A_i \mathcal{F}_{ij} = A_j \mathcal{F}_{ji} \tag{11}$$

The foregoing equations apply to radiation enclosures consisting of any combination of diffuse and specular surfaces ranging from totally diffuse to totally specular enclosures.

II. REFERENCES

1. Gebhart, B., "Unified Treatment for Thermal Radiation Transfer Processes--Gray, Diffuse Radiators and Absorbers", Paper No. 57-A-34, ASME, December 1957.

2. Gebhart, B., Heat Transfer, McGraw-Hill Book Co., Inc., 1961, pp. 117-122.

3. Gebhart, B., "Surface Temperature Calculations in Radiant Surroundings of Arbitrary Complexity--for Gray, Diffuse Radiation," International Journal of Heat Mass Transfer, Vol. 3, No. 4, 1961, pp. 341-346.

APPENDIX J

SYSTEM-DEPENDENT INFORMATION

1. Univac 1110 - JSC Houston

a. Binary Interface Tape

Capability exists to dump all data that exists in out of core storage to a binary tape at any point in the execution. In addition, radiation conductors, as output by RKCAL or RCCAL may be written to this tape.

Loading Binary Tape - Subroutine TPLOAD

When the call CALL TPLOAD (CNAME) is encountered in the operations data, all data in out of core storage plus pertinent in-core data is placed on the binary tape. This tape is written from unit USER1, so this operation requires the word USER1 in the options block.

The data written consits of the following:

Node Numbers (One record).
Node Areas (One record).
Surface Optical Properties (One record).

After this, all or part of the following are written (obtained from out of core storage) as:

Form Factors (One file).
Grey-Body Factors (One file per wave band).
Incident Flux Data (One file per DICAL execution).
Absorbed Flux Data (One file per AQCAL execution).

Each file written is identified by the configuration name as specified by argument CNAME.

After a TPLOAD call, radiation conductors (as computed by RCCAL or RKCAL) may be added to the binary tape by using 5HBTAPE as the 9th argument in RCDATA and/or RKDATA and executing RCCAL and/or RKCAL. Note the restriction that must be done subsequent to a TPLOAD call.

b. Subroutine NPRIDS

This subroutine is used to provide user identification information to the 1110 plot package so that plot output finds its way back to the user. User name, box number, phone extension, and project number are specified by the following calling sequence:

CALL NPRIDS (NAME1, NAME2, BOX, EXTN, PROJCT)

Example: CALL NPRIDS (5HR. A., 5HVOGT, 3HM34, 4H2326, 4HF261)

One to six characters are allowed for each Hollerith argument.