26821-H027-R0-00

CR 160096

# FORMULATION OF ADVANCED CONSUMABLES MANAGEMENT MODELS

JANUARY 1979 CONTRACT NO. NAS9-14264

- FINAL TECHNICAL REPORT -

## ENVIRONMENTAL CONTROL AND ELECTRICAL POWER SYSTEM PERFORMANCE MODELS REQUIREMENTS

Prepared by

J. K. Daly

J. G. Torian

Operational Systems Section

TRW
DEFENSE AND SPACE SYSTEMS GROUP

26821-H027-R0-00

# FORMULATION OF ADVANCED CONSUMABLES MANAGEMENT MODELS

JANUARY 1979

CONTRACT NO. NAS9-14264

## - FINAL TECHNICAL REPORT -

## ENVIRONMENTAL CONTROL AND ELECTRICAL POWER SYSTEM PERFORMANCE MODELS REQUIREMENTS

Prepared by

J. K. Daly

J. G. Torian

Operational Systems Section

TRW
DEFENSE AND SPACE SYSTEMS GROUP

# PREFACE

Current consumables management techniques are supported by computer programs which provide consumables analysis in support of Space Transportation System (STS) preoperational flight programs. The techniques are also supported by computer programs currently under development for space vehicles such as STS and other advanced spacecraft programs. These latter programs provide consumables management techniques which relieve skilled personnel from the routine processing associated with repetitive spacecraft mission planning and operational functions. This allows concentration on detailed trade studies and mission assessment in support of advanced missions and spacecraft. Current subsystem analysis tools used for trade studies and mission assessment are out of date with respect to computer application and flexibility to model the more advanced spacecraft subysystems. These reports present the results of a study to establish the requirements for advanced subsystem analytical tools and define the modifications and updating of current computer programs that will satisfy the requirements for future space programs.

The final report on the study to formulate advanced consumables management models is presented in two parts - an Executive Summary and a Final Technical Report.

The Executive Summary presents an overview of the study results.

This Final Technical Report presents the software design specification for development of Environmental Control and Life Support System (ECLSS) and Electrical Power System (EPS) interactive computer programs.

## CONTENTS

## FIGURES

# 1.0 INTRODUCTION

This report presents the software design specifications for developing Environmental Control and Life Support System (ECLSS) and Electrical Power System (EPS) programs into interactive computer programs.

Specifications for the ECLSS program are at the detail design level with respect to modification of an existing batch mode program. The Fortran Environmental Analysis Routines (FEAR) of Reference 1 are the subject batch mode program. Development of this interactive program requires familarity with ECLSS analysis, the FEAR program and Reference 1.

An Appendix summarizing the characteristics of the FEAR program is included in this report. This Appendix is provided as a reference for the reader with a general interest in modifying batch mode programs to form interactive programs, rather than the specific modification of FEAR.

The EPS program specifications are at the preliminary design level. The text assumes the reader is familiar with EPS analysis techniques. Emphasis is on top-down structuring in the development of an interactive program.

The TRW Systems Engineering and Integration Division (SEID) Software Development Policies (Reference 2) has been used as the guideline in the development of these specifications. These or similar policies should be used in the implementation of the modified program to make it uniform, readable, understandable and maintainable.

# 2.0 FORMULATION OF INTERACTIVE SUBSYSTEM MODELS

## 2.1 ENVIRONMENTAL CONTROL AND LIFE SUPPORT ANALYSIS PROGRAM

The objectives of ECLSS performance evaluations are best satisfied by the application of an interactive computer program with which the user accesses a library of routines simulating the performance of various components and functions common to ECLSS. These routines are assembled with a driver routine (MAIN) to simulate the particular ECLSS under consideration. The assembled program is then loaded and executed to produce the transient performance parameters of the ECLSS under prescribed boundary conditions.

The assembly procedures for such a program are shown in Figure 1. The master library of routines is extracted from a secure file. The user has the option to enter a MAIN routine (as for initial development of an ECLSS model), or extract a particular MAIN from his individual library (as for update/edit and/or additional studies with a previously developed ECLSS model). The extracted MAIN may be altered as part of the update/edit process. The program is then MAP'ed and the MAIN may be stored in the user's file for future use. The particular ECLSS program is then ready for execution.

The execution procedure including a variety of input/output options is shown in Figure 2. The component characteristic data and initial conditions may be read in from restart data stored previously or entered directly. If the user desires, the system will output a schematic of the ECLSS modeled. The user has the option to select particular nodes (component locations) to be included in tabular output* or the system will default to include all output for tabular nodes defined in the model. If plots are desired, the user simply defines the particular parameters to be plotted. Restart data may be stored for future use. Up to this point the program is executed in an interactive mode which is mandatory as noted in Figure 2. The program then transfers to a second stage of execution.

*Tabular output may be stored for input to, or directly interfaced to other resident programs requiring these parameters.

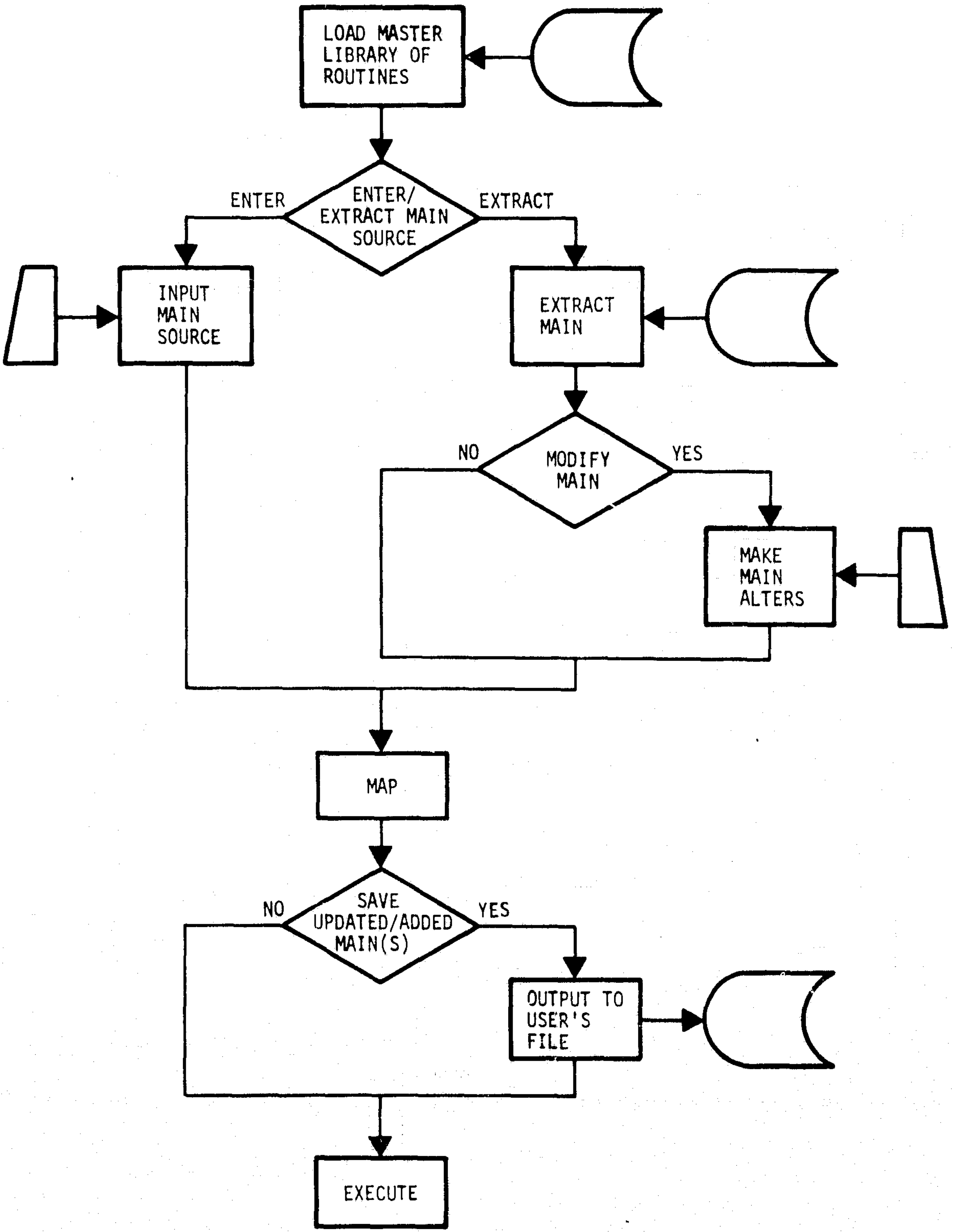


Figure 1. ECLSS Program Assembly Procedure

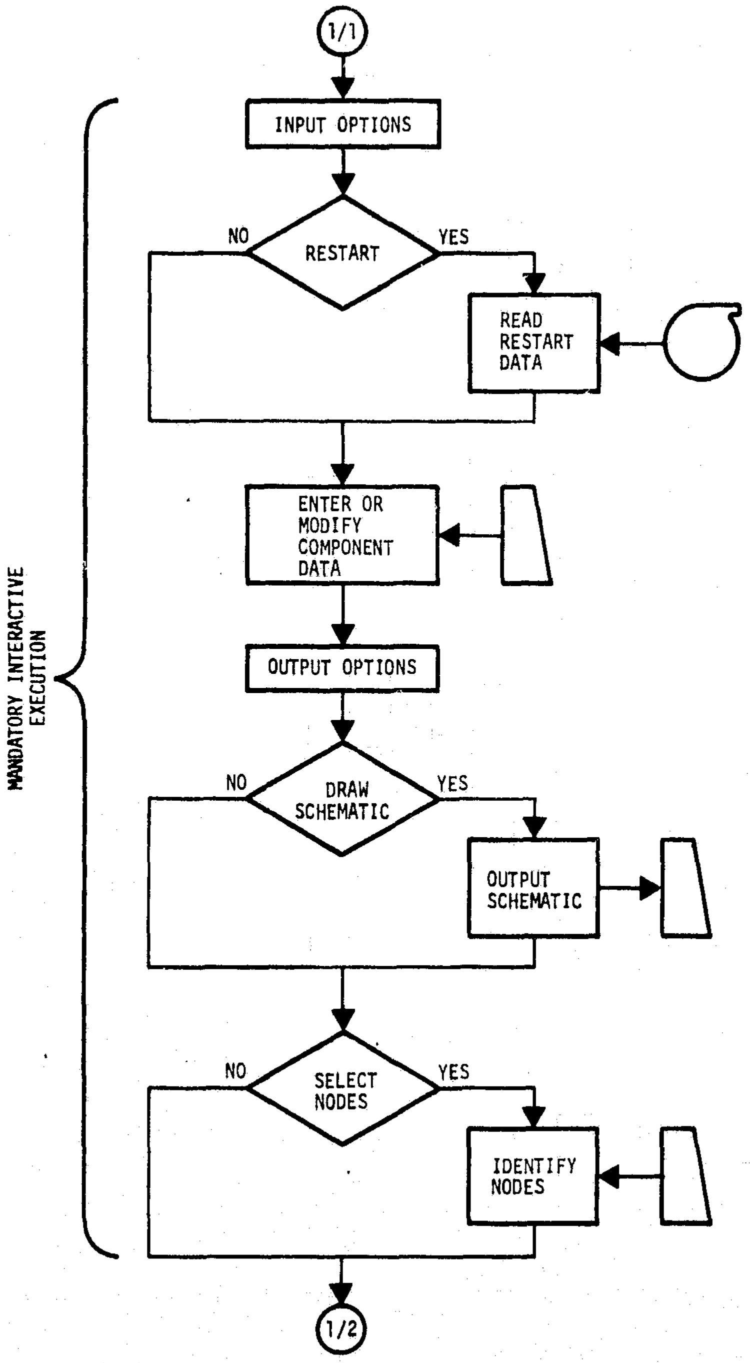


Figure 2. ECLSS Program Execution Procedure

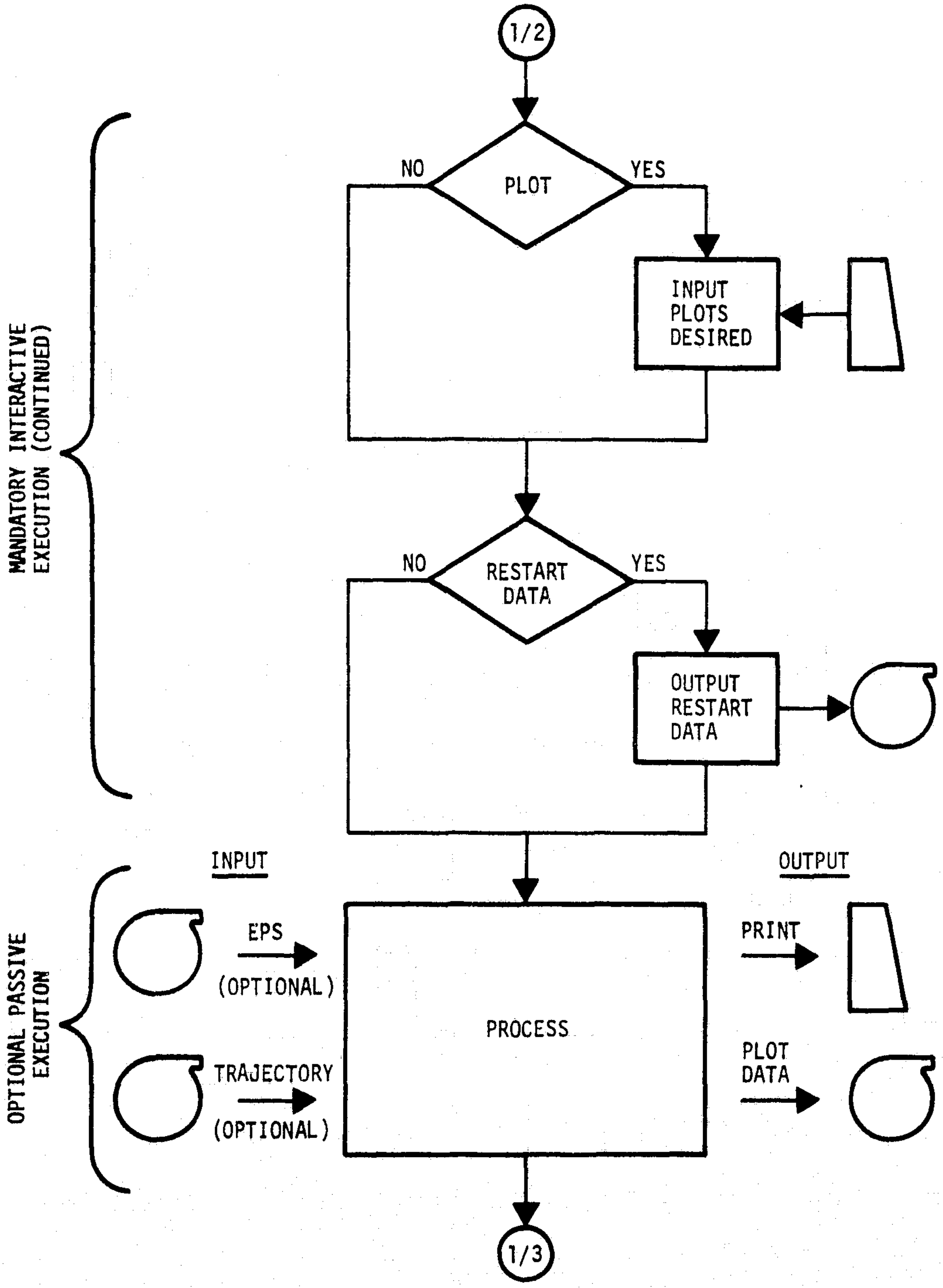


Figure 2. ECLSS Program Execution Procedure (Continued)

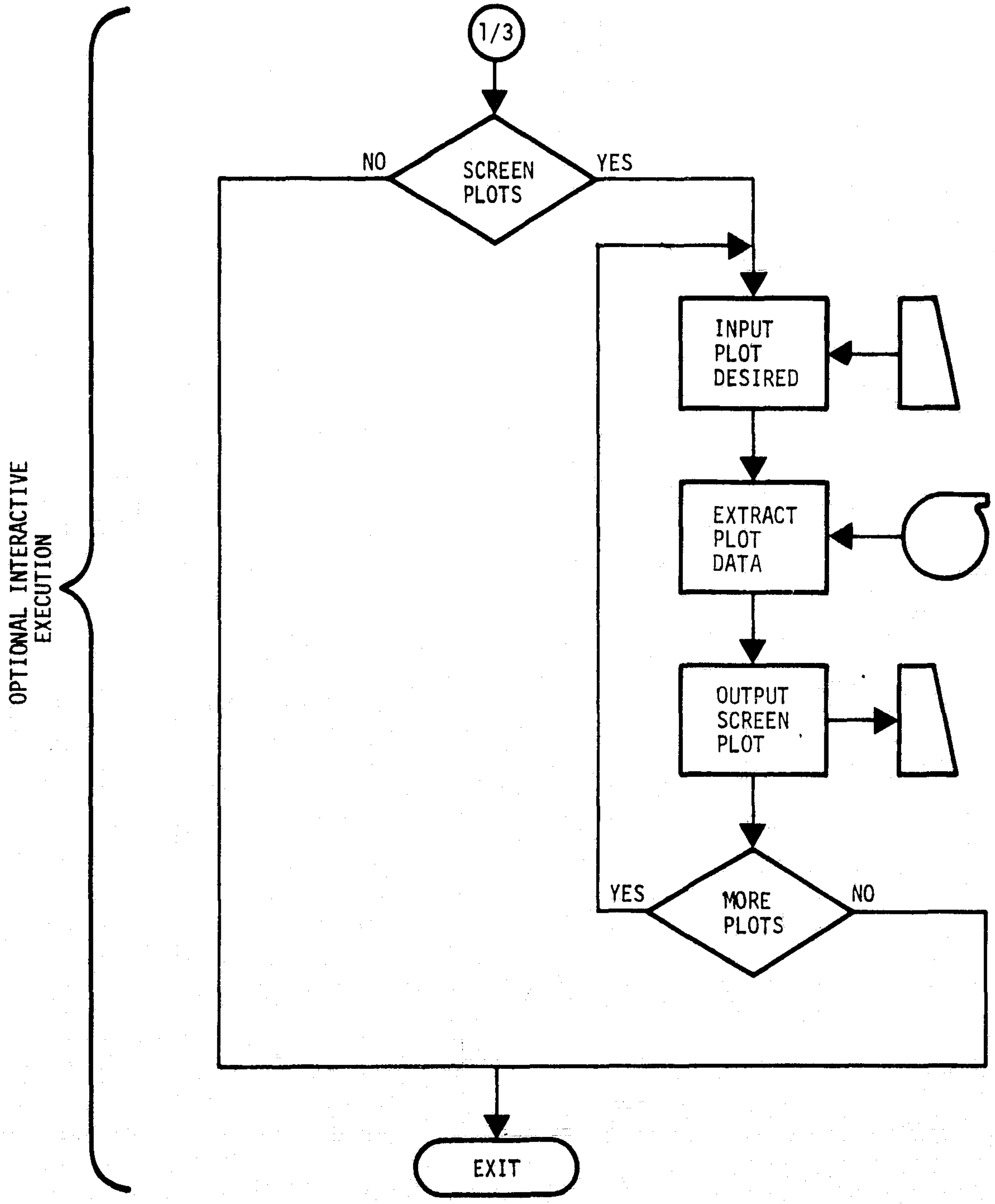


Figure 2. ECLSS Program Execution Procedure (Concluded)

The second stage of execution is passive in the sense the data is processed with no interaction on the part of the user. The processing accesses Electrical Power System (EPS) data and/or trajectory data automatically, if required. The data can be accessed from tape, secure files, or interface to other programs resident in the system. This stage of execution produces the tabular and plot data. This stage is optional in that the user may execute and witness the tabular data as it is produced, or exit at this point. Exit may be desired in a case where the user desired only to update the model and review the schematic. Exit may also be desired so that this stage of execution can be performed off line in a batch-mode rather than interactively.

A second interactive execution is available which allows the user to extract the plot data just created or previously created plot data for display on the terminal. The execution is interactive with the user simply specifying which plots are to be displayed.

A program of the type described can be developed by modifying the existing batch mode version of FEAR. Subsequent parts of this section define the requirements for such a modification. The modified program is referred to as SON OF FEAR.

A pilot model of the interactive model was developed and used as an aid in establishing these requirements. Illustrative interactive displays generated by the pilot model are used as examples throughout the subsequent text. These displays are presented as examples typifying the display requirements and should not be considered firm requirements in format nor total content.

### 2.1.1 Program Control Routines

Four routines are used for basic program control. The use of START and PRINT routines are mandatory. These routines initialize the run, provide for output of data, control the timing indicies, and sum the consumables associated with the operation of various ECLSS components. The optional LOOP and CONVRG routines, used only for convergence control in the FEAR program, are classified as basic program control routines in SON OF FEAR. The reclassification of these routines is associated with their additional role in the control of display flags in the interactive program. Requirements for the four control routines are as follows.

## Routine START

The call to routine START is the first executable statement in the MAIN. The routine accepts run mode, output option and initialization data via the terminal. In addition, the routine sets up several control parameters as in FEAR. The routine sets a display control flag to instruct subsequent routine calls to request data on their initial execution, rather than read in formatted card data as in FEAR. The flow diagram for routine START is given on Figure 3.

A typical menu of control options and indicators, as well as subsequent request and entry of data, are shown in Figure 4. These displays were executed on the pilot program. Note that the user has indicated a new start run mode, hence a program title is requested. In the restart mode the program name is contained in the restart data.

A typical initialization display, user prompting, user entry, and display response (revision) is shown in Figure 5. Note the program and run title. The initial display has zero values for all items. The user indicates a desire to enter (or change) item number 1. Subsequent prompting and entry result in the display revision. A completely revised display and entry requesting display exit is illustrated in Figure 6.

The user interface illustrated on Figures 5 and 6 is typical of line item entry/change displays used in the program. A general flow diagram for control of such displays is shown in Figure 7. This is a support routine and does not appear in the user's MAIN.

## Routine PRINT

Routine PRINT is the last routine call in the timing loop. The routine, as modified, not only provides for output control, but also is the program control for updating the timing loop, reading in boundary condition data, summation of the consumables, and program termination.

The restart data is written from this routine. In addition, the plot input/output control is internal to this routine such that the user does not have to enter a call to the PLOOT Routine in the MAIN as in FEAR.

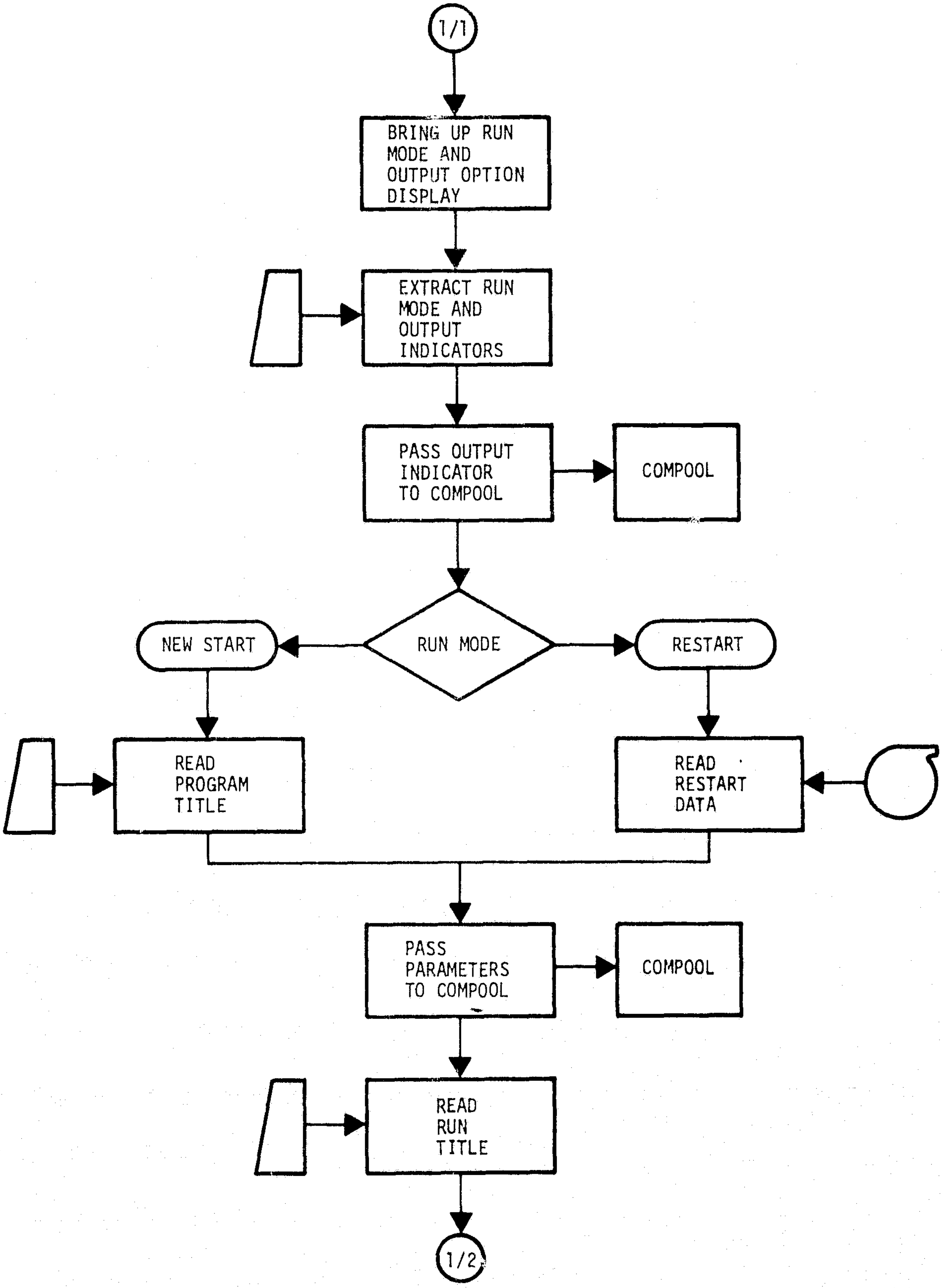

1/1
BRING UP RUN MODE AND OUTPUT OPTION DISPLAY
EXTRACT RUN MODE AND OUTPUT INDICATORS
PASS OUTPUT INDICATOR TO COMPOOL
COMPOOL
RUN MODE
NEW START
RESTART
READ PROGRAM TITLE
READ RESTART DATA
PASS PARAMETERS TO COMPOOL
COMPOOL
READ RUN TITLE
1/2

Figure 3. Routine START

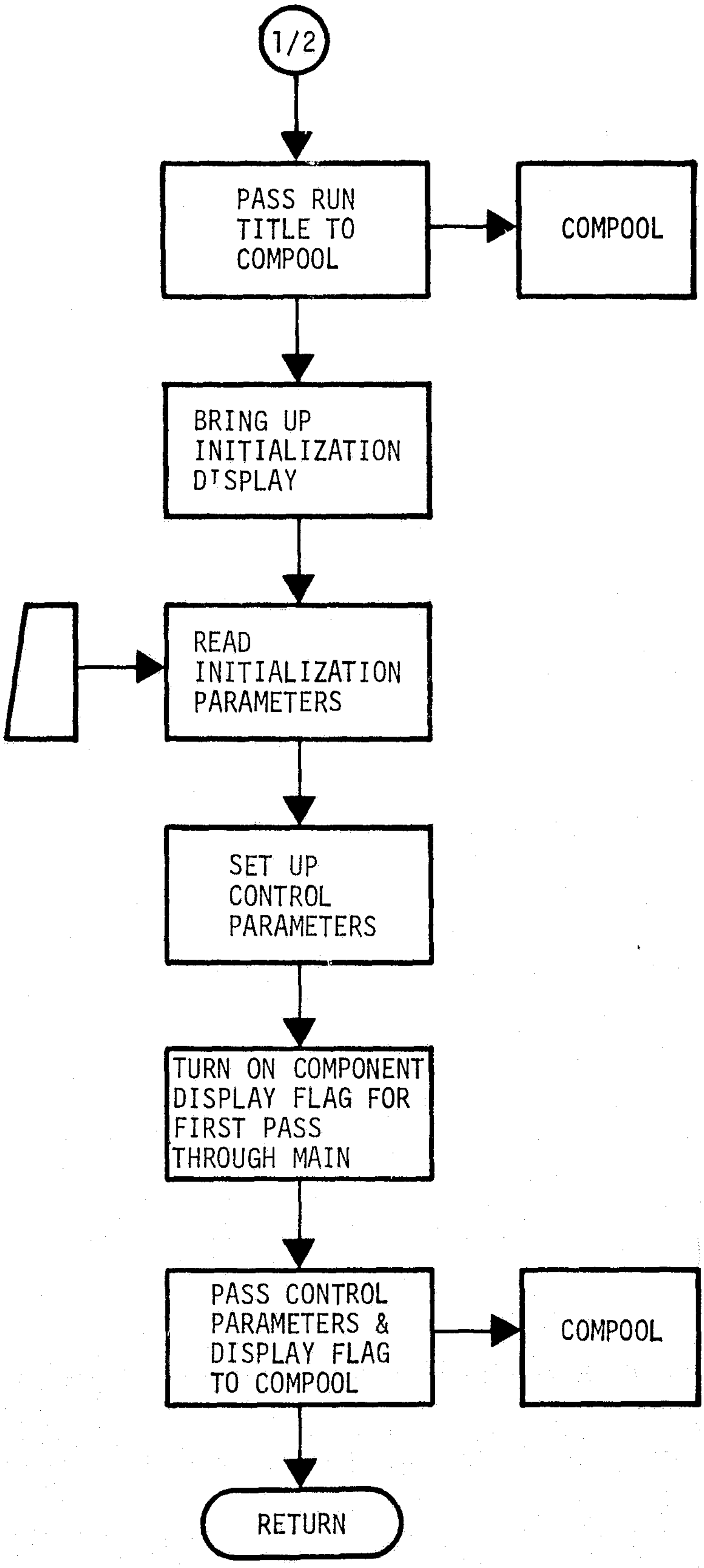
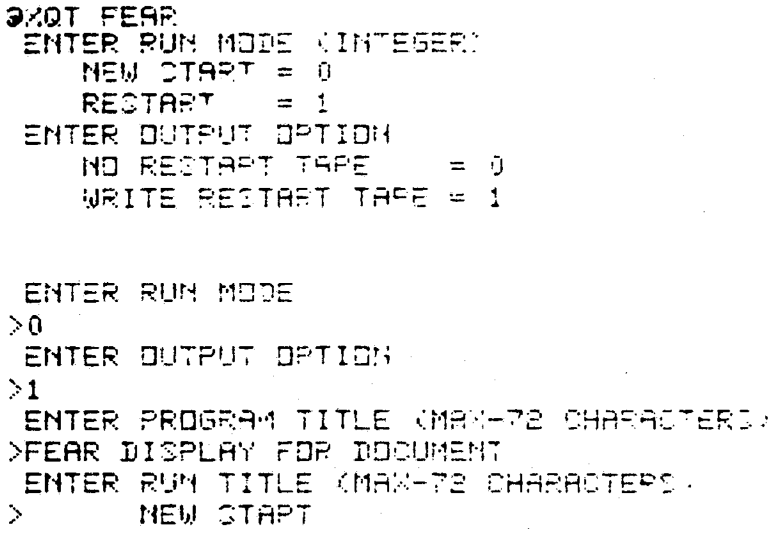
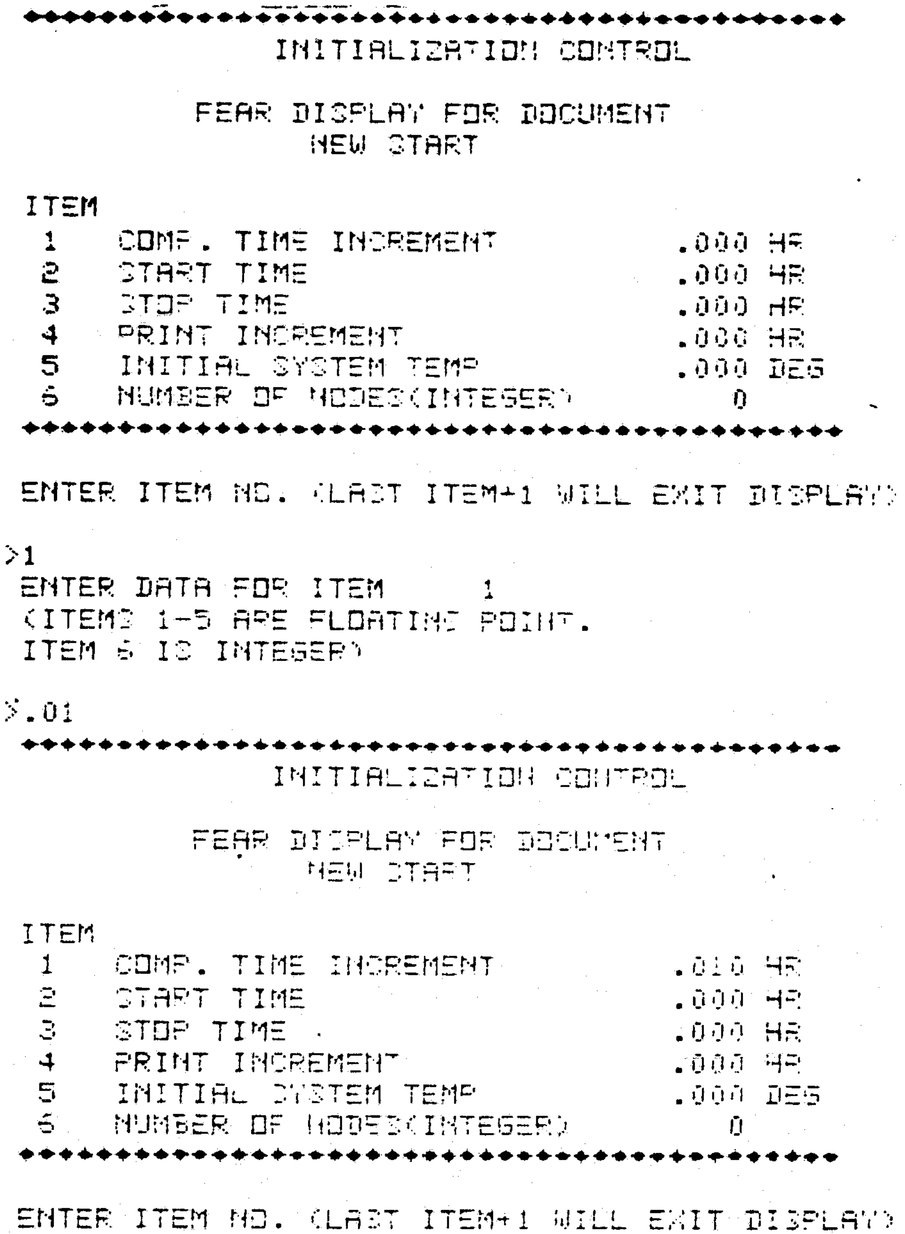

1/2
PASS RUN TITLE TO COMPOOL
COMPOOL
BRING UP INITIALIZATION DᵀSPLAY
READ INITIALIZATION PARAMETERS
SET UP CONTROL PARAMETERS
TURN ON COMPONENT DISPLAY FLAG FOR FIRST PASS THROUGH MAIN
PASS CONTROL PARAMETERS & DISPLAY FLAG TO COMPOOL
COMPOOL
RETURN

Figure 3. Routine START (Concluded)



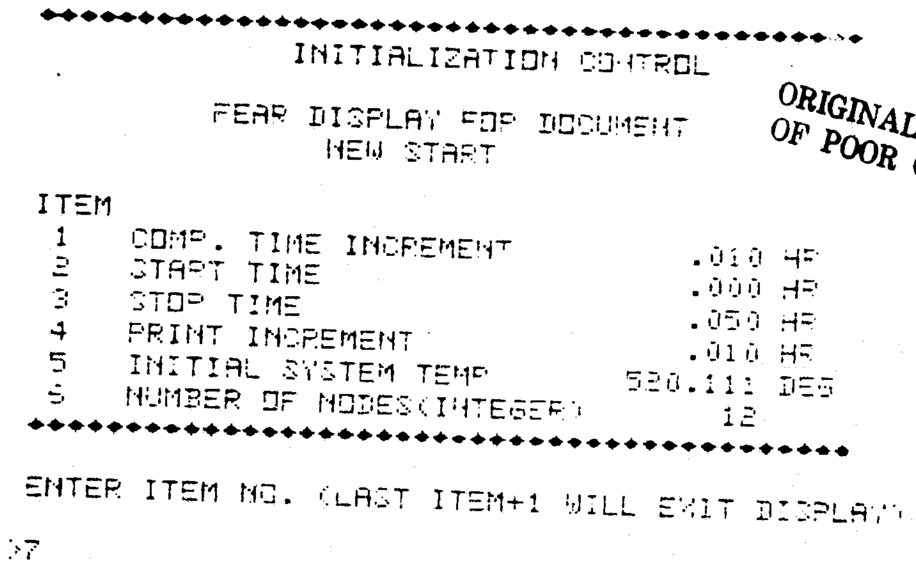
```
@XQT FEAR
 ENTER RUN MODE (INTEGER)
    NEW START = 0
    RESTART   = 1
 ENTER OUTPUT OPTION
    NO RESTART TAPE    = 0
    WRITE RESTART TAPE = 1


 ENTER RUN MODE
>0
 ENTER OUTPUT OPTION
>1
 ENTER PROGRAM TITLE (MAX-72 CHARACTERS)
>FEAR DISPLAY FOR DOCUMENT
 ENTER RUN TITLE (MAX-72 CHARACTERS)
>       NEW START
```

Figure 4.  Control Options



```
*******************************************
            INITIALIZATION CONTROL

        FEAR DISPLAY FOR DOCUMENT
              NEW START

ITEM
 1    COMP. TIME INCREMENT              .000 HR
 2    START TIME                        .000 HR
 3    STOP TIME                         .000 HR
 4    PRINT INCREMENT                   .000 HR
 5    INITIAL SYSTEM TEMP               .000 DEG
 6    NUMBER OF NODES(INTEGER)             0
*******************************************

ENTER ITEM NO. (LAST ITEM+1 WILL EXIT DISPLAY)

>1
 ENTER DATA FOR ITEM      1
 (ITEMS 1-5 ARE FLOATING POINT.
 ITEM 6 IS INTEGER)

>.01
*******************************************
            INITIALIZATION CONTROL

        FEAR DISPLAY FOR DOCUMENT
              NEW START

ITEM
 1    COMP. TIME INCREMENT              .010 HR
 2    START TIME                        .000 HR
 3    STOP TIME                         .000 HR
 4    PRINT INCREMENT                   .000 HR
 5    INITIAL SYSTEM TEMP               .000 DEG
 6    NUMBER OF NODES(INTEGER)             0
*******************************************

ENTER ITEM NO. (LAST ITEM+1 WILL EXIT DISPLAY)
```

Figure 5. Initialization Control Display and Operation



```
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
                INITIALIZATION CONTROL

          FEAR DISPLAY FOR DOCUMENT
                NEW START

ITEM
 1   COMP. TIME INCREMENT
 2   START TIME                             .010 HR
 3   STOP TIME                              .000 HR
 4   PRINT INCREMENT                        .050 HR
 5   INITIAL SYSTEM TEMP                    .010 HR
 6   NUMBER OF NODES(INTEGER)            520.111 DEG
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆   12

ENTER ITEM NO. (LAST ITEM+1 WILL EXIT DISPLAY)

>7
```

Figure 6. Revised Initialization Control Display

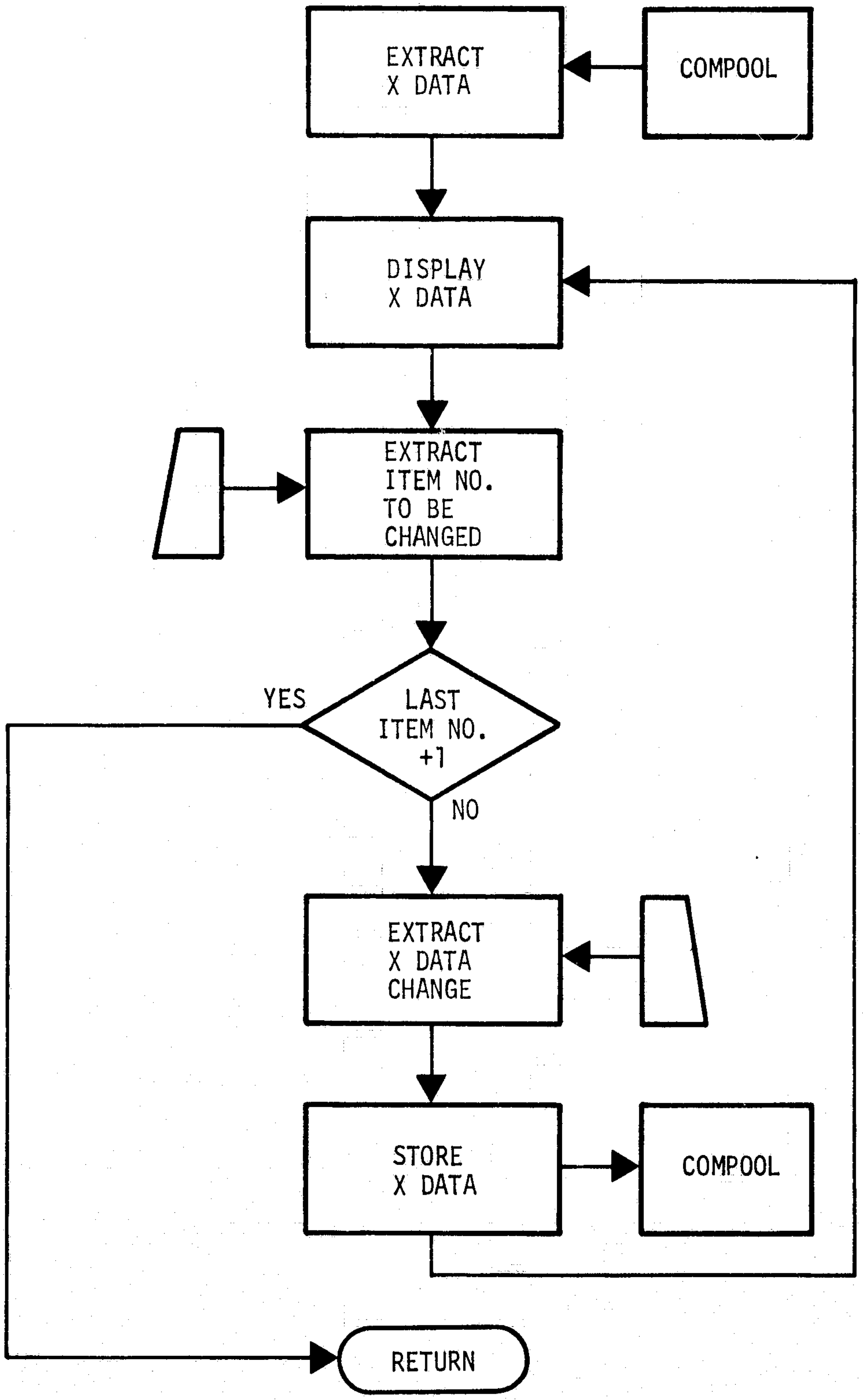


Figure 7. Routine DISPLAY X

The PRINT routine as shown in Figure 8 provides for selective nodal print out (similar to routine SELECT in FEAR) and affords a selection of generic or user assigned nodal name in the output. With the assigned name option the user can specify an alphanumeric name to a node. the assigned node name such as "AFT BAY EQUIPMENT" is printed with the nodal number in the output. The generic name option assigns a name associated with the component routines referencing that node. A typical generic name is "EVAP OUT/PLATE IN." The sample name is for a node that represents the fluid parameters at a node down stream of an evaporator (outlet node for call to EVAP) and upstream of a cold plate (inlet node for call to PLATE).

The generic name indicators are set up in the component routines and are also the cross reference indicators to construct a schematic of the coolant loop(s). Typical output display is shown in Figure 9. This particular display was produced by the pilot model which has the generic node name option set up for the evaporator and cold plate only.

The PRINT routine is set up to provide for display of a schematic of the circuit being analyzed. The process of displaying the schematic is essentially a plotting procedure controlled by the generic name indicators.

The display flag is turned off in routine PRINT such that subsequent calls to this routine and component routines will not bring up their associated input request displays.

The remainder of the PRINT routine is similar to the equivalent routine in FEAR except that the boundary condition routines are called automatically if required and provisions are made for viewing screen plots at the terminal phase of execution. The extraction of boundary condition data at this point eliminates the need for the user to enter the routines in the MAIN as in FEAR. This feature is further discussed in Section 2.1.4. The screen plots are individually selected from the menu of plots extracted during the first pass through PRINT.

<u>Routine LOOP, CONVRG</u>

Routine LOOP and CONVRG are used in conjunction with a closed loop system that controls iteration to a prescribed convergence tolerance. To obtain proper control of the display flags, the FEAR LOOP routine is modified to turn on the display flag. Whereas, CONVRG is modified to turn off

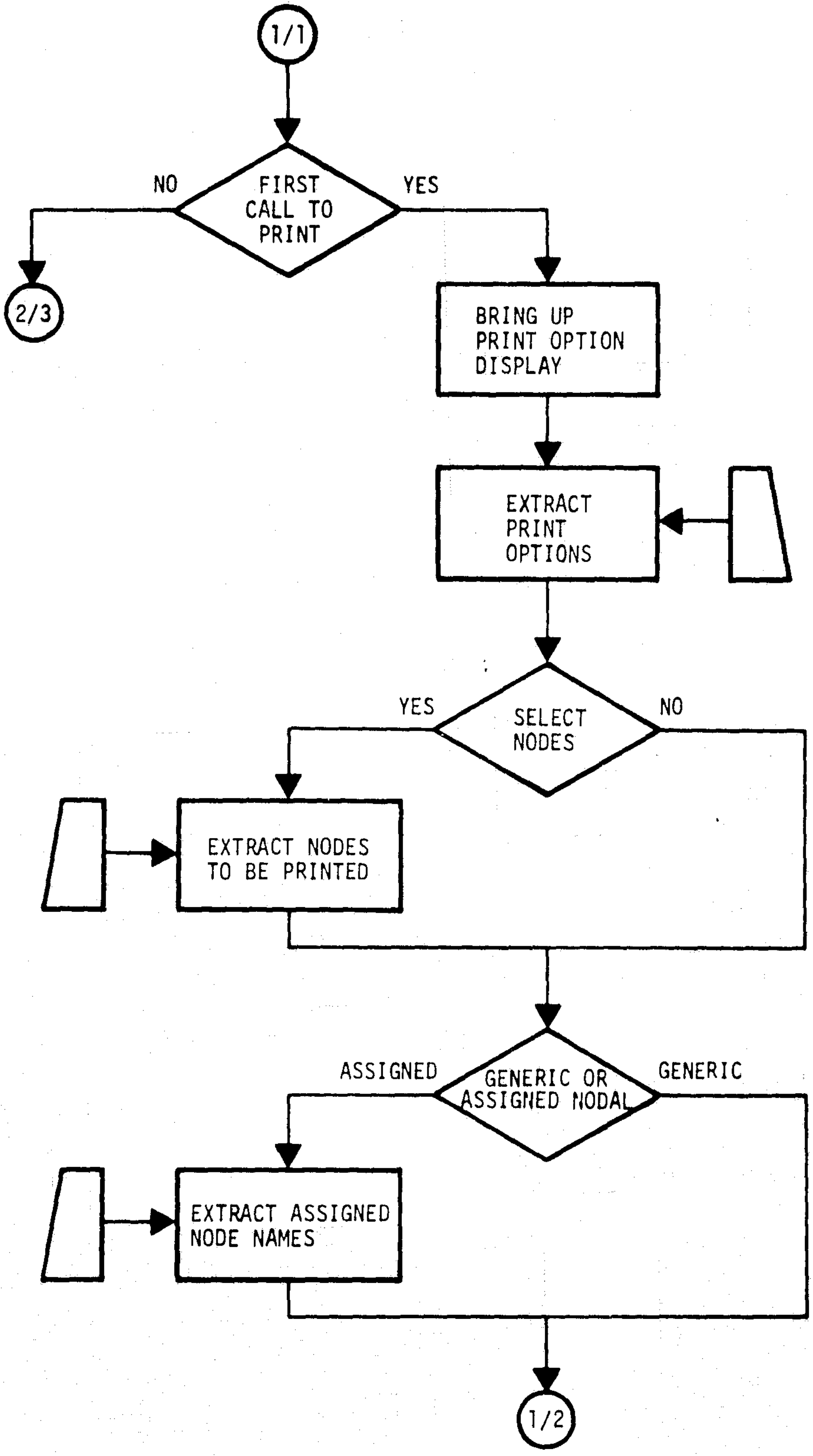

1/1
FIRST CALL TO PRINT
NO
YES
2/3
BRING UP PRINT OPTION DISPLAY
EXTRACT PRINT OPTIONS
SELECT NODES
YES
NO
EXTRACT NODES TO BE PRINTED
GENERIC OR ASSIGNED NODAL
ASSIGNED
GENERIC
EXTRACT ASSIGNED NODE NAMES
1/2

Figure 8. Routine PRINT

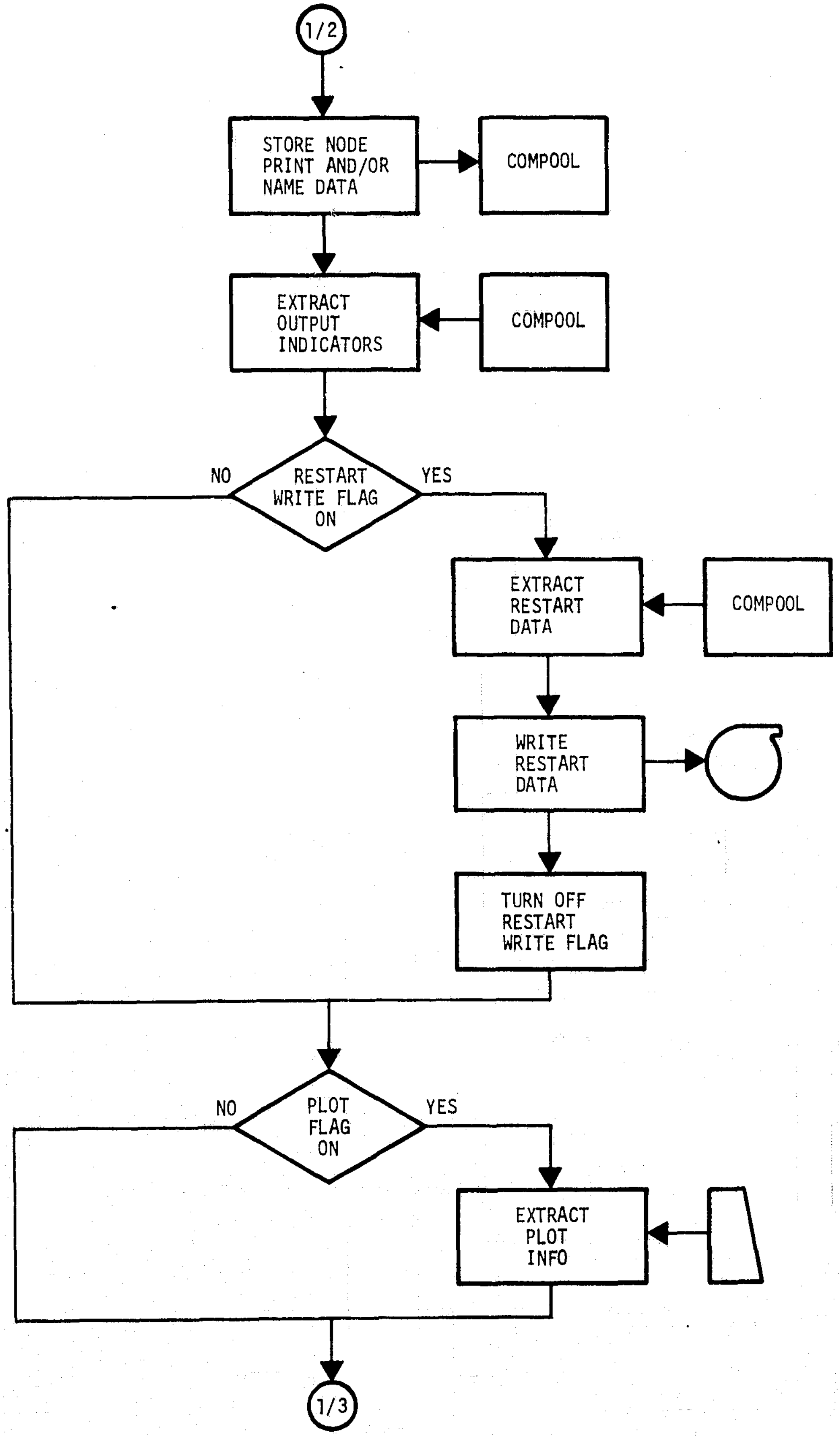


Figure 8. Routine PRINT (Continued)

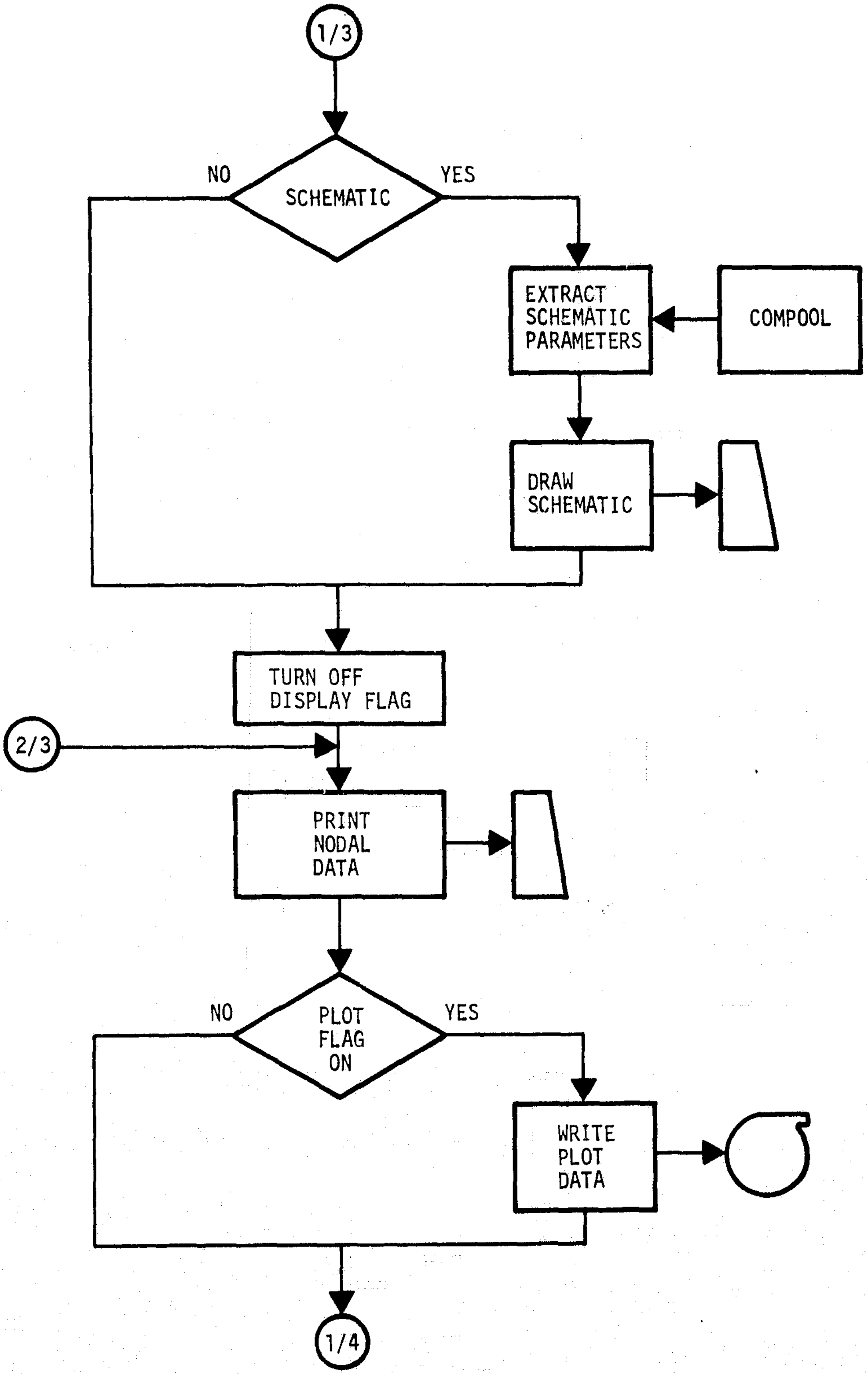


Figure 8. Routine PRINT (Continued)

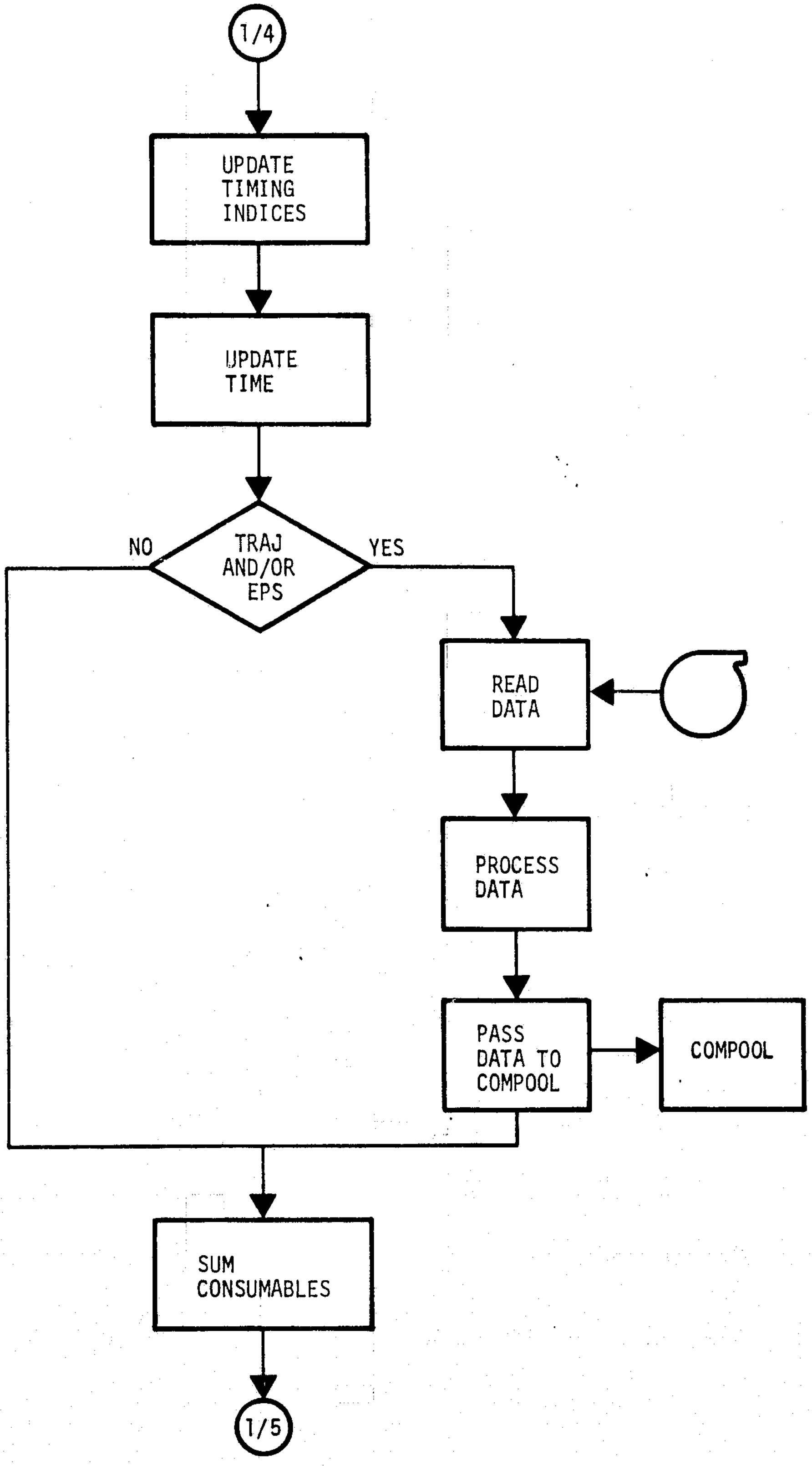


Figure 8. Routine PRINT (Continued)

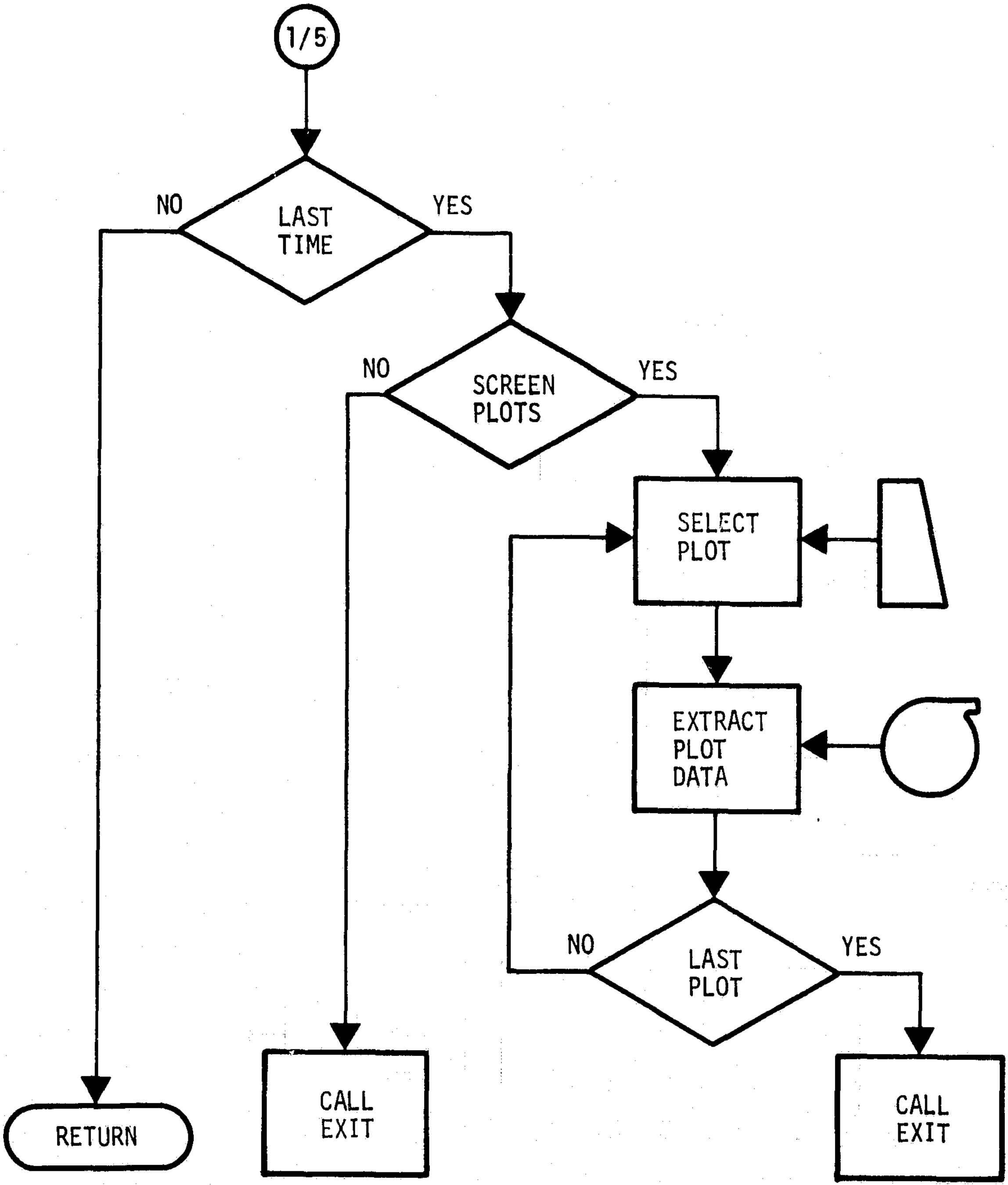
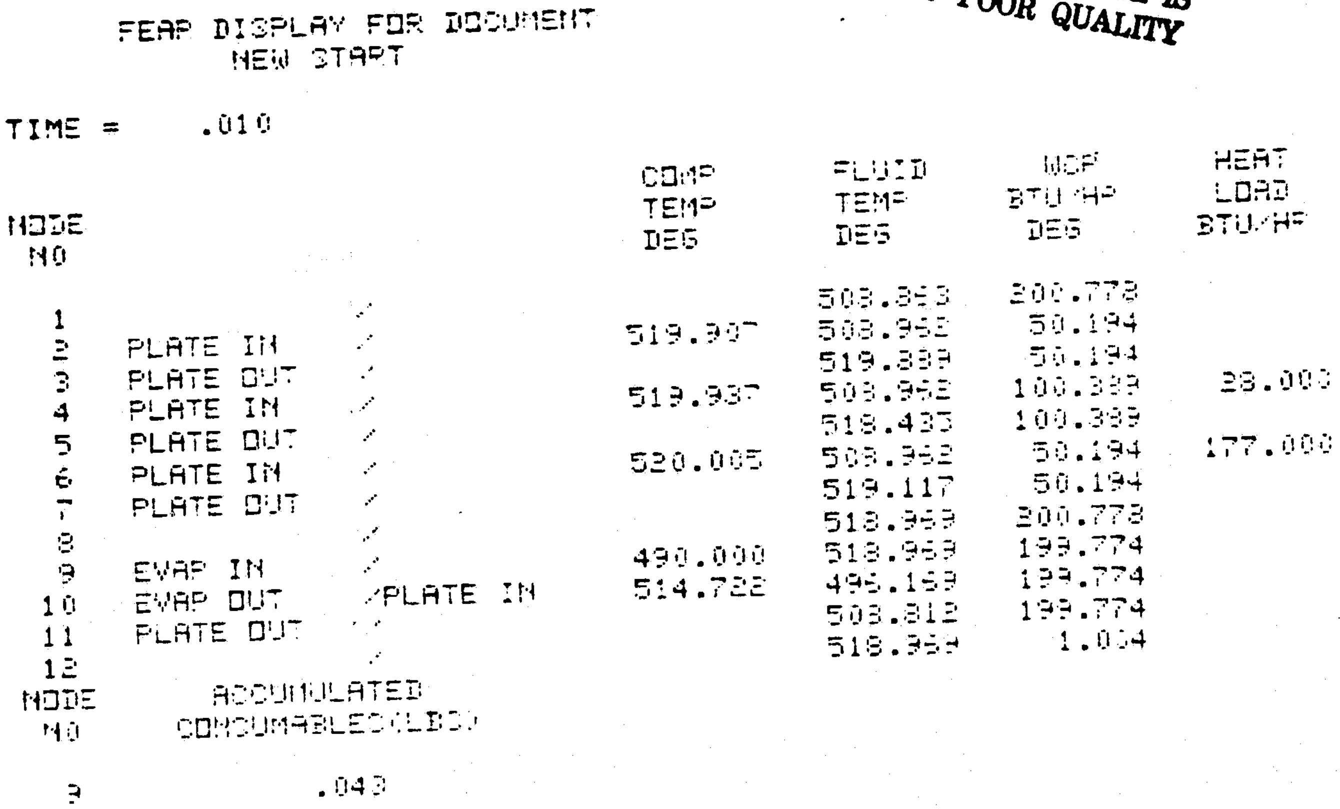


Figure 8. Routine PRINT (Concluded)



```
           FEAP DISPLAY FOR DOCUMENT
                 NEW START

TIME =     .010
                                          COMP       FLUID       WCP        HEAT
                                          TEMP       TEMP      BTU/HR       LOAD
NODE                                      DEG        DEG        DEG        BTU/HR
 NO

                                                   508.853    200.778
  1                       /
  2  PLATE IN             /              519.907   508.952     50.194
  3  PLATE OUT            /                        519.339     50.194
  4  PLATE IN             /              519.937   508.952    100.389      28.000
  5  PLATE OUT            /                        518.435    100.389
  6  PLATE IN             /              520.005   508.952     50.194     177.000
  7  PLATE OUT            /                        519.117     50.194
  8                       /                        518.969    200.778
  9  EVAP IN              /              490.000   518.969    199.774
 10  EVAP OUT             /PLATE IN      514.722   495.159    199.774
 11  PLATE OUT            /                        508.812    199.774
 12                       /                        518.954      1.004
NODE      ACCUMULATED
 NO     CONSUMABLES(LBS)

  9              .043
```

Figure 9. Typical Output Display

the display flag. It is necessary to turn off the flag in CONVRG to prevent bringing up the displays during the iteration process. The display flag is turned on in LOOP for the first call referencing that node in case a previous call to CONVRG has turned the flag off. This latter case occurs in a program with multiple loops. Note that START and PRINT also control the flag in a similar manner for cases in which LOOP and CONVRG are not used in the program.

Routine CONVRG also stores the restart data for the subject loop if requested.

Flow diagrams for LOOP and CONVRG are given in Figures 10 and 11, respectively. The "PROCESS" block (Figure 11) refers to execution of those algorithms in the current version of this routine in FEAR.

### 2.1.2 Component Performance Routines

All component characteristic and initialization data are controlled interactively through displays unique to that component and for that particular node. A typical display control routine has been illustrated in Figure 7. Typical component requirements to incorporate this interactive feature are shown in Figure 12.

When the display flag is on, all resident data on the augmented node are transferred to a dummy array for communication with the display routine. Note that the resident data contain input and initialization from a previous execution for a restart. In the case of a new start the resident data are any throughput data from previous reference to this node and a default component temperature. The latter is equal to the initial system temperature as specified in the initialization control (Figure 6).

Modification to the component data via interaction with the display are entered in the dummy array. The dummy array overlays the resident nodal data upon return from the display routine.

The remaining processing for the components is similar to the current version of FEAR except for additional processing of boundary condition data. The additional processing is discussed in Section 2.1.4.

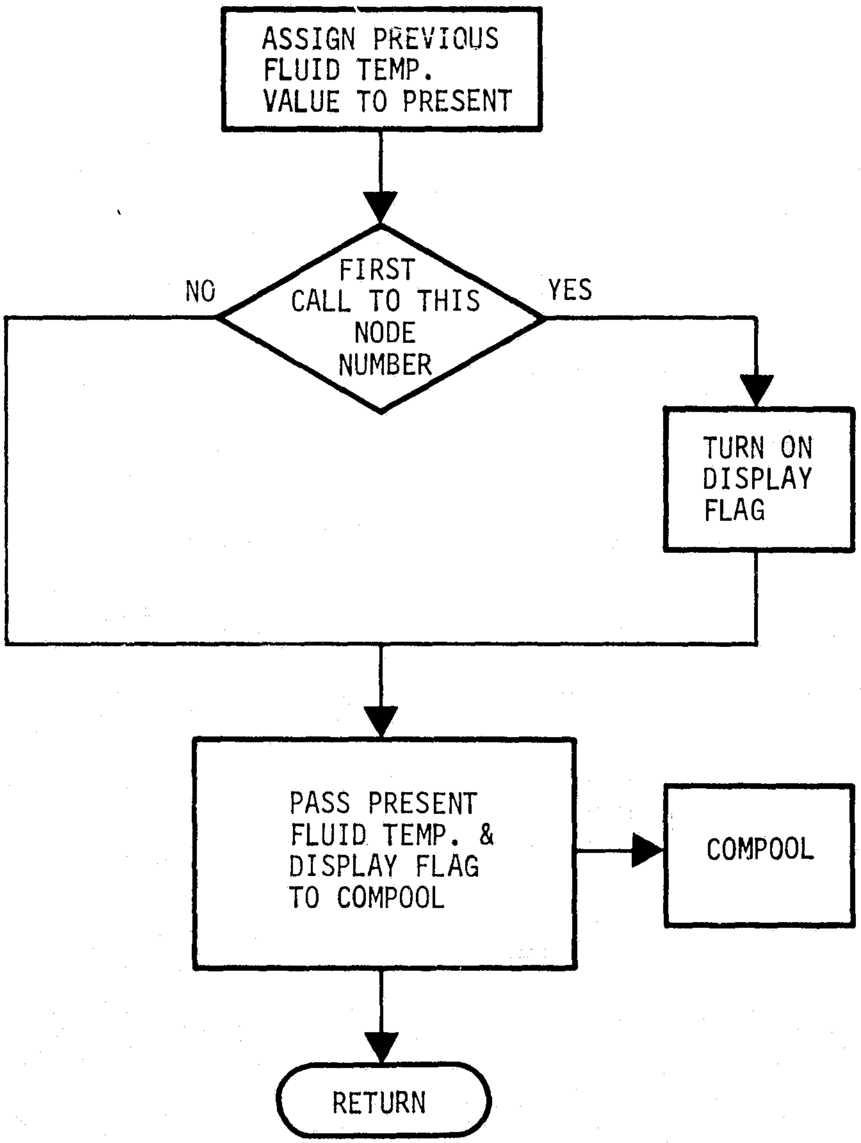

ASSIGN PREVIOUS FLUID TEMP. VALUE TO PRESENT
FIRST CALL TO THIS NODE NUMBER
NO
YES
TURN ON DISPLAY FLAG
PASS PRESENT FLUID TEMP. & DISPLAY FLAG TO COMPOOL
COMPOOL
RETURN

Figure 10. Routine LOOP

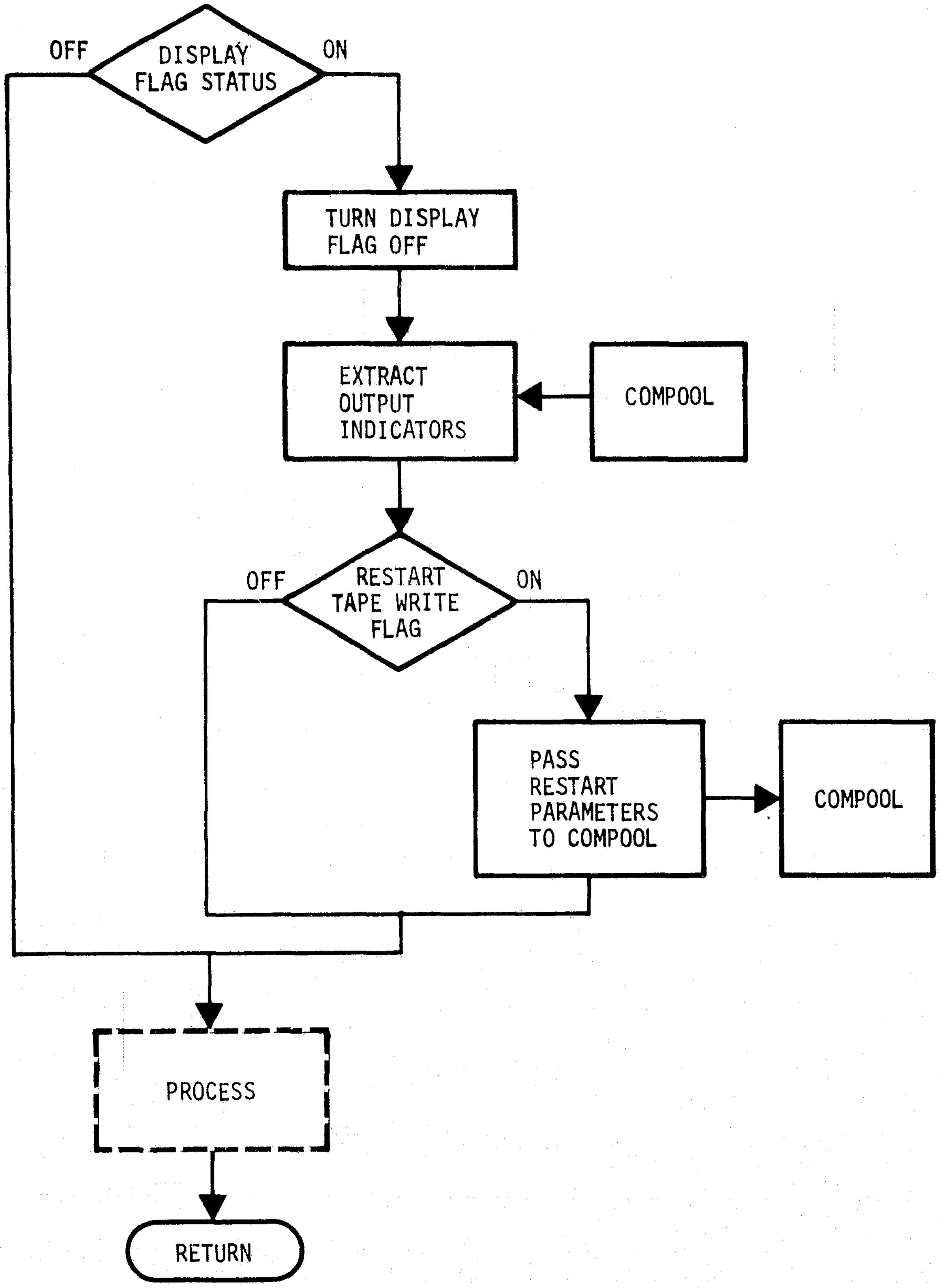


Figure 11. Routine CONVRG

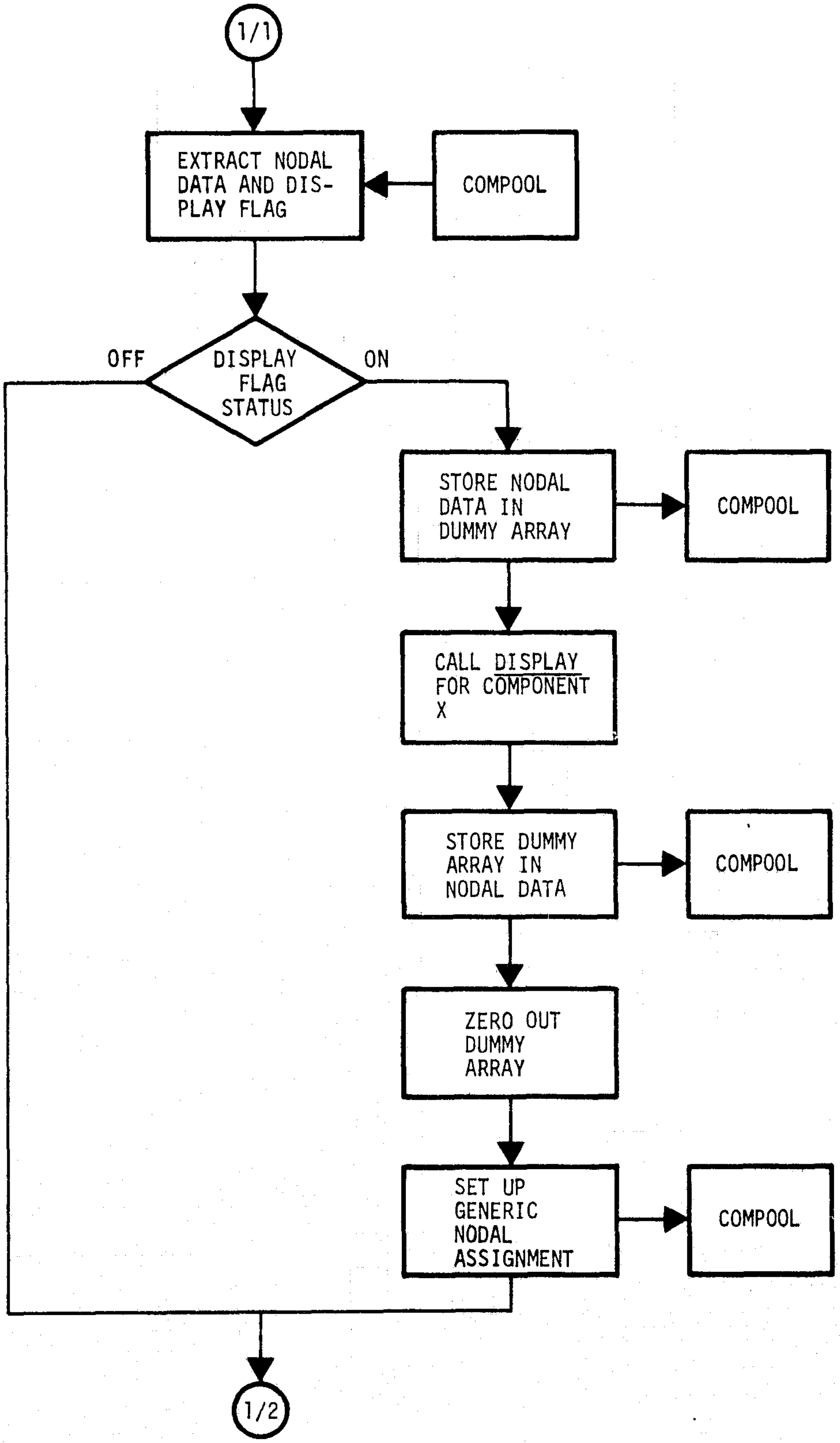

1/1
EXTRACT NODAL DATA AND DISPLAY FLAG
COMPOOL
DISPLAY FLAG STATUS
OFF
ON
STORE NODAL DATA IN DUMMY ARRAY
COMPOOL
CALL DISPLAY FOR COMPONENT X
STORE DUMMY ARRAY IN NODAL DATA
COMPOOL
ZERO OUT DUMMY ARRAY
SET UP GENERIC NODAL ASSIGNMENT
COMPOOL
1/2

Figure 12. Routine COMPONENT X

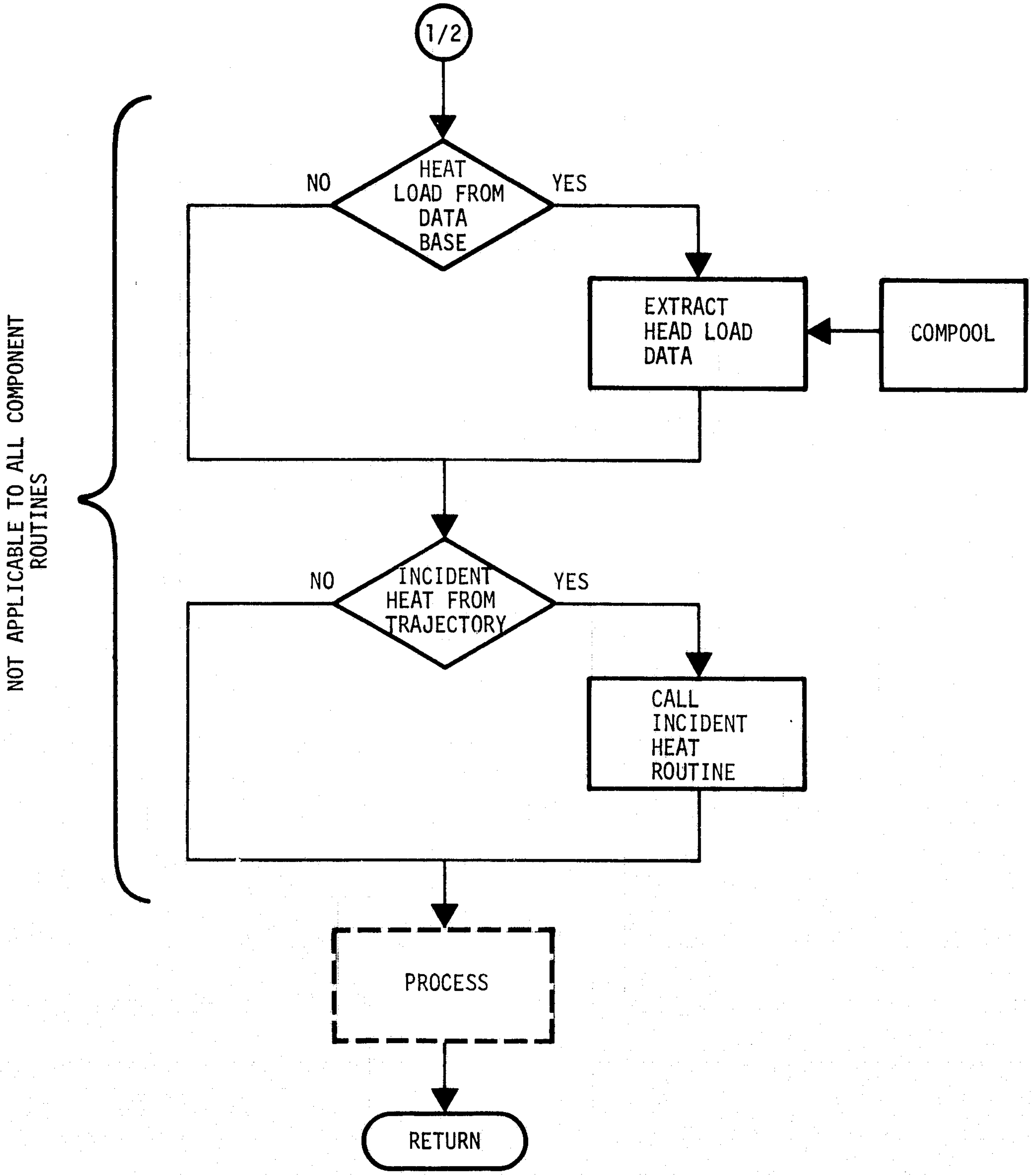

1/2
HEAT LOAD FROM DATA BASE
NO
YES
EXTRACT HEAD LOAD DATA
COMPOOL
INCIDENT HEAT FROM TRAJECTORY
NO
YES
CALL INCIDENT HEAT ROUTINE
NOT APPLICABLE TO ALL COMPONENT ROUTINES
PROCESS
RETURN

Figure 12. Routine COMPONENT X (Concluded)

The display and its operation for a typical component routine such as a cold plate is shown in Figure 13. Note that the initial display (new start) reflects the throughput coolant flow rate and coolant inlet temperature resulting from this node (10) having previously been referred to as an outlet from another processing component. The default initial component temperature (plate temperature) is also reflected. The final stages of user modification are also shown in this figure. Here, the user has specified a change to the component thermal capacitance and heat transfer coefficient. All values resident on the display at the time of exit will be overlayed onto existing nodal data (node 10).

Figure 14 presents a similar type display for a modulation valve. This is an example of a component for which numerous variables referenced in the call sequence of FEAR are loaded or modified during execution on SON OF FEAR.

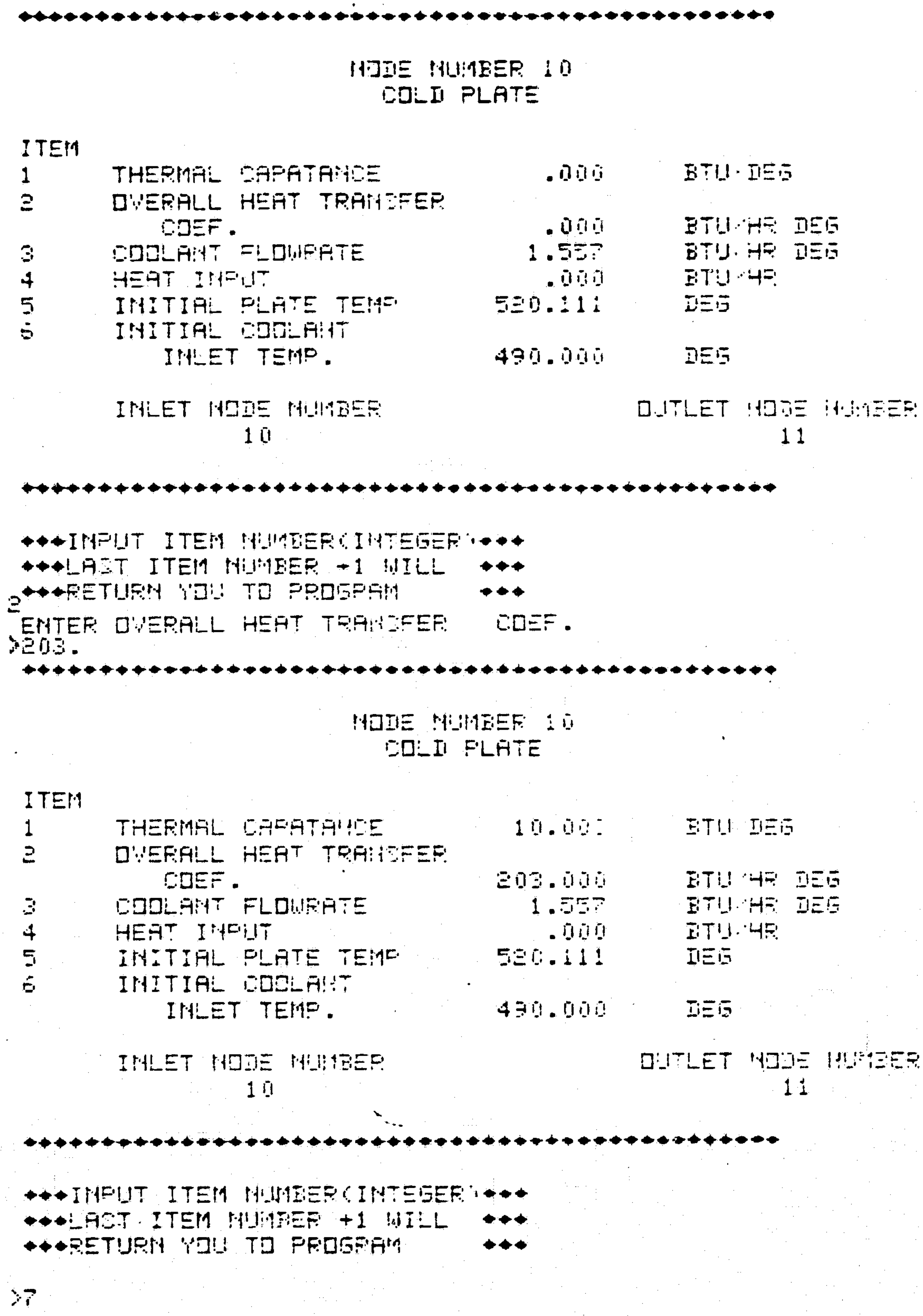
Figure 15 presents the display and its operation for the mixing of flow from two nodes (11 and 12) into a single node (1). This is an example of a component for which information was loaded by DIMENSION and DATA statements entered by the user in the MAIN of FEAR. In SON OF FEAR the information is entered or modified during execution and stored as nodal data. Note that for initial execution the user is not allowed to exit from this display until the mixed nodes are defined.

The current library of component routines in FEAR may need to be expanded to incorporate more advanced types of ECLSS components. Such advanced components fall in two categories:

Category A - Those components which are a direct processing of the atmospheric stream. (Such as the condensing heat exchanger and the lithium hydroxide carbon dioxide removal component in FEAR).

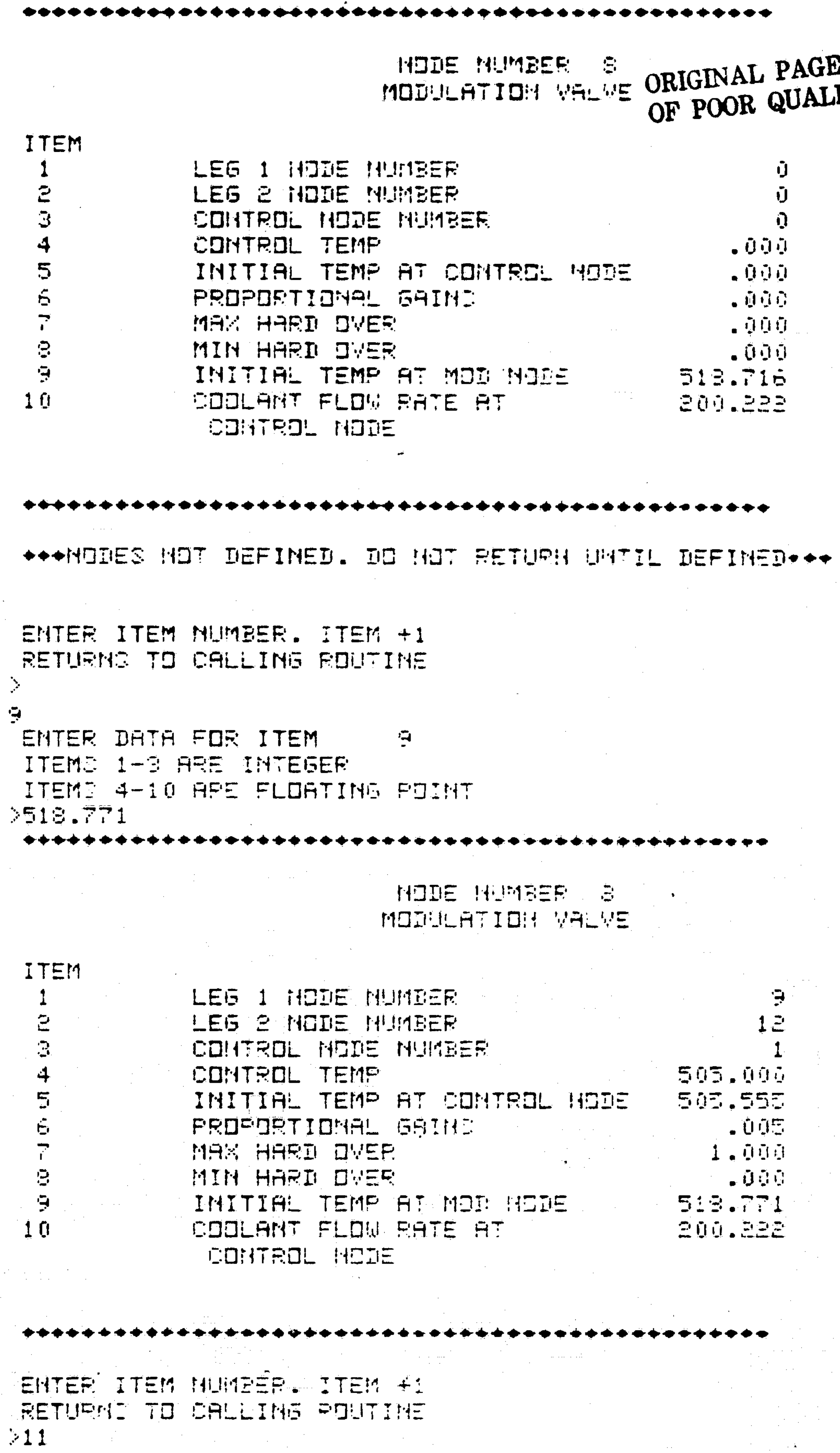
Category B - Those components whose heat of reaction interfaces with the coolant loop and consumes and/or produces chemical constituents. (Such as the evaporator component, a consumer of $H_2O$, in FEAR).

The growth potential to add performance of these advanced components to the library of subroutines is basically inherent in the FEAR since both categories are currently incorporated. Twelve of these more advanced types



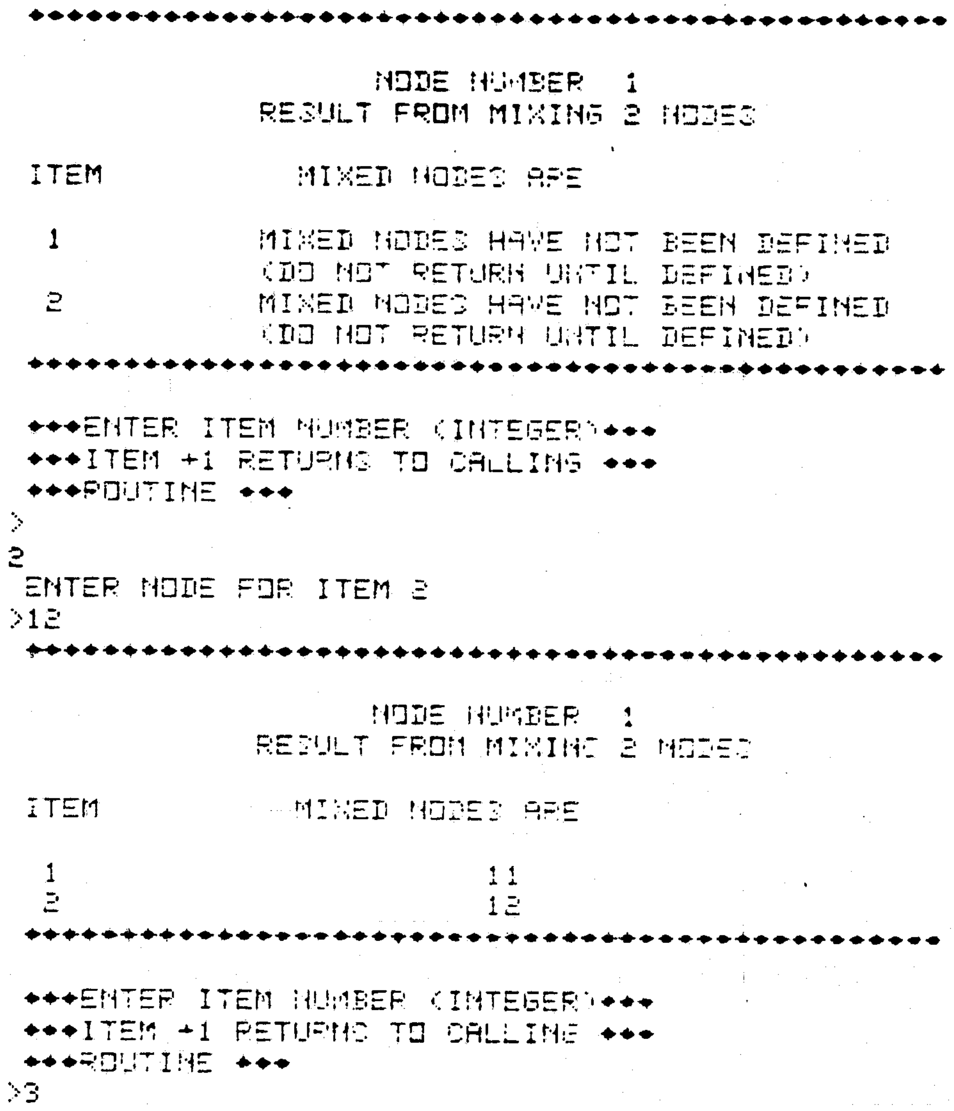
```
******************************************************

                    NODE NUMBER 10
                      COLD PLATE

ITEM
1      THERMAL CAPATANCE                .000     BTU/DEG
2      OVERALL HEAT TRANSFER
          COEF.                         .000     BTU/HR DEG
3      COOLANT FLOWRATE                1.557     BTU/HR DEG
4      HEAT INPUT                       .000     BTU/HR
5      INITIAL PLATE TEMP            520.111     DEG
6      INITIAL COOLANT
          INLET TEMP.                490.000     DEG

       INLET NODE NUMBER                     OUTLET NODE NUMBER
             10                                       11

******************************************************

***INPUT ITEM NUMBER(INTEGER)***
***LAST ITEM NUMBER +1 WILL   ***
***RETURN YOU TO PROGRAM      ***
2
ENTER OVERALL HEAT TRANSFER   COEF.
>203.
******************************************************

                    NODE NUMBER 10
                      COLD PLATE

ITEM
1      THERMAL CAPATANCE              10.001     BTU/DEG
2      OVERALL HEAT TRANSFER
          COEF.                      203.000     BTU/HR DEG
3      COOLANT FLOWRATE                1.557     BTU/HR DEG
4      HEAT INPUT                       .000     BTU/HR
5      INITIAL PLATE TEMP            520.111     DEG
6      INITIAL COOLANT
          INLET TEMP.                490.000     DEG

       INLET NODE NUMBER                     OUTLET NODE NUMBER
             10                                       11

******************************************************

***INPUT ITEM NUMBER(INTEGER)***
***LAST ITEM NUMBER +1 WILL   ***
***RETURN YOU TO PROGRAM      ***

>7
```

Figure 13. Typical Component Data Display and Operation - Cold Plate



```
****************************************************

                            NODE NUMBER  8
                          MODULATION VALVE

ITEM
 1          LEG 1 NODE NUMBER                          0
 2          LEG 2 NODE NUMBER                          0
 3          CONTROL NODE NUMBER                        0
 4          CONTROL TEMP                            .000
 5          INITIAL TEMP AT CONTROL NODE            .000
 6          PROPORTIONAL GAINS                      .000
 7          MAX HARD OVER                           .000
 8          MIN HARD OVER                           .000
 9          INITIAL TEMP AT MOD NODE             518.716
10          COOLANT FLOW RATE AT                 200.222
             CONTROL NODE


****************************************************

***NODES NOT DEFINED. DO NOT RETURN UNTIL DEFINED***


 ENTER ITEM NUMBER. ITEM +1
 RETURNS TO CALLING ROUTINE
>
9
 ENTER DATA FOR ITEM     9
 ITEMS 1-3 ARE INTEGER
 ITEMS 4-10 ARE FLOATING POINT
>518.771
****************************************************

                            NODE NUMBER  8
                          MODULATION VALVE

ITEM
 1          LEG 1 NODE NUMBER                          9
 2          LEG 2 NODE NUMBER                         12
 3          CONTROL NODE NUMBER                        1
 4          CONTROL TEMP                         505.000
 5          INITIAL TEMP AT CONTROL NODE         505.555
 6          PROPORTIONAL GAINS                      .005
 7          MAX HARD OVER                          1.000
 8          MIN HARD OVER                           .000
 9          INITIAL TEMP AT MOD NODE             518.771
10          COOLANT FLOW RATE AT                 200.222
             CONTROL NODE


****************************************************

 ENTER ITEM NUMBER. ITEM +1
 RETURNS TO CALLING ROUTINE
>11
```

Figure 14. Typical Component Data Display and Operation - Modulation Valve



```
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

                    NODE NUMBER  1
              RESULT FROM MIXING 2 NODES

 ITEM             MIXED NODES ARE

  1             MIXED NODES HAVE NOT BEEN DEFINED
                (DO NOT RETURN UNTIL DEFINED)
  2             MIXED NODES HAVE NOT BEEN DEFINED
                (DO NOT RETURN UNTIL DEFINED)
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

 ◆◆◆ENTER ITEM NUMBER (INTEGER)◆◆◆
 ◆◆◆ITEM +1 RETURNS TO CALLING ◆◆◆
 ◆◆◆ROUTINE ◆◆◆
>
2
 ENTER NODE FOR ITEM 2
>12
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

                    NODE NUMBER  1
              RESULT FROM MIXING 2 NODES

 ITEM             MIXED NODES ARE

  1                        11
  2                        12
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

 ◆◆◆ENTER ITEM NUMBER (INTEGER)◆◆◆
 ◆◆◆ITEM +1 RETURNS TO CALLING ◆◆◆
 ◆◆◆ROUTINE ◆◆◆
>3
```

Figure 15. Typical Component Data Display and Operation - Mixing of Fluid Branches

of components are presented in Reference 3. The reference includes the processing algorithm (math model) for each component. As such, these types of components as well as a fuel cell could be added to FEAR.

The growth potential for an orderly addition of these types of components to SON OF FEAR is accomplished by incorporating the following requirements:

1) Expand nodal data array to include rate of mass production (consumption) for:

| | |
|---|---|
| $H_2O$ | Potable water |
| $H_2O$ | Water |
| $CO_2$ | Carbon dioxide |
| $O_2$ | Oxygen |
| $H_2$ | Hydrogen |
| $N_2$ | Nitrogen |
| $CH_q$ | Methane |
| $H_2O_2$ | Hydrogen peroxide |
| $NH_3$ | Ammonia |

2) Provide for a general tankage routine for integration of rate of mass production (consumption).

3) Modify the current FEAR, molecular sieve, condensing heat exchanger and evaporator performance routines to interface with the nodal data array of 1) and tankage routine of 2).

### 2.1.3 Input/Output Utility Routines

Input utility routines are modified such that the data storage allocation is assigned by the routine rather than by a DIMENSION statement in the MAIN as in FEAR. In addition the data are to be entered during execution (or recovered from a restart) through an interactive display.

Typical operation with the TABLE display routine is shown in Figure 16. The user has merely specified the table number and the instantaneous storage locations for the data look-up in the MAIN. The number of points and tabular data are entered (modified) interactively during execution.

The function of the output utility routines PLØØT and SELECT as used in FEAR are incorporated in the control routines of SON OF FEAR and do not require reference in the user's MAIN.

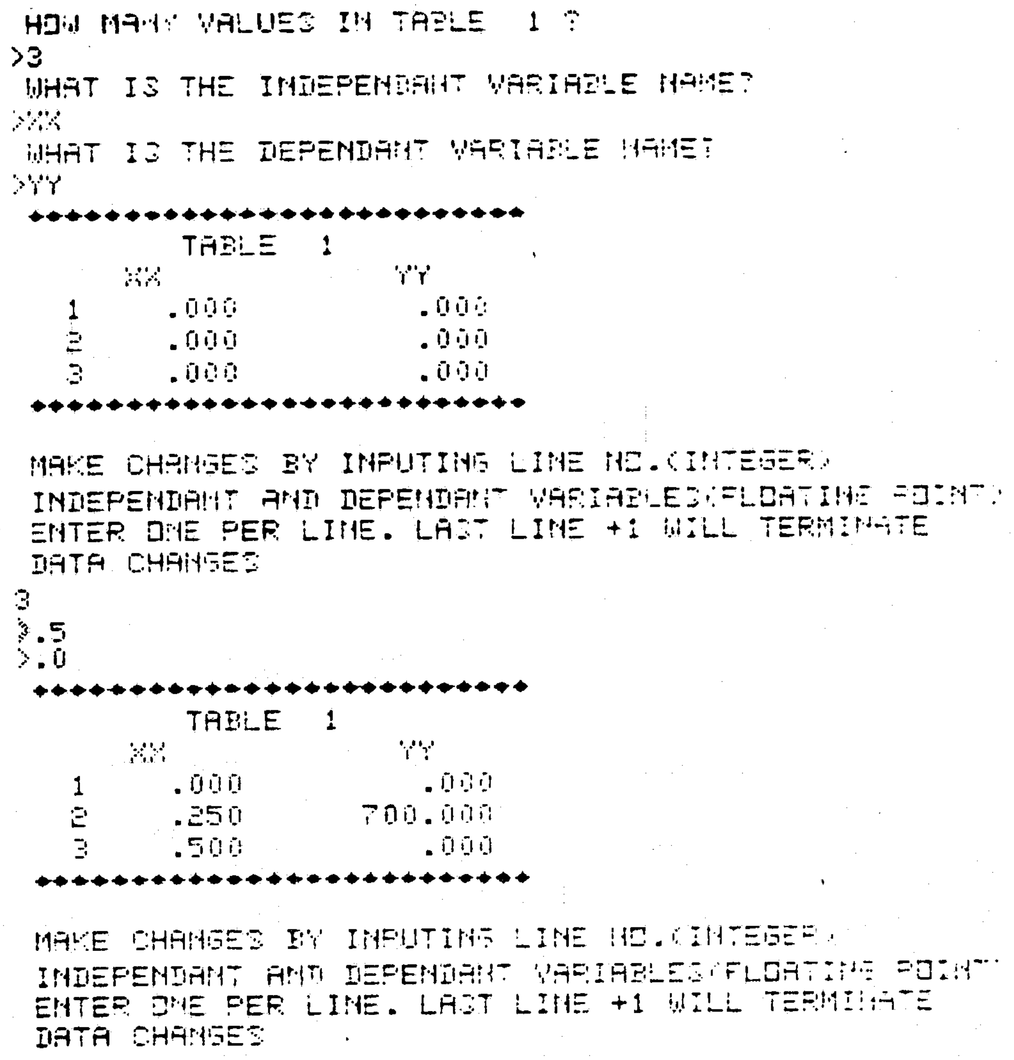
### 2.1.4 Boundary Condition Routines

The functions of the boundary condition routines from FEAR are incorporated in the control routines for SON OF FEAR. Data are entered (modified) interactively with the display for the affected component in SON OF FEAR.

The electrical power data are read into a dummy array from the PRINT routine. Each component extracts and assigns the appropriate data when processing that component. The assembly assignment is entered interactively with the component display for each node.

Node coupling data, position data, and shadowing information are entered (modified) interactively for each affected node. The appropriate assembly and subsequent calls to the orbital and incident heat routines are executed from routine PRINT. The user does not reference these routines in the MAIN as with FEAR.

The shadowing technique use in FEAR assumes that all energy incident on the shadowing node is to be subtracted from the shadowed node value. This assumption is increasingly valid as the distance between the two nodes approaches zero. The resulting evaluation is conservative with respect to shadowing since the shadowing is in effect for a solid angle of 2π steradians centered about the normal to the shadowing node. In the real physical world the finite stand-off distance and relative location results in a reduction of the solid angle for which shadowing occurs.



```
 HOW MANY VALUES IN TABLE   1 ?
>3
 WHAT IS THE INDEPENDANT VARIABLE NAME?
>XX
 WHAT IS THE DEPENDANT VARIABLE NAME?
>YY
 ***************************
          TABLE   1
      XX               YY
   1     .000            .000
   2     .000            .000
   3     .000            .000
 ***************************

 MAKE CHANGES BY INPUTING LINE NO.(INTEGER)
 INDEPENDANT AND DEPENDANT VARIABLES(FLOATING POINT)
 ENTER ONE PER LINE. LAST LINE +1 WILL TERMINATE
 DATA CHANGES
3
>.5
>.0
 ***************************
          TABLE   1
      XX               YY
   1     .000            .000
   2     .250         700.000
   3     .500            .000
 ***************************

 MAKE CHANGES BY INPUTING LINE NO.(INTEGER)
 INDEPENDANT AND DEPENDANT VARIABLES(FLOATING POINT)
 ENTER ONE PER LINE. LAST LINE +1 WILL TERMINATE
 DATA CHANGES
```

Figure 16. Typical Table Data Display and Operation

SON OF FEAR will incorporate input provisions for definition of the vector separating the shadowing and shadowed nodes. These data will be used to evaluate and apply the solid angle for which shadowing is in effect.

## 2.2 ELECTRICAL POWER SYSTEM ANALYSIS PROGRAM

The objectives of EPS analysis are best satisfied by an interactive program similar in structure to the ECLSS program of the previous section. However, EPS models, in general, do not encompass as wide a variety of components and configurations as do ECLSS models. Accordingly, it is not necessary that the user be required to develop and supply the driver (MAIN). The EPS network description, component characteristic data, and boundary condition are entered interactively during execution.

The execution procedure for an interactive EPS program is shown in Figure 17. The following defines user interface and internal processing for such a computer program.

### 2.2.1 User Interface

Circuit Description - The circuit description is entered interactively by specifying each component name, a component code number and the hook up to the rest of the circuit. A simple EPS schematic and the interactive circuit description procedure are shown in Figure 18.

The line number shown in Figure 18 is program generated and used for update and edit control. The component name is user specified. The name is used in the output to label performance data and/or the schematic in reference to that component line number. The component code is an indicator as to the type of component. In the example in Figure 18 the code ten (10) is used for batteries; six (6) for resistive line loss, and one (1) for a power consuming component. The nodal hook-up for each component is specified by junction (node) names in the order of the normal current flow direction. The junction name is used for output as for the component name. However, for components the calculations are controlled by the line number and the component name is merely a label. Consequently, two distinct components may have the same name. The junction name uniquely defines a

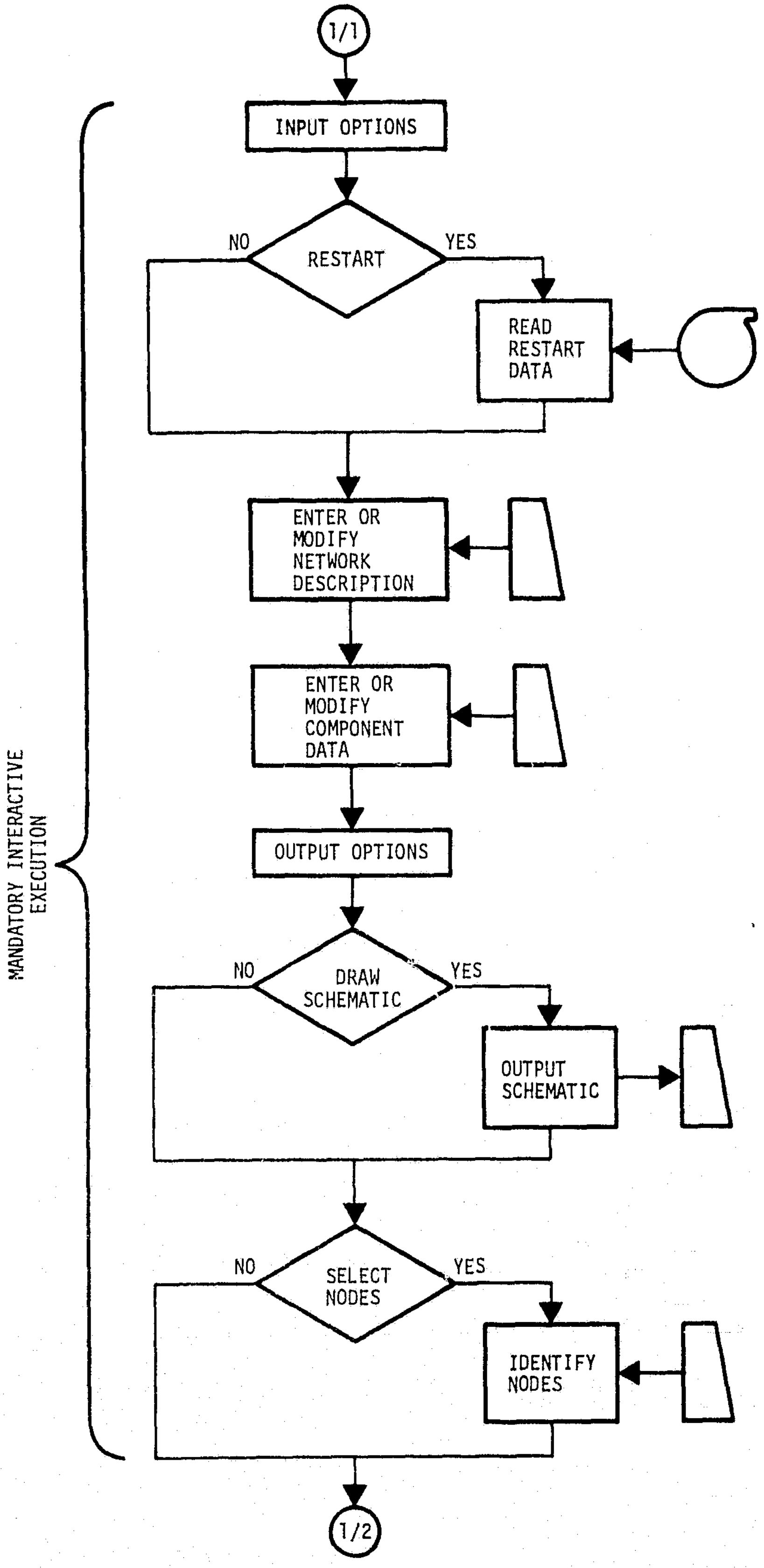


Figure 17. EPS Program Execution Procedure

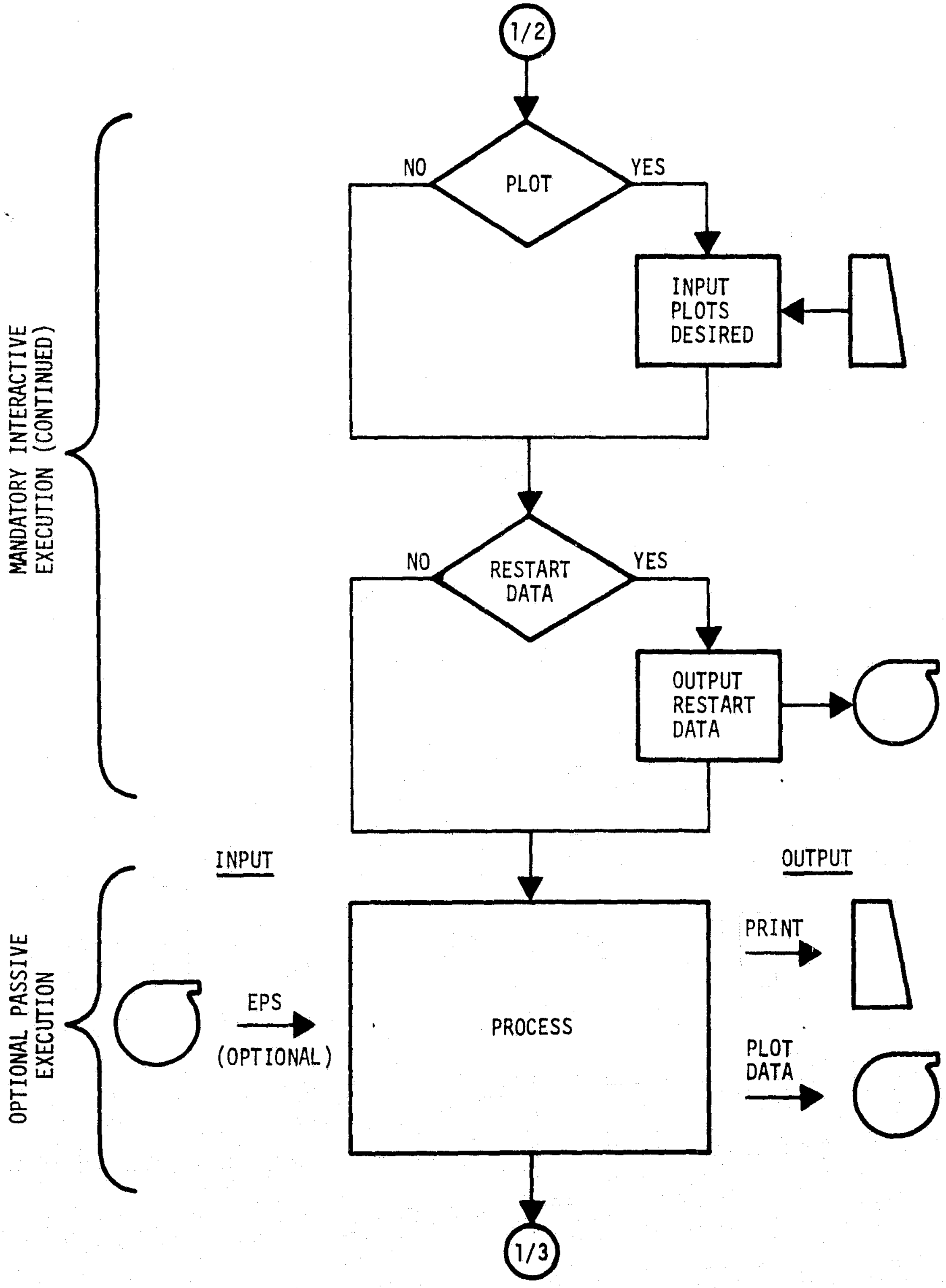


Figure 17. EPS Program Execution Procedure (Continued)

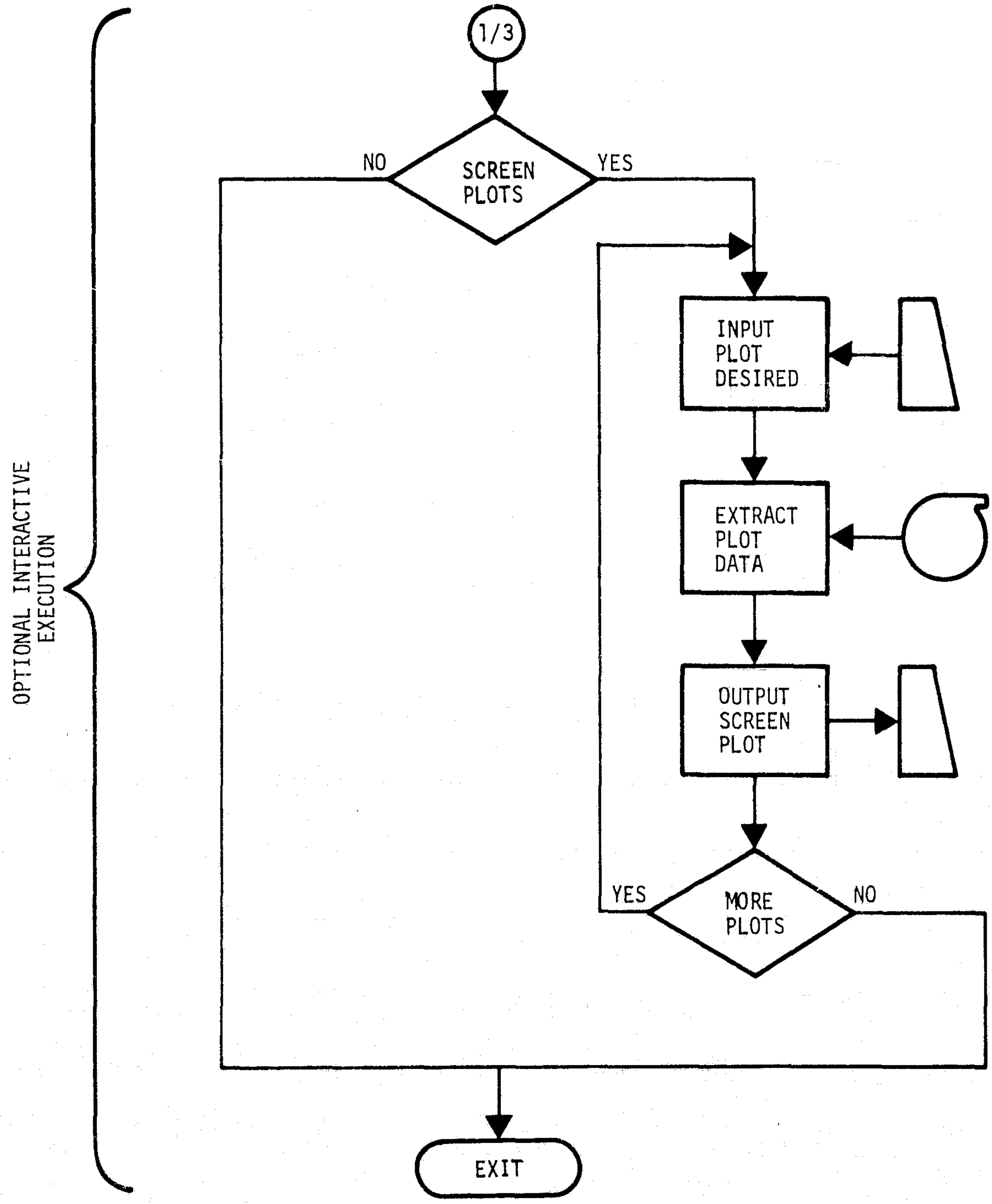
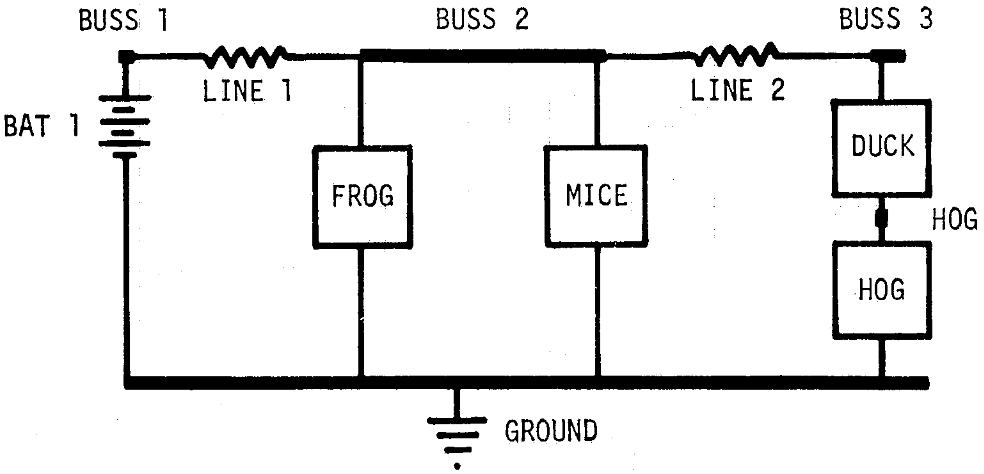


Figure 17. EPS Program Execution Procedure (Concluded)



| LINE NUMBER | COMPONENT NAME | COMPONENT CODE | NODAL HOOK UP | |
|---|---|---|---|---|
| 1 | BAT 1 | 10 | GROUND | BUSS 1 |
| 2 | LINE 1 | 6 | BUSS 1 | BUSS 2 |
| 3 | FROG | 1 | BUSS 2 | GROUND |
| 4 | MICE | 1 | BUSS 2 | GROUND |
| 5 | DUCK | 1 | BUSS 3 | HOG |
| 6 | LINE 2 | 6 | BUSS 2 | BUSS 3 |
| 7 | HOG | 1 | HOG | GROUND |

Figure 18. Typical Network Definition Procedure

junction. Once a junction name has been entered, subsequent reference to that name will assume the component (specified by that line number) is connected to the junction previously referenced by the same name.

Component Characteristic Data - Upon completion of the circuit description the component line numbers are sequenced to bring up the appropriate display for that component. The display brought up is keyed by the component code and is unique for that type of component. Typical component types include switches, fuel cells, power conditioning equipment, etc.; in addition to the types referenced in the previous paragraphs. Component characteristic data and boundary condition indicators are entered interactively as for ECLSS components in SON OF FEAR.

Boundary Condition Data - The power input to a power consuming component may be controlled by tabular data of power versus time, internal switching, or position of a switch component in series with the power consuming component. The procedural flow diagram for the boundary condition input is shown in Figure 19.

If tabular power data have been indicated in the component characteristic data, the program will request the data unless previously defined. Previously defined tabular data may be brought to the display for review and/or modification by the user. The tabular data of power versus time are referenced by table number such that there is no restriction to the number of components using the same table. The component characteristic data include a multiplying factor such that the same curve may be used for different power levels of various components. The tabular power data option will normally be used when it is desirable, to group several system components into a single nodel component.

All power consuming component models of the program have internal switching controllable by tabular data. The component is normally "off" unless one or more of the referenced tables specifies "on". This option is used to power a component with respect to a group of N activities or configurations. Note that this is an "ON IF AND/OR" logic in the sense that the component is "on" when any of the activities or configurations are in

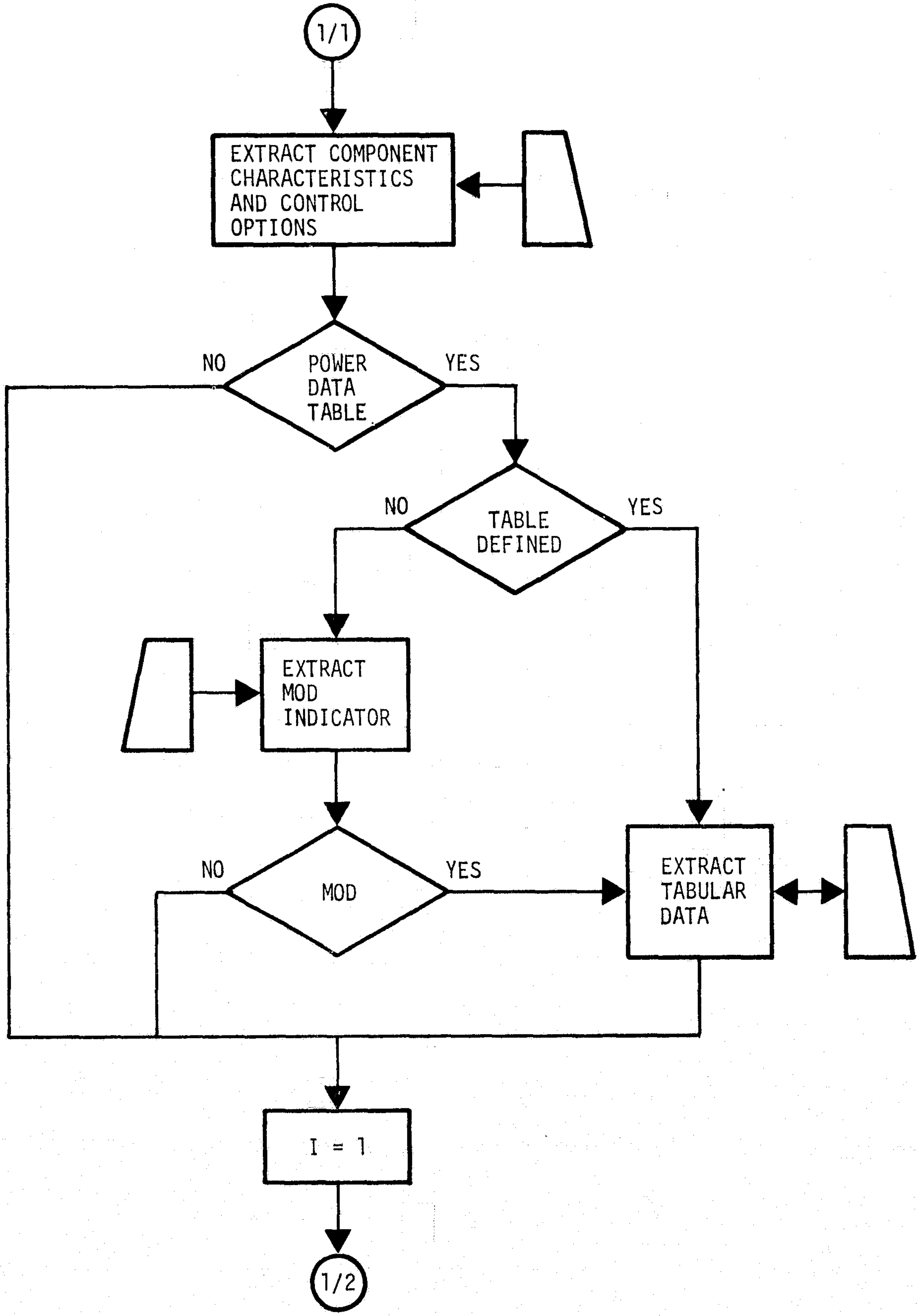


Figure 19. Boundary Condition Input Procedure

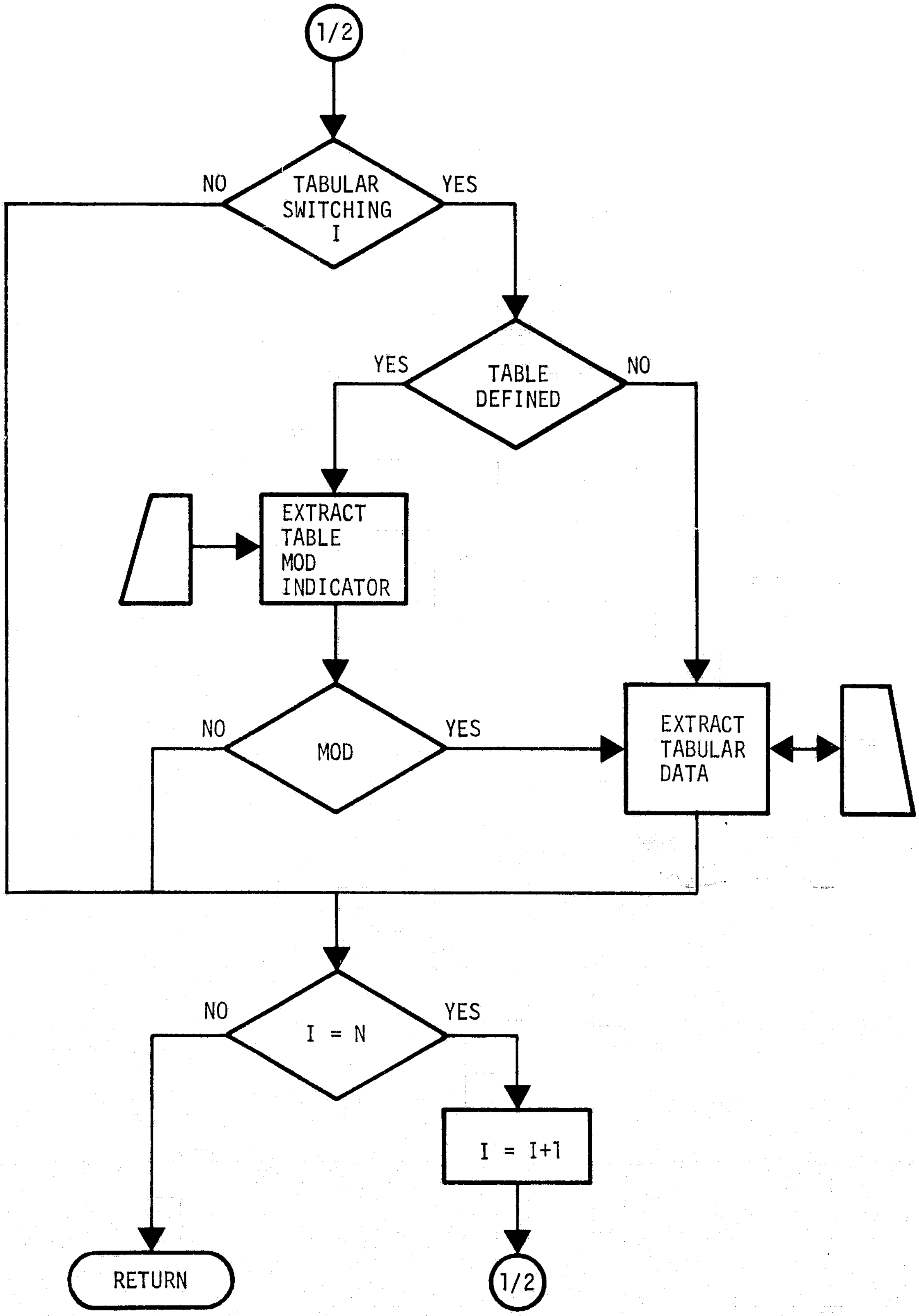


Figure 19. Boundary Condition Input Procedure (Concluded)

effect. A component may reference the converse of a table. For this option the program will use the converse of the on/off table data before applying the "ON IF AND/OR" logic. This latter option reduces the amount of tabular data required in cases where components are alternately switched.

A switch may be considered as an independent component. The switch component uses on/off tabular data as for a power consuming component except that the logic is "OFF IF AND/OR". That is, a switch is considered normally "on". It is "off" when so specified by any of the referenced tabular data. Note that an "off" positioned switch in series with power consuming components will override any "on" logic as well as tabular power data in effect for those components.

### 2.2.2 Internal Processing

The performance of the EPS defined by the circuit description, component characteristics, and boundary conditions is evaluated by the following internal processing.

1) A set of cross reference parameters is established for the elements of a characteristic determinant and boundary condition vector. These matrices model the set of simultaneous equating for a resistor equivalent network. Similar cross-reference parameters for construction of a schematic are also established.

2) The independent variables (times) for all input tables are sequenced into an event array. This array is used to establish the sequential times at which the performance is to be evaluated.

3) A quasi-steady-state solution of the set of simultaneous equations modeled by the characteristic determinant and boundary condition vector is obtained for each time point in the event array. Characteristic determinant and boundary condition vector element values are obtained by extracting equivalent resistor data at the augmented time from routines associated with each component. The particular component affecting a given column or row of the matrices is obtained from the cross-reference parameters. Iteration is performed if required to obtain the solution.

# 3.0 REFERENCES

1. "Fortran Environmental Analysis Routines (FEAR)", Volume I, User's Guide, TRW Systems, June 1974.

2. Besserer, C. W., and Walquist, R. L.: "SEID Software Development Policies," TRW IOC, December 29, 1976.

3. Graf, P., et al., "Generalized Consumables Budgeting Program for Advanced ECS Subsystems," TRW Houston, January 1971.

# APPENDIX

## FORTRAN ENVIRONMENTAL ANALYSIS ROUTINES (FEAR) BASIC PROGRAM DESCRIPTION

The FEAR computer program is a library of math models and support routines designed to analyze ECLSS. The program was originally developed in support of Apollo and Skylab consumables and subsystems/systems analysis and is currently being utilized in support of Shuttle analysis. The FEAR Program utilizes a group of related routines for which the user writes the control program (MAIN) to conduct an analysis of a particular ECLSS system. These routines are general as regards ECLSS analysis, except for those which read input tapes in a specific format. This Appendix, abstracted from Reference 1, provides a brief description of the program.

The Basic Program includes a common block, program control routines, convergence control, component simulation routines, and various utility and boundary condition routines. A summary of the general routines is given in Table A-1. Several of the component routines have gas processing entry points which expand the list on Table A-1 to include atmospheric analysis. Typical MAIN requirements are shown in Table A-2.

The common block referenced in Table A-2 serves to provide various program controls and routine communication with a minimum of variables referenced in the call sequence to routines. The first card provides timing indices (IPREV and IPRES), the calculating time increment (DELT), time (TIME), the stop time for the run (TSTOP) and the number of nodes involved in the run (NNODES). Data storage for up to 300 nodes is provided by the next card wherein the first subscript of the three variables is the node number. The first two variables P and F are time dependent variables which require storage of both the current and previous value.

The timing indices are used in the second subscript to establish the time status of that storage location. The remaining entries (C) are storage locations for instantaneous node parameters. The value in these storage locations at the time of an execution (involving that node and parameters) is in effect for that calculation. The latter card (IDUM and DUM)

Table A-1. Summary of General Routines

| | |
|---|---|
| PROGRAM CONTROL | |
| START | Sets up for reading of input data in specified format and initializes problem. |
| PRINT | Provides print-out in specified format, updates timing increment and sums the expendables. |
| BASIC COMPONENTS | |
| PLATE | Forced-cooled coldplated equipment equations. |
| PLATEC | Equations for forced-cooled coldplated equipment thermally coupled to other components or material. |
| EVAP | Evaporator performance equation. |
| EXCHC | Counter flow heat exchanger performance equation. |
| EXCHP | Parallel flow heat exchanger performance equations. |
| RAD1 | Equations for radiator panel using basic fin effectiveness equations. |
| RAD2 | Equations for radiator panel using simplified fin effectiveness equations. |
| RAD3 | Equations for radiator using constant fin effectiveness. |
| RAD1C<br>RAD2C<br>RAD3C | Same as RAD1, 2, 3 except provides for thermal coupling to other components. |
| HEATER | Equations for in-line on/off heater controlled in response to prescribed control temperature station and deadband. |
| THCAP | Equations for thermal capacitor utilizing melting media. |
| CONSUM | Equations for a condensing heat exchanger utilizing a sublimator as the ultimate heat sink. |
| CONXCH | Equations for condensing counter flow heat exchanger. |

Table A-1.· Summary of General Routines (Continued)

| | |
|---|---|
| BASIC COMPONENTS | |
| PASS | Equation for isothermal passive (structural) component. |
| ATM01 | Process internal heat, moisture and carbon dioxide generation in an oxygen and nitrogen atmospheric compartment. |
| SPLIT | Equations to branch coolant flow. |
| MIX | Equations of mixing coolant flow from various branches. |
| MOD | General equations to proportion flow split between two branches in response to prescribed temperature, location and proportional gains. |
| AVG | Sets up dummy node whose temperature is average of two prescribed nodal temperatures. |
| DIFF | Sets up dummy node with temperature equal to difference between two prescribed nodal temperatures. |
| SIEVE | Performance for a molecular sieve type of $CO_2$ removal unit. |
| TRANS | Used to transfer data to an additional node. |
| FLO | On/off pump or fan performance controlled to a set point temperature. |
| LOAD | Calculates net heat to a node. |
| ALIOH | Performance for a LiOH type $CO_2$ removal unit. |
| ATM02 | Same as ATM01 plus control of oxygen and nitrogen. |
| UTILITY (INPUT) | |
| STEP | Provides for input of dependent variable as step function of independent variable. |
| TABLE | Provides for input of curve (2-D) data points; uses linear interpolation between points. |
| TABLE2 | Provides for input of parametric curves (3-D); uses linear interpolation. |

Table A-1. Summary of General Routines (Continued)

| | |
|---|---|
| UTILITY (INPUT) | |
| PERIOD | Same as TABLE except function is periodic and need be specified over only one period. |
| CHANGE | Provides for dynamic updating of node coupling values. |
| UTILITY (OUTPUT) | |
| PLOOT | Provide CALCOMP plots of selected parameters. |
| SELECT | Replaces control Routine PRINT for selective output. |
| BOUNDARY CONDITION | |
| ABHEAT | Calculates absorbed heat using specified trajectory tape or calculates absorbed heat resulting from orbit derived from input orbital parameters. |
| SHADOW | Simplified shadowing routine referenced to node numbers. |
| SHASTR | Simplified shadowing routine referenced to location of nodes in specified array (entry point to SHADOW). |
| EPREAD | Assigns heat load to various coldplates from a specifically formatted EPS heat load tape. |
| GENEPS | Assigns heat load to various coldplates from a specifically formatted heat load tape. |
| STOREQ | Assigns prescribed heat load to additional nodes. |
| CONVERGENCE CONTROL | |
| LOOP | Defines starting node in coolant loop. |
| CONVRG | Convergence routine for closed loop simulation. |

Table A-2. Typical MAIN Requirements

```
      COMMON IPREV,IPRES,DELT,TIME,TSTOP,NNODES
      COMMON P(300,2),F(300,2),C(300,20)           COMMON BLOCK
      COMMON DUM(100),IDUM(100)

      DATA IKNOW/....                              DATA REQUIRED
                                                   BY ROUTINES
      ....                               ..../

      CALL START                                   PROGRAM CONTROL

  333 CONTINUE

      CALL ABHEAT(....    )
                                                   CALL TO UTILITY
      ....                                         OR BOUNDARY CON-
                                                   DITION ROUTINES
      CALL STEP(....    )

 3133 CONTINUE

      CALL LOOP(....    )                          CONVERGENCE CONTROL

      CALL PLATE(....    )

      ....                                         CALL TO COMPONENT
                                                   ROUTINES

      CALL MIX(....    )

      CALL CONVRG(....    )                        CONVERGENCE CONTROL

      CALL PRINT                                   PROGRAM CONTROL AND
      GO TO 333                                    TIMING UPDATE

      END
```

provides special storage arrays which are used for internal program control. The parameters storage scheme shown on Table A-3 is used unless otherwise noted in the detailed component routine description (Reference 1).

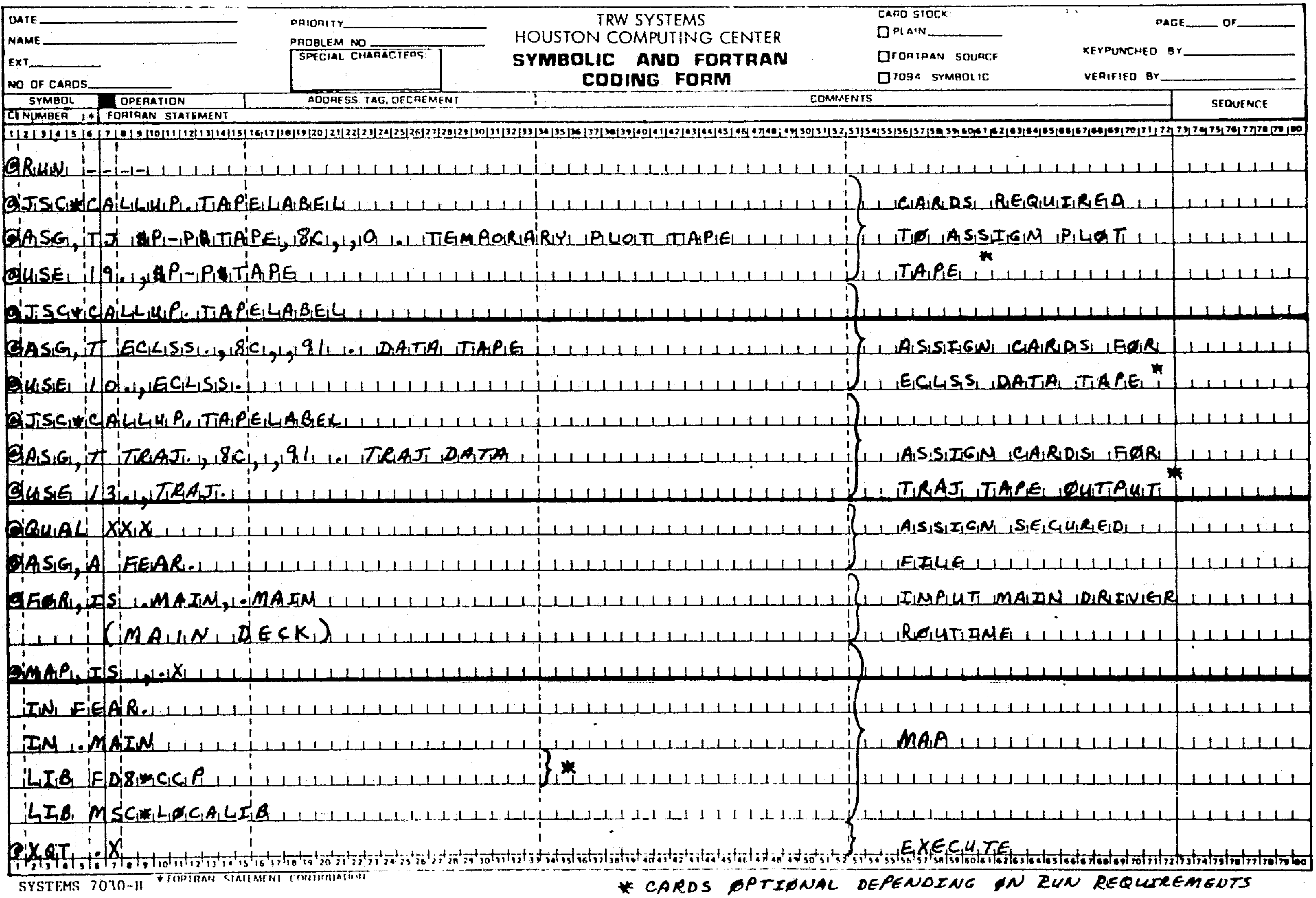
Several of the component simulation and various utility and boundary condition routines require information in addition to that provided by the common block. Entry of this type of data is indicated below the common block in Table A-2. The remaining entries shown in Table A-2 are typical of the relationship of the various routines in the FEAR library.

Typical deck set-up for a batch mode execution is shown in Table A-4. The partial set-up illustrated is for the Univac 1110 EXEC 8. The component initial data and plot control data are loaded as fixed format cards read in during execution. The node data are the initial values of the parameters shown in Table A-3 and are required for each component referenced in the MAIN.

Table A-3. General Storage Allocation

| | | |
|---|---|---|
| P(I,X) | Component Temperature | |
| F(I,X) | Inlet Temperature | |
| C(I,1) | Thermal Capacity | |
| C(I,2) | Heat Transfer Coefficient | |
| C(I,3) | Dynamic Thermal Capacity | |
| C(I,4) | Heat Rate or Heat Flux | |
| C(I,5) | Specific Heat | Gas Parameters - Applies to Gas Processing Routines Only. |
| C(I,6) | Concentration of $H_2O$ | |
| C(I,7) | Concentration of $N_2$ | |
| C(I,8) | Concentration of $O_2$ | |
| C(I,9) | Concentration of $CO_2$ | |
| C(I,10) | (Varies in Different Routines) | |
| C(I,11) | Partial Pressure of $H_2O$ | |
| C(I,12) | Partial Pressure of $N_2$ | |
| C(I,13) | Partial Pressure of $O_2$ | |
| C(I,14) | Partial Pressure of $CO_2$ | |

## Table A-4. Univac 1110 EXEC 8 Deck Set-Up for FEAR Program



DATE ____ PRIORITY ____ NAME ____ PROBLEM NO ____ EXT ____ SPECIAL CHARACTERS: ____ NO OF CARDS ____

TRW SYSTEMS
HOUSTON COMPUTING CENTER
**SYMBOLIC AND FORTRAN CODING FORM**

CARD STOCK:
□ PLAIN ____
□ FORTRAN SOURCE
□ 7094 SYMBOLIC

PAGE ____ OF ____
KEYPUNCHED BY ____
VERIFIED BY ____

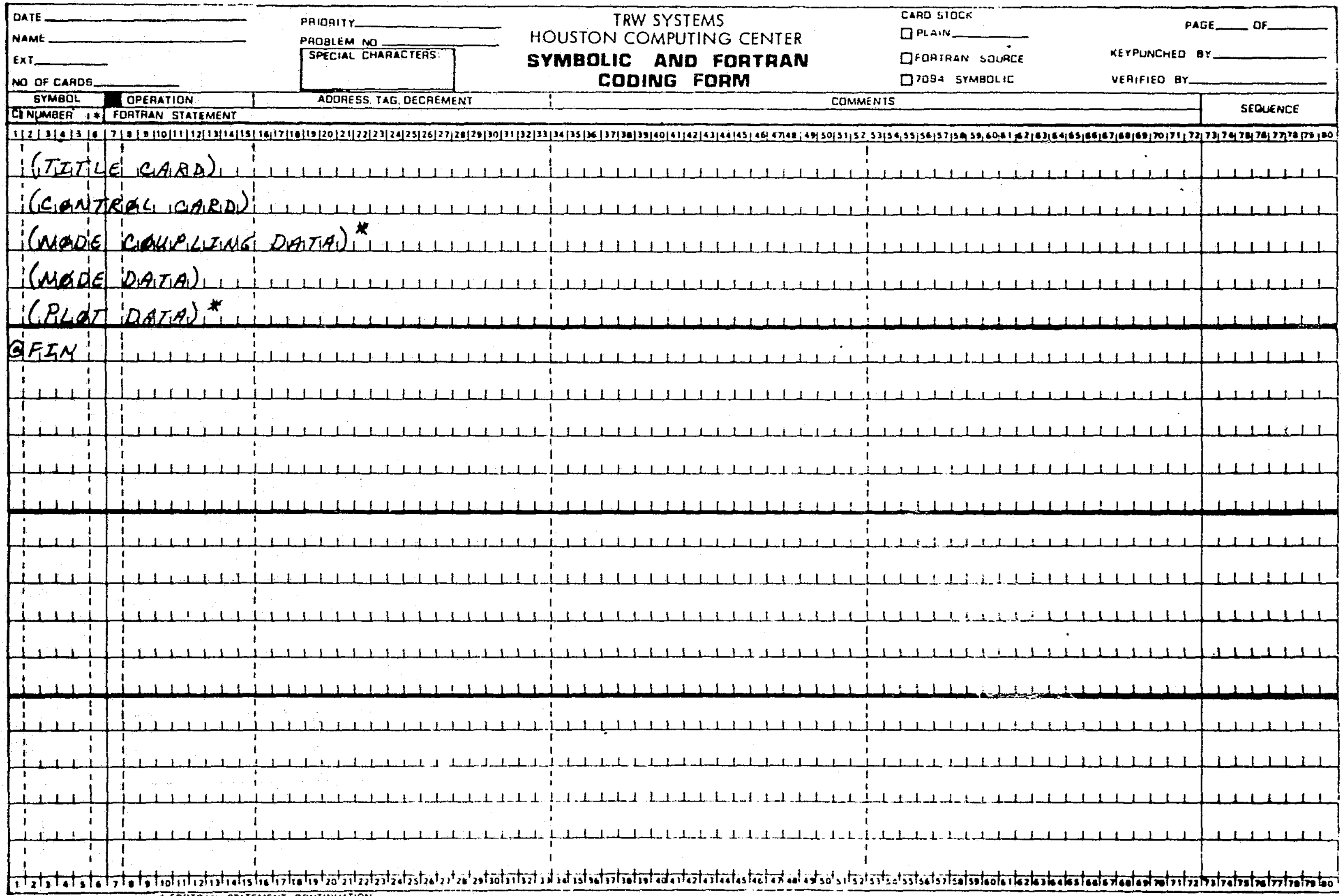
SYMBOL | OPERATION | ADDRESS, TAG, DECREMENT | COMMENTS | SEQUENCE
C NUMBER | * | FORTRAN STATEMENT

```
@RUN  - - - -
@JSC*CALLUP.TAPELABEL                                }  CARDS REQUIRED
@ASG,TJ 8P-PATAPE,8C,,0 . TEMAORARY PLOT TAPE        }  TØ ASSIGN PLØT
@USE 19.,8P-PØTAPE                                   }  TAPE *
@JSC*CALLUP.TAPELABEL                                }
@ASG,T ECLSS.,8C,,91 . DATA TAPE                     }  ASSIGN CARDS FØR
@USE 10.,ECLSS.                                      }  ECLSS DATA TAPE *
@JSC*CALLUP.TAPELABEL                                }
@ASG,T TRAJ.,8C,,91 . TRAJ DATA                      }  ASSIGN CARDS FØR
@USE 13.,TRAJ.                                       }  TRAJ TAPE ØUTPUT *
@QUAL XXX                                            }  ASSIGN SECURED
@ASG,A FEAR.                                         }  FILE
@FØR,IS .MAIN,.MAIN                                  }  INPUT MAIN DRIVER
      (MAIN DECK)                                    }  RØUTINE
@MAP,IS ,.X                                          }
 IN FEAR.                                            }
 IN .MAIN                                            }  MAP
 LIB FD8*CCP                     } *                 }
 LIB MSC*LØCALIB                                     }
@XQT .X                                              }  EXECUTE
```

SYSTEMS 7030-II   * FORTRAN STATEMENT CONTINUATION

* CARDS ØPTIØNAL DEPENDING ØN RUN REQUIREMENTS

## Table A-4. Univac 1110 EXEC 8 Deck Set-Up for FEAR Program (Concluded)



DATE ______ PRIORITY ______ PAGE ___ OF ___

NAME ______ PROBLEM NO ______

EXT ______ SPECIAL CHARACTERS:

NO OF CARDS ______

TRW SYSTEMS
HOUSTON COMPUTING CENTER
**SYMBOLIC AND FORTRAN CODING FORM**

CARD STOCK
□ PLAIN ______
□ FORTRAN SOURCE
□ 7094 SYMBOLIC

KEYPUNCHED BY ______

VERIFIED BY ______

| SYMBOL<br>C NUMBER | OPERATION<br>FORTRAN STATEMENT | ADDRESS, TAG, DECREMENT | COMMENTS | SEQUENCE |
|---|---|---|---|---|

```
 (TITLE CARD)
 (CONTROL CARD)
 (MODE COUPLING DATA)*
 (MODE DATA)
 (PLOT DATA)*
@FIN
```

SYSTEMS 7030-II   * FORTRAN STATEMENT CONTINUATION

* CARDS OPTIONAL DEPENDING ON RUN REQUIREMENTS