AD-A068 520 COMPUTER SCIENCES CORP ALBUQUERQUE N MEX F/G 19/4
HULL SYSTEM GUIDE.(U)
JAN 79 L P GABY F29601-78-C-0012
UNCLASSIFIED CSC-C4-C-4041 AFWL-TR-78-115 NL

1 OF 3
AD
A068520

AFWL-TR-78-115

(2) LEVEL III

AFWL-TR-78-115

ADE300264

DDC

AD A068520

HULL SYSTEM GUIDE

Lewis P. Gaby

Computer Sciences Corporation
Albuquerque, New Mexico 87108

January 1979

Final Report

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

This final report was prepared by the Computer Sciences Corporation, Albuquerque, New Mexico, under Contract F29601-78-C-0012, Job Order 88091821 with the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico. Dr. Clifford E. Rhoades, Jr. (DYP) was the Laboratory Project Officer-in-Charge.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Clifford E. Rhoades, Jr.

CLIFFORD E. RHOADES, JR.
Project Officer

FOR THE COMMANDER

Norman F. Roderick

NORMAN F. RODERICK
Major, USAF
Chief, Advanced Concepts Branch

Thomas W. Ciambrone

THOMAS W. CIAMBRONE
Colonel, USAF
Chief, Applied Physics Division

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWL-TR-78-115	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) HULL System Guide		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER C4-C-4041
7. AUTHOR(s) Lewis P. Gaby		8. CONTRACT OR GRANT NUMBER(s) F29601-78-C-0012
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Sciences Corporation 1400 San Mateo Boulevard SE Albuquerque, NM 87108		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F/88091821
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (DYP) Kirtland Air Force Base, NM 87117		12. REPORT DATE January 1979
		13. NUMBER OF PAGES 212
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Radiation Hydrodynamics, Elastic Plastic, Software System

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

HULL is a system of computer programs which solve the partial differential equations of continuum mechanics. It is used to solve problems in nuclear air blast and conventional weapon effects. The system is designed to tailor the program to be executed to fit the requirements of the problem and thus optimize the execution of the program. The three major functions performed by the system are the initialization of a problem (KEEL), calculating the time evolution of the problem (HULL), and plotting the results (PULL). The HULL system is currently operational on the CDC 6600

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409 656

BLOCK 20. ABSTRACT (cont'd)

(SCOPE 3.4, NOS/BE), CDC Cyber 176 (NOS/BE, CDC 7600 (SCOPE 2.0), and IBM 370 (OS/VS2).

SUMMARY

The HULL system was initiated in 1971 at the Air Force Weapons Laboratory (AFWL) as a successor to the SHELL code to solve atmospheric response to nuclear weapons. In 1973, the code was modified to handle stress, strain and discontinuous material interface for use in calculations involving conventional weapons design at the Air Force Armaments Laboratory (AFATL).

The HULL system uses SAIL as its preprocessor to select the required code for the problem. The program PLANK is used to read the input for the problem and the problem dump tape (for HULL or PULL), and provide the information that is needed by SAIL to select the necessary code.

For the HULL system operating at the Air Force Weapons Laboratory and the Air Force Armaments Laboratory, the program BOW is used to monitor the tapes used for the problem and to keep a record of their use.

PREFACE

This manual was prepared in conjunction with HULL calculations performed for the Elastic-Plastic HULL Analytical Modeling.

The author wishes to acknowledge the extensive contributions of Major Daniel A. Matuska and Major Richard E. Durrett to the HULL system, and the assistance in preparing this guide that was received from Dr. Paul Lewis, Dr. Mark Fry, and Dr. Clifford Rhoades.

TABLE OF CONTENTS

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

ILLUSTRATIONS

Illustrations (continued)

TABLES

SECTION I

INTRODUCTION

This manual describes the use of the HULL system. HULL is a system of computer programs which approximate the solutions to the equations of continuum mechanics. The approach is effectively Eulerian in that the computational mesh is not distorted with time. The technique consists of a Lagrangian calculation where the velocity, density and energy are updated from time, t, to $t + \Delta t$ using the total derivatives, followed by a fluxing or rezone calculation of $t + \Delta t$ to retain the original mesh configuration.

Development of the HULL code began in 1971 at the Air Force Weapons Laboratory (AFWL). The code was a successor to the SHELL code and was initially used to handle calculations of atmospheric response to nuclear weapons. Major Daniel Matuska, USAF, developed a stable difference technique to replace the method used in SHELL. Most of the early calculations were made using only one material, air, although it was structured to handle multimaterial problems.

During the development of the HULL code auxiliary routines were written which initialized a problem and plotted the results of a calculation. In addition, capability to perform calculations including radiation diffusion, high explosive burn and magnetic fields were added.

In 1973, the HULL code began running calculations at the Air Force Armament Laboratory (AFATL) involving conventional weapon design. Major Matuska extended the difference method of handling stress, strain, material interfaces and failure. The architecture was also modified to conform more to the vector structure of fourth generation computers, such as the Cray-1 and STAR 100.

1. BASIC EQUATION

HULL solves the continuum mechanics differential equation which describes the behavior of linear elastic-plastic material. The basic equations in tensor form are

$$\dot{\rho} + \rho u_{i,i} = 0 \tag{1}$$

$$\rho \dot{u}_j - T_{ij,i} = \rho g_j \tag{2}$$

$$\rho \dot{E} - \left(T_{ij}\, u_j\right)_{,i} = \rho u_i g_i \tag{3}$$

which reflect the conservation of mass, momentum and energy, respectively. The quantities in the equations are

- ρ - material density (g/cm^3)
- u_j - material velocity vector (cm/s)
- T_{ij} - stress tensor (dynes/cm^2)
- g_i - body force due to gravity (cm/s^3)
- E - total specific energy (ergs/g)

The dot implies the total (Lagrangian) derivative with respect to time.

Considering the definition of E and the deviator representation for T_{ij}, we have

$$E = I + 1/2\, u_i\, u_i \tag{4}$$

$$T_{ij} = S_{ij} - \delta_{ij} p \tag{5}$$

where I is the internal specific energy (erg/g), p is the hydrostatic pressure (dynes/cm^2) obtained from the equation of state $p(\rho, I)$, S_{ij} is the stress deviator tensor which is obtained by solution of

$$\dot{S}^{e}_{ij} = 2G\left(\dot{\epsilon}_{ij} - 1/3\,\delta_{ij}\dot{u}_{i,i}\right) \tag{6}$$

where G is the bulk modulas and,

$$\dot{\epsilon}_{ij} = 1/2\left(u_{i,j} + u_{j,i}\right) \tag{7}$$

Subject to the von Mises yield criterion to include plasticity which states

$$S_{ij} = S^{e}_{ij} \quad \text{where} \quad S_{ij}S_{ij} \leq \frac{2y^2}{3} \tag{8}$$

where y is the yield or flow stress. The deviator then has the form

$$S_{ij} = S^{e}_{ij}\left(\frac{2y^2}{3S_{ij}S_{ij}}\right)^{1/2} \quad S_{ij}S_{ij} > \frac{2y^2}{3} \tag{9}$$

Finally, failure occurs when the maximum principal stress (the maximum root σ of $|T_{ij} - \sigma\delta_{ij}| = 0$) exceeds the tensile stress criteria for the material.

If we assume cylindrical symmetry, the equations become

$$\dot{\rho} + \rho\left[\frac{1}{r}\frac{\partial(ru)}{\partial r} + \frac{\partial v}{\partial r}\right] = 0 \tag{10}$$

$$\rho\dot{u} - \frac{\partial rT_{rr}}{r\partial r} - \frac{\partial T_{rz}}{\partial z} + \frac{T_{\theta\theta}}{r} = 0$$

$$\rho\dot{v} - \frac{\partial rT_{rz}}{r\partial r} - \frac{\partial T_{zz}}{\partial z} = -\rho g$$

$$\rho\dot{E} - \frac{\partial}{r\partial r}\left[r\left(uT_{rr} + vT_{rz}\right)\right] - \frac{\partial}{\partial z}\left(uT_{rz} + vT_{zz}\right) = -\rho v_g \qquad (11)$$

here r is the radial coordinate

z is the axial coordinate

u is the radial velocity component

v is axial velocity component

g is body force due to gravity in downward direction (i.e., $g_z = -g$)

The only nonzero components of the stress tensor are

$$T_{rr} = S_{rr} - p \qquad (12)$$

$$T_{zz} = S_{zz} - p \qquad (13)$$

$$T_{\theta\theta} = S_{\theta\theta} - p \qquad (14)$$

$$T_{rz} = T_{zr} = S_{rz} \qquad (15)$$

where the stress deviators are calculated from

$$\dot{S}^{e}_{rr} = 2G\left(\dot{\epsilon}_{rr} - D \right) \tag{16}$$

$$\dot{S}^{e}_{zz} = 2G\left(\dot{\epsilon}_{zz} - D \right) \tag{17}$$

$$\dot{S}^{e}_{\theta\theta} = -S_{rr} - S_{zz} \tag{18}$$

$$\dot{S}^{e}_{rz} = 2G\left(\dot{\epsilon}_{rz} \right) \tag{19}$$

$$D = 1/3 \left(1/2 \frac{\partial}{\partial r} ru + \frac{\partial v}{\partial v} \right) \tag{20}$$

$$\dot{\epsilon}_{rr} = \frac{\partial u}{\partial r} \tag{21}$$

$$\dot{\epsilon}_{zz} = \frac{\partial v}{\partial} \tag{22}$$

$$\dot{\epsilon}_{rz} = 1/2 \left(\frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \right) \tag{23}$$

The yield criteria become

$$S_{\alpha\beta} = S^{e}_{\alpha\beta} \qquad \text{if} \quad J \leq 1/3\, y^2 \tag{24}$$

with

$$J = \left(S^2_{rr} + S^2_{rz} + S^2_{zz} + S_{rr} S_{zz} \right) \tag{25}$$

and

$$S_{\alpha\beta} = S^{e}_{\alpha\beta} \left(\frac{V^2}{3J} \right) \quad J > 1/3\, y^2 \tag{26}$$

Finally, failure occurs when $\sigma \geq \sigma_{frac}$, where

$$\sigma = 1/2 \left(T_{rr} + T_{zz} \right) + \left[\left(\frac{T_{rr} - T_{zz}}{2} \right)^2 + T^2_{rz} \right]^{1/2} \tag{27}$$

and σ_{frac} is the tensile yield strength for the material.

It is not the purpose of this guide to discuss the finite difference techniques. These are discussed in a separate publication.*

*The HULL Code, A Finite Difference Solution to the Equations of Continuum Mechanics, Air Force Armaments Laboratory, Eglin AFB, FL, to be published.

2. OVERVIEW

This guide consists of seven sections. Section II discusses the system utilities used by the HULL system. Section III discusses the HULL ZBLOCK and other control functions. Section IV describes the format of the mesh in core and the format of all tapes. Section V discusses the use of KEEL to initialize a HULL problem. Section VI describes how a HULL calculation is run. Section VII describes how to plot the HULL data using PULL.

There are three appendixes: appendix A lists and describes all of the utility routines used in the HULL system; appendix B lists sample procedures for HULL on CDC NOS/BE operating cycles; and appendix C lists procedures for running HULL on the IBM 360/370 systems.

SECTION II

SYSTEM UTILITIES

Users of the HULL system will soon become acquainted with the utilities used by HULL. There are three (two on IBM systems) utility programs used in HULL problems and a library of general purpose subroutines that are used by all of the programs in the system. Therefore, these routines must be maintained or the system will not function properly. The utility programs are SAIL, BOW (not used on IBM systems) and PLANK. A typical HULL run includes execution of each of the utilities in addition to the main program.

The procedures used for operation and maintenance of the HULL system must be adapted to the machines and operating systems available. Representative examples of control statement sequences are presented in following subsections as guides for users in development of procedures for individual installations. Unless otherwise specified, examples of CDC usage are for a CDC 6600/Cyber 176 installation using the NOS/BE operating system. IBM examples are for IBM 370/168 using OS/US2-Version 2.

1. SAIL UTILITY

The SAIL preprocessor program is used to generate the HULL program and all other utilities. SAIL, itself, is maintained on a separate SAIL library file for generation and modification. When running on an existing HULL system, the user need only access the current load module for SAIL or generate a new load module by running the current SAIL program to generate a new version of SAIL. In order to initiate SAIL usage at a new installation, the source code of SAIL for the particular computer must be generated at an installation where SAIL is currently running. The source code and the SAIL library file of the SAIL program then provide the new user with the capability to use and maintain SAIL.

Since the SAIL program and most of its inputs are described in another document*, only the additional feature of the alternate input file will be discussed here. The user should acquaint himself with SAIL before attempting to utilize the SAIL alternate input file.

The SAIL program receives its initial instructions from the primary input file (called INPUT by SAIL). If the SAIL mode is NORMAL, the program searches the alternate input file (this is a LFN of INPUT2 on a CDC system and DDNAME of FT09F001 on an IBM system). If there is nothing in the file (file does not exist on CDC or is dummied on IBM), then SAIL proceeds as the primary input data has instructed it. If, however, there is data in the alternate input file, then the program uses the OPTION directive to charge or add option in the option list, the PROGRAM directive to specify the selected programs or the PROSNAME directive to specify the program for which code is to be generated (all other programs will be processed for PROCS). The OPTION directive overrides any settings of the same options from the primary input file and if PROGRAM or PROSNAME is used, only the programs specified with the directives in the alternate input file are selected. Figure 1 shows the syntax of the alternate input file.

2. HULL GENERATION

The remainder of the HULL system (except for some utility subroutines) is maintained on a HULL system SAIL library file. To generate a new or to update a HULL system, the utility programs must be generated. Figure 2 shows the typical sequence of commands to generate the HULL utilities on a CDC 6600/Cyber 176 assuming the SAIL utilities are in a library called HULLIB as shown and that the system uses the tape library. Figure 3 shows the similar generation of the system for IBM computers. (Appendix C describes the procedures HLIB and PGEN.)

---

* SAIL: An Automated Approach to Software Development and Management, Air Force Weapons Laboratory, AFWL-TR-78-80 to be published.

```
OPTION      optname     value
PROGRAM     pgname
PROSNAME    psname
```

where

optname = option name

value = option value

pgname = program name

psname = name of program to be processed

Figure 1. SAIL Alternate Input Syntax.

```
      JOB,.-
      ATTACH(HULLIB,ID=DYTHULL)
      LIBRARY(HULLIB)
      DYTHUL,
      FTN(I=SAIL,OPT=2,PL=100000,0,B=BOW)
      FTN(I=SAIL,OPT=2,PL=1000000,B=PLANK)
      FTN(A,I=SAIL,OPT=2,S,S=PFMTEXT,B=LIB)
      REQUEST(NEWLIB,XPF)
      EDITLIB
      LIBRARY(NEWLIB)
      LOAD(BOW)
      NOLD(BOWF,BOW,STERN)
      LOAD(PLANK)
      NOLD(PLANKF,PLANK)
      LIBRARY.
      EDITLIB.
      CATALOG(NEWLIB, HULLIB,ID=DYTHULL,PW=XXXX,RP=999)
*EOR
      SAIL PROGRAM BOW PLANT LIBRARY ENDPROGRAM
         .
         .
         SAIL(changes)
         .
         .
*EOR
      LIBRARY(NEWLIB,NEW)
      ADD(SAILM,HULLIB,AL=1,LIB)
      ADD(COMPR,HULLIB,LIB)
      ADD(GETNUM,HULLIB,LIB)
      ADD(GTWD,HULLIB,LIB)
      ADD(STOR),HULLIB,LIB)
      ADD(INTEG,HULLIB,LIB)
```

Figure 2. CDC HULL Generation(continued)

```
      ADD(STORN,HULLIB,LIB)
      REWIND(LIB)
      ADD(*,LIB)
      FINISH.
      ENDRUN.
*EOR
      LIBRARY(NEWLIB,OLD)
      REWIND(BOWF)
      ADD(*,BOWF,AL=1)
      REWIND(PLANKF)
      ADD(*,PLANKF,AL=1)
*EOR
```

Figure 2. Continued

```
//....     JOB...
//LIB      EXEC     HLIB
//SAIL.INPUT   DD *
     SAIL  PROGRAM  LIBRARY  ENDPROGRAM
          :
          :
        SAIL (changes)
          :
          :
/*
//PLANK  EXEC   PGEN
//SAIL.INPUT   DD  *
     SAIL  PROGRAM  PLANK  ENDPROGRAM
/*
```

Figure 3. IBM HULL Generation.

3. HULL SYSTEM EXECUTION

The execution sequence for HULL is shown in figures 4 (CDC) and 5 (IBM).

On a CDC system BOW reads the input to determine the type of HULL run (KEEL, HULL or PULL). Identifiers for data tapes needed for the problem and the ZBLOCK (a table of problem parameters for the problem are written on file TAPE41. If there is no ZBLOCK for the problem in the tape library, the required data tape is requested. PLANK reads the ZBLOCK from TAPE41 file or the problem tape and the input to set up the options needed for the problem and generates an INPUT2 record. SAIL then reads the SAIL input, the HULL system SAIL file and INPUT2 and produces the appropriate source code on the file SAIL. This source code is compiled to produce the executable code for HULL, which reads the data tapes (requesting them as needed) and performs the requested function (initialize problem, run problem, or plot problem).

In IBM systems, PLANK reads the input and data tape (for HULL and PULL runs) and generates a SAIL alternate input file. Then SAIL, using the SAIL input and alternate input file, generates the source code which is compiled and executed.

The representative control cards for HULL runs on the IBM and CDC systems are shown in figures 6 and 7, respectively. (The description of IBM procedure HULL is shown in appendix C.)

4. HULL UTILITY ROUTINES

4.1 HULL Tape Library

On some CDC systems, the tapes used by the HULL problems are managed by the tape library program BOW. The tape library index file contains one three-word header record with the format shown in figure 8, followed by a variable number of 1000-word problem records, are shown in figures 9-11, ending with one 1000-word resource record as shown in figure 12.

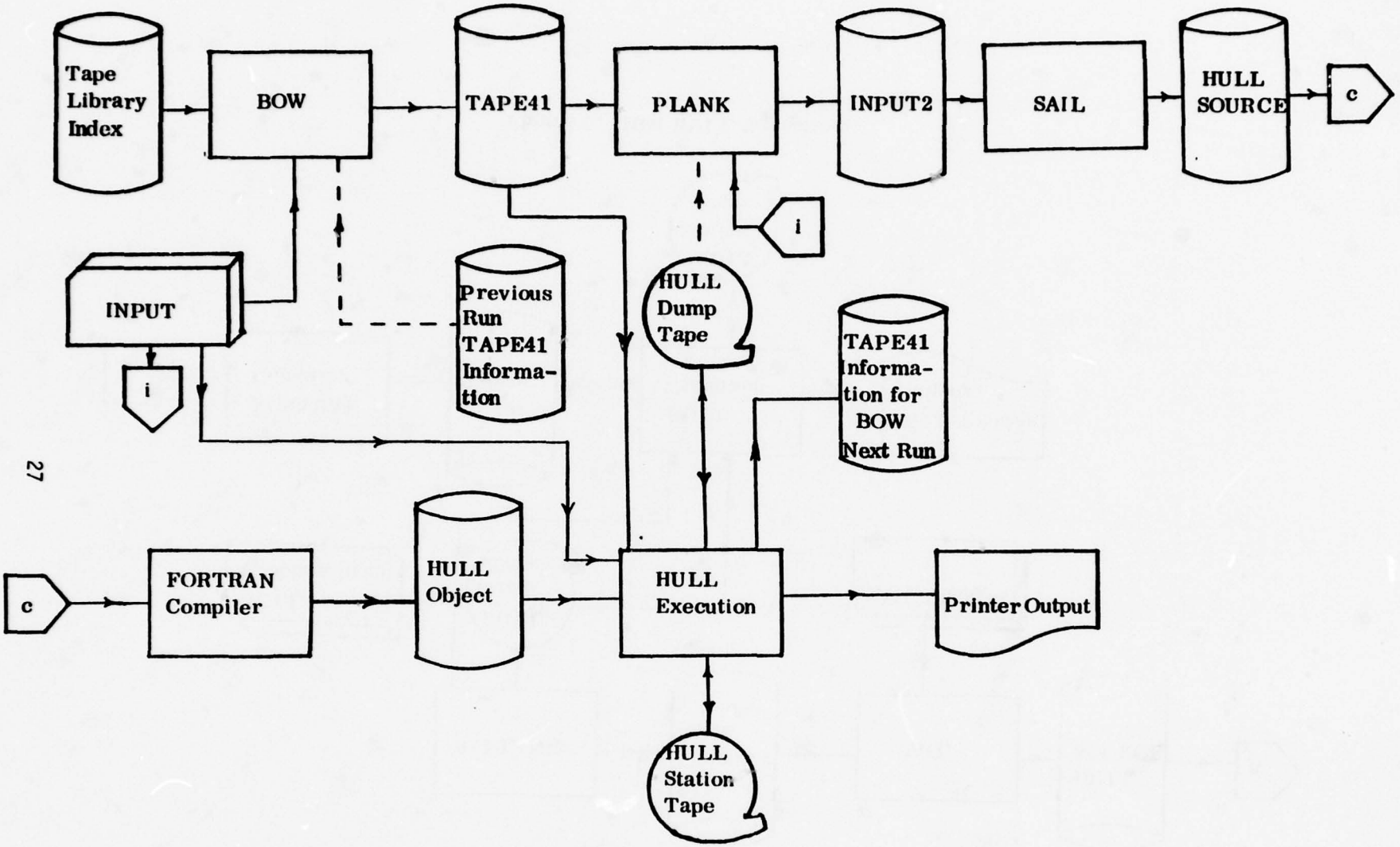


Figure 4. CDC HULL Execution.

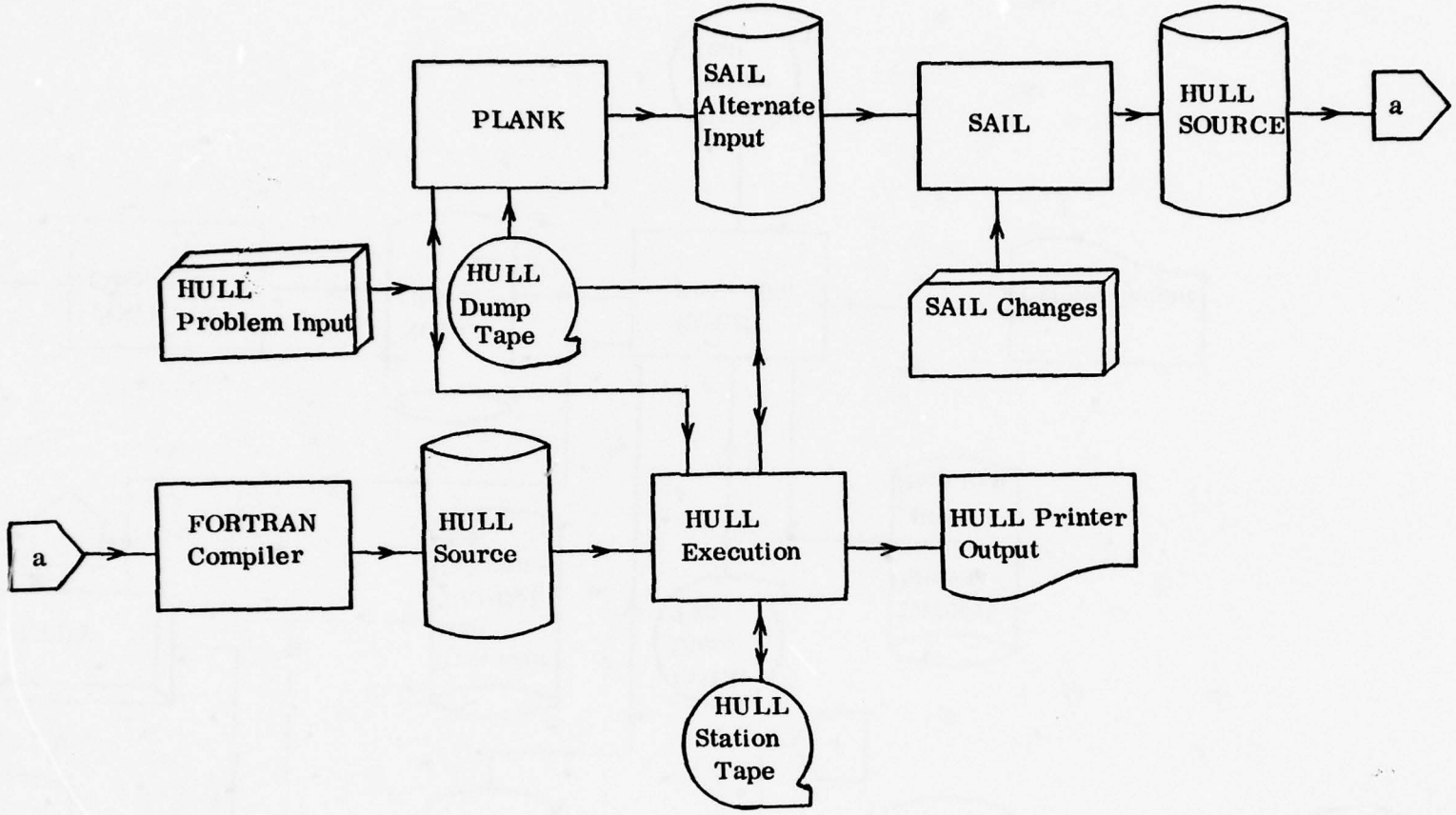


Figure 5. IBM HULL Execution.

```
JOB,...
ATTACH,HULLIB,ID=DYTHULL.
LIBRARY(HULLIB)
BOW.
PLANK.
DYTHUL.
REQUEST(SAVE,*Q)
FIN(I=SAIL,B=HULL,PL=1000000,L=SAVE,OPT=2)
REQUEST(MAP,*Q)
LDSET(MAP=SBEX/MAP,PRESETA=NGINDEF)
HULL.
EXIT(S)
DISPOSE(SAVE,PR)
DISPOSE(MAP,PR)
*EOR
SAIL
    .
    .
    SAIL changes
*EOR
HULL (or KEEL or PULL)
        .
        .
        problem data
        .
*EOI
```

Figure 6. CDC Control Cards for HULL Execution.

```
//****     JOB
//HULL     EXEC    HULL
//HULL.DATA      DD          'HULL dump tape data set.'
//HULL.STATION   DD          'HULL station tape data set.'
                             'Must be dummied if not needed.'
//HULL.INPUT     DD      *
HULL
          HULL input
/*
```

Figure 7. IBM JCL for HULL Execution.

Word	59	17	0
1	LIBTAPE		
2	Date of last update in character form ddmmmyy		
3	Time of last update in character form hh.mm.ss		
4	Julian date of last update	Time of last update inserted	

dd = day of month

mmm = month

yy = year

hh = hour

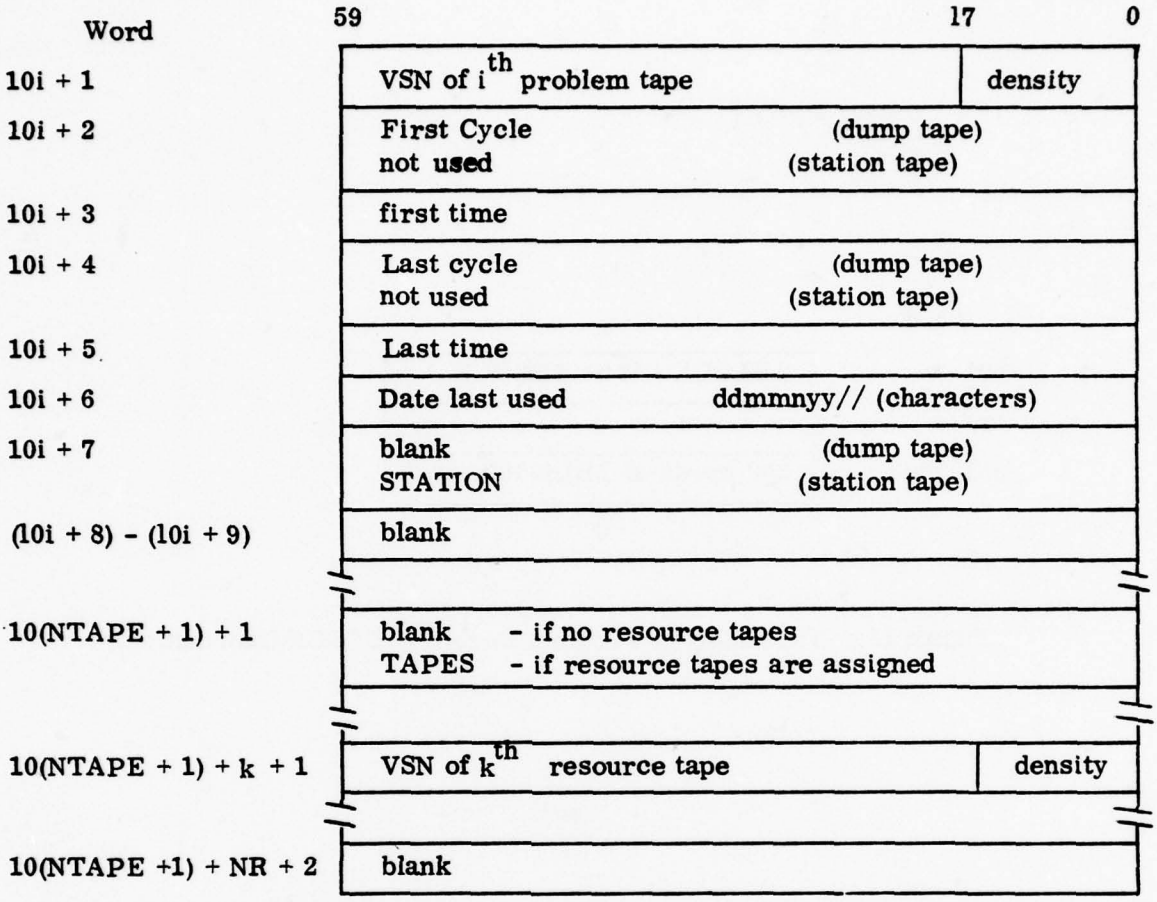
mm = minutes

ss = seconds

Figure 8. Tape Library Header.

Word	59	17	0
1	PROB		
2	Problem number		
3	CYCLE - if previous run was a cycle start blank - if not		
4	TIME - if previous run was a time start blank - if not		
5	Cycle start on last run -1.0 - if not a cycle start		
6	Time start on last run -1.0 - if not a time start		
7	MANUAL - if last run was manual blank - if not		
8	Date of last update - (old) (now) Last update Julian date		time(sec)
9	Time of last update added (new) Last resource tape Julian date		time(sec)
10	Active Inactive	Number of PF used to update Blank	

Figure 9. Tape Library Problem Record (Problem Information).



NTAPE - number of problem tapes.

NR = number of resource tapes.

Figure 10. Tape Library Problem Record (Tape Information).

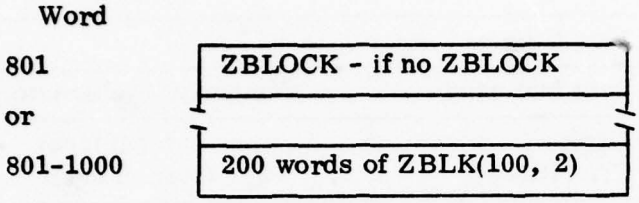
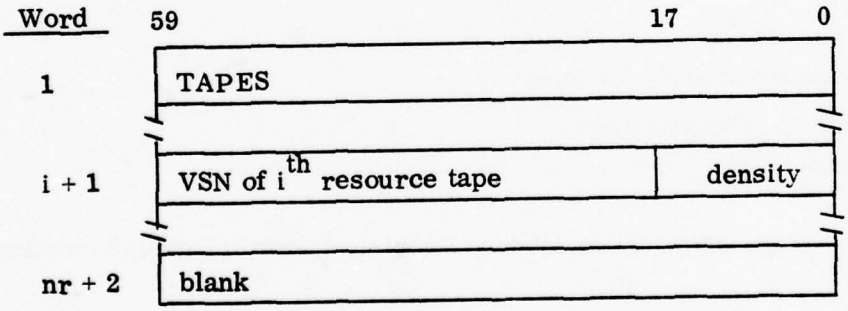


Figure 11. Tape Library Problem Record (ZBLOCK Information).



nr = number of resource tapes.

Figure 12. Tape Library Resource Record.

When BOW finds a HULL, PULL and/or KEEL problem record in input, but no BOW records, it selects tapes from those listed in the requested problem record and (for KEEL or HULL runs) resource tapes from the resource record. The type of run is determined by the problem input record that is found first. Tape identifiers are placed in the TAPE41 file.

In addition to selecting tapes BOW performs some protection functions. BOW will not permit a KEEL run for an existing problem unless it finds the keyword RERUN in the KEEL input. Also successive cycle or time selected restarts for the same cycle or time are not allowed unless sense switch three is set on. This prevents the user from accidentally restarting from the same cycle a second time. If no cycle or time restart is specified, then BOW assumes an "end of world" start and selects the last dump tape to restart the problem on a HULL run.

4.1.1 Library Maintenance

BOW performs automatic library maintenance when generated with the TAPELIB option set to 4. In this case, whenever BOW executes, it searches for a permanent file for each problem listed in the library ~~by the~~ the name yyyyyyyyyTAPE41FOxxxxPzzzz. Here yyyyyyyyy and the ID of the permanent file are installation dependent as shown in table 1. xxxx is the integer part of the problem number and zzzz is the fractional part of the problem number. Each permanent file contains information supplied by HULL during an earlier execution about the tapes that have been used and whether the problem was completed. This information is used to update the problem record in the tape library.

If BOW finds a BOW record in input then it proceeds to process the input for controlled maintenance of the tape library index file. There are two sets of primary key words which BOW recognizes, library definition and library maintenance key-words.

TABLE 1

BOW PERMANENT FILE CHARACTERISTICS

PFN = yyyyyyyTAPEINFOxxxPzzz

INST	yyyyyyyyyy	ID
1	PROBLEMMAS	DYMXCER
2	PROBLEMHUL	DYTHUL
3	PROBLEMHUL	LEWISHULL
4	PROBLEMHUL	SBEUK
5	PROBLEMHUL	SMCSAI

4.1.2 Library Definition Keywords

The library definition allows the user to select a tape library index file other than the one defined in the generation defaults of the installation selected. The keywords are

PFN. This keyword precedes the permanent file name for the index file to be used.

ID. This keyword precedes the permanent file ID for the index file.

TK, CN, MD, EX, RD. These keywords precede the turnkey, control, modify, extend and read passwords of the index permanent file, respectively.

PACKVSN, SN. These keywords precede the VSN and set name for the index permanent file (only available if generated with PF = 2).

CY. This keyword precedes the cycle number of the index permanent file.

LIBTAPE, VSN. These keywords precede the VSN of a tape which contains the index.

4.1.3 Library Maintenance Keywords

The library maintenance keywords allow the user to modify the tape index library and the problem information that it contains.

They also allow one to initialize a new library file. The maintenance keywords are

NEW. This keyword causes BOW to initialize the tape library file. It writes a three word header and an empty resource record on the file. Since BOW does not check the file it is initializing, care should be taken when using this keyword that an existing tape library is not destroyed.

ADD. This keyword causes resource tapes to be added to the resource record. There are a free and a fixed format syntax. The free format syntax is

ADD [RESOURCE] [(TAPE) (TAPES)] tapeid_1, tapeid_2, ..., tapeid_n

where tapeid_i are tape volumn serial numbers to be added to the tape library resource record. The density is the current density of tapes.

The fixed format is used to add tapes to the tape library from tape cards used at the AFWL. Each card contains two alphabetic characteristics in columns 1 and 2 and a one to three digit number right justified in columns 3-5. The syntax for this form of ADD is

ADD [RESOURCE][TAPE TAPES] {CARD CARDS}
:
:
tape cards
:
:
END [RESOURCE] [TAPE TAPES]

Note that the END directive is necessary with this form of ADD.

DELETE. This keyword allows the user to delete resource tapes from the resource records or to delete problem records. The syntax to delete resource tapes is

DELETE [RESOURCE] [TAPE TAPES] $tapeid_1$, $tapeid_2$, ..., $tapeid_n$

or

DELETE TAPE AFTER $tape_b$

where $tapeid_1$ are the resource tapes VSNs to be deleted and $tape_b$ is the VSN which precedes an unreadable tape VSN in the resource record.

To delete problems from the tape library, the syntax is

DELETE [PROB, PROBLEM, PROBLEMS] $pnum_1$, $pnum_2$, ..., $pnum_n$,

where $pnum_i$ are the problem numbers to be deleted. These numbers must be in ascending order.

LIST. This keyword causes the information from a particular problem record or from the resource record to be listed. The syntax is

LIST [[PROB PROBLEM PROBLEMS] $pnum_1$, $pnum_2$, ..., $pnum_n$]
[RESOURCE [TAPE TAPES]]

where $pnum_i$ are numbers for the problems to be listed (in ascending order), and a request for the resource list, if included, must follow all the problem numbers.

EDIT. This keyword causes a list of the information in the entire library to be produced. It has no parameters.

UPDATE. The keyword places BOW in the problem update mode and through the use of secondary keywords allows the user to add problems or to modify existing problems. The general syntax is

UPDATE [PROB PROBLEM] pnum
:
Secondary Keywords
:
{PROB PROBLEM} pnum
:
{PROB PROBLEM} pnum
:
{(END UPDATE) ENDUPDATE}

To remove tapes for a problem record from the tape library the command is

DELETE $tapeid_1$, $tapeid_2$, ... $tapeid_n$

To remove tapes for a problem and place them in the resource record for future use, the command is

RELEASE $tapeid_1$, $tapeid_2$, ... $tapeid_n$

In both cases, $tapeid_i$ are VSNs of the tapes to be deleted or released.

To add tape information to a new problem, the command is

ADD

$tapeid_i$ [den_i] fcy_i $ftime_i$ $lcyc_i$ $ltime_i$ $date_i$ (for restart tapes)

$tapeid_i$ [den_i] STATION $ftime_i$ $ltime_i$ $date_i$ (for station tapes)

and to add tape information to an existing problem, the command is

ADD AFTER $tapeid_a$

$tapeid_i$ den_i $fcyc_i$ $ftime_i$ $lcyc_i$ $ltime_i$ $date_i$ (for restart tapes)

$tapeid_i$ den_i STATION $ftime_i$ $ltime_i$ $date_i$ (for station tapes)

To delete a problem tape and replace it with one or more problem tapes, the command is

CHANGE $tapeid_d$

$tapeid_i$ den_i $fcyc_i$ $ftime_i$ $lcyc_i$ $ltime_i$ $date_i$ (for restart tapes)

$tapeid_i$ den_i STATION $ftime_i$ $ltime_i$ $date_i$ (for station tapes)

where

$tapeid_a$ is the VSN of the problem tape after which the tape information was to be added;

$tapeid_d$ is the VSN of the problem tape which is to be replaced by the tape information that follows; and

$tapeid_i$ is the VSN of the i^{th} problem tape to be added by this command.

den_i. This is an optional density for the i^{th} problem tape. If not specified, the default density is assumed.

$fcyc_i$. The first problem cycle on the i^{th} problem tape.

$ftime_i$. The first problem time on the i^{th} problem tape.

$lcyc_i$. The last problem cycle on the i^{th} problem tape.

$ltime_i$. The last problem time on the i^{th} problem tape.

$date_i$. The date the i^{th} problem tape was last used.

To add resource tapes to a problem, the command is

ADD RESOURCE TAPE TAPES $tapeid_1$, $tapeid_2$, ... $tapeid_n$

where $tapeid_i$ are the VSNs of the resource tapes to be added to the problem.

To terminate an update, the command is

END UPDATE

or

ENDUPDATE

COPY. This keyword causes the currently defined tape library to be copied to the file TAPE11. It has no other parameters.

RELEASE. This keyword allows the user to release resource tapes from problem records and return assigned tapes to the resource record. The syntax to release resource tapes from problems is

RELEASE [RESOURCE] [TAPES TAPE] FROM [PROB PROBLEM PROBLEMS] $pnum_1$, $pnum_2$, ... $pnum_n$

where $pnum_i$ are the problem numbers (in ascending order) from which the resource tapes are to be released.

To release problems from the tape library, the syntax is

RELEASE {PROB PROBLEM PROBLEMS} $pnum_1$, $pnum_2$, ... $pnum_n$

where $pnum_i$ are the problem numbers of the problems to be released (numbers in ascending order).

TRANSFER. This keyword will cause problems to be transferred from the current tape library to the tape library or file TAPE11. The syntax is

TRANSFER [PROB PROBLEM PROBLEMS] $pnum_1$, $pnum_2$, ... $pnum_n$

where $pnum_i$ are the problem numbers (in ascending order) of the problems to be transferred.

DENSITY. This keyword allows the user to change the default density when adding tapes. The syntax is

DENSITY = PE HD HY HI LO

where

PE is 1600 bpi, nine track

HD is 800 bpi, nine track

HY is 800 bpi, seven track

HI is 556 bpi, seven track

LO is 200 bpi, seven track

The original default density is defined by DENSLIB.*

DENSLIB	Default Density
3	HY
5	PE

4.2 PLANK

PLANK searches the input file for the first KEEL, HULL, or PULL problem data record and requests and generates an alternate input file with the appropriate options for SAIL to generate the program requested.

4.3 Subroutine Library

There are a number of subroutines used jointly by more than one program in the HULL system. These constitute the subroutine library. Ten of these subroutines are included in the SAIL library:

COMPR, GETNUM, GTWD, STOR, INTEG, STORN, AND(IBM), OR(IBM) COMPL(IBM), AND SHIFT(IBM).

* See Section III.1 for a description of DENSLIB

The routines which are part of the HULL library are

VALUE, NEXT, PARTOUT(CDC), PFNSET(CDC), RDL, WR, RDL2(CDC), WR2(CDC), SEARCH, SINK, OFFSW(CDC), ONSW(CDC), CDATE, DAY(IBM), DEVTYP(CDC), ENDREC(CDC), FL(CDC), INBY(CDC), PFMM(CDC), REMARK(CDC), REQUEST(CDC), RETURNS(CDC), ROUTEIT(CDC), SET(CDC) TIMTGO(CDC), WAECS(CDC), SECOND(IBM), LOCF(IBM).

The subroutines are described individually in appendix A.

SECTION III

HULL SYSTEM CONTROL

The HULL system generation is controlled by the general system SAIL options, the problem ZBLOCK, the PLANK control options for HULL and KEEL, the PLANK control for PULL, derived options, and on CDC systems by sense switch setting.

1. GENERAL SYSTEM SAIL OPTIONS

These are the master control options which select characteristics appropriate to the hardware and system software to be used when generating any of the codes in the HULL system. These options are

INST. This option defines the installation where the code is to be run. The default for this option is set in the header of the HULL system SAIL file. This option determines the default value of all other general system options. The values have the following meanings.

INST	Installation
1	Air Force Weapons Laboratory, Kirtland AFB, NM
2	Air Force Armament Laboratory, Eglin AFB, FL
3	McDonnell Douglas Astronautics, Huntington Beach, CA
4	ABMDA Research Center, Huntsville, AL
5	SAI (ABMDA Research Center), Huntsville, AL
6	Atomic Weapons Research Establishment, United Kingdom

SYS. This describes the computer on which the HULL system is to run. The values are

SYS	Computer
66	CDC 6600
76	CDC 7600
176	Cyber 176
360	IBM 360
370	IBM 370

The defaults currently set in the HULL system by installation are

INST	SYS
1	176
2	66
3	76
4	66
5	66
6	370

VER. This option defines the version of the operation system under which the HULL system is running. The values are

VER	Operating System
2	OS/VS2(IBM)
4	Scope 3.4 or NOS/BE (see option NOSBE, CDC)
20	Scope 2.0, 2.1 or 2.2 (CDC)

The defaults by installation are

INST	VER
1	4
2	4
3	20
4	4
5	4
6	2

NOSBE. Since the NOS/BE system has features additional to those of SCOPE 3.4, this option was added to indicate for CDC systems that the NOS/BE extensions can be used if the value is nonzero. The defaults by installation are

INST	NOSBE
1	1
2	1
3	0
4	0
5	0
6	not defined

PF. On CDC systems, this option defines the type of permanent files that are to be used by the HULL programs. The values are

PF	Permanent File Usage
0	no permanent file
1	permanent file on default system disk
2	permanent file on default cycle disk or other user set disk

Defaults by installation are

INST	PF
1	1
2	1
3	2
4	1
5	1
6	not defined

ECS. This option defines the availability of extended core storage (ECS) or large core memory (LCM) on the CDC system where the HULL programs run. If nonzero, ECS/LCM is available. The defaults are

INST	ECS
1	1
2	0
3	1
4	0
5	0
6	not defined

OBJLIB. This option, if nonzero, indicates that there exists a library of the HULL utility routines. If this is the case, these routines are not generated each time a program is generated. The defaults are

INST	OBJLIB
1	1
2	1
3	1

INST	OBJLIB
4	0
5	0
6	1

TAPELIB. This option defines the type of tape library that is to be used to handle the HULL system tapes. The values are

TAPELIB	Library Type
0	HULL runs with tape VSNs provided manually on control cards with no tape library index.
1	HULL runs on a single main frame with a separate BOW and STERN. Requires no permanent files. Index is maintained on tape.
2	HULL has no tape library index but creates dump tapes and station tapes as permanent files with tape staging when file length is equivalent to a full reel (CDC 7000 with no on-line tapes).
3	HULL uses a tape library index on a CDC 6600 permanent file with creation of permanent files and tape staging from the CDC 700 computer. STERN not needed but available.
4	On-line tapes for all mainframes but tape library index maintained and BOW runs only on CDC 6000 system. STERN not needed but available. Program files required.

The defaults by installation are

INST	TAPELIB
1	4
2	4
3	2

INST	TAPELIB (continued)
4	0
5	0
6	0

Note. TAPELIB = 1 generates code for STERN which has not been upgraded with the rest of the tape library.

ROUTE. This option describes the routing characteristics to be used on CDC systems. The understood values are

ROUTE	Routing Type
0	no routing
1	routing available
2	routing of generated program to another mainframe
3	routing of a HULL job following a KEEL run
4	routing of a PULL job upon completion of a dump tape in a HULL run

The defaults by installation are

INST	ROUTE
1	1
2	4
3	0
4	0
5	0
6	not defined

DENSHUL, DENSSTA, DENSLIB. These options define the default density of the HULL dump tape, the station tape and the tape library tape, respectively. Values are

DENS***	
1	LO--200 bpi seven-track tape (code not implemented)
2	HI--556 bpi seven-track tape (code not implemented)
3	HY--800 bpi seven-track tape (code not implemented)
4	HD--800 bpi seven-track tape (code not implemented)
5	PE--1600 bpi phase encoded nine-track tape
6	GE--6250 bpi group encoded nine-track tape (code not implemented)

Where *** is HUL, STA and LIB.

The defaults by installation are

INST	DENSLIB	DENSHUL	DENSSTA
1	5	5	3
2	3	3	3
3	3	3	3
4	3	3	3
5	3	3	3
6	---------- not defined ----------		

LABEL. This option, if nonzero, indicates the HULL dump tape has existing labels on CDC systems. The defaults are

INST	LABEL
1	1
2	1
3	0
4	0
5	0
6	not defined

DATE. This is the date option for CDC systems.

DATE	Date Type
0	DD/MM/YY
1	MM/DD/YY

where DD is the day of the month, MM is the month, and YY is the year. The defaults by installation are

INST	DATE
1	0
2	0
3	1
4	1
5	1
6	not defined

2. HULL PROBLEM ZBLOCK

The description of a HULL problem is found in the problem ZBLOCK. The ZBLOCK is defined in the HULL and KEEL programs in the common block /ZBLOCK/. The names of the ZBLOCK parameters are found in a common block /ZNAME/. The variable NAMEZ and the SAIL program ZNAMES contain the names of the /ZBLOCK/ parameters in the order the parameters are found in common /ZBLOCK/.

The ZBLOCK information is transferred between the HULL system programs and the problem dump tape in the form of an array ZBLK(100,2) (single precision on CDC and double precision on IBM). In the array, ZBLK(I,2) contains the name of the ZBLOCK parameter and ZBLK(I,1) contains its value. All numeric ZBLOCK parameters are stored in ZBLK as real (i.e., floating point numbers) and all logical variables as logicals. (On IBM, the logical value is in the upper half of the double precision ZBLK(I,1).

KEEL generates the ZBLOCK for a new problem from the NAMEZ array of PROC ZNAMES. Because of FORTRAN statement continuation restrictions, the NAMEZ data is divided into four arrays NAME1 (dimensioned D$NAMEA), NAME2 (dimensioned D$NAMEB), NAME3 (dimensioned D$NAMEC), and NAME4 (dimensioned D$NAMED). The actual number of parameters in the ZBLOCK is determined by the option of the problem and is D$NAMEA + D$NAMEB + D$NAMEC + D$NAMED. Paremeters are included in the ZBLOCK unless they are needed for the problem.

ZBLOCK information is transferred between common /ZBLOCK/ and the array ZBLK in the routines ZSET and SETZ. The parameter name ZBLK(I,2) is compared to NAMEZ(J), and the value is stored in the corresponding position of /ZBLOCK/ if they match. Since those parameters which are stored as integers in /ZBLOCK/ must be converted from floating point while those which are stored as floating point or logical do not need to be converted, an array ITYPZ (with division ITY1, ITY2, ITY3, and ITY4) defines whether the parameter is transferred or converted between fixed and floating point. (If ITYPZ(J) = 1, the value is converted and not if ITYPZ(J) = 0.)

2.1 ZBLOCK Parameters

The meaning of the ZBLOCK parameters and their default values (in KEEL) are described below. Parameters marked with asterisks are used by PLANK as SAIL options for program generation.

PROB. Problem number for the calculation (real).

AREF. Back reflective boundary flag in three dimensions (logical). Included only if DIMEN = 3 (default = .FALSE.).

ATMOS.* This parameter defines the type of atmosphere which will be used in the rezoning process. This value is initially set during KEEL and, if not specified, defaults in KEEL to a value of 2 (integer).

ATMOS	Atmosphere Type
1	tropical atmosphere
2	temperate atmosphere
3	arctic atmosphere
4	exponential atmosphere
5	constant atmosphere (density and pressure)

BREF. Bottom reflective boundary flag (logical, default = .FALSE.).

BURN.* This parameter is included in the ZBLOCK by KEEL only if it is set to a value not equal to 0. This describes the type of HE burn routine that is to be used (integer).

BURN	Burn Code
0	no burn code
1	TNT type burn with detonation velocity = Cs + 2 Cp, where Cs = sound speed and Cp = particle velocity.
2	programmed burn with a constant detonation velocity

*ZBLOCK parameters used by PLANK in the program generation as a SAIL option are starred in this subparagraph.

CODE.* This parameter has meaning only for codes where NOP is greater than 0. It describes the type of particle transport to be used. KEEL sets the default in the ZBLOCK to 1 (integer).

CODE	Particle Transport
1	noninteractive
2	interactive dust

COLD. Logical parameter to indicate if radiative cooling is to be done. PLANK sets the SAIL generation option HOT to 0 if COLD = .FALSE. (default = .TRUE.).

CYCLE. Current cycle number (real).

DIMEN. Number of dimensions in calculation. Set in KEEL with a default of 2 and should not be changed after KEEL (integer).

DIMEN	
1	one-dimensional mesh (not implemented)
2	two-dimensional mesh
3	three-dimensional mesh

DT. Current time step in seconds (real).

ELC. Total energy of mesh at last cycle in ergs (real).

EOS.* This parameter describes the equation of state to be used during the problem. Default in KEEL is 1. Care should be exercised if the value of this option is changed during execution (integer).

EOS	Equation of State
1	analytic equation of state for air
2	constant gamma equation of state
6	multimaterial equation of state
10	tabulated air equation of state

ETH. Total theoretical energy in the mesh in ergs (real).

EXPAND. Expansion factor for rezone (real).

FAIL.* This value is defaulted to 0 in KEEL. If it is set to a nonzero value, the multimaterial (EOS = 6) equation of state will simulate failure by inserting a gas (usually air) into a cell where a failure has occurred.

FLUXER.* This parameter describes the type of multimaterial fluxing that is to be done. The default value is set to 2 in KEEL and cannot be changed to a higher value than the KEEL value during the run.

FLUXER	
0	no flux limiter (not implemented)
1	Flux limited by mass of material. Temperature equilibrium assumed. Individual masses carried.
2	Flux limit calculated by relative volumes of individual material. Temperature equilibrium assumed and Individual masses are carried.
3	Flux limit calculated by relative volumes of individual materials. Temperature equilibrium not assumed since energies of individual materials are carried as well as their masses and volumes.

FREF. Front reflective boundary flag (logical). Included in ZBLOCK only if DIMEN = 3 (default = .FALSE.).

GEOM.* This parameter is put in ZBLOCK only for two-dimensional problems. It indicates the geometry of the two-dimensional problem (three-dimensional problems are all Cartesian). The default set by KEEL is 2.

GEOM	Geometry
1	Cartesian
2	cylindrical
3	spherical (not implemented)

HOB. Height of the weapon burst in kilometers (real, default =0).

IMAX.* This is the number of cells in the x direction of the mesh. KEEL defaults IMAX to 100. It must not be changed from the KEEL value (integer).

IQ. Current extent of active calculation in the x direction (integer, default = IMAX -1).

ISLAND. This parameter is defaulted to 0 by KEEL and PLANK and causes codes to be generated for fixed internal boundaries if set greater than 0. The value of the option should not be changed after KEEL (integer).

JMAX.* This is the number of cells in the y direction (axial in two dimension) of the mesh. KEEL defaults this to 200. It must not be changed from its KEEL value (integer).

JQ. Current extent of active calculation in the y direction (integer, default = JMAX -1).

KMAX.* This is the number of cells in the z direction of a three-dimensional mesh. It is defaulted to 1 by KEEL and also KEEL will put the option in the ZBLOCK only if DIMEN = 3 (integer, default = 1).

KQ. Current extent of active calculation in the z direction of a three-dimensional mesh (integer). Included only if DIMEN = 3 (default = KMAX - 1).

LREF. Left reflective boundary flag (logical, default = .TRUE.).

MAGFLD.* This option describes the magnetic field code that is to be included. KEEL defaults MAGFLD to 0 and only includes it in the ZBLOCK if it is set to a non-zero value (integer).

MAGFLD	
0	no magnetic field code
1	dipole magnetic field

METHOD.* This option selects the finite difference method used in the hydro-differencing. The default set by KEEL is 2.

METHOD	Difference Time
1	SHELL
2	HULL II
3	Lax-Wendroff (not implemented)

MLC. Total mass on the mesh at end of last cycle.

NTH. Total theoretical mass in mesh.

NH.* The number of words in mesh per cell (integer).

NH = 3 + DIMEN + NUM*FLUXER + NHIST + NR + MG + NSTR
+ NSTN + NEOS + NWRK

NR = 4 if RAD = 2
= 0 otherwise

NUM = NM , NM > 1 and EOS=6
= 0 , NM $\leq$ 1 or EOS =6

MG = 2 if MAGFLD $\neq$ 0

= 0 otherwise

NSTR = 3 if STRESS $\neq$ 0 and DIMEN = 2

= 5 if STRESS $\neq$ 0 and DIMEN = 3

= 0 otherwise

NSTN = 3 if STRAIN $\neq$ 0 and DIMEN = 2

= 5 if STRAIN $\neq$ 0 and DIMEN = 3

= 0 otherwise

NEOS = 1 if EOS = 7 and DIMEN = 2

= 0 otherwise

NWORK = 1 if WORK $\neq$ 0

= 0 otherwise

NHIC. Number of word of core used by HULL for calculation. This is a program generator option used by PLANK (integer).

NHIC = NROWIC*IMAX*NH

NROWIC = 3 + MG + NR + NMN if DIMEN = 2 and SW $\neq$ 5

MG = 1 if METHOD $\neq$ 0 or MAGFLD $\neq$ 0

= 0 otherwise

NR = 1 if RAD $\neq$ 0

= 0 otherwise

NMN = 1 if NM > 1

= 0 otherwise

NROWIC = JMAX if DIMEN = 2 and SW = 5

= JMAX*3 if DIMEN = 3 and NM $\leq$ 1

= JMAX*4 if DIMEN = 3 and NM > 1

NVARST.* This parameter is the number of variables per station (default 4 if STRESS = 0 and 15 otherwise). Not included in the ZBLOCK if NSTN = 0.

NM.* This parameter is the number of materials in the problem. It is only useful for EOS = 6. Set to 1 as a default by KEEL (integer).

NOP.* This is the number of particles and/or stations in the problem. It is set to the appropriate value by KEEL and is included in the ZBLOCK only if greater than 0. It is a program generation option.

NPLPB.* This parameter is the number of planes per logical record on the dump tape of a three-dimensional problem. It is included in the ZBLOCK only if DIMEN = 3.

NPLPB = MAX(K, NPLIC) such that MOD(KMAX, K) = 0

where NPLIC = 3 if NM $\leq$ 1
= 4 if NM > 1

NPP. This parameter is the number of particle parameters.

NPP = 2 if DIMEN = 2, CODE = 1, CDC $\neq$ 0
= 3 if DIMEN = 2, CODE = 1, IBM $\neq$ 0
or DIMEN = 3, CODE = 1, CDC $\neq$ 0
= 4 if DIMEN = 3, CODE = 1, IBM $\neq$ 0
= 6 if DIMEN = 2, CODE = 2, CDC $\neq$ 0
= 7 if DIMEN = 2, CODE = 2, IBM $\neq$ 0

NHIST.* This is the number of fluxed histories. KEEL defaults to 3 if STRESS is greater than 0, to 6 if STRAIN if greater than 0, and 0 otherwise. It is not included in the ZBLOCK if it is equal to 0. Care should be exercised in changing the value of this option during execution (integer).

NSTN.* This is the number of stations in the problem. It is defaulted to 0 in KEEL and not included in the ZBLOCK if it is equal to 0. KEEL will not allow the creation of more stations at any one time than is indicated if NSTN is included in input. KEEL will set the parameter to the number of stations generated (integer).

NROWPB.* This is the number of rows in a logical record on a HULL dump tape for a two-dimensional mesh (integer). It is included in the ZBLOCK only if DIMEN = 2:

NROWPB = MAX(J, NROWIC) said that MOD(J, JMAX) = 0.

PTSTOP. Stop time in terms of seconds of problem time (real, default 1.0 E 20).

RAD.* This parameter specifies the radiation transport code to be included. It is defaulted to 0 in KEEL and is included in the ZBLOCK only if it is set to a value greater than 0 (integer).

RAD	Location Transport
0	no radiation code
1	equilibrium radiation diffusion (no code)
2	nonequilibrium radiation diffusion

RADLOS. Total energy (ergs) lost from the mesh due to radiation or radiation cooling routines (real).

REZONE.* This parameter specifies the type of rezone that is to be performed at the boundary of the mesh (integer). KEEL defaults the value of this option to 1 and the acceptable values are

REZONE	Type of Rezone
0	no rezone
1	standard shock follower rezone
2	fireball follower
3	horizontal shock follower
4	vertical shock follower
5	particle follower
6	particle follower with shocks ignored
7	continuous rezone

BREF. Right reflective boundary flag (logical, default = .FALSE.).

STABF. Courant stability factor (real, default = 0.5).

STRAIN.* This option specifies that code is to be included for material strain calculation. Defaults to 0 (no strain calculation) and is not included in ZBLOCK unless it is set greater than 0. Should not be set greater than 0 unless STRESS > 0 (integer).

STRESS. This option if greater than 0 specifies the inclusion of the code to calculate the elastic-plastic strength of solids. KEEL has a default of 0 and the option is not included in the ZBLOCK unless it has a value greater than 0.

SUME. Total energy (ergs) lost from mesh (real).

SURF.* This option provides code for physical bottom boundary conditions. The default in KEEL is 0 and the option is included in the ZBLOCK only if it has a value greater than 0 in KEEL. The values acceptable are

SURF	Surface Condition
0	no special code
1	simple thermal layer
2	thermal layer for nuclear blast precursor

TERAD. Cumulative radiation energy loss (ergs, real) compared with RADLOS.

TLC. Problem time last cycle (sec, real).

TREF. Top reflective boundary flag (logical, default = .FALSE.).

TTIME. Total CPU time for problem to current cycle (real).

TTIME6. Total CDC 6600 CPU time for problem (real). Included only for INST = 1.

TTIME7. Total CDC 7600 CPU time for problem (real). Included only for INST = 1.

TTSTOP. Total CPU time limit (real).

UREZ. Velocity (cm/sec) which will trigger a rezone at boundaries of x-direction (real, default = 10).

VISC.* When this parameter has a value which is greater than 0, artificial viscosity is included in the hydro calculation. The default set in KEEL is 0 (integer).

VREZ. Velocity (cm/sec) which will trigger a rezone at boundaries of the y-direction (real, default = 10).

VOIDS.* This option specifies the code for the handling of voids that is to be included (integer). The default in KEEL is 0 and the acceptable values are

VOIDS	Void Code
0	no voids allowed
1	implicit voids
2	use of material "VOID" which must be declared

WORK.* If this option has a nonzero value then code is included to calculate the elastic-plastic work on solids (integer). The default in KEEL is 0.

WREZ. Velocity (cm/sec) which will trigger a rezone at boundaries of z-direction (real). Included in the ZBLOCK only if DIMEN = 3 (default = 10).

XOB. The x coordinate of burst (real).

YOB. The y coordinate of burst (real). Default = 0.

YGND. Location of ground. When mesh is rezoned at bottom to YGND, the BREF becomes .TRUE. (real, default = 0).

YIELD. Yield of weapon in KT (real, default = 0).

"material". If EOS = 6, then NM ZBLOCK entries are the material identifiers for the materials used in the problem (see table 2). The values are the integer number for the equation-of-state number for that material. There must be NM unique values between 1 and NM.

As noted above, ZBLOCK parameters marked with asterisks are used by PLANK in the program generation as a SAIL option. They have default values of 0 if they are not found in the ZBLOCK or are not specified in the KEEL input, except

ATMOS = 2
CODE = 1
DIMEN = 2
EOS = 1
FLUXER = 2
IMAX = 100
JMAX = 200
KMAX = 1
METHOD = 2
NVARST = 4 if STRESS = 0
NVARST = 15 if STRESS > 0
NM = 1
NPP = DIMEN

TABLE 2

HULL CODE MATERIALS

Material	Material Identifier	Ambient	
		Specific Internal Energy (ergs/g)	Density (g/cm^3)
AFX-108	AFXO	5.98005×10^9	1.54
Air	AIR	2.044×10^9	1.225×10^{-3}
Aluminum	AL	2.71578×10^9	2.71
Anfo	ANFO	2.66281×10^9	0.85
Burned Anfo	ANFOBRN	4.7×10^9	0.85
Methane	CH4	2.3×10^9	1.113×10^{-3}
Comp B	COMPB	2.615×10^9	1.72
Burned Comp B	CBBRN	4.94×10^9	1.72
Concrete	CONCRT	7.81×10^9	2.2
Copper	CU	1.13336×10^9	8.9
Iron	FE	1.26816×10^9	7.86
Granite	GRANIT	5.2×10^7	2.68
Octol	OCTOL	2.615×10^9	1.82
Burned octol	OCTBRN	5.272×10^{10}	1.82
Pentalite	PEN	2.65316×10^9	1.66
Burned pentalite	PENBRN	5.155×10^{10}	1.66
PBX	PBX	2.65316×10^9	1.84
Burned PBX	PBXBRN	5.86×10^{10}	1.84
Sand	SAND	7.81×10^8	1.6
Stainless steel	SSTEEL	1.26816×10^9	7.86
Tantalum	TA	4.02619×10^8	16.6

TABLE 2 (continued)

HULL CODE MATERIALS

Material	Material Identifier	Ambient	
		Specific Internal Energy (ergs/g)	Density (g/cm^3)
TNT	TNT	2.65516×10^9	1.56
Burned TNT	TNTBR	4.73×10^{10}	1.56
Tuff	TUFF	2.8573×10^5	1.97
Tungsten	W	3.54228×10^4	18.1
Water	WATER	2.0×10^6	1.0

2.2 Adding ZBLOCK Parameters

To add a ZBLOCK parameter, the parameter is inserted in the ZBLOCK and the name of the parameter is inserted in the same relative position in the NAMEZ subarray. The parameter may be conditionally or unconditionally included. Also a data type parameter of 0 for logical and real parameters and 1 for integer parameters must be inserted in the corresponding position of the DATA statement for the ITYPZ subarrays. Next the appropriate DEFN must be added to increase NAMEZ dimension (D$NAMEA if the name was inserted in NAME1, D$NAMEB if in NAME2, D$NAMEC if in NAME3, and D$NAMED if in NAME4). Finally, the default value must be set in KEEL.

If the new ZBLOCK parameter is to be a PLANK/SAIL generation option, the name must be added to the data statement in PLANK and the DATA statement for NVAR must be increased. The default (if not 0) must be set in PLANK.

3. PLANK/SAIL CONTROL OPTIONS

There are a number of program generated options which control the type of code that is produced. These options include HULL and KEEL options and PULL option.

3.1 HULL and KEEL Options

DEBUG. This option has a default value of 0 and may be set to nonzero in the SAIL input to cause DEBUG prints to be generated in the code.

FILMPR. This option has a default value of 0 for all installations except at INST = 1 and INST = 3, where it is 1. If set to nonzero in SAIL input, it causes the generation of additional output from HULL and KEEL which would normally be directed to a film output device.

HLEV. This option has a default value of 1. When set to 2 in SAIL input, it causes the incore H-array to be put on level 2 on CDC 7600 and Cyber 176.

SW/SWX. These options are set in PLANT to determine what storage mode will be used for the problem.

SW--Defaults to 0 for CDC and 6 for IBM. PLANK will set to 6 if sense switch 1 is not on, to 0 if both sense switches 6 and 1 are on, and to 5 if only sense switch 5 is on.

SWX--Defaults to 0. PLANK will set to 1 if sense switch 6 and sense switch 1 are both on (CDC systems).

Since sense switches are only available on CDC machines, the option SW and SWX can be set from PLANK input on IBM systems. These values control the mesh allocation.

SW	SWX	
0	0	Mesh is stored in ECS/LCM (CDC only).
0	1	Mesh is stored on direct access file.
6	N/A	Mesh is stored on sequential file.
5	N/A	Mesh is stored in core.

3.2 PULL Options

The PULL control options are set by PLANK when PULL generation is requested. They include SW and SWX discussed in the HULL and KEEL section. All others have a default of 0 unless specified otherwise.

HHST. Is set to 1 when code for horizontal histograms is to be included.

VHST. Is set to 1 when code for vertical histograms is to be included.

ELDEN. Is set to 1 if electron density plots were requested.

DUSTY. Is set to 1 if the plot is of a DUSTY tape.

PART. Is set to 1 if the code for particle plots is to be included.

CURL. Is set to 1 if the code to calculate curls is to be included.

GRAD. Is set to 1 if the code to calculate gradients is to be included.

VECTOR. Is set to 1 if vector plots are to be made.

CONT. Is set to 1 if contour plots are to be made.

MF. Specifies type of film output. Default is 16 except when PLOTPKG = 5, then MF = 105. Values are

MF	Type
16	16 mm film
35	35 mm film
105	48X microfiche

HULL. Set to 1 if the plot is of HULL data.

CONTV. Set to 1 if contour values were specified.

DDDPLOT. Set to 1 if a three-dimensional contour plot of a three-dimensional problem is requested.

STATION. Set to 1 if station plots are requested.

PUNCH. Set to 1 if code for punched output is required.

LAMBPLOT. Set to 1 if plotting of LAMB data is requested.

PLOTPKG. Plotting package to be used defaults as 5 for INST = 1, 4 for INST = 2 or INST = 3 and 0 for all other installations. The values are

PLOTPKG	Plotting Package
0	No plotting package.
1	Plotting package included to generate Calcan tape for a CDC 160A system.
2	Plotting package included to generate CDC 280 microfilm.
3	Both plotting package 1 and 2 are included.
4	SC 4020 plotting package included.
5	Meta file plotting interface included.

3.3 Derived Options

These options obtain their values based on the value of other options.

3.3.1 General Derived Options

These affect the entire code and are generally related to the system configuration.

CDC = 1 if SYS = 66, SYS = 76, SYS = 65 or SYS = 176
= 0 otherwise

IBM = 1 if SYS = 360 or SYS = 370
= 0 otherwise

CW -- Number of characters in a character variable.
= 10 if CDC $\neq$ 0
= 8 if IBM $\neq$ 0

RDEND -- Type of end of file test.
= 2 if CDC $\neq$ 0
= 1 if IBM $\neq$ 0

NW -- Number of words in a character variable.
= 1 if CDC $\neq$ 0
= 2 if IBM $\neq$ 0

CARDL -- Number of words in a CARD.
= 80/CW, MOD(80, CW) = 0
= 80/CW + 1, MOD(80, CW) = 0

CARDO -- Even length of card data.
= 80 - [80/CW] *CW

3.2.2 HULL and KEEL Derived Options

These option values are derived either in PLANK, KEEL, or SAIL.

LBUFA = max(LBUFB1, LBUFB2, JMAX)

LBUFB1 = 6*IMAX + 2 if STRESS ≠ 0

= 2*IMAX otherwise

LBUFB2 = 6*IMAX if MAGFLD ≠ 0

= 0 otherwise

NHEC. Total number of words required for problem

= IMAX*JMAX*NH if DIMEN = 2

= IMAX*JMAX*KMAX*NH if DIMEN = 3

CMECS. Amount of core required for SW = 5 version.

3.3.3 PULL Derived Options

These option values are set for PULL by PLANK.

IJKMAX. Maximum dimensions

= max(IMAX,JMAX,KMAX)

BLKCOM. Amount of blank common needed

= max(BLKCM,BLKCMSO, BLKCMV, BLKCMP, BLKCMG, BLKCMC, BLKCME)

BLKCM = max(200, IMAX + JMAX + 6, IMAX*NH*NROWPB, NPPR*NPP)

BLKCMSO = max(NVARPP*NPLPB, NNARPP + IJKMAX*NH) if DIMEN = 3

= 0 otherwise

NVARPP = IMAX*JMAX*NH

BLKCMV = 2*IMAX CONT ≠ 0 or VECTOR ≠ 0

= 0 otherwise

BLKCMP = max(NPPR + 100)*NPPX, 200*NPPX)SORT ≠ 0

= 0 otherwise

BLKCMG = 4*IMAX if GRAD ≠ 0

= 0 otherwise

BLKCMC = 7*IMAX if CURL ≠ 0

= 0 otherwise

BLKCME = 506 + max(NPPL*NPR, 11 + IMAX) ELDEN ≠ 0

= 0 otherwise

NPPR = MIN(NOP, NHIC/NPP)

NPPX = 2 if CODE = 1

= NPP CODE = 2

4. CDC SENSE SWITCH CONTROL

Criterion functions in the execution of HULL can be controlled by setting the system sense switches. Sense switch settings during PLANK will cause the setting of program generation options SW and SWX as discussed in the section on HULL and KEEL options.

During execution, the following sense switch settings are recognized.

Sense Switch	Effect
1 on	The program will terminate at the end of the current operation (HULL and PULL only).
3 on	The program displays current status (HULL and PULL only).
4 on	The program will change output tapes (HULL only).
5 on	The disk version of HULL will switch to double buffering (more CM is required).
5 off	The disk version of HULL will switch back to single buffering.
6 on	The ECS version of HULL will "roll" itself out of ECS until given a GO command by the computer operator.

SECTION IV

MESH STORAGE

The problem information is stored both in core and on tape. The respective formats are shown below.

1. IN-CORE MESH

The computational mesh is stored in a single dimensioned array called the "H-array." There are NHIC words reserved. The first word for each cell contains the pressure in the cell followed by words for the velocity components in each dimension. Then comes the total specific internal energy and the total mass of all. If the calculation is multimaterial then the masses of the individual materials are next followed by the individual material volumes if FLUXER is ≥ 2, and then the total internal energies of the individual materials if FLUXER = 3. Next come the stress components (if STRESS > 0) followed by the STRAIN component (if STRAIN > 0). There are three stress and strain components in two dimensions and five in three dimensions. Next follow the magnetic field components if MAGFLD > 0 (one component for each dimension). There are five words for radiation information if RAD = 2. Finally one word for detonation time if BURN = 2. Most of the referencing of the cell variables is done through equivalent variables. The meaning of the equivalent H-array variables are

P(N). Pressure of cell.

U(N). x component of velocity.

V(N). y component of velocity.

W(N). z component of velocity.

XI(N). Specific internal energy.

XM(N). Mass of cell.

XM(N + I). Mass of i^{th} material in cell.
NM > 1

XV(N + I). Volume of i^{th} material in cell.
NM > 1, FLUXER $\geq$ 2

XII(N + I). Total internal energy of i^{th} material in cell.
NM > 1, FLUXER = 3

SRR(N)
SZZ(N)
SZR(N)
— 2-D stresses

SXX(N)
SYY(N)
SZZ(N)
SXY(N)
SXZ(N)
SYZ(N)
— 3-D stresses

ERRH(N)
EZZH(N)
EZRH(N)
— 2-D strains

EXXH(N)
EYYH(N)
EZZH(N)
EXYH(N)
EXZH(N)
EYZH(N)
— 3-D strains

FMX(N). x-component of magnetic field. MAGFLD > 0.

FMY(N). y-component of magnetic field. MAGFLD > 0.

FMZ(N). z-component of magnetic field. MAGFLD > 0, DIMEN = 3.

ERAD(N). Radiation energy density. RAD = 2.

CV(N). Specific heat in cell. RAD = 2.

DLTA(N). Solution coefficient for radiation equation. RAD = 2.

TK(N). Temperature of cell. RAD = 2.

FYS(N). Second solution coefficient. RAD = 2.

In all cases, N points to the word which precedes the first word in the H-array for that cell.

2. HULL TAPE FORMATS

There are two tapes which are used by HULL: the dump tape and, if NSTN > 0, the station tape.

2.1 HULL Dump Tape

The dump tape contains the problem information for the entire mesh. Each problem record set contains enough information to restart the problem.

A problem record set begins with a four word header record. The first word contains 555.0, if a full problem record follows or 666.0 if it is the tape termination record. The second word contains the problem number, the third word contains the cycle number of the problem, and the time is in the fourth word.

Then follows a 200 word (character*) logical record containing the ABLOCK. This is followed by "NBLKS" logical records which are NVARPB in length, where

NVARPB=NK*1MAX*NROWPB if DIMEN=2

=NH*IMAX*JMAX*NPLPB if DIMEN=3

NBLKS =NHIC/NVARPB

If NOP 0, then follows NPREC - 1 logical records which are NPVAR words in length and one logical record which is NPVLR in length containing the particle information, where

NPVAR = (NHIC/NPP)*NPP

NPREC = 1 + (NOP*NPP)/NPVAR

NPVLR = NOP*NPP- (NPREC-1)*NPVAR

At the end of each problem record set HULL writes a header record with word 1 containing 666.0. On a subsequent data set output this record is overwritten with a regular problem header. Therefore, when the program terminates, the last logical record on the tape is a terminator record (i.e., a header record with 666.0 in word 1). This is used by HULL to identify the end of the problem when reading the dump tape.

2.2 HULL Station Tape

The station tape contains a two word header which identifies the tape. In the old format, the first word contains the current VSN and the second word contains the next VSN for a continuation tape. In the new format, the first word is STAT and the second word is the problem number. HULL can differentiate between the two formats by the manner it received the station tape (new format is used for tapes that come from BOW or on IBM systems).

The header record is followed by a 200 word logical record containing the ZBLOCK. Then comes the particle information in the same format as on the dump tape. Then follow logical records of 1024 words containing the station information using the format shown in figures 13-18. The station tape is terminated by an end of file.

---

* A character word is one machine word on CDC system and is a double word (2 machine words) on an IBM system.

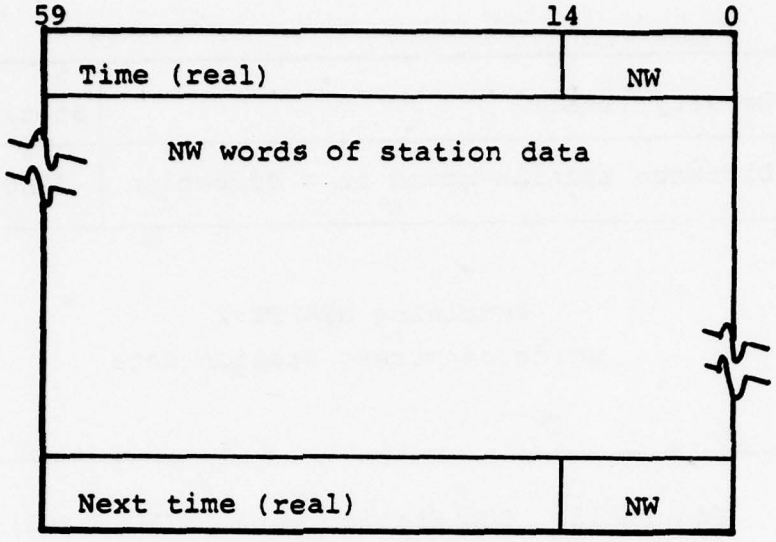


Figure 13. CDC Station Record.

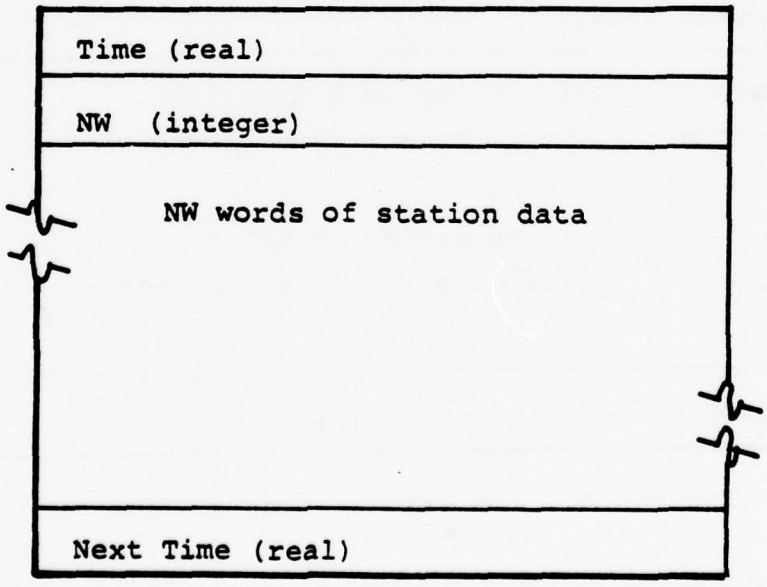


Figure 14. IBM Station Record.

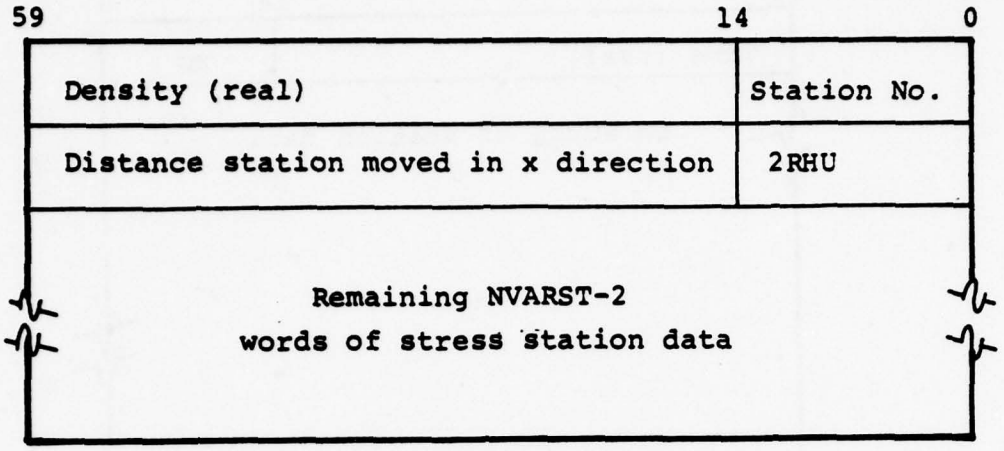


Figure 15. CDC Station Data (STRESS ≠ 0).

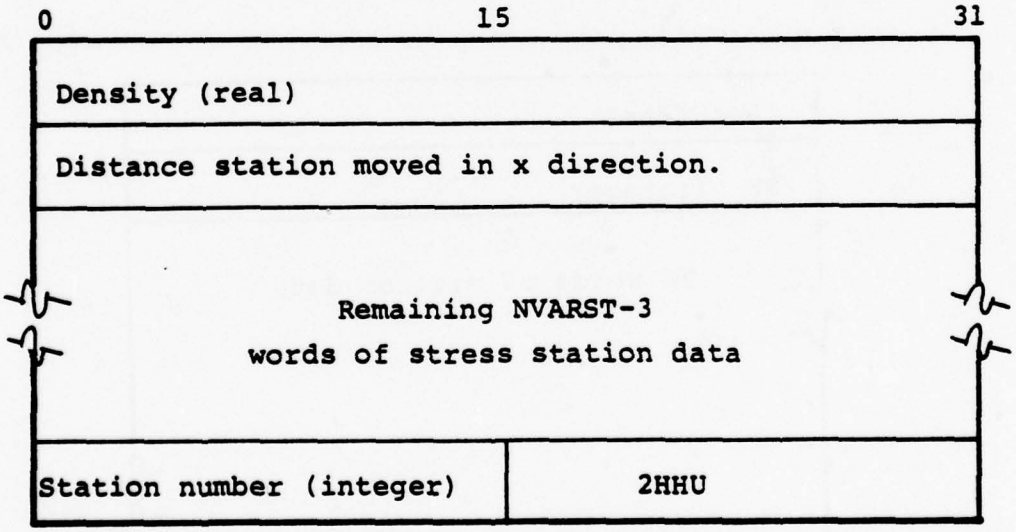


Figure 16. IBM Station Data (STRESS ≠ 0).

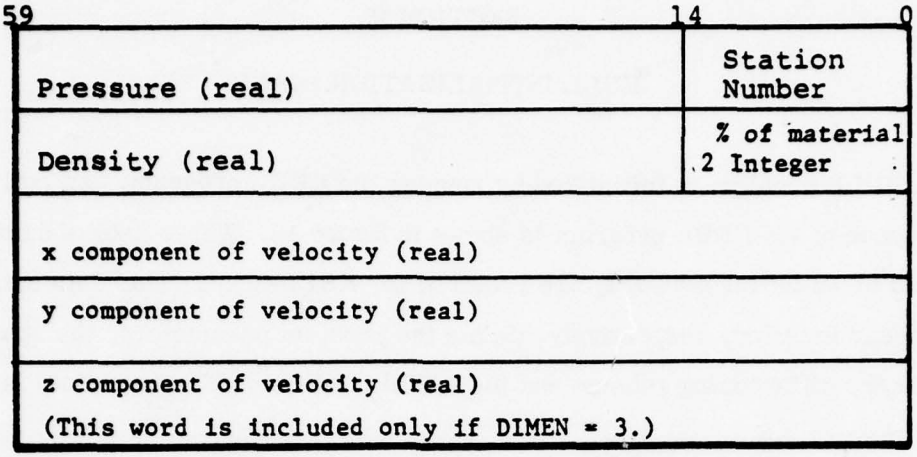


Figure 17. CDC Station Data (STRESS = 0).

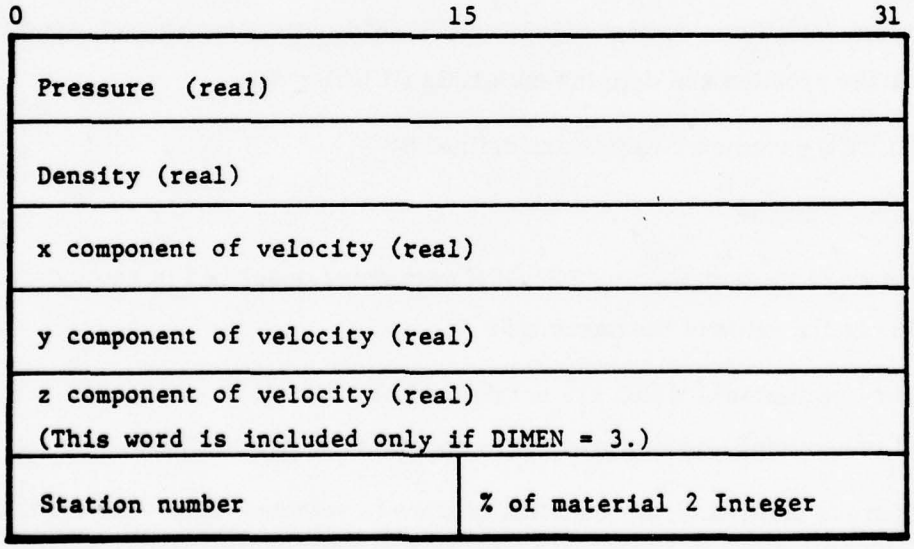


Figure 18. IBM Station Data (STRESS = 0).

SECTION V

HULL INITIALIZATION (KEEL)

A HULL problem is initialized by running the KEEL program. An outline of the sequence of the KEEL program is shown in figure 19. Three sets of data, each identified by an initial keyword, are required for KEEL input. This data set, which must appear in order, respectively, define the problem parameters, the spatial mesh for the differencing scheme and the initial condition. The input data is described as follows.

1. DEFINING THE PROBLEM

The input which follows the keyword KEEL if processed by KEEL to define the problem. The problem definition processing is terminated by the keyword MESH. The problem definitions consist of setting ZBLOCK parameter values, specifying a title for the problem and defining materials (if EOS = 6).

ZBLOCK parameter values are defined by

pname = pvalue

where pname is the name of any ZBLOCK parameter described in section III.2.1, and pvalue is the value of the parameter.

Those parameters which are not defined will default to zero or the values specified in section 2.4. However, it is necessary to define PROB.

When the multimaterial equation of state is selected (EOS = 6), the materials must be specified. The syntax is

matid = matn

where matid is a material identifier (refer to table 2 for material identifiers) and matn is the number to be assigned to the material. They must be NM materials specified, with each assigned a unique number between 1 and NM.

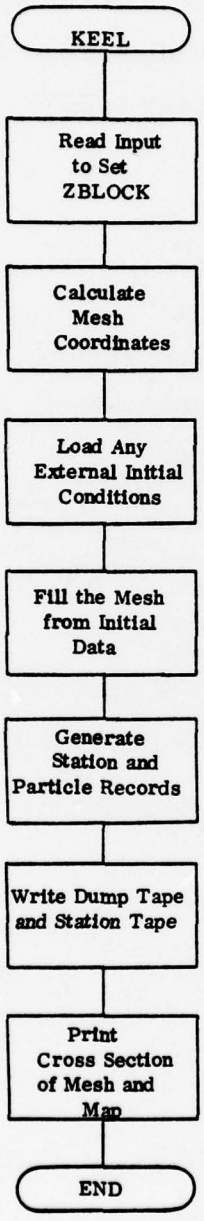


Figure 19. HULL Initialization Sequence

The parameters RERUN and DENSITY, used by BOW but ignored by KEEL, may also be included in the problem definition. RERUN overrides a protective function of BOW and permits KEEL to be executed although data already exist for the problem. DENSITY overrides the default tape density in BOW. Syntax for this parameter is described in subsection IV.1.3.

2. DEFINING THE MESH

The keyword MESH begins the mesh definition. In any direction, the mesh may be defined by a constant grid, constant subgrid, or explicit definition of the boundaries. A constant grid is defined by

MESH
[X0 = x_o] XMAX = x_m [Y0 = y_o] YMAX = y_m
[Z0 = z_o ZMAX = z_m]

Here the mesh is divided into iq cells of equal width: $dq = (q_m - q_o)/iq$, where $q = x, y, z$ and $iq = IMAX, JMAX, KMAX$, respectively.

To generate a constant subgrid, the syntax is

MESH CONSTANT SUBGRID
[X0 = x_o] XMAX = x_m [XMAXLIMIT = $x_{m\ell}$] [XOLIMIT = $x_{o\ell}$] NX = n_x
RXPOS = r_{x+} RXNEG = r_{x-}
[Y0 = y_o] YMAX = y_m [YMAXLIMIT = $y_{m\ell}$] [YOLIMIT = $y_{o\ell}$] NY = n_y
RYPOS = r_{y+} RYNEG = r_{y-}
[Z0 = z_o] ZMAX = z_m [ZMAXLIMIT = $x_{m\ell}$] [ZOLIMIT = $z_{o\ell}$] NZ = n_z
RZPOS = r_{z+} RZNEG = r_{z-}

In each direction, the region from q_o to q_m consists of nq cells of equal width. From q_m to $q_{m\ell}$, the successive cell widths have the ratio r_{q+}, and in the negative direction from q_o to $q_{o\ell}$ successive cell widths have the ratio r_{q-}, where q is x, y, or z.

To explicitly describe the mesh, the syntax is

MESH

$X0 = x_o$ $NX = n_{x1}$ $DX = d_{x1}$ ... $NX = n_{xn}$ $DX = d_{xn}$

$Y0 = y_o$ $NY = n_{y1}$ $DY = d_{y1}$ ... $NY = n_{yn}$ $DY = d_{yn}$

$Z0 = z_o$ $NZ = n_{z1}$ $DZ = d_{z1}$ ... $NZ = n_{zn}$ $DZ = d_{zn}$

This generates a mesh in the q(q = x,y,z) direction which begins at q_o and consists of successive groups of n_{qi} cells of equal width d_{qi}, i = 1, ... n. The n_{qi} are independent, but Σn_{xi} = IMAX, Σn_{yi} = JMAX, and Σn_{zi} = KMAX.

3. GENERATING THE INITIAL CONDITIONS

The actual generation of the initial conditions is initiated by the keyword GENERATE and terminated by the keyword END. In the generation process, the mesh is initialized and trace particles or sensor stations are included in the mesh. The mesh initial conditions are defined either by an isothermal sphere superimposed on ambient conditions by data from other calculations or by use of the package options.

3.1 Isothermal Sphere Initialization

The syntax for isothermal sphere initialization is

GENERATE

[ISOENERGY = e] [YIELD = y_e] [HOB = h_b]

[XOB = x_b] [YOB = y_b] [T = t_b]

where

e is the isothermal sphere energy density (ergs/g). The default is 2×10^{12} ergs/g.

y_e is the device yield in KT. Default is the value set in ZBLOCK;

h_b is the height of the burst in kilometers. Default is 0.

x_b is the x coordinate of the burst in centimeters (three dimensional mesh only. Default is 0.

y_b is the y coordinate of the burst in centimeters (three dimensional mesh only. Default is 0.

t_b is the initial time. If set in ZBLOCK then that value is used. Otherwise it is calculated.

The KEEL generation routine calculates the ambient density ρ_b at altitude h_b by using the atmosphere routine that was selected.

The radius of the isothermal sphere is

$$r_i = \frac{1 \times 10^{19} y_e}{\sqrt[3]{\rho_b e}}$$

The initial conditions then represent a sphere of air of radius r_i, uniform energy e, and density ρ_b, superimposed on air with ambient conditions defined by the atmosphere routine.

3.2 Initial Conditions from Other Calculations

KEEL can use information from other calculations as initial conditions for a HULL problem. The initial conditions can be from general card input, a KNFIRE output deck, a DYTSAP dump tape, a SAP dump tape, a SPUTTER dump tape, another two-dimensioned HULL dump tape, or an LLL RAD9 tape. Ambient conditions

defined by the atmosphere routine are used for all of the mesh outside of the region defined by the other calculation. This form of initialization is begun by the keyword FIREIN with some of the following sets of parameters.

3.2.1 General Card Input

The syntax of general card input is

CARDINPUT

followed by sets of card, each of which contains a set of values r, e, ρ, v, IF, where

r is the radius in a one-dimensioned mesh in cm;
e is the internal energy density in ergs/g;
ρ is the density in g/cm^3;
v is the radial velocity in cm/s;
IF is the termination variable;
$\neq 0$ means this is the last card.

The format for these cards is (4E12.5,I5). This input generates a sphere of air as described in the cards on a mesh of ambient air.

3.2.2 SAP, DYTSAP or RAD9 Input

KEEL will generate a sphere in a two- or three-dimensioned mesh from a SAP, DYTSAP or RAD9 tape with the following command

{SAP DYTSAP RAD9} TAPE = VSN

where VSN is the VSN of the SAP, DYTSAP or RAD9 tape.

3.2.3 SPUTTER Input

Initialization from a one-dimensional SPUTTER problem is performed by the command

SPUTTER FB = f_b

where f_b is the FB identifier of the SPUTTER problem.

3.2.4 KNFIRE Input

Initialization from a KNFIRE output deck is performed by

KNFIRE CARDS = n_c

where n_c is the number of cells in the KNFIRE mesh on the cards that follow.

3.2.5 Scaling of Input Conditions

When it is required that the input conditions be scaled from the yield and altitude of an input one-dimensional calculation to the yield and altitude of the HULL problem, KEEL will perform standard yield and altitude scaling by the command

SCALE [YIELDIN = y_i]

This command follows any of the one-dimensional calculation initializations. Here y_i is the yield of the device in the input data.

3.2.6 HULL Initialization

A three-dimensional HULL calculation can be initialized from a two-dimensional HULL calculation and a two-dimensional HULL calculation can be initialized from another two-dimensional HULL calculation with a different mesh by the command

HULL TAPE = vsn

where vsn is the VSN of the HULL dump tape to be used.

3.2.7 Geometry

In initialization from previous calculations, the region thus initialized must be defined. The default bounds for the region are determined by the maximum extent of the calculation data. Thus, for a one-dimensional previous calculation, the initialized region is a sphere centered at the burst point (x = 0.0, y = HOB in two dimensions; x = XOB, y = YOB, z = HOB in three dimensions). For a two-dimensional previous calculation, the initialized region is a cylinder with bounds corresponding to the edges of the previous calculation mesh.

The initialized region may be redefined by inclusion of a geometry description after the FIREIN data. Geometry descriptions are discussed in subsection 3.4. For this application the region described must be wholly contained within the data region.

3.3 Package Option for Complex Geometry

The most powerful method for defining the initial conditions in KEEL is the PACKAGE option. The region of the mesh to be initialized is described by beginning with the keyword PACKAGE followed by the name of the material to be inserted, optional parameters describing the initial state, and geometry descriptions defining the region being initialized by PACKAGE.

$$\text{PACKAGE } matid \left[U = v_1\right] \left[V = v_2\right] \left[W = w_3\right] \left[I = e_i\right] \left[\text{RHO} = \rho\right]$$

where matid is the HULL material identifier for the material to be inserted. This is required. (See table 2 for a list of current materials and identifiers.)

- v_i are the initial velocity components of the inserted material (in two dimensions v_1 = radial velocity and v_2 = axial velocity, while in three dimensions $v_1 = v_x$, $v_2 = v_y$, and $v_3 = v$.) Any velocity not specified receives a default value of 0 (cm/s).
- e_i is the initial internal energy of the inserted material. If this is not specified, the ambient value is used (ergs/g).

ρ is the initial density of the material being inserted. The default value is the ambient density (g/cm^3).

The material identification and initial state parameters are followed by the geometry description.

3.4 Geometry Description

Each geometry description consists of one or more geometry definitions. A geometry definition begins with one of the geometry keywords and includes parameters defining a shape associated with that keyword. The described region is the region covered by the first geometry definition, but excluding any region which is within any of the subsequent definitions. Thus, the region is described by a single region with zero or more excluded areas.

The current geometry keywords are RECTANGLE, CIRCLE, TRIANGLE, PARABOLA, and HYPERBOLA. The syntax associated with each keyword is described below. Alternative forms for parameters are grouped within curly brackets; square brackets indicate that the parameters are optional and defaults are provided.

3.4.1 RECTANGLE

This definition is used to establish a rectangle in two dimensions and a rectangular prism in three dimensions. The syntax is

RECTANGLE

$$\left[\left\{X1 = x_1 \;\; XL = x_1 \;\; X\emptyset = x_1 \;\; XLEFT = x_1\right\}\right]$$

$$\left[\left\{X2 = x_2 \;\; XR = x_2 \;\; XRIGHT = x_2\right\}\right]$$

$$\left[\left\{Y1 = y_1 \;\; YB = y_1 \;\; Y\emptyset = y_1 \;\; YBOT = y_1 \;\; YBOTTOM = y_1\right\}\right]$$

$$\left[\left\{Y2 = y_2 \;\; YT = y_2 \;\; YA = y_2 \;\; YTOP = y_2\right\}\right]$$

$$\left[\left\{Z1 = z_1 \;\; ZB = z_1 \;\; Z\emptyset = z_1 \;\; ZBOT = z_1 \;\; ZBOTTOM = z_1\right\}\right]$$

$$\left[\left\{Z2 = z_2 \;\; ZT = z_2 \;\; ZTOP = z_2\right\}\right]$$

where (refer to figures 20 and 21)

x_1 is the minimum extent of the figure in the x dimension (radial dimension in two dimensions and x dimension in three dimensions);

x_2 is the maximum extent of the rectangle in the x dimension;

y_1 is the minimum extent of the rectangle in the y dimension (axial dimension in two dimensions and y dimension in three dimensions);

y_2 is the maximum extent of the rectangle in the y dimension;

z_1 is the maximum extent of the rectangular prism in the z dimension for a three-dimensional mesh;

z_2 is the maximum extent of the rectangular prism in the z dimension.

The defaults are $x_1 = x_{min}$, $x_2 = x_{max}$, $y_1 = y_{min}$, $y_2 = y_{max}$, $z_1 = z_{min}$, and $z_2 = z_{max}$.

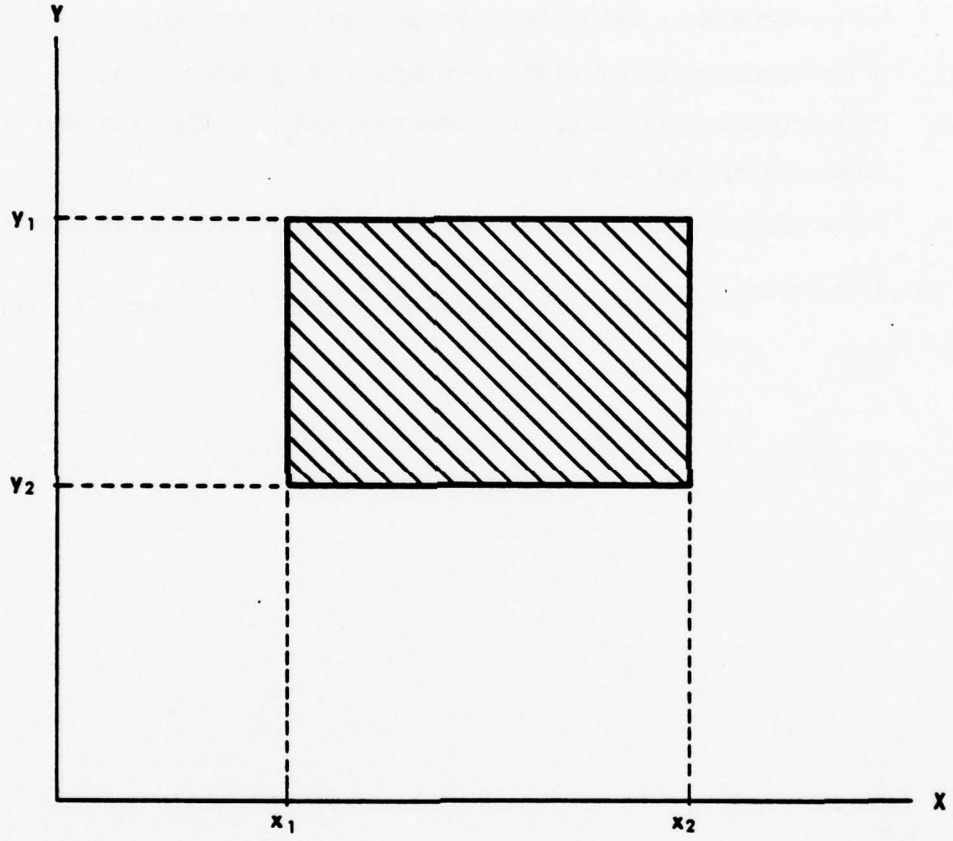


Figure 20. Two-Dimensional Rectangle

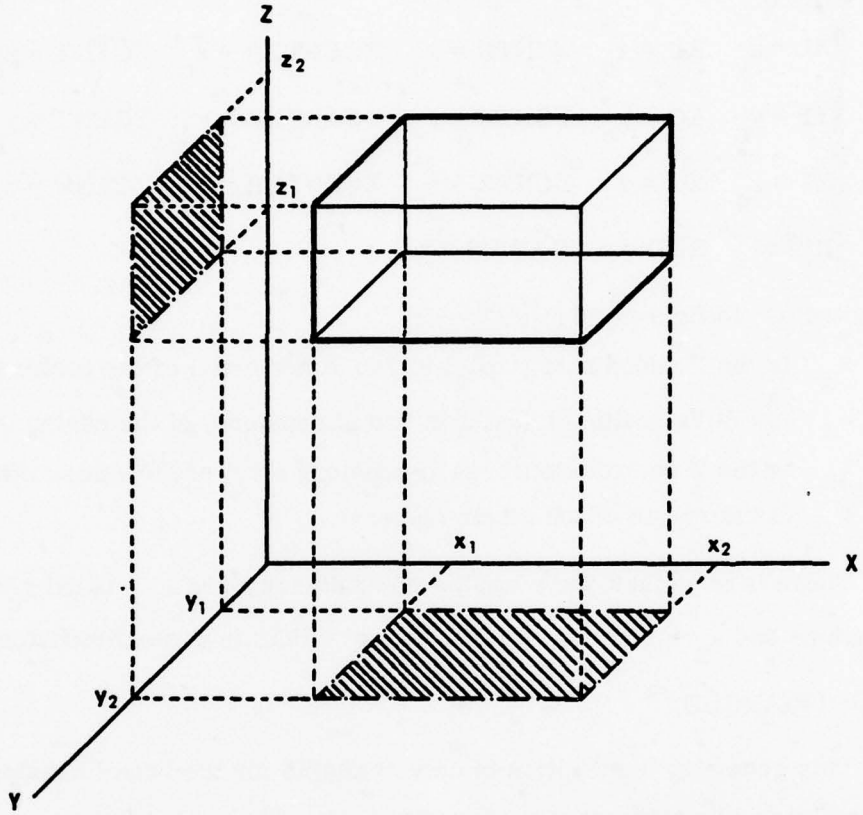


Figure 21. Three-Dimensional Rectangle

3.4.2 CIRCLE

This description defines a circle in the two-dimensional mesh and the three-dimensional mesh. The syntax for this description

CIRCLE

$$\left[\left\{X1 = x_c \quad XC = x_c \quad XCNTR = x_c \quad XCENTER = x_c \quad XCENT = x_c\right\}\right]$$

$$\left[\left\{Y1 = y_c \quad YC = y_c \quad YCNTR = y_c \quad YCENTER = y_c \quad YCENT = y_c\right\}\right]$$

$$\left[\left\{Z1 = z_c \quad ZC = z_c \quad ZCNTR = z_c \quad ZCENTER = z_c \quad ZCENT = z_c\right\}\right]$$

$$\left\{R = r \quad RAD = r \quad RADIUS = r\right\}$$

where (refer to figures 22 and 23)

x_c is the X coordinate (radial in two dimensions) of the center of the sphere;

y_c is the Y coordinate (axial in two dimensions) of the center of the sphere;

z_c is the Z coordinate (three dimensions only) of the center of the sphere;

r is the radius of the circle (sphere).

There is no default for r but the other defaults are $x_c = 0$ and y_c = HOB in two dimensions and x_x = XOB, y_c = YOB, and z_c = HOB in three dimensions.

3.4.3 TRIANGLE

This geometry description is only available for the two-dimensional mesh. The description is of a triangular area in the mesh. The syntax is

TRIANGLE

$$X1 = x_1 \quad Y1 = y_1 \quad X2 = x_2 \quad Y2 = y_2$$

$$X_3 = x_3 \quad Y3 = y_3$$

where (see figure 24) x_i, y_i are coordinates of the corner of the triangle. There are no default values for a triangle description.

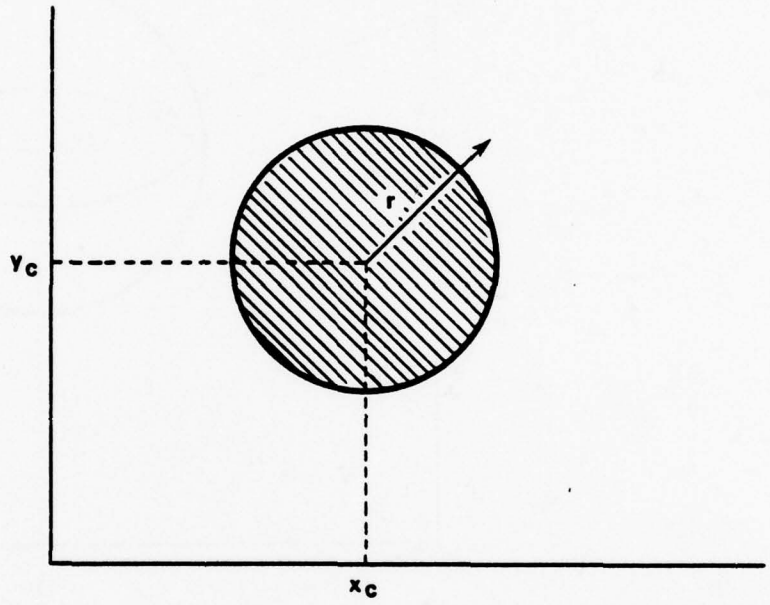


Figure 22. Two-Dimensional Circle

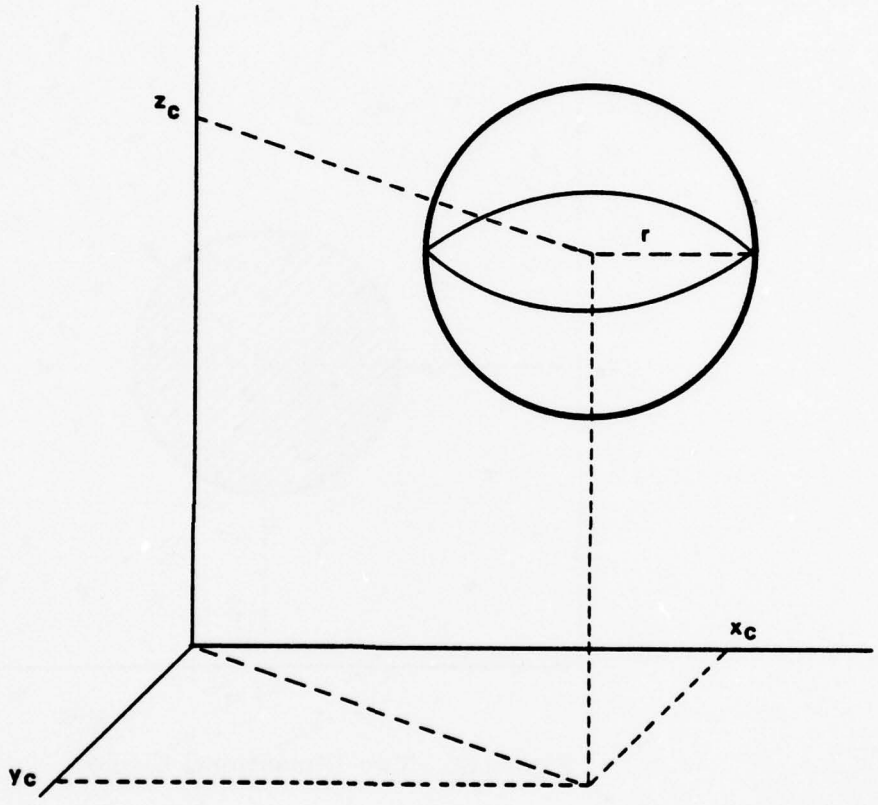


Figure 23. Three-Dimensional Circle (Sphere)

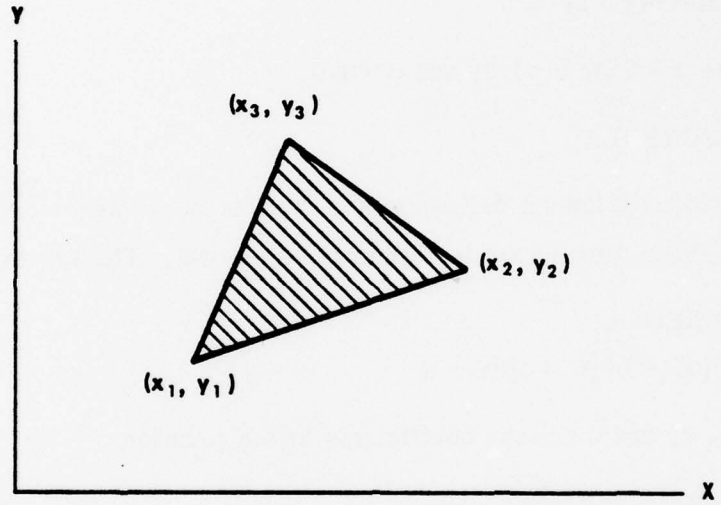
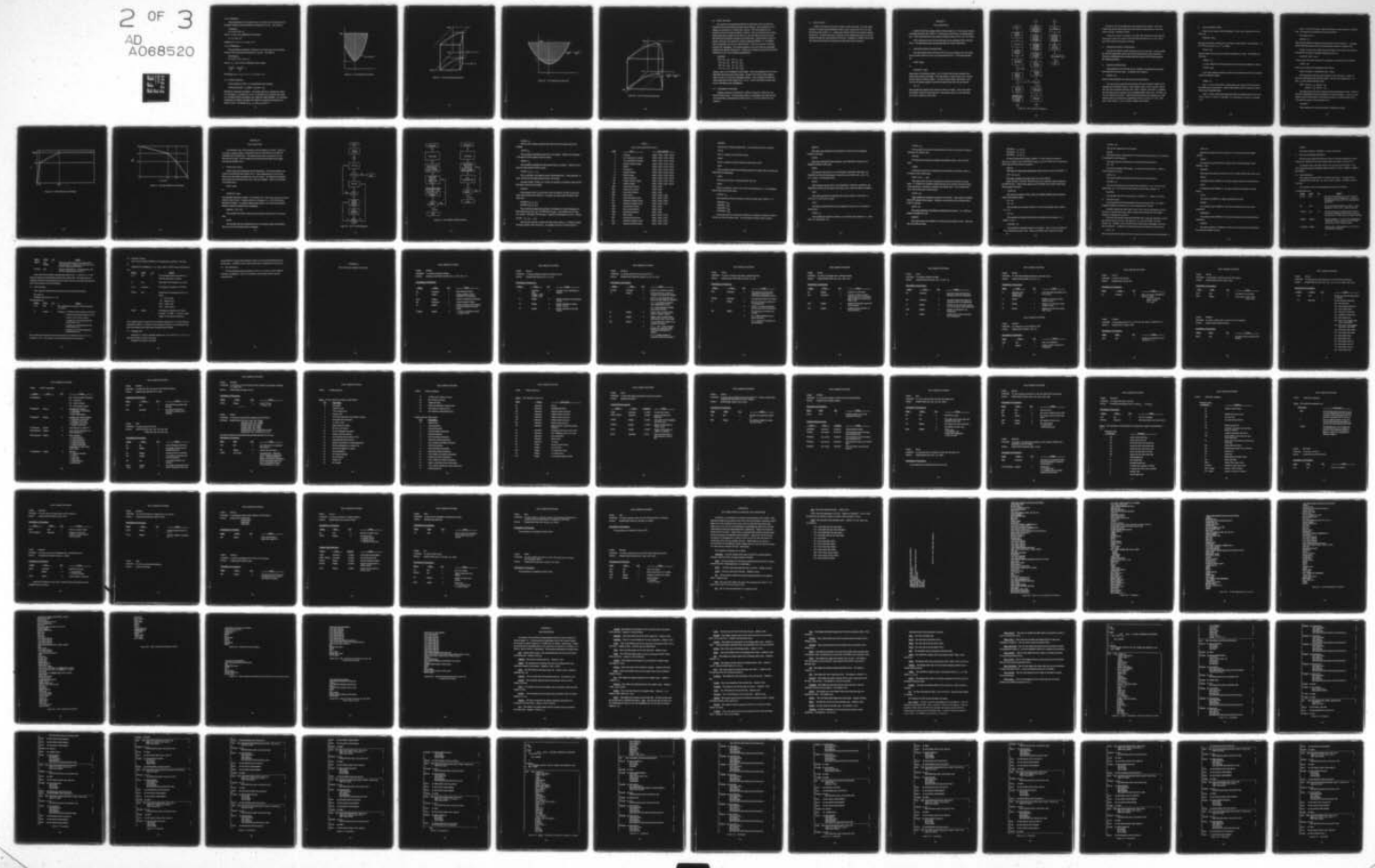


Figure 24. Triangle

AD-A068 520 COMPUTER SCIENCES CORP ALBUQUERQUE N MEX F/G 19/4
HULL SYSTEM GUIDE.(U)
JAN 79 L P GABY F29601-78-C-0012
UNCLASSIFIED CSC-C4-C-4041 AFWL-TR-78-115 NL

2 OF 3
AD
A068520



3.4.4 PARABOLA

This description is for a parabola in the Y direction in two dimensions and a parabolic cylinder in three dimensions (see figures 25 and 26). The syntax is

PARABOLA
[A = a] [B = b] [C = c]

where a, b, and c are coefficients of the equation

$$(y - a) = b(x - c)^2$$

Defaults are a = 0.0, b = 1.0, and c = 0.0.

3.4.5 HYPERBOLA

This description set defines a hyperbola in two dimensions and a hyperbolic cylinder in three dimensions (see figures 27 and 28). The syntax is

HYPERBOLA
[A = a] [B = b] [C = c] [D = d]

where a, b, c, and d are the coefficients of the equation

$$\frac{(x - a)^2}{b^2} - \frac{(x - c)^2}{d^2} = 1$$

The defaults are a = 0.0, b = 1.0, c = 0.0, and d = 1.0.

3.5 Particle Generation

Tracer particles (CODE = 1) are generated by the command

PARTICLE(S) [NSC = n_c] [NSR = n_r] [NSP = n_p]

followed by a geometry description. The routine will put n_c particles per column (I—x direction), n_r particles per row (J—y direction) and n_p particles per plane (K—I direction in three dimensions only) within the region specified. If no geometry description is provided, the region last defined by a geometry description or by default is used. The defaults for n_c, n_r, and n_p are each = 1.

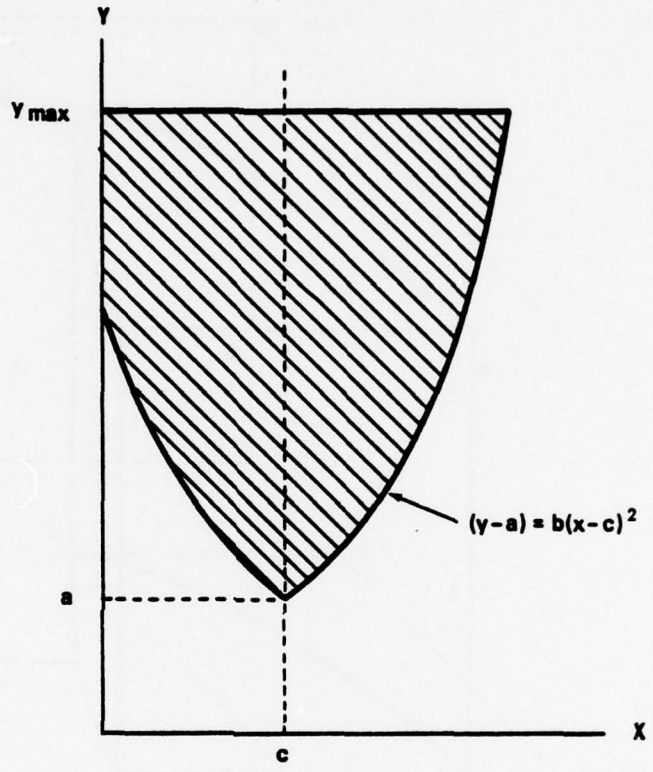


Figure 25. Two-Dimensional Parabola

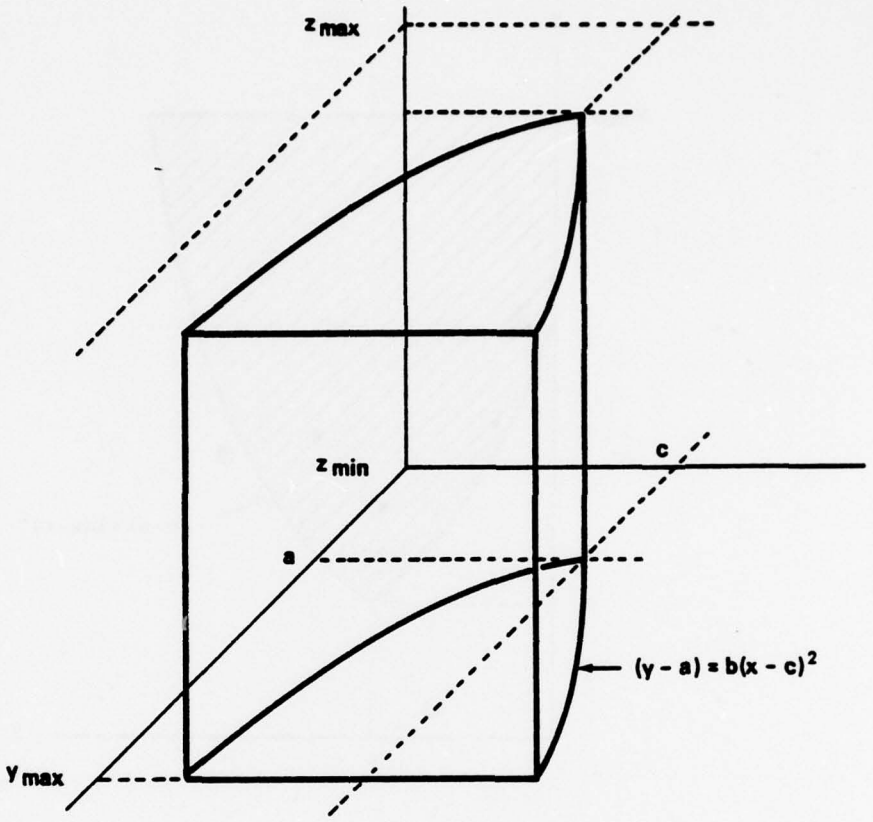


Figure 26. Three-Dimensional Parabola

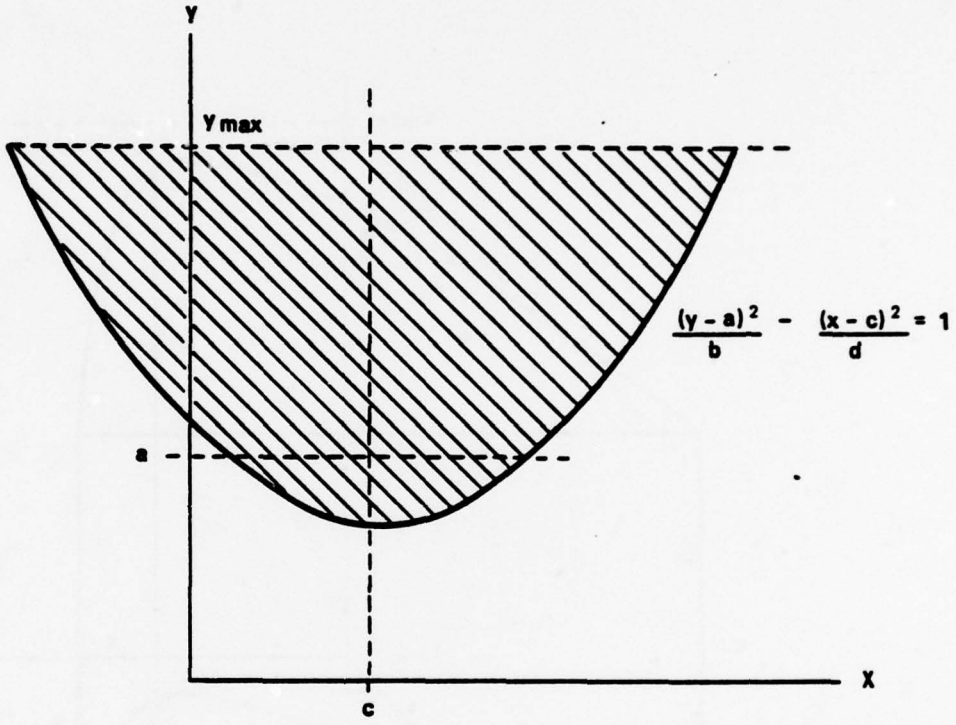


Figure 27. Two-Dimensional Hyperbola

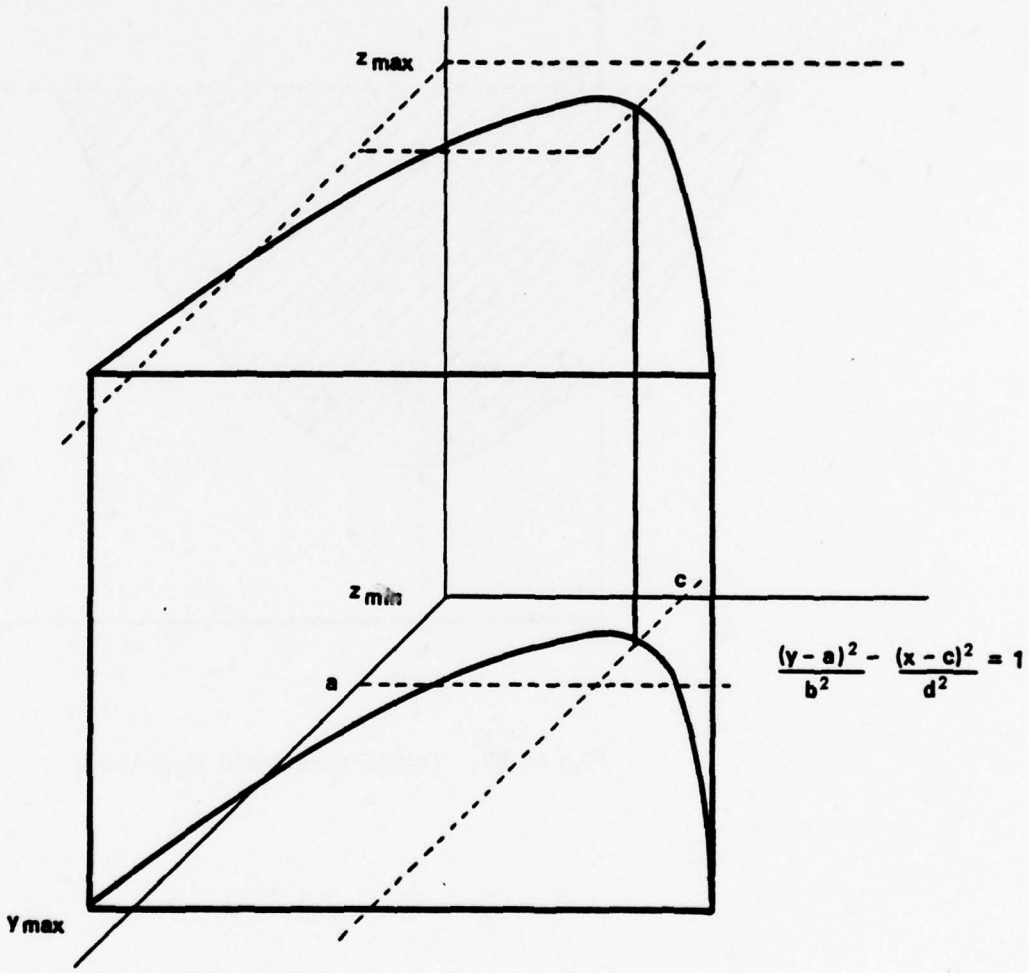


Figure 28. Three-Dimensional Hyperbola

3.6 Station Generation

For purposes of comparing calculations to experiment, HULL provides for continuous sensor monitoring at positions called stations. State parameters at the stations are recorded on a "station tape" data file. The state parameters are scanned at each cycle of the calculation. However, data are recorded only for those stations that show a change of more than 1% in any parameter from the values for that station last previously recorded on the station tape. State variables saved are density, pressure, and velocity for nonsolid problems (STRESS = 0). In addition, stress components are saved for solid problems. There are two types of stations: Eulerian and Lagrangian. The Eulerian stations do not move while the Lagrangian stations move with the material flow. Stations are created by the keyword STATIONS followed by one or more coordinate specifications:

STATIONS

$$\left\{\begin{array}{ll} (XS = x_1 \ldots x_m\ , & YS = y_1 \ldots y_n) \\ (YS = y_1 \ldots y_n\ , & XS = x_1 \ldots y_m) \\ (XL = x_1 \ldots x_m\ , & YL = y_1 \ldots y_n) \\ (YL = y_1 \ldots y_n\ , & XL = x_1 \ldots x_m) \end{array}\right\}$$

where x_1 and y_1 are coordinates of the stations. Here the parentheses delimit alternate forms and are not part of the syntax. XS and YS are used for fixed stations while XL and YL are used for Lagrangian stations. Each coordinate specification causes generation of mxn stations ($m \geq 1$, $n \geq 1$). Each x-coordinate is paired with every y-coordinate in the specification.

3.7 Termination of Generation

Multiple sequences of GENERATE, FIREIN, PACKAGE, PARTICLE, and STATION may be used. The generation process is terminated by the keyword END or the end of file as detected by the coded read (i.e., an EOR on INPUT for CDC systems).

4. KEEL OUTPUT

A KEEL run produces two types of output: printed and taped. The tape output consists of a dump tape describing the mesh in such a manner that HULL can start the problem and, if NSTN > 0, a station tape with the ZBLOCK and particle positions written on it. Printed output gives a summary of the initialization performed and the amount of energy and mass inserted in the mesh. In addition, a printout of one vertical column in the mesh and a printer plot of the entire mesh is produced.

SECTION VI

HULL EXECUTION

The flow of the Hull program itself is shown in figure 29. The program requires an existing dump tape and, if NSTN > 0 in the dump tape ZBLOCK, an existing station tape. These tapes must have been generated during a previous HULL run or by KEEL.

HULL searches for and reads the first input record beginning with the keyword HULL. The input data includes startup infomation and control information.

1. PROGRAM STARTUP INFORMATION

This information follows the keyword HULL and is terminated by an end of file (end of record on INPUT in CDC) or by the keyword INPUT. The startup keywords are

PROB = pnum

or

PROBLEM = pnum

where pnum is the problem number. HULL verifies that the tape mounted is the right problem before proceeding. This information is required since HULL will not proceed without a problem number. On those systems using a tape library, BOW will use the problem number to select the required tapes from the tape library.

[T = t]

This specifies the problem time where the restart is to begin. HULL will restart the problem at the first dump which has a time greater than t or at the last dump on the tape, whichever comes first.

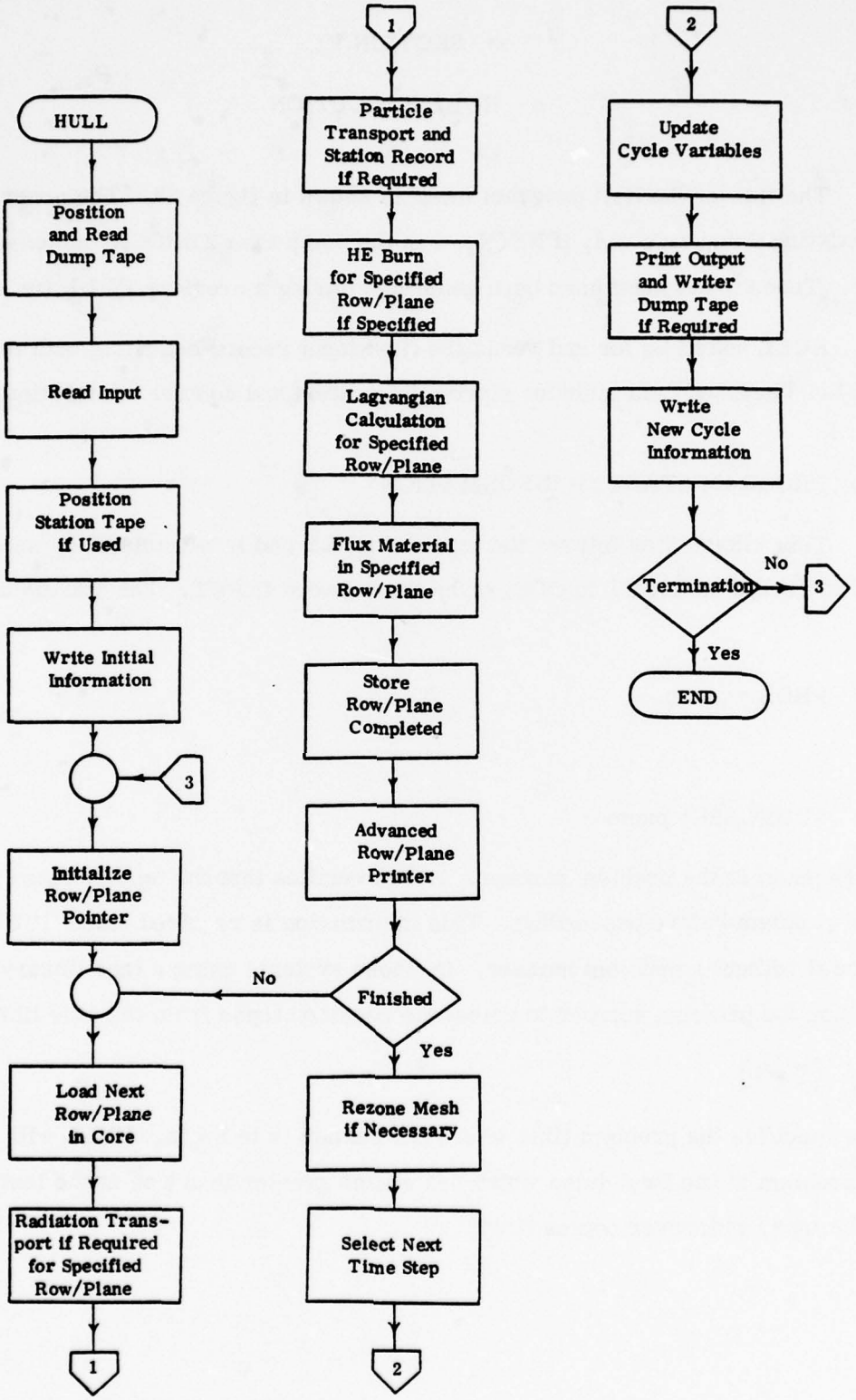


Figure 29. HULL Execution Sequence

[CYCLE = nc] This specifies the cycle number for the restart. HULL will restart from the first dump where the cycle number is greater than nc or the last dump on the tape, whichever is first.

If the cycle or time is not given, or if BOW (CDC system only) shows that the last previous start was a cycle or time start from the same cycle or time, HULL will restart from the last dump on the tape.

2. PROGRAM CONTROL INFORMATION

The user has control over the execution of HULL in two ways. He may modify the ZBLOCK parameters, and he may set local parameters for the current run. Parameter modification data are placed after the keyword INPUT that terminates the startup information.

3. ZBLOCK MODIFICATION

Any parameter in the ZBLOCK may be modified by specifying the parameter name followed by the new value. An example of the syntax is

TTSTOP = tp

where tp is the maximum CPU time in hours for this problem.

The user should realize that the following ZBLOCK parameters should not be changed from their KEEL values: CODE, DIMEN, IMAX, JMAX, ISLAND, KMAX, NH, NM, NOP, NROWPB, NPLPB, NPP, NSTN, NVARST, and PROB. In addition, the user must exercise extreme care in modifying BURN, EOS, FAIL, FLUXER, IQ, JQ, KQ, MAGFLD, and RAD. Finally the following ZBLOCK parameters are maintained by HULL and should not be modified by the user: CYCLE, DT, ELC, ETH, MLC, MTH, SUME, T, TLC, TTIME, TTIME6, and TTIME7.

4. LOCAL MODIFICATIONS

These are two types of local modification. First, local variables may be set. These are

MRELER = $\max_e$

this sets the maximum relative error allowance in total energy or mass (default = 1.0×10^{-15} for CDC and 1.0×10^{-8} for IBM).

CSTOP = ncy

This parameter sets the cycle at which the calculation is to stop. The default is no cycle stop.

RTSTOP = t_r

t_r is the maximum CPU time in hours for the current run (default is no limit).

DCYST = nscy

nscy is the maximum number of cycles of the problem that HULL will calculate in this run (default no limit).

TIMES = pt

If pt = 1, HULL will produce a printed output and a dump in TAPE4 whenever the problem time corresponds to standard high-altitude times (36 times per decade evenly spaced logarithmically).

If pt = 2, HULL will produce output and dumps at problem times of 10 ms and 30 ms, every 0.1 s from 0.1 s through 1.2 s, and every 0.5 s from 1.5 s through 6.0 s.

If pt = 3, HULL will produce output and dumps at regular intervals of problem time. The interval is specified by the keyword DMPINT.

The default is pt = 1

DMPINT = dt

where dt is the output and dump problem time interval when TIMES = 3. All dumps except termination dumps will be at problem times which are a multiple of dt.

Finally, the user may specify the yield strength for the solids in the problem. The keywords to begin the strength definition process are

MATPROP MAT = matn

where matn is the number that has been assigned to the material in the problem.

YLDST = yld

where yld is a initial yield strength for the material.

WORK [YLDMAX = yldmax] [EPLAST = emax]

This instruction allows the yield strength to vary with energy. Figure 30 shows the relationship of the internal energy ϵ and the yield strength due to Yw. Here ϵ_g is the ambient internal energy.

SOFT YFRAC1 = yf_1 IFRAC1 = ef_1
YFRAC2 = yf_2 IFRAC2 = ef_2

This input allows the user to specify the thermal softening of solids. Figure 31 shows the relationship of the yield strength to the internal energy, e. Here e_r is the internal energy which produces 1 atmosphere pressure at ambient density and e_m is the internal energy at which melting occurs.

ENDPROP

This terminates the material properties modification section.

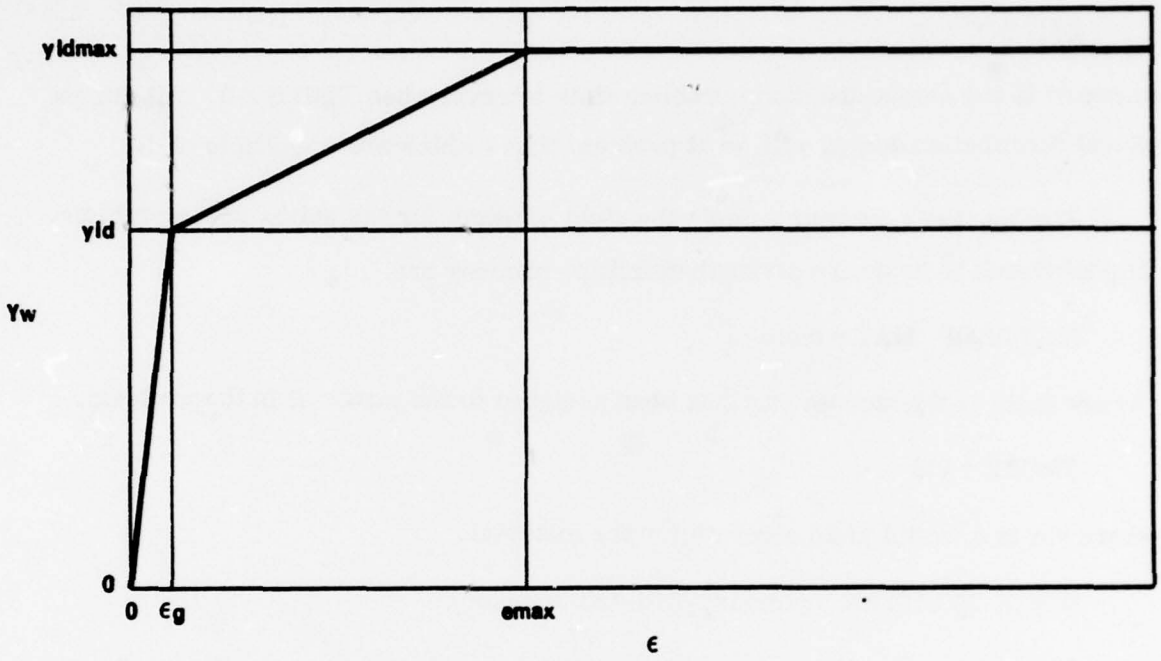


Figure 30. Work Hardening Yield Strength

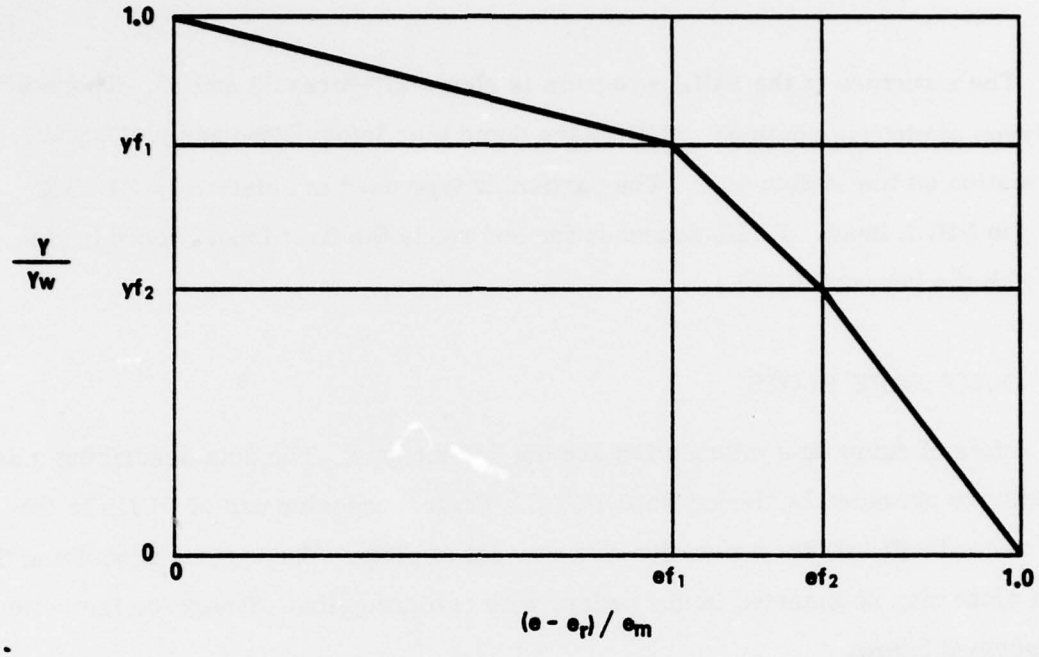


Figure 31. Thermal Softening Yield Strength

SECTION VII

PULL EXECUTION

The structure of the PULL program is shown in figures 32 and 33. There are two types of plotting routines: plotting the dump tape information and plotting the information on the station tape. The particular type used is selected by PLANK from the PULL input. PULL searches for and reads the first input record beginning with the keyword PULL.

1. DUMP TAPE PLOTS

Plots of dump tape information are the default type. The data describing a set of plots are preceded by the keyword PULL. Each successive use of PULL in the input record will initiate generation of a new set of plots. The options describing the set of plots may be inserted in any order, with one exception. Syntax for the options is described below.

PROB = pnum

or

PROBLEM = pnum

This specifies the problem number. If it is given, PULL checks this against the problem number in the ZBLOCK. Otherwise there is no checking, i.e., PULL will plot the tape that is provided. On systems using the tape library, BOW will use the problem number to select the required tapes for plotting.

DENSITY = {HY PE}

This specifies the density of tape when not using the tape library (CDC systems only).

STIME

This specifies that only standard times are to be plotted; dumps at nonstandard times such as run termination times are skipped.

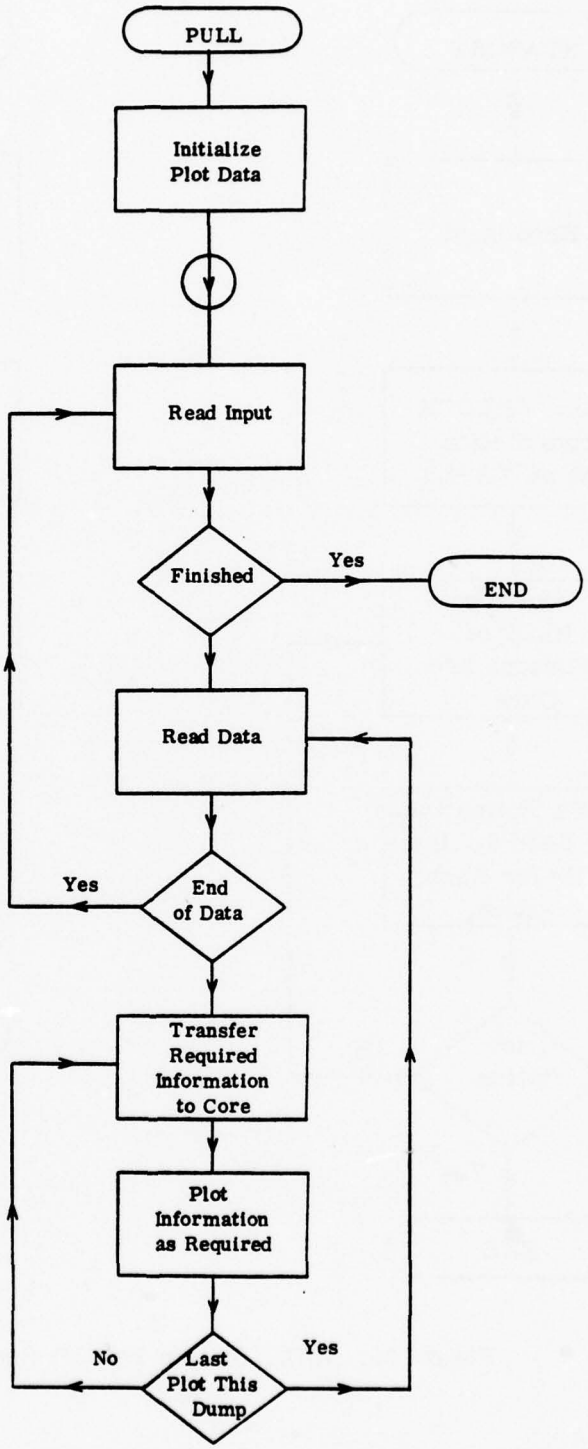


Figure 32. HULL Plotting Sequence

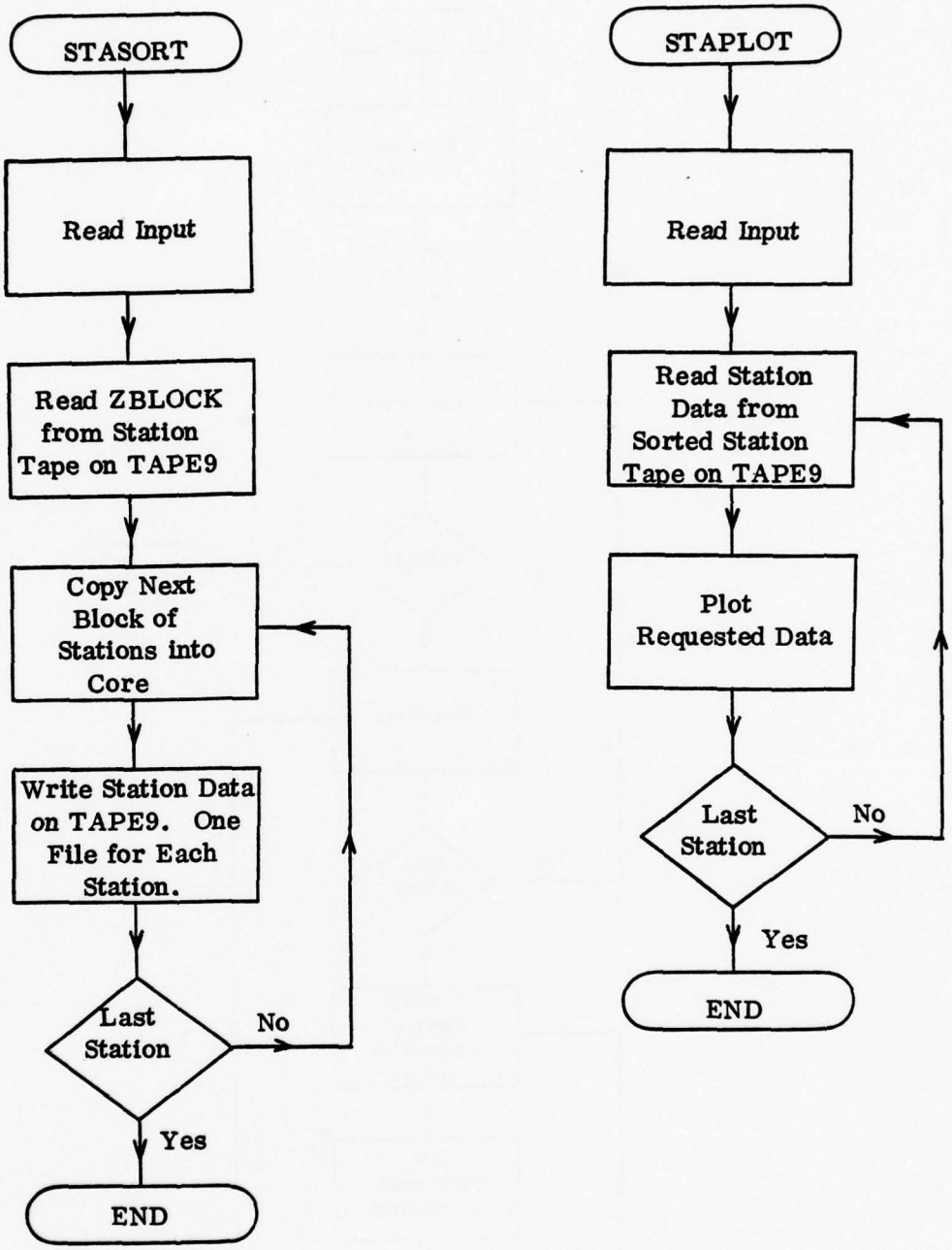


Figure 33. HULL Station Plotting Sequence.

DTIME = d_t

Here d_t is the minimum problem time interval between dumps selected for plotting.

FTIME = t_f

This parameter specifies the first time to be plotted. Default is the beginning of the tape (or of the problem if BOW is used).

LTIME = t_e

This parameter specifies the last problem time to be plotted. Default is end of tape (or of the problem if BOW is used).

CTIME = $t_1, t_2, \ldots t_n$

The t_i are times to be plotted in order of increasing value. This parameter, if used, must be the last option inserted for the set of plots.

If neither STIME, CTIME, nor LTIME are specified, all dumps on tape from the first time to last time are plotted.

HEADER

This instructs PULL to use the next card in the input as the title for the plots rather than the title in the ZBLOCK. No options may appear after HEADER on the same card.

xxxHHST $[y_i, y_c, y_d]$
xxxVHST $[x_i, x_f, x_d]$

These keywords request the generation of horizontal or vertical histograms at lines between $y_i(x_i)$ and $y_c(x_c)$ at increments of $y_d(x_d)$. xxx specifies the data tape to be plotted. See table 3 for data types. Default is one histogram for row 1 (column).

xxCONT $[c_1, c_2, \ldots c_n]$

This keyword requests a contour plot with contour values c_i. (Default: program calculates contour values from data.) xx specifies the data to be plotted (table 3).

TABLE 3

PLOT DATA SELECTION TABLE

XXX	Data	Plot Available
P	pressure	HHST, VHST, CONT, GRAD
U	x(r) component of velocity	HHST, VHST, CONT, GRAD
V	y(r) component of velocity	HHST, VHST, CONT, GRAD
E	specific ambient energy	HHST, VHST, CONT, GRAD
D	density	HHST, VHST, CONT, GRAD
T	temperature	HHST, VHST, CONT, GRAD
M	material density	HHST, VHST, CONT, GRAD
K	kinetic energy	HHST, VHST, CONT, GRAD
RP	relative pressure	HHST, VHST, CONT, GRAD
RE	relative internal energy	HHST, VHST, CONT, GRAD
RD	relative density	HHST, VHST, CONT, GRAD
SRR	radial stress deviator	HHST, VHST
SZZ	axial stress deviator	HHST, VHST
SHH	hoop stress deviator	HHST, VHST
SRZ	shear stress	HHST, VHST, CONT
S2I	second invariant stress	HHST, VHST, CONT
SP1	maximum principle stress	HHST, VHST, CONT
SP2	minimum principle stress	HHST, VHST, CONT
ERR	radial strain deviator	HHST, VHST
EZZ	axial strain deviators	HHST, VHST
EHH	hoop strain deviators	HHST, VHST
ERZ	shear strain	HHST, VHST, CONT
E2I	second invariant strain	HHST, VHST, CONT
EP1	maximum principle strain	HHST, VHST, CONT
EP2	minimum principle strain	HHST, VHST, CONT

xxxGRAD

This keyword requests gradient plots. xxx specifies the data to be plotted.

VVECT

This is a request to plot velocity vectors.

SVECT

This is a request to plot the axial plus radial stress vector.

PART

This parameter requests that the tracer particles be plotted with all contour and vector plots and separately.

PART1

This keyword requests a separate particle plot only.

DUSTY

This is a request for radar cross section and absorption only. It is applicable only for plots from DUSTY tapes.

YDFRAC = y_f

This specifies the fission fraction for electron density plots (default = 1.0).

XMINGRD = x_{mn}
YMINGRD = y_{mn}
XMAXGRD = x_{mx}
YMAXGRD = y_{mx}

These keywords are to specify the minimum or maximum coordinate values for contour and vector (gradient) plots. If not specified the entire mesh is plotted.

REDUCE

This option sets XMAXGRD and YMAXGRD to include only the significant portions of the mesh.

BOTOK

When used with REDUCE this parameter sets YMINGRD to include only the significant portions of the mesh.

FIREB

This keyword causes PULL to set XMAXGRD, YMAXGRD, XMINGRD, and YMINGRD such that the plotting grid is centered on the fireball with a size 2 r by 3.5 r, where r is the fireball radius.

SHOCK

This parameter causes PULL to set XMAXGRD, YMAXGRD, XMINGRD, and YMINGRD to include only the shock at the bottom of the mesh (overridden by FIREB).

LOGV

This option specifies that logarithmically spaced contours, with values 1.0×10^{n} and 3.0×10^{n}) are to be used.

LOGD

This option specifies that the natural logarithm of the data is to be plotted as contour plots.

ICTRS = n_c

This specifies the number of contours in the contour plots (minimum = 2, maximum = 20, and default = 10).

PLTINC = p_i

This specifies the step size of the plotter and is used by the contour routines to determine its minimum step.

NOCURV

This parameter disables the nonlinear interpolation within a cell on contour plots.

COORD

This keyword causes the coordinates of the minimum and maximum values to be printed on the contour plots.

NMAT = mni, ... mnn

This option allows the user to specify the material to be included in density plots. A -1 (minus one) separates material groups and is plotted as the total density of the materials in each group or plotted in the density plots. If not specified then PULL will produce plots of total density.

NPART = np

This specifies the selection of particles to be plotted. If this option is specified every np^{th} particle will be plotted. Otherwise, every particle will be plotted unless SORTP is nonzero.

SORTP = ps

This option eliminates overstrikes in particle plots if nonzero. ps = 1020/max points to be plotted in a row.

MASSIVE

This option causes all variables to be saved from the particle record. Otherwise only two words are saved.

XPLANES = x_1, x_f, x_d

YPLANES = y_1, y_f, y_d

ZPLANES = z_1, z_f, z_d

For three-dimensional problems (DIMEN = 3), this causes the planes for x_i(y_i or z_i) to x_f(y_f or z_f) in increments of x_d(y_d or z_d) to be plotted as two-dimensional plots using all the previous plot requests.

3DPLOT

This keyword selects three-dimensional contour plots for cases with DIMEN = 3.

EYE = ex, ey, ez

This option specifies the position of the eye of the 3DPLOT.

(Note: RDCONT, RBCONT, RECONT are the only allowable contour options for use with 3DPLOT. Each of these options may be followed by the contour value which will be opaque in the plot.)

LAMBPLOT

This allows the interface of the LAMB (low altitude multiple burst) simulation model to the plotting routine.

XL = xm

YL = ym

This specifies the maximum dimension of a hard copy plotting device (default = 10.0 in.).

BD = bs

This specifies the width of the border for microfilm plots (minimum = 1.5, default = 2.2 in.).

PLOTPKG = np

This specifies the plotting package to be included. This is used by PLANK and SAIL to generate the proper code. Refer to PLOTPKG in the section on PLANK options.

FACTPL = fp

This set the scaling factor for the plots.

SLIDES

This option causes a blank frame to be left between plotted frames for convenience in mounting for slide projection.

This option causes a blank frame to be left between plotted frames.

MF = mf

This selects the type of film output. It is used by PLANT and SAIL. Refer to PLANK options for values.

THETA = th

This option specifies the rotation of the entire plot on film plotters.

CAMERA = nc

This sets the camera option on the SC 4020 (PLOTPKG = 4) nc = 35 for 35 mm camera only, nc = 9 for Xerox only and both for all other values (default = 0).

PLOTVSN

This specifies the VSN of the plot tape is PLOTPKG = 1. Default is PLOTAP.

2. STATION PLOTS

If the keyword STATION immediately follows the keyword PULL, the station program STASORT and STAPLOT are set for generators by PLANK.

STASORT unpacks the station tape on unit 4 and writes the information on unit 9. The data is converted from time order to station order with data for each station written on a separate file on unit 9.

STAPLOT will plot the output from STASORT and it reads the same PULL record that would be used by PLANK. The option XL, YL, BD, PLOTPKA, FACTPL, SLIDES, MF, CAMERA, and PLOTVSN which are described in paragraph 1 are available in STAPLOT. In addition, the following options are available for station plots:

FSTA = nf

This option specifies the number of the first station to be sorted and/or plotted (default = 1).

LSTA = ne

Specifies the number of the last station to be sorted and/or plotted (default = NSTN).

PUNCH

This keyword causes STAPLOT to punch a NEAT deck for problems where STRESS = 0 only.

ENGLISH

This causes the output from STAPLOT to be in English rather than metric units.

RADIAL

In problems where STRESS >0, this option requests the plots of radial motion.

AXIAL

In problems where STRESS >0, this option requests the plotting to be axial motion.

TOTAL

This option, if STRESS >0, requests plotting total motion.

STRESS

For problems where STRESS >0, this option requests plots of total stress components.

STRESSDEV

For problems where STRESS >0, this option requests plots of deviation stress components.

TENSOR

This option requests, if STRESS >0, plots of the second invariant and minimum and maximum principal stresses.

FLUXM

This option requests, if STRESS > 0, plots of mass flux.

3. PULL PLOT DRIVER INTERFACE

The plot driver option PLOTPKG has values of 1 through 5 supply plot drivers on the driver interface for the CDC plotting routine described in section III.3.2.

To interface PULL with other drivers, the user must set PLOTPKG = 0 (or a new value) and write five interface routines--INIPLT, UPLOT, USYMBL, NEXPLT, and TERPLT.

3.1 Plot Initialization

PULL calls the routine INIPLT to initialize the plotter. The entry form is:

SUBROUTINE INIPLT (XL, YL, BO, PLTING, MODPL, ICAMERA, THETA, FACTPL).

This routine is to set up the mapping and initializes the plot file.

The parameters are:

NAME	TYPE	I/O	USAGE
XL, YL	real	I	The size of the plotting area in the X- and Y-direction (in inches) respectively. These are the parameters set by the PULL input keyword XL and YL.
BD	real	I	The size of film plot border (in inches). This parameter is set by PULL input keyword BD.
PLTINC	real	I/O	The plot resolution (in inches) can be input (is set by PULL input keyword PLTINC) or can be set by INIPLT. Value is used in contour routines.
MODPL	integer	I	The plot situation. It is set to value specified in PLOTPKG in PULL input (or from PLOTPKG option value at PULL generate time).
ICAMERA	integer	I	Camera type. This parameter is set by PULL input keyword CAMERA.

NAME	TYPE	I/O	USAGE
THETA	real integer	I	Total plot rotation must be a multiple of 90°. This parameter is defaulted to 0 and may be set by PULL input keyword THETA.
FACTPL	real	I	The plot scaling factor. The parameter is set by PULL input keyword FACTPL.

Upon return PULL expects a plotting area which is XL + 2 * BD by YL = 2 * BD where the origin and current plotter position are at (BD, BD). The dimensions are nominally inches but can represent any unit as long as they are consistent with the units used in the routines UPLOT and USYMBL.

3.2 Vector Plotting

PULL calls the routine UPLOT to draw vector and position the plotter.

The entry is:

SUBROUTINE UPLOT (X, Y, IP)

The arguments are:

NAME	TYPE	I/O	a USAGE
X, Y	real	I	The coordinates (in inches) to which the plotter is to move.
IP	integer	I	Plot type = 1 performs same functions as last line. = 2 draw from camera position to (X, Y). = 3 move to (X, Y) (do not draw). = -1 same as 1 but resulting position then becomes (0, 0). = -2 same as 2 but resulting position then becomes (0, 0). = -3 same as 3 but resulting position then becomes (0, 0).

PULL expects this routine to move to the position (X, Y) and draw a line if /IP/ = 2. In addition, if IP < 0 the origin is reset to the final position of the plotter.

3.3 Character Plotting

PULL call the routine USYMBL to plot alphanumeric symbols. The entry is:

SUBROUTINE USYMBL(X, Y, H, LAB, THETA, NCHR) where the perimeters are:

NAME	TYPE	I/O	USAGE
X, Y	real	I	The coordinate where the plotter is to draw the characters in inches.
H	real	I	The height of the characters in inches.
LAB	character	I	The character (symbols) to be drawn.
THETA	real	I	The direction the characters are to be drawn. 0. - left to right 90. - bottom to top 1180. - right to left 270. - top to bottom
NCHR	integer	I	The number of characters in LAB to be drawn. If NCHR 0 LAB is a single integer (A) for special characters.

PULL expects routine to draw the lint of symbols starting at (X, Z) in the direction specified by THETA. In addition, PULL expects the plotter to be positioned at the end of the character just drawn when returning from USYMBL.

3.4 Changing Plots

When PULL is ready to generate another plot it calls VPLOT (L + 2.5, 0.0, -3) then calls NEXPLT which has the entry:

SUBROUTINE NEXPLT (SLIDES)

where SLIDE is a logical input parameter which is set by the SLIDES keyword from the full input. If SLIDES is true, then an extra frame is generated for 35 mm plots.

3.5 Plot Termination

The last plotting operation performed by PULL is to call the routine TERPLT which has no arguments. This is the interface to the plotting system to end the plotting session.

APPENDIX A

SAIL AND HULL LIBRARY ROUTINES

NAME: COMPR

PURPOSE: To compare character strings.

ENTRY: LOGICAL FUNCTION COMPR (M1, I1, NCR, M2, I2)

Description of Parameters

Name	Type	I/O	Usage
M1	character	I	first character string
I1	integer	I	relative position of first character to be compared
NCR	integer	I	number of compared characters
M2	character	I	second character string
I2	integer	I	relative positive of first character to be compared in second string
COMPR	logical	O	= .TRUE. if strings are equal = .FALSE. if not

NAME: FETCH

PURPOSE: To extract integers packed in a character word.

ENTRY: SUBROUTINE FETCH (N, I1, I2, I3)

Description of Parameters

Name	Type	I/O	Usage
N	character word (integer - CDC) (real*8 - IBM)	I	character word containing the integers
I1	integer	O	integer contained in first quarter of word
I2	integer	O	integer contained in second quarter of word
I3	integer	O	integer contained in last half of word

NAME: GETNUM

PURPOSE: To decode numbers of the form XXX.YYY.

ENTRY: SUBROUTINE GETNUM (NCARD, IS, IB, IC, IFN)

Description of Parameters

Name	Type	I/O	Usage
NCARD	character	I	field to be scanned for number
IS	integer	I	the first character position in NCARD to be scanned for number
		O	if IFN < 0, the character position of the first nonblank character
			if < 0, no nonblank characters were encountered in scan
			if > 0, first character position past number decoded
IB	integer	O	integer value of number found prior to a blank or period
IC	integer	O	integer value of number found after period; zero if none found
IFN	integer	I	the last character position to be scanned for number
		O	if < 0, IFN is the character position of a character which could not be decoded into a number
			if > 0, contains number of characters found in decoding IC

NAME: GTWD

PURPOSE: To parse a character unit from a character string.

ENTRY: SUBROUTINE GTWD (IWD, NCARD, IS, IFN)

Description of Parameters

Name	Type	I/O	Usage
IWD	character	O	the character unit generated (maximum 20 characters on CDC and 16 characters on IBM)
NCARD	character	I	the character string to be parsed
IS	integer	I	the first character position for the parsing scan
		O	if IFN > 0, the first character position past the unit assembled
IFN	integer	I	last character of NCARD to be scanned
		O	if < 0, field contained no non-blank characters
			if > 0, number of characters in unit assembled

NAME: INTEG

PURPOSE: To convert an integer into a character string.

ENTRY: SUBROUTINE INTEG (NX, IFL, NCR, IFC)

Description of Parameters

Name	Type	I/O	Usage
NX	integer	I	number to be converted
IFL	character	O	character string to contain number; must be 24 characters long on IBM and 30 characters long on CDC
NCR	integer	O	number of characters generated for the number
IFC	integer	O	position of first character of string generated

NAME: STOR

PURPOSE: To transfer character strings.

ENTRY: SUBROUTINE STOR (M1, M2, I1, NCR, I2)

Description of Parameters

Name	Type	I/O	Usage
M1	character	I	character string into which the characters are to be transferred
M2	character	O	character string from which the characters are to be transferred
I1	integer	I	position in M2 where first character is to be placed when transferred
NCR	integer	I	number of characters to be transferred
I2	integer	I	position in M2 of first character to be transferred

SAIL LIBRARY ROUTINES

NAME: STORN

PURPOSE: To store three integers packed in a character word.

ENTRY: SUBROUTINE STORN (N, I1, I2, I3)

Description of Parameters

Name	Type	I/O	Usage
N	character word (integer CDC) (real*8 IBM)	O	word into which the integers are to be stored
I1	integer	I	integer to be stored in first quarter of word
I2	integer	I	integer to be stored in second quarter of word
I3	integer	I	integer to be stored in second half of word

HULL LIBRARY ROUTINES

NAME: BACKUP

PURPOSE: To backspace a record written by RDL.

ENTRY: SUBROUTINE BACKUP (LFN, N)

Description of Parameters

Name	Type	I/O	Usage
LFN	unit	I	unit to be backspaced
N	integer	I	number of RDC records to be backspaced

NAME: CDATE

PURPOSE: To return current date.

ENTRY: SUBROUTINE CDATE (DT)

Description of Parameters

Name	Type	I/O	Usage
DT	character	O	date is in format BDDMMMYY where B = blank DD = day of month MMM = month YY = year

NAME: CLOSEP

PURPOSE: To terminate partial I/O on a CDC file and return to FORTRAN I/O.

ENTRY: SUBROUTINE CLOSEP (LFN)

Description of Parameters

Name	Type	I/O	Usage
LFN	unit	I	the file to be returned to CDC FORTRAN I/O

NAME: DEVTP

PURPOSE: To determine if a file is a tape on CDC system.

ENTRY: SUBROUTINE DEVTP (FILE, IFLG)

Description of Parameters

File	Type	I/O	Usage
FILE	unit	I	file which is to be tested
IFLG	integer	O	device flag = 1 file is tape = 2 file is disk

NAME: ENDRCD

PURPOSE: To write a SCOPE end of record on a CDC coded file.

ENTRY: SUBROUTINE ENDRCD (IFILE)

Description of Parameters

Name	Type	I/O	Usage
IFILE	unit	I	file where end of record is to be written

NAME: FILETY

PURPOSE: To determine the characteristics of a CDC file from FORTRAN.

ENTRY: SUBROUTINE FILETY (LFN, DT, FP, TC, OF, RMS, PER, TYP)

Description of Parameters

Name	Type	I/O	Usage
LFN	unit	I	the file for which the information is required
DT	character	O	device hardware mnemonic
			AH - 819 disk drive
			AM - 841 multiple disk drive
			AX - ECS resident files
			AY - 705/844-21 disk drive
			AZ - 7054/844-41 disk drive
			LM - link medium file
			MT - 657 or 667 magnetic tape drive, seven track
			NT - 659, 669 or 679 magnetic tape drive, nine track
			TR - 3691 paper tape reader
			TP - 3691 paper tape punch
			LP - line printer (any)
			LQ - line printer (512)
			LR - line printer (580-12)
			LS - line printer (580-16)
			LT - line printer 580-20)
			CR - card reader (405)

NAME: FILETY (continued)

Name	Type	I/O	Usage
DT(continued)			KB - remote terminal keyboard CP - card punch DC - console display GC - 252-2 graph console
FP(optional)	integer	O	sequential file position = 1 beginning of information = 2 end of file = 3 end of information
TC(optional)	integer	O	magnetic tape characteristics = 1 unlabeled, seven track = 2 labeled, seven track = 3 unlabeled, nine track = 4 labeled, nine track
OF (optional)	integer	O	open flag; file is open if $\neq 0$
RMS (optional)	integer	O	mass storage flag; file is on mass storage if $\neq 0$
PER (optional)	integer	O	file permission = 0 no permission = 1 read permission = 2 write permission = 4 extend permission = 8 modify permission combination or sums of the above values
TYP(optional)	integer	O	file tape = 0 local file (scratch) = 1 INPUT = 2 print file = 3 punch file = 4 permanent file = 63 all others

NAME: GETVSN

PURPOSE: To obtain the VSN of a tape on CDC NOS/BE systems.

ENTRY: SUBROUTINE GETVSN (LFN, VSN)

Description of Parameters

Name	Type	I/O	Usage
LFN	unit	I	the local file associated with the tape
VSN	character	O	the VSN is the first five characters; reading zeros included

NAME: INBY

PURPOSE: *To insert bits in CDC word.*

ENTRY: FUNCTION INBY (M1, IB1, NB, IB2, M2)
ENBY (M1, IB1, NB, IB2, M2)

Description of Parameters

Name	Type	I/O	Usage
M1	any	I	word containing source bits
IB1	integer	I	bit position of first bit to be inserted from M1
NB	integer	I	number of bit to be inserted
IB2	integer	I	bit position in M2 where bits are to be inserted
M2	any	I	word where bits are to be inserted
INBY ENBY	integer real	O	the result of NR bits from IB1 and M1 inserted at IB2 in M2

NAME: MACHNT

PURPOSE: To determine on CDC NOS/BE system whether the program is running on a Cyber 176.

ENTRY: SUBROUTINE MACHNT (ITYP)

Description of Parameters

Name	Type	I/O	Usage
ITYPE	integer	O	Cyber 176 type = 0 if not a 176

NAME: PFMM

PURPOSE: To perform permanent file functions from FORTRAN.

ENTRIES: SUBROUTINE ATTACH (LFN, NR, PARM)
CATALOG (LFN, NR, PARM)
EXTEND (LFN, NR, PARM)
PURGE (LFN, NR, PARM)
ALTER (LFN, NR, PARM)
RENAME (LFN, NR, PARM) (not on 7600)
PERM (LFN, NR, PARM) (not on 7600)

Each entry performs the permanent file operation indicated by its name.

Description of Parameter

Name	Type	I/O	Usage
LFN	unit	I	local file name for permanent file operator
NR	integer	O	return code (see note 1)
PARM	any (array)	I	Parameter list. First four words contain permanent file name. Optional parameters follow two words per parameter; first is parameter name (see note 2) and second is value. List terminated by a zero.

NAME: PFMM(continued)

Note 1. Return code (for all entries except PERM).

NR	Description
ø	successful
1	PFN/FD error
2	LFN already in use
3	unknown LFN
4	cycle limit exceeded (more than five cycles)
5	permanent file catalog full
6	no LFN or PFN
8	latest index not written
9	file not on PF device
10	file not cataloged (not found)
11	archive retrieval aborted
13	cycle number limit reached (> 999)
14	permanent file directory full
15	function attempted on a nonpermanent file
16	function attempted on a nonlocal file
18	file never assigned to a device
19	cycle incomplete
20	PF already attached
21	file unavailable
23	illegal LFN
24	file dumped

NAME: PFMM (continued)

27	ALTER needs exclusive access
29	file already in system
32	illegal set name
33	device not mounted at control point
34	RBJ chain too large for PFC
35	file resides on unavailable device

Return code (for PERM entry)

NR	Description
ø	unknown file
1	read permission
2	extend permission
3	read and extend permission
4	modify permission
5	read and modify permission
6	extend and modify permission
7	read, extend and modify permission
8	control permission
9	read and control permission
10	extend and control permission
11	read, extend, and control permission
12	modify and control permission
13	read, modify and control permission
14	extend, modify and control permission
15	read, extend, modify and control permission
<0	nonpermanent file

NAME: PFMM (continued)

Note 2. The parameter names are

Name	Value	Description
IO	character	file ID
PW	character	submitted password
TK	character	defined turnkey password
CN	character	defined control password
MD	character	defined modify password
EX	character	defined extend password
RD	character	defined read password
XR	character	defined control, modify and extend password
AC	character	accounting code (not on CDC 7600)
EC	character	ECS buffering (not on CDC 7600)
FO	character	file organization
ST	character	MMF station
SN	character	set name
RP	integer	retention period (days)
CY	integer	cycle number
MR	integer	$\neq 0$ multiread access
LC	integer	$\neq 0$ lowest cycle
RW	integer	$\neq 0$ multiread single rewrite

NAME: NEXT

PURPOSE: To scan card image and generate next parsed element.

ENTRY: SUBROUTINE NEXT

Common Blocks Accessed

Name	Type	Common	Usage
CARD	character	/CARD/	current card image
TEST, TESTC	character	/CARD/	parsed character unit generated
IPOINT	integer	/CARD/	scan point at both beginning and end of scan
NCHAR	integer	/CARD/	number of characters in TEST, TESTC
NCOL	integer	/CARD/	number of characters of CARD to be scanned
EOFC	character	/EOF/	test set to this value if read produces an end of file

NAME: OPENP

PURPOSE: To open a file on SCOPE records for partial I/O. Must be called before using READP and WRITEP on CDC systems.

ENTRY: SUBROUTINE OPENP (LFN, MRL)

Description of Parameters

Name	Type	I/O	Usage
LFN	unit	I	the file to be opened for partial I/O
MRL	integer	I	the maximum length of record on file in words

NAME: PFNSET

PURPOSE: To set up PF parameter labels for HULL permanent files.

ENTRY: SUBROUTINE PFNSET (PROB)

Description of Parameters

Name	Type	I/O	Usage
PROB	real	I	problem number to be set in permanent file name

Common Blocks Accessed

Name	Type	Common	Usage
PROBPFN	character	/PARAM/	coded problem number
PARAM	any (array)	/PARAM/	PFN parameter array for dump tape permanent file
PARAM2	any (array)	/PARAM/	PFN parameter array for station tape permanent file
PARAM3	any (array)	/PARAM/	PFN parameter array for TAPE41 files
PARAM4	any (array)	/PARAM/	PFN parameter array for tape library

NAME: RDL

PURPOSE: To read a block of data and return the length read.

ENTRY: SUBROUTINE RDL (NT, DT, LN, IRTN)

Description of Parameters

Name	Type	I/O	Usage
NT	any	I	the file to be read
DT	any	O	the origin of the block which will contain the data read
LN	integer	I	the number of words in the block
		O	the number of words read
IRTN	integer	O	return code -1 successful read 0 end of file encountered +1 error in read

NAME: RDL2

PURPOSE: To read into level 2 variables on CDC 7600 and Cyber 176

ENTRY: SUBROUTINE RDL2 (PT, LN, IRTN)

Description of Parameters

The parameters are identical with those for RDL.

NAME: READP

PURPOSE: To read a partial record from a CDC file with SCOPE record tape.

ENTRY: SUBROUTINE READP (LFN, WS, LWS, LRD, IER)

Description of Parameters

Name	Type	I/O	Usage
LFN	unit	I	file to be read
WS	any	O	the area to contain the data read
LWS	integer	I	the number of words in the data area
LRD	integer	O	the number of words read
IER	integer	O	the read position = 0 mid record = 20B end of record

NAME: REMARK

PURPOSE: To generate a 40-character message on CDC system B display plus, optionally, in the job dayfile.

ENTRY: SUBROUTINE REMARK (MSG, FLAG)

Description of Parameters

Name	Type	I/O	Usage
MSG	character	I	message to be displayed; if less than 40 characters, it must be terminated by 0
FLAG (optional)	integer	I	display tape = 0 B display only ≠ 0 B display plus job dayfile (default if not present)

NAME: REQUEST

PURPOSE: To assign CDC files to devices

ENTRY: SUBROUTINE REQUEST (LFN, $PARAM_i$, ... $PARAM_n$)

Description of Parameter

Name	Type	I/O	Usage
LFN	unit	I	file to be assigned
$PARAM_i$	character	I	file characteristic (see note 1 and 2)

Note 1. The characters in $PARAM_i$ that are recognized as single word parameters are

Character Value of $PARAM_i$	Meaning
MT	select seven-track tape
NT	select nine-track tape
LO	select 200 BPI seven-track tape
HI	select 556 BPI seven-track tape
HY	select 800 BPI seven-track tape
HD	select 800 BPI nine-track tape
PE	select 1600 BPI nine-track tape
U	ANSI labeled tape
N	new labeled tape
E	existing labeled tape
Z	Z labeled tape (density is labeled
Y	Y labeled tape (3000 series labeled)
S	stranger tape
L	long stranger tape

NAME: REQUEST (continued)

Character Value of $PARAM_i$	Meaning
Q	assign to queue device
Q*	
IB	inhibit noise byte on tape
MN	any tape unit
A	any mass storage device
Ax	
EC	buffer through ECS
OV	overflow to any other mass storage device is allowed
PF	assign to permanent file device
EB	select EBCDIC conversion for nine-track coded tape
US	select ASC II conversion for nine-track coded tape
NS	nonstandard tape label
NR	inhibit normal parity array recovery
MF	multiple set
SV	same tape
IV	inhibit physical unload of tape
CK	check point tape
RING	request write ring on tape
NO RING	request no write ring on tape
VSN = xxxxxx	request a VSN of xxxxxx
SN = xxxxxx	request a set name of xxxxxx

NAME: REQUEST (continued)

Note 2. The multiword parameters are

First Word	Description
MSG	The following eight words will be used as the message instead of the standard message. If less than eight words, it must be terminated with a zero word.
L	The last eight characters of the first word and the first nine characters of second word represent the HDR1 label.
LABEL	The following four words contain the standard label block as described on page 7-37 of the NOS/BE reference manual or page 12-27 of SCOPE 3.4 reference manual.

NAME: RETURNS

PURPOSE: To return a CDC file.

ENTRY: SUBROUTINE RETURNS (LFN)

Description of Parameters

Name	Type	I/O	Usage
LFN	unit	I	file to be returned

NAME: ROUTEIT

PURPOSE: To route a file to the input queue on CDC systems.

ENTRY: SUBROUTINE ROUTEIT (FILE, STAT)

Description of Parameters

Name	Type	I/O	Usage
FILE	unit	I	file to be routed to input
STAT (optional)	character	I	station to receive file (default is current station if not specified)

NAME: SEARCH

PURPOSE: To locate a data record beginning with a specified keyword.

ENTRY: SUBROUTINE SEARCH (WORD, I, NREC)

Description of Parameters

Name	Type	I/O	Usage
WORD	character	I	keyword for search
I	integer	O	= 1 record found = 2 record not found
NREC	integer	O	record number of keyword

Input file is positioned so that calls to NEXT and value will process record beginning with the keyword.

NAME: SETREG

PURPOSE: To set the NOS/BE CLL registers R1, R2, and R3.

ENTRY: SUBROUTINE SETREG (NREG, NVAL)

Description of Parameters

Name	Type	I/O	Usage
NREG	integer	I	register number must be 1, 2, or 3
NVAL	integer	I	value for register; maximum $2^{18} - 1$

NAME: SINK

PURPOSE: To force an abnormal termination.

ENTRY: SUBROUTINE SINK

NAME: SWITCH

PURPOSE: To manipulate pseudo-sense switches on CDC system.

ENTRY: SUBROUTINE OFFSS (ISW)
OFFSW (ISW)
ONSS (ISW)
ONSW (ISW)

Description of Parameter

Name	Type	I/O	Usage
ISW	integer	I	sense switch number (must be in range 1-6)

NAME: TIMTGO

PURPOSE: To get the remaining job CPU time on CDC system.

ENTRY: SUBROUTINE TIMTGO (SEC)

Description of Parameters

Name	Type	I/O	Usage
SEC	integer	O	The amount of CPU time (sec) remaining until the current time limit is reached.

NAME: VALUE

PURPOSE: To decode a number or a logical constant.

ENTRY: SUBROUTINE VALUE (WS, ITYPE)

Description of Parameters

Name	Type	I/O	Usage
WS	real	O	number or logical value
ITYPE	integer	O	conversion type = 1 number WS set = 2 logical WS set = 3 character WS not set

Common Block Reference

Name	Type	Common	Usage
CARD	character	/CARD/	card image being scanned
TEST, TESTC	character	/CARD/	parsed character unit
IPOINT	integer	/CARD/	current character pointer
NCHAR	integer	/CARD/	number of characters in TEST, TESTC
NCOL	integer	/CARD/	number of characters to scan on card

NAME: WRE

PURPOSE: To wait for the completion of asynchronous output.

ENTRY: SUBROUTINE WRE (NU)

Description of Parameters

Name	Type	I/O	Usage
NU	unit	I	file for which the completion of the output operation is desired

NAME: WR

PURPOSE: To write a block of data.

ENTRY: SUBROUTINE WR (NO, BLOCK, LN, IRTN)

Description of Parameters

Name	Type	I/O	Usage
NO	unit	I	file to be written
BLOCK	any	I	origin of the block to be written
LN	integer	I	number of words to be written
IRTN	integer	O	return code = -1 successful write = 1 error in write

NAME: WRR

PURPOSE: To write a block to a file but return to CPU execution immediately on systems that have asynchronous I/O capability in FORTRAN.

ENTRY: SUBROUTINE WRR (NO, BLOCK, LN, IRTN)

Description of Parameters

The parameters are identical to those in WR.

NAME: WRR2

PURPOSE: To write a block from level 2 on CDC 7600 and Cyber 176 system with immediate return.

ENTRY: SUBROUTINE WRR2 (NO, BLOCK, LN, IRTN)

Description of Parameters

The parameters are identical to those in WR.

NAME: WR2

PURPOSE: To write a block for level 2 on CDC 7600 and Cyber 176 systems.

ENTRY: SUBROUTINE WR2 (NO, BLOCK, LN, IRTN)

Description of Parameters

The parameters are identical to those in WR.

NAME: WRITEP

PURPOSE: To write a partial record on CDC file with SCOPE type record.

ENTRY: SUBROUTINE WRITEP (LFN, WSA, LWT, IWT)

Description of Parameters

Name	Type	I/O	Usage
LFN	unit	I	file to be written
WSA	any	I	first word of area to be written
LWT	integer	I	number of words to be written
IWT	integer	I	type of writing = 0 continue ≠ 0 end of record

APPENDIX B

CDC CYBER CONTROL LANGUAGE (CCL) PROCEDURES

In NOS/BE, it is possible to use control card procedures in the system. This appendix describes the procedures used at the Air Force Weapons Laboratory (AFWL). Figure B-1 is the procedure used to copy a HULL system file from tape to disk. Figures B-2, B-3, and B-4 are procedures for generating the entire HULL library, regenerating PLANK and regenerating BOW, respectively. Figure B-5 is the procedure used to run HULL. Figure B-6 is a permanent-file retrieval procedure used by other procedures and exterior to the procedures. Figures B-7 and B-8 are the procedures for executing HULL, KEEL or PULL on the CDC 6600 and Cyber 176, respectively (used only by procedure HULLR). Finally figures B-9 and B-10 are procedures for executing the station plotting routines on the CDC 6600 and Cyber 176 (used only by procedure HULLR), respectively.

The procedure parameters are as listed.

CHANGES. The SAIL changes if file exists and INPUT current position is assumed, otherwise BOI is assumed (default CHANGES).

HLIB. The ID parameter for the library permanent file and the HULL permanent file (HCOPY) during generation or modification.

HPFN. The HULL file permanent file name is HCOPY. Default is MAST.

TAPE. The HULL tape VSN is HCOPY. Default is HULL.

FIL. The local file to which the requested permanent file is to be copied is GETC. Default is FIL.

PFN. This parameter defines the name of the permanent file which is to be copied to the local file in the procedure GETC.

PID. The ID of the permanent file to be copied by GETC.

DEN. HULL tape density (HCOPY). Default is HY.

ABTD. The abort parameter in HULLR. Default is COMMENT. If set to DMP the procedure will produce a dump on an abort of the execution of HULL.

TYPE. The meta plot system plotting media. Default is P8 and values may be as follows.

CF – Cal/COMP full size white paper

CG – Cal/COMP full size red ruled paper

CH – Cal/COMP half size white paper

CI – Ca/COMP half size red ruled paper

G4 – Gould 4800

B1 – B/W 16mm film (FR80)

B3 – B/W 35mm film (FR80)

K1 – color 16mm film (FR80)

K3 – color 35mm film (FR80)

MB – B/W 16mm movies (FR80)

MK – color 16mm movies (FR80)

F8 – 48X microfiche (FR80)

```
.PROC,HCOPY,TAPE=HULL,DEN=HY,HPFN=MAST,HID=DYMXCER.
REQUEST(OLD,DEN,VSN=TAPE,E) .
DYMAST(I=LOCAL)
CATALOG(NEW,HPFN,ID=HID,PW=HULLP,XR=HULLP,RP=999)
RETURN(OLD,NEW,LOCAL)
.DATA,LOCAL
SAIL COPY
OPTIONS
ENDOPTIONS
```

Figure B-1. HULL Tape Copy Procedure

```
.PROC,GENH,CHANGES,HLIB=HULLIB,HID=DYMXCER.
COPYCR(LOCAL,HHH)
RETURN(LOCAL)
IFE,FILE(CHANGES,IN),CPR.
COPYCR(CHANGES,HHH)
ENDIF,CPR.
IFE,FILE(CHANGES,AS.AND..NOT.IN),CPY.
REWIND(CHANGES)
COPYCF(CHANGES,HHH)
BKSP(HHH)
ENDIF,CPY.
DYMAST(I=HHH)
RETURN(HHH)
FTN(I=SAIL,B=BOW,PL=1000000,OPT=2)
ROUTE,OUTPUT,DC=PR,FID=*.
FTN(I=SAIL,B=PLANK,PL=1000000,OPT=2)
ROUTE,OUTPUT,DC=PR,FID=*.
FTN(A,I=SAIL,B=LIB,S,S=PFMTEXT,S=IPTEXT,OPT=2)
ROUTE,OUTPUT,DC=PR,FID=*.
REQUEST(NEWLIB,*PF)
EDITLIB(I=LOCAL1)
RETURN,LOCAL1.
LIBRARY(NEWLIB)
FILE(TAPE10,SBF=NO)
FILE(TAPE11,SBF=NO)
FILE(TAPE41,SBF=NO)
FILE(TAPE42,SBF=NO)
LDSET(PRESETA=NGINDEF,MAP=SBEX)
LDSET(FILES=TAPE10/TAPE11/TAPE41/TAPE42)
LOAD(BOW)
NOGO(BOWF,BOW,STERN)
RETURN(BOW)
FILE(TAPE41,SBF=NO)
FILE(TAPE4,SBF=NO)
LDSET(PRESETA=NGINDEF,MAP=SBEX,FILES=TAPE41/TAPE4)
LOAD(PLANK)
NOGO(PLANKF,PLANK)
RETURN,PLANK.
ROUTE,OUTPUT,DC=PR,FID=*.
FTN(A,I=SAIL,B=STRIP,OPT=2,L=SAVE)
RETURN(SAVE)
COPYCR(SAIL,CCLP)
LDSET(MAP=N)
STRIP.
RETURN,CCLP.
FTN,I=SAIL,B=ACCOUN,OPT=2.
FTN,I=SAIL,B=DUMPA,OPT=2.
LDSET,PRESETA=NGINDEF,MAP=SBEX.
LOAD,ACCOUN.
NOGO,ACCOUNF,ACCOUN.
LDSET,PRESETA=NGINDEF,MAP=SBEX.
LOAD,DUMPA.
NOGO,DUMPAF,DUMPA,SUMA.
RETURN,SUMA,ACCOUN,DUMPA.
ROUTE,OUTPUT,DC=PR,FID=*.
```

Figure B-2. HULL Library Generation Procedure

```
FTN,I=SAIL,B=MONITOR,OPT=2,PL=1000000.
FTN,I=SAIL,B=CNT,OPT=2.
LDSET,PRESETA=NGINDEF,MAP=SBEX.
LOAD,MONITOR.
NOGO,MON,MONITOR.
LDSET,PRESETA=NGINDEF,MAP=SBEX.
LOAD,CNT.
NOGO,CONTRF,CONTR.
RETURN,MONITOR,CNT.
ROUTE,OUTPUT,DC=PR,FID=*.
RETURN(SAIL)
LIBRARY.
EDITLIB(I=LOCAL2)
CATALOG(NEWLIB,HLIB,ID=HID,PW=HULLP,XR=HULLP,RP=999)
RETURN,BOWF,PLANKF,OLD,PLANK,LIB,PROC,BOW.
RETURN,MON,CONTRF.
RETURN,ACCOUNF,SUMAF,DUMPAF.
LIBRARY,NEWLIB.
ACCOUN.
LIBRARY.
RETURN,NEWLIB.
REVERT.
EXIT,S.
RETURN,ACCOUN.
LIBRARY,HULLIB.
ACCOUN.
REVERT,ABORT.
.DATA,LOCAL
SAIL LINENO PROGRAM BOW PLANK LIBRARY
  ACCOUNT
  CONTROL
  ENDPROGRAM
.DATA,LOCAL1
LIBRARY(NEWLIB,NEW)
ADD(SAILM,HULLIB,AL=1,LIB)
REWIND(LIB)
ADD(*,LIB)
FINISH.
ENDRUN.
.DATA,LOCAL2
LIBRARY(NEWLIB,OLD)
REWIND(PROC)
ADD(*,PROC,AL=1)
REWIND(BOWF)
ADD(*,BOWF,AL=1)
REWIND(PLANKF)
ADD(*,PLANKF,AL=1)
REWIND(ACCOUNF)
ADD(*,ACCOUNF,AL=1)
REWIND(DUMPAF)
ADD(*,DUMPAF,AL=1)
REWIND(MON)
ADD(*,MON,AL=1)
REWIND(CONTRF)
ADD(*,CONTRF,AL=1)
```

Figure B-2. (Continued)

```
.PROC,PLANKG,CHANGES,HIB=HULLIB,HID=DYMXCER.
LIBRARY.
RETURN(HULLIB)
ATTACH(HULLIB,HLIB,ID=HID,PW=HULLP)
LIBRARY(HULLIB)
COPYCR(LOCAL,HHH)
RETURN(LOCAL)
IFE,FILE(CHANGES,IN),CPR.
COPYCR(CHANGES,HHH)
ENDIF,CPR.
IFE,FILE(CHANGES,AS.AND..NOT.IN),CPY.
REWIND(CHANGES)
COPYCF(CHANGES,HHH)
BKSP(HHH)
ENDIF,CPY.
GETC,HHH.
DYMAST(I=HHH)
RETURN(HHH)
FTN(I=SAIL,B=PLANK,PL=1000000,OPT=2)
FILE(TAPE41,SBF=NO)
FILE(TAPE4,SBF=NO)
LDSET(PRESETA=NGINDEF,MAP=SBEX,FILES=TAPE41/TAPE4)
LOAD(PLANK)
NOGO(PLANKF,PLANK)
RETURN(SAIL,PLANK)
LIBRARY.
EDITLIB(I=LOCAL1)
RETURN(LOCAL1)
EXTEND(HULLIB)
RETURN,OLD,PLANK,PLANKF.
LIBRARY,HULLIB.
ACCOUN.
REVERT.
EXIT,S.
LIBRARY,HULLIB.
ACCOUN.
REVERT,ABORT.
.DATA,LOCAL
SAIL PROGRAM PLANK ENDPROGRAM
.DATA,LOCAL1
LIBRARY(HULLIB,OLD)
REWIND(PLANKF)
REPLACE(*,PLANKF,AL=1)
FINISH.
ENDRUN.
```

Figure B-3. PLANK Regeneration Procedure

```
.PROC,BOWG,CHANGES,HLIB=HULLIB,HID=DYMXCER.
LIBRARY.
RETURN(HULLIB)
ATTACH(HULLIB,HLIB,ID=HID,PW=HULLP)
LIBRARY(HULLIB)
COPYCR(LOCAL,HHH)
RETURN,LOCAL.
IFE,FILE(CHANGES,IN),CPR.
COPYCR(CHANGES,HHH)
ENDIF,CPR.
IFE,FILE(CHANGES,AS.AND..NOT.IN),CPY.
REWIND(CHANGES)
COPYCF(CHANGES,HHH)
BKSP(HHH)
ENDIF,CPY.
GETC,HHH.
DYMAST(I=HHH)
RETURN(HHH)
FTN(I=SAIL,B=BOW,PL=1000000,OPT=2)
RETURN(SAIL)
FILE(TAPE10,SBF=NO)
FILE(TAPE11,SBF=NO)
FILE(TAPE41,SBF=NO)
FILE(TAPE42,SBF=NO)
LDSET(PRESETA=NGINDEF,MAP=SBEX)
LDSET(FILES=TAPE10/TAPE11/TAPE41/TAPE42)
LOAD(BOW)
NOGO(BOWF,BOW,STERN)
RETURN(BOW)
LIBRARY.
EDITLIB(I=LOCAL1)
RETURN,LOCAL1.
EXTEND(HULLIB)
RETURN,BOWF.
LIBRARY,HULLIB.
ACCOUN.
REVERT.
EXIT,S.
LIBRARY,HULLIB.
ACCOUN.
REVERT,ABORT.
.DATA,LOCAL
SAIL PROGRAM BOW ENDPROGRAM
.DATA,LOCAL1
LIBRARY(HULLIB,OLD)
REWIND(BOWF)
ADD(*,BOWF,AL=1)
FINISH.
ENDRUN.
```

Figure B-4. BOW Regeneration Procedure

```
.PROC,HULLR,CHANGES,ABTD=COMMENT,TYPE=F8.
COPYCR(LOCAL,HHH)
RETURN(LOCAL)
IFE,FILE(CHANGES,IN),CPR.
COPYCR(CHANGES,HHH)
ENDIF,CPR.
IFE,FILE(CHANGES,AS.AND..NOT.IN),CPY.
REWIND(CHANGES)
COPYCF(CHANGES,HHH)
BKSP(HHH)
ENDIF,CPY.
GETC,HHH.
CONTR,BOW.
FILE(TAPE10,SBF=NO)
FILE(TAPE11,SBF=NO)
FILE(TAPE41,SBF=NO)
FILE(TAPE42,SBF=NO)
LDSET(FILES=TAPE10/TAPE11/TAPE41/TAPE42)
BOW.
CONTR,PLANK.
FILE(TAPE4,SBF=NO)
FILE(TAPE41,SBF=NO)
LDSET(FILES=TAPE4/TAPE41)
PLANK.
CONTR,SAIL.
DYMAST(I=HHH)
RETURN(HHH)
RETURN(SAVE)
REQUEST(SAVE,*Q)
CONTR,COMPILE.
IFE,R2.EQ.2,COMPS.
FTN(A,I=SAIL,B=STASORT,PL=100000,OPT=2,L=SAVE)
FTN(A,I=SAIL,B=STAPLOT,PL=100000,OPT=2,L=SAVE)
ELSE,COMPS.
FTN(A,I=SAIL,PL=1000000,B=HULL,OPT=2,L=SAVE)
ENDIF,COMPS.
RETURN(SAIL,LOCAL)
IFE,R2.NE.0,MET.
LIBRARY(HULLIB,METALIB)
RETURN(TAPE16)
REQUEST(TAPE16,*Q)
DISPOSE(TAPE16,*MF=P+TYPE)
ENDIF,MET.
RETURN(MAP)
REQUEST(MAP,*Q)
CONTR,LOAD.
IFE,R2.EQ.2,RUNS.
IFE,R1.NE.0,R176.
EXST176,ABTD.
ELSE,R176.
EXSTAP,ABTD.
ENDIF,R176.
ELSE,RUNS.
IFE,R1.NE.0,H176.
HULE176,ABTD.
```

Figure B-5. HULL Execution Procedure

```
ELSE,H176.
HULLEX,ABTD.
ENDIF,H176.
ENDIF,RUNS.
ACCOUN.
REVERT.
EXIT,S.
DISPOSE(SAVE,PR)
DISPOSE(MAP,PR)
CONTR,ABORT.
ACCOUN.
REVERT,ABORT.
.DATA,LOCAL
SAIL LINENO
```

Figure B-5. HULL Execution Procedure (Cont'd)

```
.PROC,GETC,FIL,PFN=CH1,PID=DYMXCER.
ATTACH(XXX,PFN,ID=PID)
COPYCR(XXX,FIL)
RETURN(XXX)
REVERT.
EXIT.
RETURN(XXX)
REVERT.
EXIT,S.
RETURN(XXX)
REVERT.
```

Figure B-6. Permanent Copy Procedure

```
.PROC,HULLEX,ABTD=COMMENT.
LDSET(PRESETA=NGINDEF,MAP=SBEX/MAP)
HULL.
RETURN(TAPE4,HULL)
REVERT.
EXIT.
ABTD,377000.
RETURN(TAPE4,HULL)
REVERT,ABORT.
```

Figure B-7. HULL Execution Procedure for CDC 6600 (for use by HYLLR Only).

```
.PROC,HULE176,ABTD=COMMENT.
FILE(TAPE4,SBF=NO)
FILE(TAPE41,SBF=NO)
FILE(TAPE9,SBF=NO)
FILE(TAPE40,SBF=NO)
FILE(TAPE10,SBF=NO)
FILE(TAPE44,SBF=NO)
FILE(TAPE45,SBF=NO)
LDSET(PRESETA=NGINDEF,MAP=SBEX/MAP)
LDSET(FILES=TAPE4/TAPE41/TAPE9/TAPE40/TAPE10)
LDSET(FILES=TAPE44/TAPE45)
HULL.
RETURN(TAPE4,HULL)
REVERT.
EXIT.
ABTD,377000.
RETURN(HULL,TAPE4)
REVERT,ABORT.
```

Figure B-8. HULL Execution Procedure for Cyber 176 (used by HULLR only)

```
.PROC,EXSTAP,ABTD=COMMENT.
LDSET(PRESETA=NGINDEF,MAP=SBEX/MAP)
STASORT.
LDSET(PRESETA=NGINDEF,MAP=SBEX/MAP)
STAPLOT.
RETURN(STAPLOT,STASORT,TAPE4,TAPE9)
REVERT.
EXIT.
ABTD,377000.
RETURN(STAPLOT,STASORT,TAPE4,TAPE9)
REVERT,ABORT.
```

Figure B-9. Station Plotting Procedure for CDC 6600 (used by HULLR only)

```
.PROC,EXST176,ABTD=COMMENT.
FILE(TAPE4,SBF=NO)
FILE(TAPE9,SBF=NO)
FILE(TAPE62,SBF=NO)
FILE(TAPE63,SBF=NO)
FILE(TAPE64,SBF=NO)
FILE(TAPE65,SBF=NO)
FILE(TAPE71,SBF=NO)
FILE(TAPE72,SBF=NO)
FILE(TAPE73,SBF=NO)
FILE(TAPE74,SBF=NO)
FILE(TAPE41,SBF=NO)
LDSET(PRESETA=NGINDEF,MAP=SBEX/MAP)
LDSET(FILES=TAPE4/TAPE9/TAPE61/TAPE62/TAPE63/TAPE64/TAPE65)
LDSET(FILES=TAPE71/TAPE72/TAPE73/TAPE74/TAPE41)
STASORT.
FILE(TAPE9,SBF=NO)
LDSET(PRESETA=NGINDEF,MAP=SBEX/MAP,FILES=TAPE9)
STAPLOT.
RETURN,STASORT,STAPLOT,TAPE4,TAPE9.
REVERT.
EXIT.
ABTD,377000.
RETURN,STASORT,STAPLOT,TAPE4,TAPE9.
REVERT,ABORT.
```

Figure B-10. Station Plotting Procedure for Cyber 176
(used by HULLR only)

APPENDIX C

IBM PROCEDURES

The control card procedures for generating the library of SAIL routines are shown in figure C-1. The procedures for generation of the HULL library routines and PLANK are shown in figures C-2 (HLIB) and C-3 (PGEN), respectively. Finally, the procedures for executing KEEL, HULL and PULL are shown in C-4 (KEEL), C-5 (HULL), and C-6 (PULL), respectively. The procedure parameters are listed below.

LIB. Primary library name. The proportional data set name for the library is PRELIB/LIB. Default is HULLIB.

LIBPRE. The data set prefix for library. Default is SAIL.

LIBDS. The depositor for the library data sets an all linkage-editor step which adds routines to the library. Default is (OLD, KEEP).

LIBU. The UNIT field for the library data set. Default is null, assuming a cataloged data set.

LIBVOL. The VOLUME field for the library data set. The default is null.

APROG. The assembler program name for procedures SLIB and HLIB. Default is IFOX00.

AREG. The REGION size for the assembly step in procedures SLIB and HLIB. Default is 187K.

ATIME. The time limit for the assembly step in procedure SLIB and HLIB. Default is (0,15).

AMACL. The data set name for the system assembler macrolibrary in procedures SLIB and HLIB. Default is SYS1.MACLIB.

AP1. This defines the printed output from the assembly step in procedures SLIB and HLIB. Default is SYSOUT = A.

APARM. This defines the parameters for the assembly step in procedures SLIB and HLIB. Default is LOAD,NODECK.

CHNBLK. This is the block size for the SAIL change file. Default is 3521.

CHNLRL. This is the record length for the SAIL change file. Default is 3517.

CREG. This is the REGION size for the copy step in procedures KEEL, HULL and PULL. Default is 100K. (Note the copy use IEBGENER).

FILO. This is the file number for the OLD SAIL file. Default is null.

FLIB. The FORTRAN library data set name in procedures PGEN, KEEL, HULL and PULL. Default is SYS1.FORTLIB.

FPARM. This defines the parameters for all FORTRAN compiler steps. The default is MAP.

FPROG. This is the name of the FORTRAN compiler. Default is IFEAAB.

FREG. This is the REGION size for all compiled steps in the procedures. Default is 512K.

FP1. This defines the printed output from all compiler steps. Default is SYSOUT = A.

FSPACE. This defines the SYSLIN space for all compiler steps. Default is (TRK, (10,5),RLSE).

FTIME. This is the time limit for all compiler steps. Default is (1,0), except in HLIB where it is (0,5).

GENO. This defines the version of the old SAIL file. The data set name for the old SAIL file is OLDPRE/OLD/GENO. Note. The old SAIL file for SLIB is the one containing SAIL while it is the one containing HULL for all other procedures. Default is (0).

LABO. The table type for the old SAIL file data set. Default is null.

LNAME. This defines member name for the main program in the procedures KEEL, HULL and PULL. Default is the procedure name.

LPARM. This defines the parameters for all linkage editor steps. Default is MAP. Note in procedures SLIB and HLIB, the parameter NCAL is always included.

LPROG. This is the name of the linkage editor. Default is IEWL.

LREG. This is the REGION size for all linkage editor steps. Default is 250K.

LP1. This defines the printed output from all linkage editor steps. Default is SYSOUT = A.

LTIME. This defines the time limit for all linkage editor steps. Default is (0,45), except in HLIB where it is (0,10).

OLD. This is the primary old SAIL file name (see GENO). Default is SAIL for SLIB and HULL for all other procedures.

OLDDCB. This defines the DCB parameters of the old SAIL file. Default is null.

OLDDS. This is the deposition of the old SAIL file. Default is SHR.

OLDPRE. The prefix for the old SAIL file (see GENO). Default is SAIL.

OLDU. The UNIT field for the old SAIL file. Default is null.

OLDVOL. The VOLUME field for the old SAIL file. Default is null.

PRROG. The member name in the HULL library for program PLANK. Default is PLANK in KEEL, HULL and PULL.

PNAME. The member where the program PLANK is to be stored in PGEN. Default is PLANK.

PLENSP. This is the space for the source output from SAIL in the procedure PGEN. Default is (CYL,(15,5),RLSE).

PP1. This defines the printed output from PLANK in procedures KEEL, HULL and PULL.

PLREG. This is the REGION size for the execution step of procedure PULL. Default is 512K.

PLTIME. This is the time limit for the execution step of procedure PULL. Default is 3.

PRCN. This defines the number of records in the random SAIL procedure file. The default is 5000, but should be set to the value of NUMREC used to generate SAIL.

PRCL. This defines the length of the random SAIL records. The default is 3644 and should be set to 8*LBUFF, where LBUFF is the LBUFF value used to generate SAIL.

PS1. This defines the primary printed output from SAIL. The default is SYSOUT = A.

PS2. This defines the error output from SAIL. The default is SYSOUT = A.

PULLSP. This defines the space allocated for the source output data set for the SAIL step of PULL. The default is (CYL,(10,5),RLSE).

SAILB2K. This defines the block size for the source data sets which are generated by SAIL in all procedures. The default is 800.

SAILR. This defines the record length of the source data sets which are generated by SAIL. The default is 80.

SCRTC. The UNIT field which defines the scratch disk. Default is SYSDA.

SREG. The REGION size for the SAIL execution step. Default is 175K.

STIME. The time limit for the SAIL step. The default is (2,0).

WORKSP. The SPACE definition of all scratch data sets used by all the procedures. The default is (CYL,(5,5)).

Step names used in the procedures are listed.

SAIL. The SAIL execution step.

HULL. The copy step in procedure HULL.

KEEL. The copy step in the procedure KEEL.

PULL. The copy step in the procedure PULL.

ASN. The assembler step in procedures SLIB and HLIB.

FORT. The FORTRAN compiler step in procedures PGEN, KEEL, HULL and PULL.

LKED. The linkage editor step in procedures PGEN, KEEL, HULL and PULL.

LKEDA. The linkage editor step for the assembly language routines in procedures SLIB and HLIB.

$FORT_n$. The FORTRAN compiler steps for procedures SLIB ($1 \geq n \geq 6$) and HLIB ($1 \geq n \geq 9$).

$LKED_n$. The linkage editor step for the routine compiled in the $FORT_n$ step of procedures SLIB and HLIB.

PLANK. The step in procedures KEEL, HULL and PULL, where PLANK is executed.

GO. The step in procedures KEEL, HULL and PULL, where the main routine is executed.

The required DD cards in the procedures are listed.

SAIL, INPUT. The SAIL directives and changes for all procedures. It may be dummied in procedures KEEL, HULL, and PULL if there are no changes. When the procedures SLIB, HLID, and PGEN are envoked, the proper programs from the system must be selected by SAIL directives here. In SLIB the required program is LIB; in HLIB, it is LIBRARY; and in PGEN, it is PLANK.

KEEL.INPUT. This data set contains the KEEL input as described in section V used in the procedure KEEL.

KEEL.DATA. This is the data set which will contain the HULL dump tape described in section IV. This DD card is used in procedure KEEL.

KEEL.STATION. This DD card defines the station tape file procedure in KEEL as described in section IV. It may be dummied if there are no stations in the problem.

HULL.INPUT. This DD card defines the HULL input as described in section VI as used in procedure HULL.

HULL.DATA. This DD card defines the HULL dump tape data set for the problem to be run as used in procedure HULL.

HULL.STATION. This DD card defines the station tape data set for the problem as used in procedure HULL. It may be dummied if there are no stations.

PULL.INPUT. This DD card defines the PULL input as described in section VII as nanoseconds.

PULL.DATA. This DD card defines the HULL dump tape data set for the problem to be plotted in procedure PULL.

```
//* * * * * * * * * * * * * * S L I B * * * * * * * * * * * * * * * * * *
//*
//* NAME
//*   SLIB
//*
//* FORMAT
//*   //         EXEC    SLIB,...(OPTIONAL PARAMETERS AS REQUIRED)
//*   //SAIL.INPUT  DD  *
//*
//*       SAIL CHANGES
//*
//*   /*
//*
//* FUNCTION
//*   THIS PROCEDURE EXECUTES THE SAIL PROGRAM AND GENERATES A NEW
//*   SAIL LIBRARY
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//SLIB    PROC LIB=HULLIB,                                              X
//             LIBDSP='(OLD,KEEP)',                                     X
//             LIBPRE='SAIL.',                                          X
//             LIBU=,                                                   X
//             LIBVOL=,                                                 X
//             APROG=IFOX00,                                            X
//             AREG=187K,                                               X
//             ATIME='(0,15)',                                          X
//             APARM='LOAD,NODECK',                                     X
//             AMACL='SYS1.MACLIB',                                     X
//             AP1='SYSOUT=A',                                          X
//             CHNBLK=3521,                                             X
//             CHNLRL=3517,                                             X
//             FILO=,                                                   X
//             FPARM=MAP,                                               X
//             FPROG=IFEAAB,                                            X
//             FREG=512K,                                               X
//             FP1='SYSOUT=A',                                          X
//             FSPACE='(TRK,(10,5),RLSE)',                              X
//             FTIME='(1,0)',                                           X
//             GENO='(0)',                                              X
//             LABO=,                                                   X
//             LPARM='MAP',                                             X
//             LPROG=IEWL,                                              X
//             LREG=250K,                                               X
//             LP1='SYSOUT=A',                                          X
//             LTIME='(0,45)',                                          X
//             SLIBSP='(TRK,(5,5),RLSE)',                               X
//             OLD=SAIL,                                                X
//             OLDDCB=,                                                 X
//             OLDDS=SHR,                                               X
//             OLDPRE='SAIL.',                                          X
//             OLDU=,                                                   X
//             OLDVOL=,                                                 X
//             PRCN=5000,                                               X
//             PRCL=3440,                                               X
//             PS1='SYSOUT=A',                                          X
```

Figure C-1. SLIB. Procedure to Add SAIL Routine to Library

```
//             PS2='SYSOUT=A',                                          X
//             SAILBLK=800,                                             X
//             SAILR=80,                                                X
//             SCRTC=SYSDA,                                             X
//             SPROG=SAIL,                                              X
//             SREG=175K,                                               X
//             STIME='(2,0)',                                           X
//             WORKSP='(CYL,(5,5))'
//*-------------------------------------------------------------------*
//SAIL    EXEC PGM=&SPROG,TIME=&STIME,REGION=&SREG
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=SHR
//*
//FT01F001  DD DUMMY
//*
//FT02F001  DD DSN=&OLDPRE&OLD&GENO,                                   X
//             UNIT=&OLDU,                                             X
//             LABEL=(&FILO,&LABO,,IN),                                X
//             DISP=&OLDDS,                                            X
//             VOL=&OLDVOL,                                            X
//             DCB=&OLDDCB
//*
//FT03F001  DD UNIT=&SCRTC,                                            X
//             DISP=(NEW,DELETE),                                      X
//             DCB=(RECFM=VBS,LRECL=&CHNLRL,BLKSIZE=&CHNBLK),          X
//             SPACE=(TRK,(20,20))
//*
//FT04F001  DD &PS2,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT05F001  DD DDNAME=INPUT
//*
//FT06F001  DD &PS1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT08F001  DD DSN=&&COMPR,                                            X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&SLIBSP,                                          X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F002  DD DSN=&&GETNUM,                                           X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&SLIBSP,                                          X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F003  DD DSN=&&GTWD,                                             X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&SLIBSP,                                          X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
```

Figure C-1. (Continued)

```
//*
//FT08F004  DD DSN=&&INTEG,                                          X
//             DISP=(NEW,PASS),                                      X
//             UNIT=&SCRTC,                                          X
//             SPACE=&SLIBSP,                                        X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F005  DD DSN=&&STOR,                                           X
//             DISP=(NEW,PASS),                                      X
//             UNIT=&SCRTC,                                          X
//             SPACE=&SLIBSP,                                        X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//FT08F006  DD DSN=&&STORN,                                          X
//             DISP=(NEW,PASS),                                      X
//             UNIT=&SCRTC,                                          X
//             SPACE=&SLIBSP,                                        X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F007  DD DSN=&&ASSEM,                                          X
//             DISP=(NEW,PASS),                                      X
//             UNIT=&SCRTC,                                          X
//             SPACE=&SLIBSP,                                        X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F008  DD DSN=&&LINK1,                                          X
//             DISP=(NEW,PASS),                                      X
//             UNIT=&SCRTC,                                          X
//             SPACE=&SLIBSP,                                        X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F009  DD DSN=&&LINK2,                                          X
//             DISP=(NEW,PASS),                                      X
//             UNIT=&SCRTC,                                          X
//             SPACE=&SLIBSP,                                        X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT09F001  DD DUMMY
//*
//FT10F001  DD UNIT=&SCRTC,                                          X
//             DISP=(NEW,DELETE),                                    X
//             SPACE=(&PRCL,(&PRCN))
//*
//FT11F001  DD DUMMY
//*
//FT12F001  DD DUMMY
//*-------------------------------------------------------------------*
//ASM     EXEC PGM=&APROG,PARM='&APARM',REGION=&AREG,                X
//             TIME=&ATIME,                                          X
//             COND=(8,LT,SAIL)
//*
//SYSLIB    DD DSN=&AMACL,DISP=SHR
//*
//SYSIN     DD DSN=&&ASSEM,DISP=(OLD,DELETE)
//*
//SYSPRINT  DD &AP1,
```

Figure C-1 (Continued)

```
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT3    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSPUNCH  DD SYSOUT=B
//*
//SYSGO     DD  DDNAME=SYSLIN
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             UNIT=&SCRTC,                                            X
//             DISP=(MOD,PASS),                                        X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*-------------------------------------------------------------------*
//LKEDA   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,ASM),(8,LT,SAIL)),                          X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LINK1,DISP=(OLD,DELETE)
//SAILD     DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT1   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&COMPR,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
```

Figure C-1 (Continued)

```
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED1   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT1),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(COMPR),                                 X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT2   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&GETNUM,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED2   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT2),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(GETNUM),                                X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
```

Figure C-1 (Continued)

```
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT3   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//S.SIN     DD DSN=&&GTWD,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED3   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT3),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(GTWD),                                  X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT4   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&INTEG,DISP=(OLD,DELETE)
//*
```

Figure C-1 (Continued)

```
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED4   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT4),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(INTEG),                                 X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT5   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&STOR,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED5   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT5),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
```

Figure C-1 (Continued)

```
//SYSLMOD   DD DSN=&LIBPRE&LIB(STOR),                                    X
//             UNIT=&LIBU,                                               X
//             VOL=&LIBVOL,                                              X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT6   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,       X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                           X
//             DISP=(MOD,PASS),                                         X
//             UNIT=&SCRTC,                                             X
//             SPACE=&FSPACE,                                           X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&STORN,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED6   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                     X
//             COND=((4,LT,FORT6),(8,LT,SAIL)),                         X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB,                                         X
//             UNIT=&LIBU,                                              X
//             VOL=&LIBVOL,                                             X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD DSN=&&LINK2,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//       PEND
```

Figure C-1 (Continued)

```
//* * * * * * * * * * * * * * H L I B * * * * * * * * * * * * * * * * * * *
//*
//* NAME
//*   HLIB
//*
//* FORMAT
//*   //         EXEC    HLIB,...(OPTIONAL PARAMETERS AS REQUIRED)
//*   //SAIL.INPUT  DD  *
//*
//*       SAIL CHANGES
//*
//*   /*
//*
//* FUNCTION
//*   THIS PROCEDURE EXECUTES THE SAIL PROGRAM AND GENERATES A NEW
//*   SAIL LIBRARY
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//HLIB    PROC LIB=HULLIB,                                                X
//             LIBDSP='(OLD,KEEP)',
//             LIBPRE='SAIL.',                                            X
//             LIBU=,                                                     X
//             LIBVOL=,                                                   X
//             APROG=IFOX00,                                              X
//             AREG=187K,                                                 X
//             ATIME='(0,10)',                                            X
//             APARM='LOAD,NODECK',                                       X
//             AMACL='SYS1.MACLIB',                                       X
//             AP1='SYSOUT=A',                                            X
//             CHNBLK=3521,                                               X
//             CHNLRL=3517,                                               X
//             FILO=,                                                     X
//             FPARM=MAP,                                                 X
//             FPROG=IFEAAB,                                              X
//             FREG=512K,                                                 X
//             FP1='SYSOUT=A',                                            X
//             FSPACE='(TRK,(10,5),RLSE)',                                X
//             FTIME='(0,5)',                                             X
//             GENO='(0)',                                                X
//             LABO=,                                                     X
//             LPARM='MAP',                                               X
//             LPROG=IEWL,                                                X
//             LREG=250K,                                                 X
//             LREG=250K,                                                 X
//             LP1='SYSOUT=A',                                            X
//             LTIME='(0,10)',                                            X
//             HLIBSP='(TRK,(5,5),RLSE)',                                 X
//             OLD=SAIL,                                                  X
//             OLDDCB=,                                                   X
//             OLDDS=SHR,                                                 X
//             OLDPRE='SAIL.',                                            X
//             OLDU=,                                                     X
//             OLDVOL=,                                                   X
//             PRCN=5000,                                                 X
//             PRCL=3440,                                                 X
```

Figure C-2. HLIB. Procedure to Add HULL Routines to Library

```
//             PS1='SYSOUT=A',                                          X
//             PS2='SYSOUT=A',                                          X
//             SAILBLK=800,                                             X
//             SAILR=80,                                                X
//             SCRTC=SYSDA,                                             X
//             SPROG=SAIL,                                              X
//             SREG=256K,                                               X
//             STIME='(2,0)',                                           X
//             WORKSP='(CYL,(5,5))'
//*-------------------------------------------------------------------*
//SAIL    EXEC PGM=&SPROG,TIME=&STIME,REGION=&SREG
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=SHR
//*
//FT01F001  DD DUMMY
//*
//FT02F001  DD DSN=&OLDPRE&OLD&GENO,                                   X
//             UNIT=&OLDU,                                             X
//             LABEL=(&FILO,&LABO,,IN),                                X
//             DISP=&OLDDS,                                            X
//             VOL=&OLDVOL,                                            X
//             DCB=&OLDDCB
//*
//FT03F001  DD UNIT=&SCRTC,                                            X
//             DISP=(NEW,DELETE),                                      X
//             DCB=(RECFM=VBS,LRECL=&CHNLRL,BLKSIZE=&CHNBLK),          X
//             SPACE=(TRK,(20,20))
//*
//FT04F001  DD &PS2,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT05F001  DD DDNAME=INPUT
//*
//FT06F001  DD &PS1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT08F001  DD DSN=&&BACKUP,                                           X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&HLIBSP,                                          X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F002  DD DSN=&&VALUE,                                            X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&HLIBSP,                                          X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F003  DD DSN=&&CDATE,                                            X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&HLIBSP,                                          X
```

Figure C-2. (Continued)

```
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F004  DD DSN=&&NEXT,                                          X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F005  DD DSN=&&SEARCH,                                        X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//FT08F006  DD DSN=&&SINK,                                          X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F007  DD DSN=&&RDL,                                           X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F008  DD DSN=&&WR,                                            X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F009  DD DSN=&&SECOND,                                        X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F010  DD DSN=&&ASSEM,                                         X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F011  DD DSN=&&LINK1,                                         X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F012  DD DSN=&&LINK2,                                         X
//             DISP=(NEW,PASS),                                     X
//             UNIT=&SCRTC,                                         X
//             SPACE=&HLIBSP,                                       X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
```

Figure C-2. (Continued)

```
//FT08F013  DD DSN=&&LINK3,                                          X
//             DISP=(NEW,PASS),                                      X
//             UNIT=&SCRTC,                                          X
//             SPACE=&HLIBSP,                                        X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT08F014  DD DSN=&&LINK4,                                          X
//             DISP=(NEW,PASS),                                      X
//             UNIT=&SCRTC,                                          X
//             SPACE=&HLIBSP,                                        X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT09F001  DD DUMMY
//*
//FT10F001  DD UNIT=&SCRTC,                                          X
//             DISP=(NEW,DELETE),                                    X
//             SPACE=(&PRCL,(&PRCN))
//*
//FT11F001  DD DUMMY
//*
//FT12F001  DD DUMMY
//*-------------------------------------------------------------------*
//ASM     EXEC PGM=&APROG,PARM='&APARM',REGION=&AREG,                X
//             TIME=&ATIME,                                          X
//             COND=(8,LT,SAIL)
//*
//SYSLIB    DD DSN=&AMACL,DISP=SHR
//*
//SYSIN     DD DSN=&&ASSEM,DISP=(OLD,DELETE)
//*
//SYSPRINT  DD &AP1,
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT3    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSPUNCH  DD SYSOUT=B
//*
//SYSGO     DD  DDNAME=SYSLIN
//*
//SYSLIN    DD DSN=&&LOADSET,                                        X
//             UNIT=&SCRTC,                                          X
//             DISP=(MOD,PASS),                                      X
//             SPACE=&FSPACE,                                        X
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*-------------------------------------------------------------------*
//LKEDA   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                  X
//             COND=((4,LT,ASM),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                 X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
```

Figure C-2. (Continued)

```
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB,                                          X
//             UNIT=&LIBU,                                               X
//             VOL=&LIBVOL,                                              X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LINK1,DISP=(OLD,DELETE)
//*
//SAILD     DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT1   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,        X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                     X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                            X
//             DISP=(MOD,PASS),                                          X
//             UNIT=&SCRTC,                                              X
//             SPACE=&FSPACE,                                            X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&BACKUP,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED1   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                      X
//             COND=((4,LT,FORT1),(8,LT,SAIL)),                          X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                     X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(BACKUP),                                  X
//             UNIT=&LIBU,                                               X
//             VOL=&LIBVOL,                                              X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT2   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,        X
//             COND=(8,LT,SAIL)
//*
```

Figure C-2. (Continued)

```
//SYSPRINT  DD &FP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                           X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&VALUE,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED2   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                     X
//             COND=((4,LT,FORT2),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(VALUE),                                  X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT3   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,       X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                           X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&CDATE,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
```

Figure C-2. (Continued)

```
//LKED3   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT3),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(CDATE),                                 X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT4   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&NEXT,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED4   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT4),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
```

Figure C-2. (Continued)

```
//          DD DSN=&&LINK2,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT5   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&SEARCH,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED5   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT5),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(SEARCH),                                X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT6   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&SINK,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
```

Figure C-2. (Continued)

```
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED6   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT6),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(SINK),                                  X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT7   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&RDL,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED7   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT7),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(RDL),                                   X
```

Figure C-2. (Continued)

AD-A068 520 COMPUTER SCIENCES CORP ALBUQUERQUE N MEX F/G 19/4
HULL SYSTEM GUIDE.(U)
JAN 79 L P GABY F29601-78-C-0012
UNCLASSIFIED CSC-C4-C-4041 AFWL-TR-78-115 NL

3 OF 3
AD
A068520

END
DATE
FILMED
6-79
DDC

```
//             UNIT=&LIBU,                                              X
//             VOL=&LIBVOL,                                             X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT8   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&UR,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED8   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT8),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD DSN=&&LINK3,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//FORT9   EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,      X
//             COND=(8,LT,SAIL)
//*
//SYSPRINT  DD &FP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                          X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
```

Figure C-2. (Continued)

```
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&SECOND,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED9   EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT9),(8,LT,SAIL)),                        X
//             PARM='NCAL,&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DUMMY
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=&LIBDSP
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD DSN=&&LINK4,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//       PEND
```

Figure C-2. (Continued)

```
//* * * * * * * * * * * * * P G E N * * * * * * * * * * * * * * * * * * *
//*
//* NAME
//*   PGEN
//*
//* FORMAT
//*   //         EXEC    PGEN,...(OPTIONAL PARAMETERS AS REQUIRED)
//*   //SAIL.INPUT  DD  *
//*
//*       SAIL CHANGES
//*
//*   /*
//*
//* FUNCTION
//*   THIS PROCEDURE EXECUTES THE SAIL PROGRAM AND GENERATES A NEW
//*   SAIL LIBRARY
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//PGEN    PROC LIB=HULLIB,                                              X
//             LIBDS='(OLD,KEEP)',                                      X
//             LIBPRE='SAIL.'  ,                                        X
//             LIBU=,                                                   X
//             LIBVOL=,                                                 X
//             CHNBLK=3521,                                             X
//             CHNLRL=3517,                                             X
//             FILO=,                                                   X
//             FLIB='SYS1.FORTLIB',                                     X
//             FPARM=MAP,                                               X
//             FPROG=IFEAAB,                                            X
//             FREG=512K,                                               X
//             FP1='SYSOUT=A',                                          X
//             FSPACE='(CYL,(10,5),RLSE)',                              X
//             FTIME='(1,0)',                                           X
//             GENO='(0)',                                              X
//             LABO=,                                                   X
//             LPARM='MAP',                                             X
//             LPROG=IEWL,                                              X
//             LREG=250K,                                               X
//             LP1='SYSOUT=A',                                          X
//             LTIME='(0,45)',                                          X
//             PGENSP='(CYL,(5,5),RLSE)',                               X
//             OLD=HULL,                                                X
//             OLDDCB=,                                                 X
//             OLDDS=SHR,                                               X
//             OLDPRE='SAIL.'  ,                                        X
//             OLDU=,                                                   X
//             OLDVOL=,                                                 X
//             PNAME=PLANK,                                             X
//             PRCN=5000,                                               X
//             PRCL=3440,                                               X
//             PS1='SYSOUT=A',                                          X
//             PS2='SYSOUT=A',                                          X
//             SAILBLK=800,                                             X
//             SAILR=80,                                                X
//             SCRTC=SYSDA,                                             X
```

Figure C-3. PGEN. Procedure to Generate PLANK

```
//             SPROG=SAIL,                                              X
//             SREG=175K,                                               X
//             STIME='(2,0)',                                           X
//             WORKSP='(CYL,(5,5))'
//*-------------------------------------------------------------------*
//SAIL    EXEC PGM=&SPROG,TIME=&STIME,REGION=&SREG
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=SHR
//*
//FT01F001  DD DUMMY
//*
//FT02F001  DD DSN=&OLDPRE&OLD&GENO,                                   X
//             UNIT=&OLDU,                                             X
//             LABEL=(&FILO,&LABO,,IN),                                X
//             DISP=&OLDDS,                                            X
//             VOL=&OLDVOL,                                            X
//             DCB=&OLDDCB
//*
//FT03F001  DD UNIT=&SCRTC,                                            X
//             DISP=(NEW,DELETE),                                      X
//             DCB=(RECFM=VBS,LRECL=&CHNLRL,BLKSIZE=&CHNBLK),          X
//             SPACE=(TRK,(20,20))
//*
//FT04F001  DD &PS2,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT05F001  DD DDNAME=INPUT
//*
//FT06F001  DD &PS1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT08F001  DD DSN=&&PLANK,                                            X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&PGENSP,                                          X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT09F001  DD DUMMY
//*
//FT10F001  DD UNIT=&SCRTC,                                            X
//             DISP=(NEW,DELETE),                                      X
//             SPACE=(&PRCL,(&PRCN))
//*
//FT11F001  DD DUMMY
//*
//FT12F001  DD DUMMY
//*-------------------------------------------------------------------*
//FORT    EXEC PGM=&FPROG,                                             X
//             REGION=&FREG,                                           X
//             PARM='NAME=SAIL,&FPARM',                                X
//             TIME=&FTIME,                                            X
//             COND=(8,LT,SAIL)
```

Figure C-3. (Continued)

```
//*
//SYSPRINT  DD &FP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                           X
//             DISP=(MOD,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=&FSPACE,                                          X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&PLANK,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=&WORKSP
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED    EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                     X
//             COND=((4,LT,FORT),(8,LT,SAIL)),                         X
//             PARM='&LPARM'
//*
//SYSPRINT  DD &LP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DSN=&FLIB,DISP=SHR
//          DD DSN=&LIBPRE&LIB,DISP=SHR
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&LIBPRE&LIB(&PNAME),                                 X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                           X
//             DISP=&LIBDS
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//       PEND
```

Figure C-3. (Continued)

```
//* * * * * * * * * * * * * * K E E L * * * * * * * * * * * * * * * * * * *
//*
//* NAME
//*   KEEL
//*
//* FORMAT
//*   //         EXEC    KEEL,...(OPTIONAL PARAMETERS AS REQUIRED)
//*   //         EXEC    KEEL,...(OPTIONAL PARAMETERS AS REQUIRED)
//*   //KEEL.STATION   DD   ..HULL STATION DATA SET (NEW)..
//*   //KEEL.INPUT DD  *
//*
//*      KEEL INPUT CARDS
//*
//*   /*
//*   //SAIL.INPUT  DD  *
//*
//*       SAIL CHANGES
//*
//*   /*
//*
//* FUNCTION
//*   THIS PROCEDURE EXECUTES THE PROBLEM INIATIALIZATION PROGRAM
//*   (KEEL) FROM THE HULL HYDRO-CODE SYSTEM
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//KEEL    PROC LIB=HULLIB,                                                  X
//             LIBPRE='SAIL.',                                              X
//             LIBU=,                                                       X
//             LIBVOL=,                                                     X
//             CHNBLK=3521,                                                 X
//             CHNLRL=3517,                                                 X
//             CREG=100K,                                                   X
//             FLIB='SYS1.FORTLIB',                                         X
//             FILO=,                                                       X
//             FPARM=MAP,                                                   X
//             FPROG=IFEAAB,                                                X
//             FP1='SYSOUT=A',                                              X
//             FSPACE='(CYL,(10,5),RLSE)',                                  X
//             FTIME='(1,0)',                                               X
//             GENO='(0)',                                                  X
//             GOSPACE='(CYL,(20,5,1))',                                    X
//             LABO=,                                                       X
//             LNAME=KEEL,                                                  X
//             LPARM='MAP',                                                 X
//             LPROG=IEWL,                                                  X
//             LREG=250K,                                                   X
//             LP1='SYSOUT=A',                                              X
//             LTIME='(0,45)',                                              X
//             KEELSP='(CYL,(10,5),RLSE)',                                  X
//             KREG=512K,                                                   X
//             KP1='SYSOUT=A',                                              X
//             KTIME='(2,0)',                                               X
//             OLD=HULL,                                                    X
//             OLDDCB=,                                                     X
//             OLDDS=SHR,                                                   X
```

Figure C-4. KEEL. Procedure to Run KEEL

```
//             OLDPRE='SAIL.',                                          X
//             OLDU=,                                                   X
//             OLDVOL=,                                                 X
//             PPROG=PLANK,                                             X
//             PP1='SYSOUT=A',                                          X
//             PRCN=5000,                                               X
//             PRCL=3440,                                               X
//             PREG=100K,                                               X
//             PTIME='(1,0)',                                           X
//             PS1='SYSOUT=A',                                          X
//             PS2='SYSOUT=A',                                          X
//             SAILBLK=800,                                             X
//             SAILR=80,                                                X
//             SCRTC=SYSDA,                                             X
//             SPROG=SAIL,                                              X
//             SREG=175K,                                               X
//             STIME='(2,0)',                                           X
//             WORKSP='(CYL,(5,5))'
//*-------------------------------------------------------------------*
//KEEL    EXEC PGM=IEBGENER,REGION=&CREG
//SYSPRINT  DD DUMMY
//SYSIN     DD DUMMY
//SYSUT1    DD DDNAME=INPUT
//SYSUT2    DD DSN=&&KEELI,                                            X
//             DISP=(NEW,PASS),                                         X
//             UNIT=&SCRTC,                                             X
//             SPACE=(TRK,(5,5),RLSE),                                  X
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*-------------------------------------------------------------------*
//PLANK   EXEC PGM=&PPROG,TIME=&PTIME,REGION=&PREG
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=SHR
//*
//FT05F001  DD DSN=&&KEELI,DISP=(OLD,PASS)
//*
//FT06F001  DD &PP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT07F001  DD DSN=&&ALTI,                                             X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=(TRK,(5,5),RLSE),                                 X
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*-------------------------------------------------------------------*
//SAIL    EXEC PGM=&SPROG,TIME=&STIME,REGION=&SREG,                    X
//             COND=(8,LT,PLANK)
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=SHR
//*
```

Figure C-4. (Continued)

```
//FT01F001  DD DUMMY
//*
//FT02F001  DD DSN=&OLDPRE&OLD&GENO,                                    X
//             UNIT=&OLDU,                                              X
//             LABEL=(&FILO,&LABO,,IN),                                 X
//             DISP=&OLDDS,                                             X
//             VOL=&OLDVOL,                                             X
//             DCB=&OLDDCB
//*
//FT03F001  DD UNIT=&SCRTC,                                             X
//             DISP=(NEW,DELETE),                                       X
//             DCB=(RECFM=VBS,LRECL=&CHNLRL,BLKSIZE=&CHNBLK),           X
//             SPACE=(TRK,(20,20))
//*
//FT04F001  DD &PS2,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT05F001  DD DDNAME=INPUT
//*
//FT06F001  DD &PS1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT08F001  DD DSN=&&KEEL,                                              X
//             DISP=(NEW,PASS),                                         X
//             UNIT=&SCRTC,                                             X
//             SPACE=&KEELSP,                                           X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT09F001  DD DSN=&&ALTI,DISP=(OLD,DELETE)
//*
//FT10F001  DD UNIT=&SCRTC,                                             X
//             DISP=(NEW,DELETE),                                       X
//             SPACE=(&PRCL,(&PRCN))
//*
//FT11F001  DD DUMMY
//*
//FT12F001  DD DUMMY
//*-------------------------------------------------------------------*
//FORT    EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,       X
//             COND=((8,LT,PLANK),(8,LT,SAIL))
//*
//SYSPRINT  DD &FP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                           X
//             DISP=(MOD,PASS),                                         X
//             UNIT=&SCRTC,                                             X
//             SPACE=&FSPACE,                                           X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&KEEL,DISP=(OLD,DELETE)
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(CYL,(2,2))
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=(CYL,(2,2))
```

Figure C-4. (Continued)

```
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED    EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT),(8,LT,SAIL),(8,LT,PLANK)),            X
//             PARM='&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DSN=&LIBPRE&LIB,DISP=SHR
//          DD DSN=&FLIB,DISP=SHR
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&&GOSET(&LNAME),                                    X
//             UNIT=&SCRTC,                                            X
//             DISP=(,PASS),                                           X
//             SPACE=&GOSPACE
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//GO      EXEC PGM=*.LKED.SYSLMOD,TIME=&KTIME,REGION=&KREG,            X
//             COND=((4,LT,LKED),(4,LT,FORT),(8,LT,SAIL),(8,LT,PLANK))
//*
//FT04F001  DD DSN=*.KEEL.DATA,DISP=(OLD,KEEP)
//*
//FT05F001  DD DSN=&&KEELI,DISP=(OLD,DELETE)
//*
//FT06F001  DD &KP1,                                                   X
//           DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT09F001  DD DSNAME=*.KEEL.STATION,DISP=(OLD,KEEP)
//*
//FT10F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.KEEL.DATA
//*
//FT11F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.KEEL.DATA
//*
//FT14F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.KEEL.DATA
//*
//FT21F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.KEEL.DATA
//*
//FT44F001  DD UNIT=&SCRTC,                                            X
```

Figure C-4. (Continued)

```
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.KEEL.DATA
//*
//FT45F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.KEEL.DATA
//*
//       PEND
```

Figure C-4. (Continued)

```
//* * * * * * * * * * * * * * H U L L * * * * * * * * * * * * * * * * * * *
//*
//* NAME
//*   HULL
//*
//* FORMAT
//*   //         EXEC    HULL,...(OPTIONAL PARAMETERS AS REQUIRED)
//*   //HULL.DATA DD ..HULL PROBLEM DATA SET..
//*   //HULL.STATION DD ..HULL STATION DATA SET..
//*   //HULL.INPUT DD  *
//*
//*      HULL INPUT CARDS
//*
//*   /*
//*   //SAIL.INPUT  DD  *
//*
//*       SAIL CHANGES
//*
//*   /*
//*
//* FUNCTION
//*   THIS PROCEDURE EXECUTES THE PROBLEM PROGRAM
//*   (HULL) FROM THE HULL HYDRO-CODE SYSTEM
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//HULL    PROC LIB=HULLIB,                                                  X
//             LIBPRE='SAIL.',                                              X
//             LIBU=,                                                       X
//             LIBVOL=,                                                     X
//             CHNBLK=3521,                                                 X
//             CREG=100K,                                                   X
//             CHNLRL=3517,                                                 X
//             FILO=,                                                       X
//             FLIB='SYS1.FORTLIB',                                         X
//             FPARM=MAP,                                                   X
//             FPROG=IFEAAB,                                                X
//             FREG=512K,                                                   X
//             FP1='SYSOUT=A',                                              X
//             FSPACE='(CYL,(10,5),RLSE)',                                  X
//             FTIME='(1,0)',                                               X
//             GENO='(0)',                                                  X
//             GOSPACE='(CYL,(20,5,1))',                                    X
//             LABO=,                                                       X
//             LNAME=HULL,                                                  X
//             LPARM='MAP',                                                 X
//             LPROG=IEWL,                                                  X
//             LREG=512K,                                                   X
//             LP1='SYSOUT=A',                                              X
//             LTIME='(0,45)',                                              X
//             HULLSP='(CYL,(10,5),RLSE)',                                  X
//             HP1='SYSOUT=A',                                              X
//             HREG=175K,                                                   X
//             HTIME=2,                                                     X
//             OLD=HULL,                                                    X
//             OLDDCB=,                                                     X
```

Figure C-5. HULL. Procedure to Run HULL

```
//             OLDDS=SHR,                                                X
//             OLDPRE='SAIL.',                                           X
//             OLDU=,                                                    X
//             OLDVOL=,                                                  X
//             PPROG=PLANK,                                              X
//             PP1='SYSOUT=A',                                           X
//             PRCN=5000,                                                X
//             PRCL=3440,                                                X
//             PREG=100K,                                                X
//             PTIME='(1,0)',                                            X
//             PS1='SYSOUT=A',                                           X
//             PS2='SYSOUT=A',                                           X
//             SAILBLK=800,                                              X
//             SAILR=80,                                                 X
//             SCRTC=SYSDA,                                              X
//             SPROG=SAIL,                                               X
//             SREG=175K,                                                X
//             STIME='(2,0)',                                            X
//             WORKSP='(CYL,(5,5))'
//*-------------------------------------------------------------------*
//HULL    EXEC PGM=IEBGENER,REGION=&CREG
//SYSPRINT  DD DUMMY
//SYSIN     DD DUMMY
//SYSUT1    DD DDNAME=INPUT
//SYSUT2    DD DSN=&&HULLI,                                             X
//             DISP=(NEW,PASS),                                         X
//             UNIT=&SCRTC,                                             X
//             SPACE=(TRK,(5,5),RLSE),                                  X
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*-------------------------------------------------------------------*
//PLANK   EXEC PGM=&PPROG,TIME=&PTIME,REGION=&PREG
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                         X
//             UNIT=&LIBU,                                              X
//             VOL=&LIBVOL,                                             X
//             DISP=SHR
//*
//FT04F001  DD DSNAME=*.HULL.DATA,DISP=(OLD,KEEP)
//*
//FT05F001  DD DSN=&&HULLI,DISP=(OLD,PASS)
//*
//FT06F001  DD &PP1,                                                    X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT07F001  DD DSN=&&ALTI,                                              X
//             DISP=(NEW,PASS),                                         X
//             UNIT=&SCRTC,                                             X
//             SPACE=(TRK,(5,5),RLSE),                                  X
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*-------------------------------------------------------------------*
//SAIL    EXEC PGM=&SPROG,TIME=&STIME,REGION=&SREG,                     X
//             COND=(8,LT,PLANK)
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                         X
//             UNIT=&LIBU,                                              X
```

Figure C-5. (Continued)

```
//             VOL=&LIBVOL,                                              X
//             DISP=SHR
//*
//FT01F001  DD DUMMY
//*
//FT02F001  DD DSN=&OLDPRE&OLD&GENO,                                     X
//             UNIT=&OLDU,                                               X
//             LABEL=(&FILO,&LABO,,IN),                                  X
//             DISP=&OLDDS,                                              X
//             VOL=&OLDVOL,                                              X
//             DCB=&OLDDCB
//*
//FT03F001  DD UNIT=&SCRTC,                                              X
//             DISP=(NEW,DELETE),                                        X
//             DCB=(RECFM=VBS,LRECL=&CHNLRL,BLKSIZE=&CHNBLK),            X
//             SPACE=(TRK,(20,20))
//*
//FT04F001  DD &PS2,                                                     X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT05F001  DD DDNAME=INPUT
//*
//FT06F001  DD &PS1,                                                     X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT08F001  DD DSN=&&HULL,                                               X
//             DISP=(NEW,PASS),                                          X
//             UNIT=&SCRTC,                                              X
//             SPACE=&HULLSP,                                            X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT09F001  DD DSN=&&ALTI,DISP=(OLD,DELETE)
//*
//FT10F001  DD UNIT=&SCRTC,                                              X
//             DISP=(NEW,DELETE),                                        X
//             SPACE=(&PRCL,(&PRCN))
//*
//FT11F001  DD DUMMY
//*
//FT12F001  DD DUMMY
//*-------------------------------------------------------------------*
//FORT    EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,        X
//             COND=((8,LT,PLANK),(8,LT,SAIL))
//*
//SYSPRINT  DD &FP1,                                                     X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                            X
//             DISP=(MOD,PASS),                                          X
//             UNIT=&SCRTC,                                              X
//             SPACE=&FSPACE,                                            X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&HULL,DISP=(OLD,DELETE)
//*
```

Figure C-5. (Continued)

```
//SYSUT1     DD UNIT=&SCRTC,SPACE=(CYL,(2,2))
//*
//SYSUT2     DD UNIT=&SCRTC,SPACE=(CYL,(2,2))
//*
//SYSTERM    DD DUMMY
//*-------------------------------------------------------------------*
//LKED    EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT),(8,LT,SAIL),(8,LT,PLANK)),            X
//             PARM='&LPARM'
//*
//SYSPRINT   DD &LP1,                                                  X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB     DD DSN=&LIBPRE&LIB,DISP=SHR
//           DD DSN=&FLIB,DISP=SHR
//*
//SYSUT1     DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD    DD DSN=&&GOSET(&LNAME),                                   X
//             UNIT=&SCRTC,                                            X
//             DISP=(,PASS),                                           X
//             SPACE=&GOSPACE
//*
//SYSLIN     DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//GO      EXEC PGM=*.LKED.SYSLMOD,TIME=(&HTIME,0),REGION=&HREG,        X
//             PARM='  &HTIME  ',                                      X
//             COND=((4,LT,LKED),(4,LT,FORT),(8,LT,SAIL),(8,LT,PLANK))
//*
//FT04F001   DD DSN=*.HULL.DATA,DISP=(OLD,KEEP)
//*
//FT05F001   DD DSN=&&HULLI,DISP=(OLD,DELETE)
//*
//FT06F001   DD &HP1,                                                  X
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT09F001   DD DSNAME=*.HULL.STATION,DISP=(OLD,KEEP)
//*
//FT10F001   DD UNIT=&SCRTC,                                           X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.HULL.DATA
//*
//FT11F001   DD UNIT=&SCRTC,                                           X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.HULL.DATA
//*
//FT12F001   DD UNIT=&SCRTC,                                           X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.HULL.DATA
//*
//FT13F001   DD UNIT=&SCRTC,                                           X
//             SPACE=&WORKSP,                                          X
```

Figure C-5. (Continued)

```
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.HULL.DATA
//*
//FT14F001  DD UNIT=&SCRTC,                                          X
//             SPACE=&WORKSP,                                        X
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.HULL.DATA
//*
//FT15F001  DD UNIT=&SCRTC,                                          X
//             SPACE=&WORKSP,                                        X
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.HULL.DATA
//*
//FT21F001  DD UNIT=&SCRTC,                                          X
//             SPACE=&WORKSP,                                        X
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.HULL.DATA
//*
//FT22F001  DD UNIT=&SCRTC,SEP=FT21F001,                             X
//             SPACE=&WORKSP,                                        X
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.HULL.DATA
//*
//FT23F001  DD UNIT=&SCRTC,SEP=(FT21F001,FT22F001),                  X
//             SPACE=&WORKSP,                                        X
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.HULL.DATA
//*
//       PEND
```

Figure C-5. (Continued)

```
//* * * * * * * * * * * * * * P U L L * * * * * * * * * * * * * * * * * *
//*
//* NAME
//*   PULL
//*
//* FORMAT
//*   //         EXEC    PULL,...(OPTIONAL PARAMETERS AS REQUIRED)
//*   //PULL.DATA  DD   DATA SET TO BE PLOTTED
//*   //PULL.INPUT DD  *
//*
//*      PULL INPUT CARDS
//*
//*   /*
//*   //SAIL.INPUT  DD  *
//*
//*       SAIL CHSSNGES
//*
//*   /*
//*
//* FUNCTION
//*   THIS PROCEDURE EXECUTES THE PLOTTING PROGRAM
//*   (PULL) FROM THE HULL HYDRO-CODE SYSTEM
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//PULL    PROC LIB=HULLIB,                                              X
//             LIBPRE='SAIL.',                                          X
//             LIBU=,                                                   X
//             LIBVOL=,                                                 X
//             CHNBLK=3521,                                             X
//             CHNLRL=3517,                                             X
//             CREG=100K,                                               X
//             FILO=,                                                   X
//             FLIB='SYS1.FORTLIB',                                     X
//             FPARM=MAP,                                               X
//             FPROG=IFEAAB,                                            X
//             FREG=512K,                                               X
//             FP1='SYSOUT=A,'                                          X
//             FSPACE='(CYL,(10,5),RLSE)',                              X
//             FTIME='(1,0)',                                           X
//             GENO='(0)',                                              X
//             GOSPACE='(CYL,(20,5,1))',                                X
//             LABO=,                                                   X
//             LNAME=PULL,                                              X
//             LPARM='MAP',                                             X
//             LPROG=IEWL,                                              X
//             LREG=250K,                                               X
//             LP1='SYSOUT=A,'                                          X
//             LTIME='(0,45)',                                          X
//             PLREG=512K,                                              X
//             PLTIME=3,                                                X
//             PULLSP='(CYL,(10,5),RLSE)',                              X
//             PLP1='SYSOUT=A,'                                         X
//             OLD=HULL,                                                X
//             OLDDCB=,                                                 X
//             OLDDS=SHR,                                               X
```

Figure C-6. Procedure to Run PULL

```
//             OLDPRE='SAIL.',                                          X
//             OLDU=,                                                   X
//             OLDVOL=,                                                 X
//             PPROG=PLANK,                                             X
//             PP1='SYSOUT=A,'                                          X
//             PRCN=5000,                                               X
//             PRCL=3440,                                               X
//             PREG=100K,                                               X
//             PTIME='(1,0)',                                           X
//             FTIME='(1,0)',                                           X
//             PS1='SYSOUT=A,'                                          X
//             PS2='SYSOUT=A,'                                          X
//             SAILBLK=800,                                             X
//             SAILR=80,                                                X
//             SCRTC=SYSDA,                                             X
//             SPROG=SAIL,                                              X
//             SREG=175K,                                               X
//             STIME='(2,0)',                                           X
//             WORKSP='(CYL,(5,5))'
//*-------------------------------------------------------------------*
//PULL    EXEC PGM=IEBGENER,REGION=&CREG
//SYSPRINT  DD DUMMY
//SYSIN     DD DUMMY
//SYSUT1    DD DDNAME=INPUT
//SYSUT2    DD DSN=&&PULLI,                                            X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=(TRK,(5,5),RLSE),                                 X
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*-------------------------------------------------------------------*
//PLANK   EXEC PGM=&PPROG,TIME=&PTIME,REGION=&PREG
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
//             VOL=&LIBVOL,                                            X
//             DISP=SHR
//*
//FT04F001  DD DSNAME=*.PULL.DATA,DISP=(OLD,KEEP)
//*
//FT05F001  DD DSN=&&PULLI,DISP=(OLD,PASS)
//*
//FT06F001  DD &PP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT07F001  DD DSN=&&ALTI,                                             X
//             DISP=(NEW,PASS),                                        X
//             UNIT=&SCRTC,                                            X
//             SPACE=(TRK,(5,5),RLSE),                                 X
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*-------------------------------------------------------------------*
//SAIL    EXEC PGM=&SPROG,TIME=&STIME,REGION=&SREG,                    X
//             COND=(8,LT,PLANK)
//*
//STEPLIB   DD DSN=&LIBPRE&LIB,                                        X
//             UNIT=&LIBU,                                             X
```

Figure C-6. (Continued)

```
//             VOL=&LIBVOL,                                              X
//             DISP=SHR
//*
//FT01F001  DD DUMMY
//*
//FT02F001  DD DSN=&OLDPRE&OLD&GEN0,                                     X
//             UNIT=&OLDU,                                               X
//             LABEL=(&FILO,&LABO,,IN),                                  X
//             DISP=&OLDDS,                                              X
//             VOL=&OLDVOL,                                              X
//             DCB=&OLDDCB
//*
//FT03F001  DD UNIT=&SCRTC,                                              X
//             DISP=(NEW,DELETE),                                        X
//             DCB=(RECFM=VBS,LRECL=&CHNLRL,BLKSIZE=&CHNBLK),            X
//             SPACE=(TRK,(20,20))
//*
//FT04F001  DD &PS2,                                                     X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT05F001  DD DDNAME=INPUT
//*
//FT06F001  DD &PS1,                                                     X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT08F001  DD DSN=&&PULL,                                               X
//             DISP=(NEW,PASS),                                          X
//             UNIT=&SCRTC,                                              X
//             SPACE=&PULLSP,                                            X
//             DCB=(RECFM=FB,LRECL=&SAILR,BLKSIZE=&SAILBLK)
//*
//FT09F001  DD DSN=&&ALTI,DISP=(OLD,DELETE)
//*
//FT10F001  DD UNIT=&SCRTC,                                              X
//             DISP=(NEW,DELETE),                                        X
//             SPACE=(&PRCL,(&PRCN))
//*
//FT11F001  DD DUMMY
//*
//FT12F001  DD DUMMY
//*-------------------------------------------------------------------*
//FORT    EXEC PGM=&FPROG,REGION=&FREG,PARM='&FPARM',TIME=&FTIME,        X
//             COND=((8,LT,PLANK),(8,LT,SAIL))
//*
//SYSPRINT  DD &FP1,                                                     X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIN    DD DSN=&&LOADSET,                                            X
//             DISP=(MOD,PASS),                                          X
//             UNIT=&SCRTC,                                              X
//             SPACE=&FSPACE,                                            X
//             DCB=(RECFM=BF,LRECL=80,BLKSIZE=1600)
//*
//SYSIN     DD DSN=&&PULL,DISP=(OLD,DELETE)
//*
```

Figure C-6. (Continued)

```
//SYSUT1    DD UNIT=&SCRTC,SPACE=(CYL,(2,2))
//*
//SYSUT2    DD UNIT=&SCRTC,SPACE=(CYL,(2,2))
//*
//SYSTERM   DD DUMMY
//*-------------------------------------------------------------------*
//LKED    EXEC PGM=&LPROG,REGION=&LREG,TIME=&LTIME,                    X
//             COND=((4,LT,FORT),(8,LT,SAIL),(8,LT,PLANK)),            X
//             PARM='&LPARM'
//*
//SYSPRINT  DD &LP1,                                                   X
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//SYSLIB    DD DSN=&LIBPRE&LIB,DISP=SHR
//          DD DSN=&FLIB,DISP=SHR
//*
//SYSUT1    DD UNIT=&SCRTC,SPACE=(1024,(200,20))
//*
//SYSLMOD   DD DSN=&&GOSET(&LNAME),                                    X
//             UNIT=&SCRTC,                                            X
//             DISP=(,PASS),                                           X
//             SPACE=&GOSPACE
//*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//*-------------------------------------------------------------------*
//GO      EXEC PGM=*.LKED.SYSLMOD,TIME=(&PLTIME,0),REGION=&PLREG,      X
//             PARM='  &PLTIME  ',                                     X
//             COND=((4,LT,LKED),(4,LT,FORT),(8,LT,SAIL),(8,LT,PLANK))
//*
//FT04F001  DD DSN=*.PULL.DATA,DISP=(OLD,KEEP)
//*
//FT05F001  DD DSN=&&PULLI,DISP=(OLD,DELETE)
//*
//FT06F001  DD &PLP1,                                                  X
//           DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//*
//FT22F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.PULL.DATA
//*
//FT40F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.PULL.DATA
//*
//FT48F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.PULL.DATA
//*
//FT50F001  DD UNIT=&SCRTC,                                            X
//             SPACE=&WORKSP,                                          X
//             DISP=(NEW,DELETE),                                      X
//             DCB=*.PULL.DATA
```

Figure C-6. (Continued)

```
//*
//FT51F001  DD UNIT=&SCRTC,                                          X
//             SPACE=&WORKSP,                                        X
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.PULL.DATA
//*
//FT52F001  DD UNIT=&SCRTC,                                          X
//             SPACE=&WORKSP,                                        X
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.PULL.DATA
//*
//FT54F001  DD UNIT=&SCRTC,                                          X
//             SPACE=&WORKSP,                                        X
//             DISP=(NEW,DELETE),                                    X
//             DCB=*.PULL.DATA
//*
//       PEND
```

Figure C-6. (Continued)