NASA CR-137900

FLEXSTAB ON THE IBM 360

R.S. Pyle, R.G. Sykora, and S.C. Denman

July 1976

Prepared under contract NAS2-6912 by

Computer Sciences Corporation
Applied Technology Division
1101 San Antonio Road, #202
Mountain View, California 94043

for
Ames Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

1. Report No. NASA CR-137900

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

FLEXSTAB ON THE IBM 360

5. Report Date July, 1976

6. Performing Organization Code

7. Author(s)

R.S. Pyle, R.G. Sykora, and S.C. Denman

8. Performing Organization Report No.

9. Performing Organization Name and Address

Computer Sciences Corporation
Applied Technology Division
1101 San Antonio Road, #202, Mt. View, CA 94043

10. Work Unit No.

11. Contract or Grant No.

NAS2-6912

12. Sponsoring Agency Name and Address

National Aeronautics & Space Administration
Washington, D.C.

13. Type of Report and Period Covered

Contractor Report

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

FLEXSTAB, an array of computer programs developed on CDC equipment, has been converted by Computer Sciences Corporation, to operate on the IBM 360 computation system.

This report contains instructions for installing, validating, and operating FLEXSTAB on the IBM 360. Hardware requirements are itemized and supplemental materials describe JCL sequences, the CDC to IBM conversion, the Input/Output subprograms, and the interprogram data flow.

17. Key Words (Suggested by Author(s))

Computer Program
Aerodynamics
Aeroelasticity, Loads
Stability and Control

18. Distribution Statement

Unclassified - Unlimited

19. Security Classif. (of this report) Unclassified

20. Security Classif. (of this page) Unclassified

21. No. of Pages 55

22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161

Introduction

Originally developed for execution on CDC computational equipment, FLEXSTAB has recently been converted to the IBM360 by Computer Sciences Corporation. Detailed instructions for installing, validating, and operating the IBM version of FLEXSTAB are presented in this report. At the time of installation, FLEXSTAB programs are read from libraries on magnetic tape and stored within the user's IBM computing hardware. Subsequent program validation requires execution of FLEXSTAB with a series of demonstration cases. Both input cards and output listings for each demonstration case are provided. Supplemental materials contained within the Appendices define Job Control Language (JCL) sequences, describe the CDC to IBM software conversion, detail IBM FLEXSTAB Input/Output subprograms, and outline interprogram data flow.

Concepts and terminology within this report presume knowledge of:

1. OS Job Control Language; both generally and specifically for FORTRAN applications.

2. OS Utilities; specifically IEHMOVE and IEBPTPCH.

3. OS Linkage Editor.

Minimum IBM 360 hardware requirements for successful execution of FLEXSTAB include:

1. Model 75 or larger; a smaller model will generally require excessive execution times for a realistic airframe analysis.

2. 408K storage partition.

3. Up to 4.6 megabytes of disk storage for scratch use during execution.

4. Two magnetic tape drives.

5. One additional tape drive is required if plots are generated.

FLEXSTAB ON THE IBM 360

FLEXSTAB Delivery Package

The FLEXSTAB program package has been converted for use on the IBM 360 from a CDC version. It is delivered on two labeled magnetic tapes as a group of unloaded partitioned data sets (PDS) and several ancillary sequential data sets. Table 1 details the contents of reels FLX360 and FLX361, the delivery tapes. The items on these tapes are grouped as follows:

1. Code; both source and object code are provided (reel FLX360, files 1-24),

2. Job Control Language decks for link editing and executing the FLEXSTAB programs. (reel FLX360, file 26),

3. Validation materials consisting of the inputs and outputs for a standard set of demonstration cases, viz.,

   a. card-image input decks (reel FLX360, file 25),

   b. structural matrix inputs, represented by several unformatted data sets (reel FLX360, files 27-31), and,

   c. printer output listings (reel FLX361, files 1-32).

FLEXSTAB Code

Files 1-10 of reel FLX360 contain source code for the 10 primary FLEXSTAB programs. File 11, DSM=PLOTS, contains source code for the four independent plot programs. File 12 contains source code for a library of various common subprograms. In total there are 583 members in these partitioned data sets. Files 13-24 of reel FLX360 contain object code for the respective source code files 1-12, eliminating the requirement to compile or assemble each source member.

Three subprograms are coded in assembler language, the remainder in FORTRAN. Seven frequently used FORTRAN programs were compiled with the level 21.7 IBM FORTRAN H compiler using option 2; the remainder of the FORTRAN subprograms were compiled with the level 21 IBM FORTRAN G compiler. Table 2 itemizes the ten exceptional subprograms and gives the location by source library and member name. Note that the subprogram name and member name may differ.

Job Control Language Decks

Job Control Language Decks for link editing and executing the various FLEXSTAB programs are provided in a single sequential file, viz., file 26 (DSN=JCLDECKS) of reel FLX360. Each deck begins with a prototype job card to assist in separation. The link edit decks are listed in Appendix I; the execution decks are listed in Appendix II.

The link edit and execution decks are provided as example jobs that contain all essential statements. Each deck should be examined and edited to suit local requirements.

TABLE 1

Contents of FLEXSTAB Delivery Tapes (IBM 360)
Density is 800 bpi

Tape (VOL=SER)	File No.	Tape"Unloaded" LRECL/BLKSIZE	DSNAME	DSORG (PDS/Seq)	#Members (If PDS)	Disk(3330)Space (Trks-Dir Blks)	Disk LRECL/BLKSIZE
FLX360	1	80/800 (FB)	GD	PDS	36	52-3	80/2480
	2	"	AIC	"	54	76-4	"
	3	"	ISIC	"	56	77-4	"
	4	"	NM	"	27	21-3	"
	5	"	ESIC	"	35	36-3	"
	6	"	SDSS	"	162	307-9	"
	7	"	TH	"	47	34-4	"
	8	"	ALDS	"	11	10-2	"
	9	"	SLDS	"	6	13-2	"
	10	"	CAIC	"	4	5-1	"
	11	"	PLOTS	"	37	32-3	"
	12	"	ALIB	"	108	80-7	"
	13	80/800 (FB)	GDOBJ	PDS	36	20-3	80/2480
	14	"	AICOBJ	"	54	25-4	"
	15	"	ISICOBJ	"	56	32-4	"
	16	"	NMOBJ	"	27	10-3	"
	17	"	ESICOBJ	"	35	18-3	"
	18	"	SDSSOBJ	"	162	132-9	"
	19	"	THOBJ	"	47	16-4	"
	20	"	ALDSOBJ	"	11	6-2	"
	21	"	SLDSOBJ	"	6	6-2	"
	22	"	CAICOBJ	"	4	3-1	"
	23	"	PLOTSOBJ	"	37	15-3	"
	24	"	ALIBOBJ	"	108	35-7	"
	25	80/800 (FB)	DEMOPROB	PDS	38	26-3	80/2480

TABLE 1 (Continued)

Tape (VOL-SER)	File No.	Tape LRECL/BLKSIZE	DSNAME	DSORG (PDS/SEQ)	RECFM	#Blocks (Tape)	#Records Approximate
FLX360	26	80/800	JCLDECKS	Seq	FB	10	950
	27	7996/8000	NASTAP.DART1.EA	Seq	VBS	1	–
	28	"	NASTAP.SST.EA1	"	"	31	–
	29	"	NASTAP.SST.E2	"	"	2	–
	30	"	NASTAP.SST.E3	"	"	1	–
	31	"	NASTAP.SST.E4	"	"	1	–
FLX361	1	133/7980	PRINT.DART1.GD	Seq	FBA	10	600
	2	"	PRINT.DART1.AIC	"	"	6	360
	3	"	PRINT.DART1.ISIC	"	"	31	1860
	4	"	PRINT.DART1.NM	"	"	8	480
	5	"	PRINT.DART1.SDSS5	"	"	39	2340
	6	"	PRINT.DART1.SDSS6	"	"	40	2400
	7	"	PRINT.DART1.TH	"	"	9	540
	8	"	PRINT.DART1.SLDS	"	"	6	360
	9	"	PRINT.DART1.ESIC	"	"	8	480
	10	"	PRINT.DART1.SDSS10	"	"	32	1920
	11	"	PRINT.DART1.SDSS11	"	"	33	1980
	12	"	PRINT.DART1.ALDS	"	"	6	360
	13	"	PRINT.DART2.GD	"	"	17	1020
	14	"	PRINT.DART2.AIC	"	"	6	360
	15	"	PRINT.DART2.SDSS	"	"	32	1920
	16	"	PRINT.B707.GD1	"	"	44	2640
	17	"	PRINT.B707.AIC	"	"	6	360
	18	"	PRINT.B707.ISIC	"	"	99	5940
	19	"	PRINT.B707.NM	"	"	15	900
	20	"	PRINT.B707.SDSS5	"	"	154	9240
	21	"	PRINT.B707.TH	"	"	61	3660
	22	"	PRINT.B707.SLDS	"	"	15	900
	23	"	PRINT.B707.GD8	"	"	44	2640
	24	"	PRINT.B707.SDSS9	"	"	59	3540

TABLE 1 (Continued)

Tape (VOL-SER)	File No.	Tape LRECL/BLKSIZE	DSNAME	DSORG (PDS/SEQ)	RECFM	#Blocks (Tape)	#Records Approximate
FLX361	25	133/7980	PRINT.SST.GD	Seq	FBA	45	2700
	26	"	PRINT.SST.AIC	"	"	7	420
	27	"	PRINT.SST.ESIC	"	"	20	1200
	28	"	PRINT.SST.SDSS4	"	"	51	3060
	29	"	PRINT.SST.SDSS5	"	"	88	5280
	30	"	PRINT.SST.ALDS	"	"	11	660
	31	"	PRINT.SST.CAIC	"	"	5	300
	32	"	PRINT.SST.SDSS8	"	"	88	5280

TABLE 2

FLEXSTAB Object Code Not Created
by FORTRAN G Compiler

Subprogram or Control Section	Source Code Location PDS(Member)	Language Processor
CMCS	TH(CMCS)	FORTRAN H,OPT=2
GINO	ALIB(LOCATE)	FORTRAN H,OPT=2
MACHFUNC	ALIB(MCHFNS)	ASSEMBLER
BLOCK	ALIB(RDBLK)	ASSEMBLER
STRING	ALIB(READS)	ASSEMBLER
RECVEC	ALIB(RECVEC)	FORTRAN H,OPT=2
SQUEZE	ALIB(SQUEZE)	FORTRAN H,OPT=2
VIP	ALIB(VIP)	FORTRAN H,OPT=2
VLIN	ALIB(VLIN)	FORTRAN H,OPT=2
ZAP	ALIB(ZAP)	FORTRAN H,OPT=2

Installation of FLEXSTAB Code

To install FLEXSTAB code perform the following steps:

1. Pre-allocate space for data set FLEXSTAB.LOADMODL.LIBRARY, which requires the equivalent of three hundred fifty tracks on a 3330 series disk. The space requirements for the individual members of this load module library are listed in Table 3.

2. Load the 12 object partitioned data sets to a suitable direct access device (using IEHMOVE). Space requirements are provided in Table 1.

3. Catalog the FLEXSTAB.LOADMODL.LIBRARY and the 12 object data sets. If cataloging is not desired, UNIT and VOLUME parameters must be added to JCL referencing these data sets.

4. Execute link edit jobs for the ten primary FLEXSTAB programs. (These are the first ten jobs listed in Appendix I.) Link editing the four FLEXSTAB plot programs requires a special procedure. Only one JCL deck is provided; it must be modified and used in turn for each plot program. Consult the JCL sample on page I-14 of Appendix I for details regarding the modifications for each plot program. In addition, successful link editing of the plot programs requires a library of CALCOMP routines. This library should be concatenated to the SYSLIB data set for either step LKPLOTS1 or LKPLOTS2, depending on whether it is in object or load module form, respectively.

5. Create and keep the FLEXSTAB COPY program load module using the JCL below. (The FLEXSTAB COPY program is used during FLEXSTAB executions for tape-to-disk and disk-to-tape copy operations.

```
//S1  EXEC PGM=IEWL,REGION=100K
//SYSPRINT DD SYSOUT=A
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(20,10))
//SYSLMOD  DD DSN=FLEXSTAB.LOADMODL.LIBRARY(COPY),DISP=OLD
//SYSLIN   DD DSN=ALIBOBJ(COPYTAPE),DISP=OLD
```

Execution of FLEXSTAB Programs

The JCL decks provided for execution of the FLEXSTAB programs contain all of the essential statements for FLEXSTAB runs. Note the following items when using or modifying this execution JCL.

1. Data set GDTAPE is created by the GD program. Since GDTAPE is used by several other FLEXSTAB programs, it is cataloged by the GD execution JCL.

2. To keep core requirements to the sizes quoted in Table 3, BUFNO=1 is specified within the DCB parameter of all FORTRAN DD statements.

3. Many execution decks contain job steps to copy data between tape and disk. Volume information must be supplied for each tape involved.

TABLE 3

CORE MEMORY AND DISK STORAGE REQUIREMENTS FOR FLEXSTAB LOAD MODULES

Module Name	Execution Core Requirement	Execution Scratch Disk Req'mt.(3330 Tracks)*	Load Module Disk Storage Required (3330 Tracks)
GD	296 K	15	27
AIC	386 K	2310	43
ISIC	366 K	3820	43
NM	252 K	3520	20
ESIC	312 K	2010	29
SDSS	408 K	3220	112
TH	382 K	1020	24
ALDS	226 K	880	17
SLDS	214 K	1120	20
CAIC	140 K	1060	15
			350 Total

* If installation scratch disk space is not plentiful, module scratch requirements can be reduced by changing the supplied execution JCL. For all but the largest models, scratch space reductions of from 50% - 80% will probably allow satisfactory runs.

FLEXSTAB Validation Products

The IBM version of FLEXSTAB has been executed with demonstration inputs from the DART I, DART II, Boeing 707-320B, and Boeing 2707-300PT(SST) cases. The inputs and outputs for these runs are provided for validation purposes. Newly installed IBM code can be executed with these inputs. The results can be compared with those provided. Table 4 lists the program, the input and the output for each validation run. Sections 3 and 4 of Reference 1 (FLEXSTAB 1.02.00 Demonstration Cases and Results) describes the validation models thoroughly, indicates modeling and analysis options selected, and itemizes the unformatted inputs required for each run. Table 4 quotes CPU times for runs on an IBM 360 Model 91 to help estimate timing for other models.

Card Image Inputs

Card image inputs are provided in file 25 (DSN=DEMOPROB) of reel FLX360. Since DEMOPROB is a PDS, it must be loaded onto disk prior to any validation runs. Each member of PDS DEMOPROB is an input deck for a particular model and program --- see Table 4.

Columns 74-77 of each input card image contain a number identifying the card by type. These numbers correspond to the card and cardset numbers used in Ref. 2 (FLEXSTAB User's Manual) to document input deck structure. Some cards in the decks provided are data deck comment cards: They are optional and no card or cardset number is given in Ref. 2. The digits 9999 appear in columns 74-77 of such cards.

Structural Matrix Inputs

Structural matrices are provided for runs of the ESIC and ALOADS programs. Table 4 correlates runs to structural matrix requirements by listing the file number on reel FLX360 for the matrix required. Table 5 lists the matrices by their file numbers, giving (1) the DSNAME as it appears in the label, (2) the FLEXSTAB name as it appears in the appendices of Ref. 2, and (3) a short description of the matrix.

Printer Outputs

The validation problem output files (files 1-32 of reel FLX361) can be printed as needed for comparison with local validation runs.

TABLE 4

FLEXSTAB Validation Cases

Model (Airframe)	Member Name for Card Image Input (DSN=DEMO PROB)	Structural Matrix Input(s), If Used, by File Number (Reel FLX360)	Program	CPU Time Model 91 (Seconds)	Output Listing by File Number (Reel FLX361)
DART I	DK01-01	-	GD	1	1
DART I	DK01-02	-	AIC	6	2
DART I	DK01-03	-	ISIC	11	3
DART I	DK01-04	-	NM	6	4
DART I	DK01-05	-	SDSS	10	5
DART I	DK01-06	-	SDSS	14	6
DART I	DK01-07	-	TH	3	7
DART I	DK01-08	-	SLOADS	3	8
DART I	DK01-09	27	ESIC	3	9
DART I	DK01-10	-	SDSS	6	10
DART I	DK01-11	-	SDSS	9	11
DART I	DK01-12	27	ALOADS	2	12
DART II	DK02-01	-	GD	2	13
DART II	DK02-02	-	AIC	14	14
DART II	DK02-03	-	SDSS	9	15
707-320B	DK03-01	-	GD	6	16
707-320B	DK03-02	-	AIC	626	17
707-320B	DK03-03	-	ISIC	830	18
707-320B	DK03-04	-	NM	130	19
707-320B	DK03-05	-	SDSS	571	20
707-320B	DK03-06	-	TH	85	21
707-320B	DK03-07	-	SLOADS	5	22
707-320B	DK03-08	-	GD	6	23
707-320B	DK03-09	-	SDSS	54	24
707-320B	DK03-10	-	GDPLOT	Not Supplied	
707-320B	DK03-11	-	EAPLOT	Not Supplied	
707-320B	DK03-12	-	NMPLOT	Not Supplied	
707-320B	DK03-13	-	PDPLOT	Not Supplied	
707-320B	DK03-14	-	THPLOT	Not Supplied	

TABLE 4 (Continued)

Model (Airframe)	Member Name for Card Image Input (DSN=DEMO PROB)	Structural Matrix input by File Number(Reel FLX360)	Program	CPU Time Model 91 (Seconds)	Output Listing by File Number (Reel FLX361)
SST	DK04-01	-	GD	6	25
SST	DK04-02	-	AIC	1311	26
SST	DK04-03	28,29,30,31	ESIC	234	27
SST	DK04-04	-	SDSS	705	28
SST	DK04-05	-	SDSS	1124	29
SST	DK04-06	28	ALOADS	5	30
SST	DK04-07	-	CAIC	117	31
SST	DK04-08	-	SDSS	1113	32
SST	DK04-09	-	GDPLOT	Not Supplied	

TABLE 5

Structural Matrices

File No. (Reel FLX360)	DSNAME	FLEXSTAB NAME	MATRIX DESCRIPTION
27	NASTAP.DART1.EA	(CAA)-S	Clamped Flexibility Matrix for Dart 1
28	NASTAP.SST.EA1	(CAA)-S	Clamped Flexibility Matrix for the SST
29	NASTAP.SST.E2	(PHI A1)-S	Free Vibration Mode Shape Matrix for the SST
30	NASTAP.SST.E3	(K 1)-S	Generalized Stiffness Matrix for the SST
31	NASTAP.SST.E4	(M 1)-S	Generalized Mass Matrix for the SST

Appendix I

JOB CONTROL LANGUAGE (JCL)
to Link Edit each
FLEXSTAB load module

Duplicates of this JCL are supplied on tape
Volume FLX360 in Data Set JCLDECKS

```
//LNK@GD   JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//LKGD1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=GDOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB   DD DSN=GDOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN   DD *
   INCLUDE LIB(GD)
   INCLUDE LIB(MCHFNS,VERSON)
    CHANGE FLUSH(EXIT)
   INCLUDE LIB(DATA)
//LKGD2 EXEC PGM=IEWL,PARM='MAP,LIST,OVLY,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(GD)
//SYSLIN   DD *
   INCLUDE LIB(ALL)
  OVERLAY A
   INSERT GEOMTY,INITAL,MONITR,SLNBOD,INTBOD,TINBOD,PCKGE,CEGAR
   INSERT BODCP,BODCC,BODYZC,BPARA,BPTHO,CONTPP,PRINTB,FILEB
   INSERT LOFTA,YINTA,LDDGPT,ADJUST,REDO
   INSERT DATA,INTURP,BLOKER,MARITE,PAGE,STATUS,VCONTL
   INSERT GDS60,GDS55,GDS50,GDSFIT,GDSCNT,GDSORG,GDSTBI,GDSUSA,GDS3
   INSERT GDS4,GDS5,GDS31,GDS48,GDS49,GDS56,GDS59,GDS1,GDS2,GDS7,GDS28
   INSERT ICARD,LOFT1,LOFT2,LOFT3,GDS8,VCNTL
  OVERLAY A
   INSERT GDPLOT,CONTRL,RGDTAP,SEARCH,TBLOCK,CENTER,BASTRD,MARGIN
   INSERT PAXIS,PLOTSB,PLOTIB,PLOTTB
   INSERT FTNLOC,OPTFIL
   INSERT GDP3,GDP2,GDP1,GDP8,GDP9,GDP10,GDP11,GDP13
/*
```

Link Edit JCL-AIC

```
//LNK@AIC   JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//LKAIC1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=AICOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB   DD DSN=AICOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD  DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN   DD *
   INCLUDE LIB(AIC)
   INCLUDE LIB(BLKDTA,MCHFNS,VERSON)
//LKAIC2 EXEC PGM=IEWL,PARM='MAP,LIST,OVLY',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD  DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(AIC)
//SYSLIN   DD *
   INCLUDE LIB(ALL)
  OVERLAY A
   INSERT VCONTL,MARITE,BLOKER,VCNTL
  OVERLAY A
   INSERT CPTGEN
  OVERLAY A
   INSERT SETUP,AICGEN,NERFER,ARCTAN,IPANEL,COMPUT,REGNA
   INSERT VI,WI,SUPINT,REGNAA,VII,WII,INTER,ELINE
   INSERT ROTATE,SUBINT,VB,VBAR,W,WBAR,LPREP,LSING
   INSERT SPANEL,SUBSRC,VEL,ASINH,COMP,QUAD1
   INSERT FFLAG,AC01,CP,EDGES,LINE,EX,COMV,VELOC
  OVERLAY A
   INSERT TRNOVR
  OVERLAY A
   INSERT CAMTHK
  OVERLAY B
   INSERT CAMBER,EXPAND
  OVERLAY B
   INSERT THICK
   INSERT TMABT
  OVERLAY A
   INSERT FIELD,MATRIX,SHRINK,STUFF
   INSERT SAIF
  OVERLAY A
   INSERT UNSTDY,WRTROW,DELTAA
   INSERT TMAD,TMDA
    INSERT  TM01
/*
```

Link Edit JCL-ISIC

```
//LNK@ISIC JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//ISIC1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET,SIZE=(178K,16K)',REGION=180K
//SYSPRINT DD SYSOUT=A
//LIB     DD DSN=ISICOBJ,DISP=SHR
//        DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB  DD DSN=ISICOBJ,DISP=SHR
//        DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1  DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN  DD *
   INCLUDE LIB(ISIC)
   INCLUDE LIB(MCHFNS,BLKDTA,VERSON)
//ISIC2 EXEC PGM=IEWL,PARM='MAP,LIST,OVLY,SIZE=(178K,16K)',REGION=180K
//SYSPRINT DD SYSOUT=A
//LIB     DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB  DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1  DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(ISIC)
//SYSLIN  DD *
   INCLUDE LIB(ALL)
  OVERLAY A
   INSERT VCONTL,MARITE,BLOKER,VCNTL
  OVERLAY A
   INSERT GDC,PGD
  OVERLAY A
   INSERT OPTION,SIC1OP
  OVERLAY A
   INSERT FMAT,CMAT,KSUBE,KSUBM,PMAT
   INSERT BCKSUB,MINVER,RECVEC,REDUCE,STDIN,STDOUT,TAAB,TAINV
   INSERT TINVER,TMAD,TMDA,TSAB,ZAP
   INSERT S1S1S1,S3S3S3,S4S4S4
  OVERLAY A
   INSERT SIC
   INSERT CM07,ST08,JP01,JP02,JP03,JP04,JP05,JP06,JP07,JP08,JP09,SB01
   INSERT ST17,ST18,ST19,ST20,TB01,TB06,TB07,TM91,TM92,TM03,TM04,TM05
   INSERT TM06,TM07,TM08,TM10,TM11,TS02,TS03,TS04,TS05,TS06,TS07,TS08
   INSERT TS35,TS34,TS36,ST07,TS22,TS21,TS20,ST02,TS19,ST14,ST15,TS18
   INSERT TS23,ST03,ST04,ST05,ST06,TB05,ST16
  OVERLAY B
   INSERT SDEF,GYRORD,LOADTP,NMINF,PATH,PLOTAP,SBDATA,SICPRT
   INSERT SICRD,TBDATA
   INSERT ST09,JP12,JP11,JP10,TM09,AT03,AT02,AT01,NA01,CM13
  OVERLAY B
   INSERT TMAT,AMTMAT,ATJP,ATMAT,ATNODE,GTMAT,MTJP,MTMAT,MTNODE
   INSERT TFORM
   INSERT RWVEC,WVE
   INSERT TS16,TS15,TS14,TS11,TS28,TS27,TS25,TS26,TS24,TS17,TS13,TS12
   INSERT TS10,TS09,TS01
  OVERLAY B
   INSERT MMAT,CGRAV,MSUBE,DMASS,MSUBR,MSUBX,PHIBAR,PHIBST
   INSERT SUMMAT,TTRANS
   INSERT SM01,LM03,CG02,CG01
```

(CONTINUED)

ISIC-JCL cont.

```
OVERLAY B
 INSERT SAFMAT.CONST,CPMAT,CSMATX,JPMAT,JPTRAN
 INSERT KBGEN,FXMATX,SATRAN,SBIC,SBSIC,TBIC,TBSIC,TBFX
 INSERT CAAB.CMABT,CMATB,CPRINT,CSAB.CSMA
 INSERT CTRANS.IADARY,IARRAY,IBLOCK,IFNDAR,INUSE,IRLEAS
 INSERT TM01,SB02,ST35,ST34,ST33,ST32,ST31,ST30,ST10,JP13,TB08
 INSERT ST11,ST12,TB04,SB03,ST13,ST01.TB02,TB03
/*
```

Link Edit JCL-NM

```
//LNK@NM   JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//LKNM1  EXEC PGM=IEWL,PARM='NOMAP,LIST,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB     DD DSN=NMOBJ,DISP=SHR
//        DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB  DD DSN=NMOBJ,DISP=SHR
//        DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1  DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN  DD *
   INCLUDE LIB(NM)
   INCLUDE LIB(MCHFNS,VERSON)
//LKNM2  EXEC PGM=IEWL,PARM='MAP,LIST,OVLY',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB     DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB  DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1  DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(NM)
//SYSLIN  DD *
   INCLUDE LIB(ALL)
  OVERLAY A
   INSERT VCONTL,MARITE,BLOKER,VCNTL
  OVERLAY A
   INSERT INCONT,CTINIT,NMOPT,DSN
   INSERT DATA,INTURP,PAGE
   INSERT ICARD
  OVERLAY A
   INSERT SHAPE,DYNA,CHOLIN,CHOLT,EIGEN,HSHLDR,VALUE,VECTRI
   INSERT SCALEV,RECOVR
   INSERT NM02,NM03,NM04
  OVERLAY A
   INSERT FNMAT,DIAG,CMAT,MODE,RESID
   INSERT COPY,CTFADD,CTMADD,TSAB
  OVERLAY A
   INSERT NMOUT,DISPSC,NMPRNT
   INSERT ST01,ST02,ST03
/*
```

Link Edit JCL-ESIC

```
//LNK@ESIC JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//ESIC1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET,SIZE=(178K,16K)',REGION=180K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=ESICOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB   DD DSN=ESICOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN   DD *
   INCLUDE LIB(ESIC)
   INCLUDE LIB(MCHFNS,VERSON)
//ESIC2 EXEC PGM=IEWL,PARM='MAP,LIST,OVLY,SIZE=(178K,16K)',REGION=180K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(ESIC)
//SYSLIN   DD *
   INCLUDE LIB(ALL)
  OVERLAY A
   INSERT INTMOP,VCONTL,MARITE,BLOKER,VCNTL,MOP5
  OVERLAY A
   INSERT PREP,GDRD,MOMENT,OPT,PAD,PINF,PLOAD,QUADCH
   INSERT SICPRT,SICRD,SRTPNL,SYNTAX,TRICH
   INSERT DATA,INTURP,PGD,STATUS
   INSERT ST07,ST08,ST09,CM13,CM14,ICARD
  OVERLAY A
   INSERT MATGEN,MOMOUT,MSUBAA,PHIBAR,PHIBST,PMAT,QUINT,TRINT
   INSERT RWVEC,TPRVSP,WVE,ZERO
  OVERLAY A
   INSERT FMAT,CMAT,INRTLR,QINVER,RMAT
   INSERT CINVER,CPRINT,TMAB,TMABT,TMATB,TSAB,TSIA
   INSERT TTRANS,VIP,VLIN
   INSERT TM02,CTMSB
/*
```

Link Edit JCL-SDSS

```
//LNK@SDSS JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//SDSS1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LFT,SIZE=(300K,16K)',REGION=300K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=SDSSOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB   DD DSN=SDSSOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN   DD *
   INCLUDE LIB(SDSS)
   INCLUDE LIB(MCHFNS,BLKDTA,VERSON)
//SDSS2 EXEC PGM=IEWL,PARM='MAP,LIST,OVLY,SIZE=(300K,16K)',REGION=300K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(SDSS)
//SYSLIN   DD *
   INCLUDE LIB(ALL)
  OVERLAY A
   INSERT INITAL,IOUNTS,INTAL1,INTAL2,INTAL3
   INSERT CLRTAB,MARITE,BLOKER,VCONTL,VCNTL
  OVERLAY A
   INSERT PREPAR,BLDTAB,FILF
   INSERT GD02,INPT
  OVERLAY B
   INSERT CARDIN,SPECS,MATPRT,MATTAP,DTRST,DGYRO,DCONRL
   INSERT DSTAB,DWT,DPERT,DDYN,DEXDW,DPRESS,DAREA,DSTRU,RCYCLE
   INSERT ATM062,DATA,INTURP
   INSERT PD02,ICARD
  OVERLAY B
   INSERT TAPEIN,RGD,SUM,ARETAB,MATCAT
   INSERT PGD
   INSERT AR05,AR04,AR03,AR02,AR01
  OVERLAY B
   INSERT FDATA,SICPAR
  OVERLAY A
   INSERT ENGINE,TDATA,TMAT,GYRO
   INSERT TR13,TR12,TR11,TR03,TR02,TR01,TR14,TR15,TR16,TR17,TR18,TR19
  OVERLAY A
   INSERT TRANS,CDATA,CTRANS,CINT,GINT,HTRANS,DTRANS,PTRANS
   INSERT CB11,CB10,CB09,CB08,CB07,CB06,CB05,CB04,CB03,CB02,CB01
   INSERT CB12,CB14
  OVERLAY A
   INSERT DUAL,FORCE,CONTRL,CONSUR
  OVERLAY A
   INSERT BASIC,BDATA,BPSI,PHIBST,CPTRAN,STFTAB
   INSERT BA08,BA07,BA06,BA03,BA02,BA01
  OVERLAY A
   INSERT BCKSUB,COPYM,MINVER,MOP,NEEDS,RECVEC,REDUCE
   INSERT STDIN,STDOUT,TAAB,TAINV,TEMAB,TINVER,TMAB,TMABT,TMATB
   INSERT TRNFER,TSAB,TSATB,TSIA,TSMA,TTRANS,VLIN,ZAP
   INSERT CTMSB,S1S1S1,S3S3S3,S4S4S4
```

(CONTINUED)

```
 OVERLAY C
  INSERT STACON,DER,SDM,MGW
  INSERT CAAB,CPRINT,CSMA
  INSERT PT01,PT02,PT03,PT04,PT05,PT06,PT07,PT08
  INSERT TM05,TM12,TM13,TM14,TM15,TM19,TM24
 OVERLAY D
  INSERT DONE,RAIC,DS,DA
 OVERLAY D
  INSERT TRIM,TMDATA,TS,TA,FS,FA,FT,TRIMCC,TRIMIT,WTDATA,WTDER
  INSERT F2F3,TMPRT,FTOTAL
  INSERT CSAB,LINEIN
  INSERT TM22,TM21,TM16,TM23
 OVERLAY D
  INSERT SHAPE,SMAT,DFP,SD,SDLIST,PRES,SBPRES,TBPRES,LOADS
 OVERLAY D
  INSERT INTDW,RISC,LEC,DMAT
 OVERLAY D
  INSERT SDSP,STADER,AERDER,SPEED,STAPAR
  INSERT SB01,SB02,SB03,SB04,SP01
 OVERLAY D
  INSERT PERT1,FUS,RFS,RQS,RFA,FRA,RQA,EFS,EQS,EFA,EQA
 OVERLAY D
  INSERT PERT2,RDULSC,UAICS,UAICA,UPRES,PPRES
 OVERLAY D
  INSERT PERT3,PDER,STABIT,PSIC,PMAT,PGYRO
  INSERT PMAT1,PMAT2,PMAT3,PMAT4
  INSERT SSPART,STPART
  INSERT PD30,PT11,PT10
 OVERLAY D
  INSERT PERT4,CONDER,GAMF,PLOAD,PELOAD,PGLOAD
 OVERLAY C
  INSERT GUST,PGUST
  INSERT GU01
 OVERLAY A
  INSERT CER,DIGEST,HMAT,LONG1,LAT1,COUP1,LONG2,LAT2,COUP2
  INSERT CMFIND,DMINIT,DMFIND,DMADD,DMDEL,SEC,EIGENC,GENEIG
  INSERT EIGENP,HESQR,SCALF,REALVE,COMPVE,CERPRT,CSHELL,FSHELL,AMPHAS
  INSERT CH01,CH03,CH04,CH05,CH06,CT01,CT02,IV01,IV02,IV03,IV04,IV05
  INSERT IV06,IV07,IV08,IV09,IV10,IV11,IV12,IV13,IV14,IV15,IV16,IV17
  INSERT IV20,IV21,MD01,CH02,CP01,CH07
 OVERLAY A
  INSERT POST,PDATA,MSIZES,SDTAPE,SUMARY
/*
```

Link Edit JCL-TH

```
//LNK@TH     JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//LKTH1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET,SIZE=(178K,16K)',REGION=180K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=THOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB   DD DSN=THOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN   DD *
   INCLUDE LIB(TH)
   INCLUDE LIB(MCHFNS,BLKDTA,VERSON)
//LKTH2 EXEC PGM=IEWL,PARM='MAP,LIST,OVLY,SIZE=(178K,16K)',REGION=180K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(TH)
//SYSLIN   DD *
   INCLUDE LIB(ALL)
  OVERLAY A
   INSERT SETUP,INITAL,DIGEST,UNSET,M,MM1C
   INSERT MARITE,BLOKER,VCONTL
   INSERT GM01,VCNTL
  OVERLAY A
   INSERT TABRD,DERT,DFI,DIFFER,SKI
  OVERLAY A
   INSERT RDRIVE,NRKR,STEP,RBM,THPRIG
  OVERLAY A
   INSERT REDRIV,NRKE,STEPE,REM,EVDUDE,THPRES
   INSERT CAMAB,CSAD,CSDB
   INSERT GC05,GC04,AV03
  OVERLAY A
   INSERT GUSTIN,GMIN,GETM,THGDRD
   INSERT GF06,GF05,GF03,GD01
/*
```

Link Edit JCL-ALDS

```
//LNK@ALDS JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//LKALDS1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB     DD DSN=ALDSOBJ,DISP=SHR
//        DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB  DD DSN=ALDSOBJ,DISP=SHR
//        DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1  DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN  DD *
   INCLUDE LIB(ALOADS)
   INCLUDE LIB(MCHFNS,VERSON)
//LKALDS2 EXEC PGM=IEWL,PARM='MAP,LIST',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB     DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB  DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1  DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(ALDS)
//SYSLIN  DD *
   INCLUDE LIB(ALL)
/*
```

Link Edit JCL-SLDS

```
//LNK@SLDS JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//LKSLDS1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=SLDSOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB   DD DSN=SLDSOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN   DD *
   INCLUDE LIB(SLOADS)
   INCLUDE LIB(MCHFNS,VERSON)
//LKSLDS2 EXEC PGM=IEWL,PARM='MAP,LIST',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(SLDS)
//SYSLIN   DD *
   INCLUDE LIB(ALL)
/*
```

Link Edit JCL-CAIC

```
//LNK@CAIC JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//LKCAIC1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=CAICOBJ,DISP=SHR
//         DD DSN=ALIBOBJ.DISP=SHR
//SYSLIB   DD DSN=CAICOBJ.DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN   DD *
   INCLUDE LIB(CAIC)
   INCLUDF LIB(MCHFNS,VERSON)
//LKCAIC2 EXEC PGM=IEWL,PARM='MAP,LIST',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=&T,DISP=(OLD.DELETE)
//SYSLIB   DD DSN=SYS1.FORTLIB.DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(CAIC)
//SYSLIN   DD *
   INCLUDE LIB(ALL)
/*
```

Link Edit JCL-PLOTTERS

```
//LNK@PLOT JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//LKPLOTS1 EXEC PGM=IEWL,PARM='NOMAP,LIST,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=PLOTSOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSLIB   DD DSN=PLOTSOBJ,DISP=SHR
//         DD DSN=ALIBOBJ,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD  DD UNIT=SYSDA,SPACE=(TRK,(100,20,5)),DISP=(,PASS),DSN=&T(ALL)
//SYSLIN   DD *
   INCLUDE LIB(THPLOT) PDPLOT NMPLOT EAPLOT
   INCLUDE LIB(SCALEX) Special SCALE Routine for THPLOT (remove for others)
   INCLUDE LIB(MCHFNS,VSNTH),VSNPD ,VSNNM ,VSNEA
//LKPLOTS2 EXEC PGM=IEWL,PARM='MAP,LIST,LET',REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB      DD DSN=&T,DISP=(OLD,DELETE)
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(30,20))
//SYSLMOD  DD DISP=OLD,DSN=FLEXSTAB.LOADMODL.LIBRARY(THPLT)
//SYSLIN   DD *                                      PDPLT
   INCLUDE LIB(ALL)                                  NMPLT
/*                                                   EAPLT
```

Use this JCL to Link Edit all 4 plotter programs; change only the INCLUDE and SYSLMOD cards; remove INCLUDE-SCALEX except when creating THPLT load module.

Appendix II

JOB CONTROL LANGUAGE (JCL)
to Execute each
FLEXSTAB program (load module)

Duplicates of this JCL are supplied on tape
Volume FLX360 in Data Set JCLDECKS

The named files which communicate data between separate FLEXSTAB programs are shown as Data Set Names enclosed in □.

Execution JCL-GD

```
//RUN@GD    JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//FLEXSTAB EXEC PGM=GD,REGION=296K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//FT01F001 DD DSN=GDTAPE,SPACE=(TRK,(5,5),RLSE),DISP=(NEW,CATLG),
//            DCB=(BUFNO=1),UNIT=????,VOL=SER=??????
//FT08F001 DD UNIT=SYSDA,SPACE=(TRK,(10,10)),
//            DCB=(RECFM=VBS,LRECL=2044,BLKSIZE=2048,BUFNO=1)
//PLOTTAPE DD UNIT=TP7,DISP=(NEW,KEEP),LABEL=(,BLP),...
//FT05F001 DD *
/*
```

Execution JCL-AIC

```
//RUN@AIC   JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//FLEXSTAB EXEC PGM=AIC,REGION=386K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//FT20F001 DD DSN=GDTAPE,DISP=OLD,DCB=(BUFNO=1)
//UT20 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT21 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT22 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT23 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT24 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT25 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT26 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT27 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT28 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT29 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT30 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT31 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT32 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT33 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT34 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//HD35 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT35 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//UT36 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT37 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT38 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//FT05F001 DD *
/*
//CPT EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD35
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT35
//T DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=AICTAP,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=01
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=02
/*
```

Execution JCL-ISIC

```
//RUN@ISIC JOB   (ACCT,INFO).'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//FLEXSTAB EXEC PGM=ISIC,REGION=366K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//FT25F001 DD DSN=GDTAPE,DISP=OLD,DCB=(BUFNO=1)
//UT01 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT02 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT03 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT04 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT07 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT08 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT09 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT10 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT11 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT12 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT13 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT14 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT15 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT16 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT17 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT18 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT19 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT20 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//HD21 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT21 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//HD22 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT22 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//HD23 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT23 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//HD24 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT24 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//HD25 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT25 DD UNIT=SYSDA,SPACE=(TRK,(100,35)),DISP=(NEW,PASS)
//FT11F001 DD UNIT=SYSDA,SPACE=(TRK,(10,10)),
//            DCB=(RECFM=VBS,LRECL=2044,BLKSIZE=2048,BUFNO=1)
//FT05F001 DD *
/*
//CPT EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT19
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD21
//FT01F003 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT21
//FT01F004 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD22
//FT01F005 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT22
//FT01F006 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD23
//FT01F007 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT23
//FT01F008 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD24
//FT01F009 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT24
//FT01F010 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD25
//FT01F011 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT25
```

(CONTINUED)

ISIC-JCL Cont.

```
//T   DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=SICTAP,VOL=SER=??????
//T2 DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=EATAPE,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=01
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=02
//FT02F003 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=03
//FT02F004 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=04
//FT02F005 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=05
//FT02F006 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=06
//FT02F007 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=07
//FT02F008 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=08
//FT02F009 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=09
//FT02F010 DD DSN=*.T2,VOL=REF=*.T2,DCB=*.D,DISP=(NEW,KEEP),LABEL=10
//FT02F011 DD DSN=*.T2,VOL=REF=*.T2,DCB=*.D,DISP=(NEW,KEEP),LABEL=11
/*
```

Execution JCL-NM

```
//RUN@NM    JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//CP EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD,DCB=(RECFM=U,BLKSIZE=2048),
//   DSN=SICTAP,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(01,,,IN)
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(02,,,IN)
//FT01F003 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(03,,,IN)
//FT01F004 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(04,,,IN)
//FT01F005 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(05,,,IN)
//FT01F006 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(06,,,IN)
//FT01F007 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(07,,,IN)
//FT01F008 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(08,,,IN)
//FT01F009 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(09,,,IN)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F003 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FT02F004 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F005 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FT02F006 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F007 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FT02F008 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F009 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FLEXSTAB EXEC PGM=NM,REGION=252K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//UT01 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT02 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT03 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT04 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT07 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT08 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT09 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT10 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT11 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT12 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//HD13 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT13 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//UT19 DD DISP=(OLD,PASS),DSN=*.CP.FT02F001
//HD21 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F002
//UT21 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F003
//HD22 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F004
//UT22 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F005
//HD23 DD DISP=(OLD,PASS),DSN=*.CP.FT02F006
//UT23 DD DISP=(MOD,PASS),DSN=*.CP.FT02F007
//HD24 DD DISP=(OLD,PASS),DSN=*.CP.FT02F008
//UT24 DD DISP=(MOD,PASS),DSN=*.CP.FT02F009
//FT05F001 DD *
/*
```

(CONTINUED)

NM-JCL Cont.

```
//CPT1 FXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT19
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD23
//FT01F003 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT23
//FT01F004 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD24
//FT01F005 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT24
//T DD UNIT=(TP9,,DEFER),DISP=(,PASS),DSN=SICTP3,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,PASS),LABEL=01
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,PASS),LABEL=02
//FT02F003 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,PASS),LABEL=03
//FT02F004 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,PASS),LABEL=04
//FT02F005 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,PASS),LABEL=05
//CPT2 EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD13
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT13
//T DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=NMTAPE,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=RFF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=06
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=07
/*
```

Execution JCL-ESIC

```
//RUN@ESIC JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB   DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//FLEXSTAB EXEC PGM=ESIC,REGION=312K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//FT01F001 DD DSN=GDTAPE,DISP=OLD,DCB=(BUFNO=1)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(50,20)),
//          DCB=(RECFM=VBS,LRECL=2044,BLKSIZE=2048,BUFNO=1)
//FT03F001 DD UNIT=SYSDA,SPACE=(TRK,(50,20)),DCB=*.FT02F001
//UT04 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT07 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT08 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT09 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//NASTAP DD UNIT=(TP9,,DEFER),DISP=OLD,VOL=SER=??????
//FT09F001 DD DSN=MATRIX.NAME,VOL=REF=*.NASTAP,DISP=OLD,DCB=(BUFNO=1)
//FT09F002 DD DUMMY
//UT10 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT11 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT12 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT13 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT14 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//FT15F001 DD UNIT=SYSDA,SPACE=(TRK,(50,20)),DCB=*.FT02F001
//UT19 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//HD23 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT23 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//HD24 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT24 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//FT05F001 DD *
/*
//CPT EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT19
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD23
//FT01F003 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT23
//FT01F004 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD24
//FT01F005 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT24
//T DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=SICTP3,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=01
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=02
//FT02F003 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=03
//FT02F004 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=04
//FT02F005 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=05
/*
```

Execution JCL-SDSS

```
//RUN@SDSS JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB   DD DSN=FLEXSTAR.LOADMODL.LIBRARY,DISP=SHR
//CP1 EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD,DCB=(RECFM=U,BLKSIZE=2048),
//   DSN=AICTAP,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(01,,,IN)
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(02,,,IN)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//CP2 EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD,DCB=(RECFM=U,BLKSIZE=2048),
//   DSN=SICTP3,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(01,,,IN)
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(02,,,IN)          A
//FT01F003 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(03,,,IN)        (See
//FT01F004 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(04,,,IN)         note
//FT01F005 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(05,,,IN)        below)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(100,35)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F003 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FT02F004 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F005 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FLEXSTAB EXEC PGM=SDSS,REGION=408K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//FT15F001 DD SYSOUT=B
//FT20F001 DD DSN=GDTAPE,DISP=OLD,DCB=(BUFNO=1)
//HD21 DD DISP=(OLD,DELETE),DSN=*.CP1.FT02F001
//UT21 DD DISP=(OLD,DELETE),DSN=*.CP1.FT02F002
//UT22 DD DISP=(OLD,DELETE),DSN=*.CP2.FT02F001
//HD23 DD DISP=(OLD,DELETE),DSN=*.CP2.FT02F002    B
//UT23 DD DISP=(OLD,DELETE),DSN=*.CP2.FT02F003  (See note
//HD24 DD DISP=(OLD,DELETE),DSN=*.CP2.FT02F004    below)
//UT24 DD DISP=(OLD,DELETE),DSN=*.CP2.FT02F005
//UT01 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT02 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT03 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT04 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT07 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT08 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT09 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT10 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT11 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT12 DD UNIT=SYSDA,SPACE=(TRK,(05,05)),DISP=(NEW,PASS)
//UT13 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//HD14 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT14 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//UT20 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//FT05F001 DD *
/*
```

(CONTINUED)

SDSS-JCL Cont.

```
//CPT EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT12
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD14
//FT01F003 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT14
//T DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=SDSSTP,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=01
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=02
//FT02F003 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=03
/*
```

NOTE - Second Copy step at A should be removed and JCL cards at B should be replaced by the following cards for a Rigid case (no SICTP3 data):

```
//UT22 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT23 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT24 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
```

```
//RUN@TH    JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//CP EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD,DCB=(RECFM=U,BLKSIZE=2048),
//   DSN=SDSSTP,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(01,,,IN)
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(02,,,IN)
//FT01F003 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(03,,,IN)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(05,05)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F003 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FLEXSTAB EXEC PGM=TH,REGION=382K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//HD01 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F002
//UT01 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F003
//HD02 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT02 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//FT03F001 DD DSN=GDTAPE,DISP=OLD,DCB=(BUFNO=1)
//UT04 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F001
//FT05F001 DD *
/*
//CPT EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD02
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT02
//T DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=THTAPE,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=01
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=02
/*
```

Execution JCL-ALDS

```
//RUN@ALDS JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB   DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//CP EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD,DCB=(RECFM=U,BLKSIZE=2048),
//  DSN=SDSSTP,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(01,,,IN)
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(02,,,IN)
//FT01F003 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(03,,,IN)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(05,05)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F003 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FLEXSTAB EXEC PGM=ALDS,REGION=226K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//FT01F001 DD DSN=GDTAPE,DISP=OLD,DCB=(BUFNO=1)
//UT02 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F001
//HD03 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F002
//UT03 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F003
//UT04 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//NASTAP DD UNIT=(TP9,,DEFER),DISP=OLD,VOL=SER=??????
//FT07F001 DD DSN=MATRIX.NAME,VOL=REF=*.NASTAP,DISP=OLD,DCB=(BUFNO=1)
//FT07F002 DD DUMMY
//UT07 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT08 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//FT10F001 DD UNIT=SYSDA,SPACE=(TRK,(10,20)),
//          DCB=(RECFM=VBS,LRECL=2044,BLKSIZE=2048,BUFNO=1)
//FT10F002 DD UNIT=SYSDA,SPACE=(TRK,(10,20)),DCB=*.FT10F001
//FT10F003 DD UNIT=SYSDA,SPACE=(TRK,(10,20)),DCB=*.FT10F001
//FT10F004 DD UNIT=SYSDA,SPACE=(TRK,(10,20)),DCB=*.FT10F001
//FT10F005 DD UNIT=SYSDA,SPACE=(TRK,(10,20)),DCB=*.FT10F001
//FT10F006 DD UNIT=SYSDA,SPACE=(TRK,(10,20)),DCB=*.FT10F001
//FT05F001 DD *
/*
```

Execution JCL-SLDS

```
//RUN@SLDS JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//CP1 EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER).DISP=OLD,DCB=(RFCFM=U,BLKSIZE=2048),
//   DSN=EATAPE,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(10,,,IN)
//FT01F002 DD DSN=*.T,VOL=RFF=*.T,DISP=OLD,LABEL=(11,,,IN)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//CP2 EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD,DCB=(RFCFM=U,BLKSIZE=2048),
//   DSN=SDSSTP.VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(01,,,IN)
//FT01F002 DD DSN=*.T,VOL=RFF=*.T,DISP=OLD,LABEL=(02,,,IN)
//FT01F003 DD DSN=*.T,VOL=RFF=*.T,DISP=OLD,LABEL=(03,,,IN)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(05,05)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F003 DD UNIT=SYSDA,SPACE=(TRK.(500.50)),DCB=*.T.DISP=(NEW,PASS)
//FLEXSTAB FXEC PGM=SLDS,REGION=214K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZF=2480,BUFNO=1)
//HD01 DD DISP=(OLD,DELETE).DSN=*.CP1.FT02F001
//UT01 DD DISP=(OLD,DELETE),DSN=*.CP1.FT02F002
//UT02 DD DISP=(OLD,DELETE),DSN=*.CP2.FT02F001
//HD03 DD DISP=(OLD,DELETE),DSN=*.CP2.FT02F002
//UT03 DD DISP=(OLD,DELETE),DSN=*.CP2.FT02F003
//UT04 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//FT05F001 DD *
/*
```

Execution JCL-CAIC

```
//RUN@CAIC JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//CP EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=(OLD,PASS),DCB=(RECFM=U,BLKSIZE=2048),
//   DSN=AICTAP,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(01,,,IN)
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=(02,,,IN)
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DCB=*.T,DISP=(NEW,PASS)
//FLEXSTAB EXEC PGM=CAIC,REGION=140K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(50,10),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//HD01 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F001
//UT01 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F002
//HD02 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT02 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//FT05F001 DD *
/*
//CPT EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD02
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT02
//T DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=AICTAP.CORR,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=03
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=04
/*
```

Execution JCL-EAPLOT

```
//RUN@EA#P JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//CP EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD,DCB=(RECFM=U,BLKSIZE=2048),
//  DSN=EATAPE,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=10
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=11
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(100,35)),DCB=*.T,DISP=(NEW,PASS)
//FLEXSTAB EXEC PGM=EAPLT,REGION=200K,COND=(0,LT)
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//HD03 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F001
//UT03 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F002
//PLOTTAPE DD UNIT=TP7,DISP=(NEW,KEEP),LABEL=(,BLP),...
//FT05F001 DD *
/*
```

Execution JCL-NMPLOT

```
//RUN@NM#P JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//CP EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD.DCB=(RECFM=U,BLKSIZE=2048),
//   DSN=NMTAPE.VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T.DISP=OLD.LABEL=01
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD.LABEL=02
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW.PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(100,35)),DCB=*.T,DISP=(NEW,PASS)
//FLEXSTAB EXEC PGM=NMPLT.REGION=150K,COND=(0,LT)
//FT06F001 DD SYSOUT=A
//UT01 DD UNIT=SYSDA,SPACE=(TRK,(100.35))
//UT02 DD UNIT=SYSDA,SPACE=(TRK,(100.35))
//HD13 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F001
//UT13 DD DISP=(OLD.DELETE),DSN=*.CP.FT02F002
//PLOTTAPE DD UNIT=TP7,DISP=(NEW,KEEP),LABEL=(,BLP),...
//FT05F001 DD *
/*
```

Execution JCL-PDPLOT

```
//RUN@PD#P JOB   (ACCT.INFO),'NAME',MSGLEVEL=(2.0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//FLEXSTAB EXEC PGM=PDPLT.REGION=100K
//FT06F001 DD SYSOUT=A
//PLOTTAPE DD UNIT=TP7.DISP=(NEW.KEEP),LABEL=(.BLP),...
//FT05F001 DD *
/*
```

Execution JCL-THPLOT

```
//RUN@TH#P JOB  (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB   DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//CP EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//T DD UNIT=(TP9,,DEFER),DISP=OLD,DCB=(RECFM=U,BLKSIZE=2048),
//   DSN=THTAPE,VOL=SER=??????
//FT01F001 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=01
//FT01F002 DD DSN=*.T,VOL=REF=*.T,DISP=OLD,LABEL=02
//FT02F001 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DCB=*.T,DISP=(NEW,PASS)
//FT02F002 DD UNIT=SYSDA,SPACE=(TRK,(100,35)),DCB=*.T,DISP=(NEW,PASS)
//FLEXSTAB EXEC PGM=THPLT,REGION=150K,COND=(0,LT)
//FT06F001 DD SYSOUT=A
//UT01 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//HD02 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F001
//UT02 DD DISP=(OLD,DELETE),DSN=*.CP.FT02F002
//PLOTTAPE DD UNIT=TP7,DISP=(NEW,KEEP),LABEL=(,BLP),...
//FT05F001 DD *
/*
```

APPENDIX III

The CDC to IBM Conversion of FLEXSTAB Software

The FLEXSTAB program package was initially developed on CDC 6000 and 7000 series computers. No effort was made to code the programs in a machine dependent fashion. FORTRAN usage was not restricted to ANSI statements and machine dependent code was scattered among a large number of routines. Also, the system consists of 14 separate programs with no executive superstructure. Therefore, all communication between programs is established by transmitting files of information.

In order that existing user and programmer documentation be equally useful with the IBM version, no change in program architecture or data communication techniques was permitted. Since program maintenance updates were expected, the source code line numbers within each version were to be as alike as possible.

Operating under these constraints, the changes necessary to effect the conversion can be broadly grouped into: (1) changes due to FORTRAN dialect differences, (2) changes made necessary by the smaller 32-bit IBM computer word and, (3) changes due to FORTRAN compiler or operating system functional differences.

The CDC FORTRAN dialect differences were quickly isolated through diagnostic messages from the IBM FORTRAN compiler. They included:

a) The PROGRAM, OVERLAY, and CALL OVERLAY statements removed during conversion.

b) Multiple assignment statements (e.g., I=J=K=1), replaced by separate assignment statements.

c) The IF (EOF(m)) statement replaced by FORTRAN I/O END= or GINO (See Appendix IV) logical END parameters.

d) Hollerith literals in assignment statements (e.g., A=4HCASE), replaced by statement function definition and appropriate calls (e.g., A=LIT(4HCASE) with Statement Function LIT(L)=L.)

e) DECODE command, replaced by BACKSPACE and a (re)READ.

f) BUFFER IN and BUFFER OUT, replaced by standard FORTRAN I/O or calls to GINO. (See Appendix IV.)

g) Octal constants in DATA and assignment statements, replaced by integer or hexadecimal constants.

h) Use of logical operators .AND. and .OR. with operands not typed logical, replaced by function calls to an assembly language subprogram.

i) References to first element of an array without explicit subscript (e.g., X=A for X=A(1)), replaced by a subscripted reference (e.g., X=A(1)).

The 32-bit IBM computer word did not provide sufficient precision for FLEXSTAB floating-point computations. The remedy was to add an "IMPLICIT REAL*8 (A-H, O-Z)" statement to each subroutine. This statement forces all floating point arithmetic and data storage to double precision (64-bit) format. The resultant computation precision slightly exceeds that provided by the 60-bit CDC word. The standard 32-bit format was retained for INTEGER variables. Consequently, the contents of many common blocks which contained both real and integer type variables had to be re-ordered to assure proper IBM storage alignment.

Also, one functional difference between the FORTRAN object code produced by the CDC compiler and by the IBM compiler caused several errors that were quite difficult to diagnose. The difference involved the subprogram to subprogram parameter passing methods. In the CDC object code all parameters are passed with calls-by-reference. In the IBM object code only dimensioned parameters are passed in this fashion; simple variables are passed in calls-by-value. In the FLEXSTAB FORTRAN code, there are several occurrences of the following;

Subroutine A calls subroutine B with a parameter dimensioned only in A. Subroutine B, where the parameter is not dimensioned or used, passes the parameter by calling subroutine C. Subroutine C again contains a dimension statement for the parameter and uses several elements of the parameter array.

With the CDC method, no problem occurred. However, with the IBM method the call from B to C was made as a call-by-value. Only the first element of the parameter array was properly passed. Dimensioning the parameter within B corrected the problem.

A second CDC versus IBM functional difference involved FORTRAN I/O and its interface with the data management routines and the JCL of the respective operating systems. As developed for CDC equipment under the SCOPE and KRONOS operating systems, FLEXSTAB performs all I/O by use of standard FORTRAN language I/O statements (READ, WRITE, END FILE, REWIND): an unlimited number of logical files may be written on each FORTRAN unit. On the IBM 360, each logical file is a physical file requiring a Data Definition statement in the JCL. The FLEXSTAB analysis of an airframe such as the Boeing 707 requires over 450 logical files when a maximum of 255 Data Definition statements are allowed by OS/360. This situation mandated the development of an IBM FLEXSTAB I/O package. This package, named GINO for Generalized Input/Output, is described in Appendix IV.

Appendix IV

GINO - A Generalized Input/Output package for FLEXSTAB

GINO is a collection of subprograms, coded in both FORTRAN and assembly language, which handles most unformatted input and output operations within OS/360 FLEXSTAB. The basic unit of I/O required by FLEXSTAB, and thus implemented in GINO, is a variable length logical record. The length of the logical record is completely variable; it depends only on the number and size of items in the read/write list. In FLEXSTAB, a matrix row or vector is generally written and read as a record. A table or other collection of logically related data may also be handled as a record.

The variable length GINO logical record must be stored on a disk device and the characteristics of the device must be taken into consideration. For the disk, the physical unit of organization is the track which stores a fixed number of words. Thus, there is a mismatch between the variable length GINO record and the fixed length disk track.

The mismatch is resolved through blocking. Blocking creates fixed length blocks from variable length records by segmenting and storing the records so as to create: (1) multiple records and record segments per block, or (2) multiple blocks per record. The blocking design parallels quite closely with the Spanned Variable Length Record logic described in Reference 3 (OS Data Management Services). The physical output and subsequent input of these blocks is accomplished through the IBM Basic Sequential Access Method (BSAM). The size of these blocks is a compromise between the effective utilization of core memory and disk storage. The larger the blocks, the fewer the physical I/O operations but the more core memory that is required. To limit the maximum FLEXSTAB execution core requirements to approximately 400K bytes, a block size of 2048 bytes was chosen for the delivered version.

The GINO design also provides a random-sequential technique which permits selection and reading of any file on a multifile unit without reading or passing any other. This random-sequential design utilizes a logical file mark created and sensed only by GINO to replace the physical file mark normally created by the FORTRAN ENDFILE Statement. This arrangement combines groups of data consisting of many files into a single file containing the same data files separated by GINO-specific logical file marks. These logical file marks are a unique data word whose position is recorded in a GINO file locator table.

The capabilities and operation of GINO can best be illustrated with the following example: A data file is created through a series of calls to a GINO output entry point. GINO moves the variable length records to a fixed length block. As each block is filled, GINO writes it to disk. A particular record may fit within a block, or span one or more blocks. Thus a single call to GINO with one logical record may initiate none, one, or many disk writes. When the file is completed, a call is made to the GINO end-of-file entry point. This call stores a logical file mark in the current block, writes it to disk and records its position in the GINO file locator table. The position or disk address of this final block of the file is obtained through the BSAM NOTE macro. Only the disk address of this last block is recorded. The location of the logical file mark within the last block is detected only when the data are subsequently read. Using the preceding sequence other logical files can be written on this unit and additional entries made in the file locator table.

With a multiple file unit thus created, data may be retrieved as follows: Using the file locator table, which contains the disk addresses of the last block of each logical file, the beginning of any file can be selected by pointing to a disk address one unit greater than the last block of the previous file. This selection is performed through the BSAM POINT macro. Once a particular file is located, the logical records within it may be read through a series of calls to a GINO input entry point. Read logic reverses the write logic, reading blocks from disk as necessary and reconstructing the variable length records. If the logical file mark is sensed during a read, a logical parameter in the calling sequence is set to true. This provides the capability to branch on end-of-file in the calling FORTRAN program.

This GINO package is contained within the FLEXSTAB ALIB Partitioned Data Set. Table IV-1 lists the GINO routines, the coding language, the location by member name, and the entry points. Accordingly, FORTRAN I/O statements, in the bulk of the FLEXSTAB code, have been replaced with appropriate calls to GINO entry points. In selected instances, where a single-file unit is written with a collection of unrelated simple variables, FORTRAN I/O statements were not replaced.

Although the primary motivation for the conversion of the IBM FLEXSTAB from FORTRAN I/O to GINO was to overcome the IBM OS/360 multi-file limitation, use of the GINO package substantially decreased program execution time. This decrease can be attributed to two factors: (1) an I/O package such as GINO with a narrowly defined purpose requires much less logic than the general purpose FORTRAN I/O package, and (2) the random-sequential file location technique eliminates wasteful file search time.

TABLE IV-1

GINO Package Contents

Subroutine or Control Section Name	Language Used	Member Name in ALIB PDS*	Entry Point names
GINO	FORTRAN	LOCATE	WRITE READ REWIND ENDFIL LOCATE FILESO FILESI FLUSH SINGLE
BLOCK	ASSEMBLER	RDBLK	RDBLK WRTBLK NOTE POINT RDHDR WRTHDR PUNT
STRING	ASSEMBLER	READS	WRITES READS

*NOTE: These member names are different from the subroutine/control section names. Each has been chosen to duplicate an entry point name, thereby facilitating use of the automatic call library mechanism within the Linkage Editor.

APPENDIX V

Data Flow in FLEXSTAB

FLEXSTAB consists of 14 separately executed programs. The only communication between these programs is by groups of one or more files of unformatted information. The term named file is used to refer to such a group of files. These named files are called GDTAPE, AICTAP, SDSSTP, etc., which suggest the program creating them. In the example JCL given in Appendix II and in References 1 and 2, named files are saved between programs on tape, a fact also suggested by their names. Section 3 of Reference 2 (FLEXSTAB User's Manual) provides an overview of the FLEXSTAB programs and the named files with which they communicate. It should be studied before FLEXSTAB runs are made.

The following data flow discussion, using the AIC program as an example, is representative of most FLEXSTAB programs. Figure V-1 duplicates execution JCL appearing in Appendix II; each statement is numbered for reference. The job consists of two steps, the first executing the AIC program, the second copying output files from disk to tape.

Step one (step name FLEXSTAB) extends from statement 3 to statement 28, executing the AIC load module located in FLEXSTAB.LOADMODL.LIBRARY. The data definition statements 4 through 27 define the data sets used during the step. The data sets for the AIC program (and most other FLEXSTAB programs) are of two types: those handled by FORTRAN I/O routines and those handled by GINO routines. (GINO is an I/O package used by FLEXSTAB for most unformatted I/O. See Appendix IV.) Statements 4-6 and 27 describe the data sets handled by FORTRAN I/O routines. Statement 6 describes the input data created in a prior run of the GD program.

Statements 7 through 26 define the GINO data sets used by AIC. GINO data set references are by number, and just as for FORTRAN the number is reflected in the DDNAME for the data set, e.g., UT20 at statement 7. In a fashion parallel to FORTRAN, the group of logical files referred to by a GINO data set number is called a unit. In fact, the GINO units correspond in content with similarly numbered FORTRAN units in the CDC version of FLEXSTAB. Each GINO unit is a single physical file with logical files delimited by a special GINO marker.

All GINO data sets but those in statements 22 and 23 are scratch data sets. Statement 23 (DDNAME:UT35) defines a data set destined for use in subsequent programs. Statement 22 (DDNAME:HD35) defines a directory file for UT35 --- it provides the information necessary for random access to each logical file on UT35. Both data sets must be passed.

Step two (stepname CPT) executes the FLEXSTAB COPY program, which simply copies data sets one-for-one from FORTRAN unit one to FORTRAN unit two until all data sets on unit one are copied. In this case statements 32 and 33 define HD35 and UT35, passed from jobstep one, to be files one and two of unit one, with dummy statement 31 serving as a reference for the DCB parameter. Statements 35 and 36 define data sets one and two of the destination tape, using dummy statement 34 as a reference for DSN and VOLUME parameters. On

FIGURE V-1

Execution JCL - AIC

(From Appendix II)

```
//RUN@AIC   JOB   (ACCT,INFO),'NAME',MSGLEVEL=(2,0)
//JOBLIB    DD DSN=FLEXSTAB.LOADMODL.LIBRARY,DISP=SHR
//FLEXSTAB EXEC PGM=AIC,REGION=386K
//FT06F001 DD SYSOUT=A
//FT50F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480,BUFNO=1)
//FT20F001 DD DSN=GDTAPE,DISP=OLD,DCB=(BUFNO=1)
//UT20 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT21 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT22 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT23 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT24 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT25 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT26 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT27 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT28 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT29 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT30 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT31 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT32 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT33 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT34 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//HD35 DD UNIT=SYSDA,SPACE=(TRK,(01,01)),DISP=(NEW,PASS)
//UT35 DD UNIT=SYSDA,SPACE=(TRK,(500,50)),DISP=(NEW,PASS)
//UT36 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT37 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//UT38 DD UNIT=SYSDA,SPACE=(TRK,(100,35))
//FT05F001 DD *
/*
//CPT EXEC PGM=COPY,REGION=50K
//FT06F001 DD DUMMY
//D DD UNIT=SYSDA,SPACE=(TRK,1),DCB=(RECFM=U,BLKSIZE=2048)
//FT01F001 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.HD35
//FT01F002 DD DCB=*.D,DISP=(OLD,DELETE),DSN=*.FLEXSTAB.UT35
//T DD UNIT=(TP9,,DEFER),DISP=(,KEEP),DSN=AICTAP,VOL=SER=??????
//FT02F001 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=01
//FT02F002 DD DSN=*.T,VOL=REF=*.T,DCB=*.D,DISP=(NEW,KEEP),LABEL=02
/*
```

completion of this step these data sets are direct images of HD35 and UT35. Note that both data sets on the tape have the data set name AICTAP, which is the named file output from AIC.

This pattern of JCL is typical of that for each of the FLEXSTAB programs The most general form is:

(1) Copy named file input created by a prior FLEXSTAB program from tape to scratch disk, using the COPY program.

(2) Execute the FLEXSTAB program with all data residing on disk.

(3) Copy named file data intended for a subsequent FLEXSTAB program from scratch disk to tape.

Further information on data flow in FLEXSTAB can be found in Appendix B of Reference 2 (FLEXSTAB User's Manual), especially Tables B.1-1 through B.1-22. These tables describe the contents of the named files involved with each program. However, the terminology is that of the CDC operating systems and requires some translation for IBM use. Table V-1 will assist with this translation.

Table V-1 lists the units for each program that are both (1) handled by the GINO package and (2) involved in inter-program communication. Translate all FORTRAN file names from TAPEij (the standard CDC name for unit ij) to UTij if ij appears in Table V-1 for the program, or FTijF001 if ij does not appear. Ignore all positioning requirements --- positioning is attained automatically on the IBM 360.

TABLE V-1

GINO Units and Named Files

PROGRAM	TABLE No. (Ref. 2)	GINO UNITS	NAMED FILE
GD	B.1-1	None	None
AIC	B.1-2	35	AICTAP(OUT)
		36	
ISIC	B.1-3	19,21,22,23,24	SICTAP(OUT)*
		25	EATAPE(OUT)
NM	B.1-4	19,21,22,23,24,	SICTAP(IN)*
		19,23,24	SICTP3(OUT)
		13	NMTAPE
ESIC	B.1-5	19,23,24	SICTP3(OUT)
SDSS	B.1-6	21	AICTAP(IN)
		22,23,24	SICTP3(IN)
		12,14	SDSSTP(OUT)
TH	B.1-7	1,4	SDSSTP(IN)**
		2	THTAPE(OUT)
EAPLOT	B.1-8	3	EATAPE(IN)
NMPLOT	B.1-9	13	NMTAPE(IN)
PDPLOT	B.1-10	None	None
THPLOT	B.1-11	2	THTAPE(IN)
CAIC	B.1-12	1	AICTAP(IN)
		2	AICTAP(OUT)
SLOADS	B.1-13	1	EATAPE(IN)
		2,3	SDSSTP(IN)
ALOADS	B.1-14	2,3	SDSSTP(IN)

*Named file SICTAP is the aggregate of SICTP1 and SICTP2 data. SICTP1, SICTP2, and NMTAP2 are not used as named files for the IBM version of FLEXSTAB. Data corresponding to NMTAP2 is merged with SICTP2 data from SICTAP by the NM job control language.

**This differs from the CDC version where all SDSSTP data for TH is on unit 4.

REFERENCES

1. Dornfeld, G.M.; Bhatia, K.G.; Maier, R.E.; Snow, R.N.; and Van Rossum, D.A.: A Method for Predicting the Stability Characteristics of an Elastic Airplane Vol. IV - FLEXSTAB 1.02.00 Demonstration Cases and Results. NASA CR 114715, 1974.

2. Hink, G.R.; Snow, R.M.; Bhatia, K.G.; Maier, R.E.; Bills, G.R., Henderson, D.M.; Bailey, D.C.; Dornfeld, G.M.; and D'Auria, P.V.: A Method for Predicting the Stability Characteristics of an Elastic Airplane, Vol. II - FLEXSTAB 1.02.00 User's Manual. NASA CR 114713, 1974.

3. IBM System/360 Operating System, Data Management Services Guide, GC26-3746, IBM Corporation.