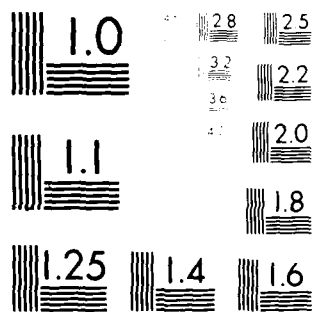


AD-A110 334 GENERAL MOTORS CORP INDIANAPOLIS IN DETROIT DIESEL A--ETC F/6 21/5
EXPERIMENTAL INVESTIGATION OF TURBINE ENDWALL HEAT TRANSFER. VO--ETC(U)
SEP 81 L D HYLTON, M S MIHELIC, E R TURNER F33615-77-C-2030
UNCLASSIFIED DDA-EDR-10363-VOL-3 AFWAL-TR-81-2077-VOL-3 NL

10-1
AD-A
10-514

END
DATE
FILMED
02-82
DTIC



MICROCOPY RESOLUTION TEST CHART
 NATIONAL BUREAU OF STANDARDS-1963-A

AD A110334

AFWAL-TR-81-2077
Volume III

LEVEL III

12

A110334

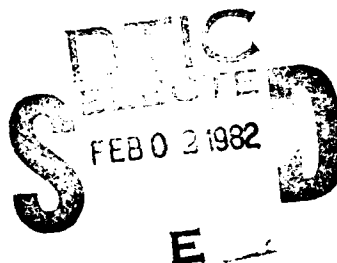


EXPERIMENTAL INVESTIGATION OF TURBINE ENDWALL HEAT TRANSFER

Volume III. Data Base System

L. D. Hylton, M. S. Mihelc, E. R. Turner, and R. E. York

Detroit Diesel Allison
Division of General Motors Corporation
P. O. Box 894
Indianapolis, Indiana 46206



September 1981

FINAL REPORT for period 1 August 1977 - 31 August 1981

Approved for public release; distribution unlimited

**AERO PROPULSION LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433**

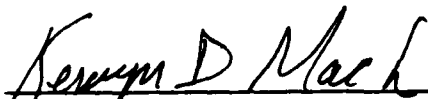
DTIC FILE COPY

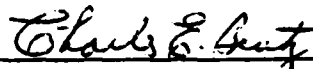
NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

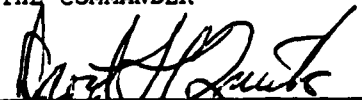
This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

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


KERVYN D. MACH
Project Engineer


CHARLES E. BENTZ, Tech Area Manager
Hot Section Technology Group

FOR THE COMMANDER


DAVID H. QUICK, Lt Col, USAF
Deputy Director, Turbine Engine Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/POTC, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWAL-TR-81-2077, Vol III	2. GOVT ACCESSION NO. AD-4000334	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) EXPERIMENTAL INVESTIGATION OF TURBINE ENDWALL HEAT TRANSFER (VOL. III, DATA BASE SYSTEM)	5. TYPE OF REPORT & PERIOD COVERED Final 1 Aug 77 - 31 Aug 81	
7. AUTHOR(s) L. D. Hylton, M. S. Mihelc, E. R. Turner, R. E. York	6. PERFORMING ORG. REPORT NUMBER EDR 10363, Vol. III	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Detroit Diesel Allison Division of General Motors Corporation Indianapolis, Indiana 46206	8. CONTRACT OR GRANT NUMBER(s) F33615-77-C-2030	
11. CONTROLLING OFFICE NAME AND ADDRESS Aero Propulsion Laboratory (AFWAL/POTC) Wright-Patterson AFB, Ohio 45433	10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS 62203F 3066-06-30	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE September 1981	
	13. NUMBER OF PAGES	
	15. SECURITY CLASS. of this report Unclassified	
	16. 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) Turbine Heat Transfer, Turbine Secondary Flow, Turbine Endwall Heat Transfer, Turbine Cascades, Heat Transfer Data Base		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Two turbine cascades were tested at simulated engine conditions to provide a data base of endwall heat transfer data. This data base is intended to be sufficiently complete to provide verification data for refined computational models developed to predict first-stage stator endwall heat transfer in ad- vanced turbine engines.		

(Continued)

DD FORM 1473

1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

029200

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Cont)

A linear, two-dimensional cascade provided the bulk of the data. This cascade provided data to separate the effects of exit Mach number, exit Reynolds number, inlet boundary layer thickness, gas-to-wall temperature ratio, inlet pressure gradients, and inlet temperature gradients. In addition, adiabatic wall temperature and inlet turbulence intensity data are available for the linear cascade runs. A computerized data base was generated. This data base, with its associated software management system, provides the user with relatively easy access to the vast amount of data generated.

A full annular, three-dimensional cascade was used to acquire data for identifying the radial pressure gradient effects. Tests in the annular cascade were run over a wide range of exit Mach and Reynolds numbers and gas-to-wall temperature ratios, all at levels typical of advanced engines.

7

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

VOLUME III

<u>Section</u>	<u>Title</u>	<u>Page</u>
1.0	Introduction.	1
2.0	System Overview	3
2.1	Data Content	3
2.2	Data Organization.	4
2.3	Management Software.	5
3.0	User's Manual	9
3.1	Software Descriptions/Usage Instructions	9
3.1.1	Main Programs	9
	Program 1--Maintenance (MAINT)	9
	Program 2--Bulk Data Listing (BLIST)	13
3.1.2	Interpolation Functions	17
	Function 1--Endwall Interpolation (ANTRP)	27
	Function 2--Exit Plane Interpolation (BNTRP)	27
3.1.3	File Access Subroutines	28
	Subroutine DBLOAD.	28
	Subroutine DBOPEN.	30
	Subroutine DBSET	30
	Subroutine DBRUN	30
	Subroutine DBSTDM.	38
	Subroutine DBDMNS.	38
	Subroutine DBSTRC.	39
	Subroutine DBFCHC.	40
	Subroutine DBSTRP.	40
	Subroutine DBFCHP.	40
	Subroutine DBDONE.	41
	Subroutine DBSAVE.	41
3.1.4	User-Written Analysis/Report Programs	41
3.2	General Operating Procedures	42
Appendix.	Replacement Data Bank Routines for IBM System Operation	47



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution	
Availability Codes	
Avail and/or	
Dist	Special
A	

LIST OF ILLUSTRATIONS

VOLUME III

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	Endwall heat transfer data base system.	2
2	Maintenance program commands.	10
3	Listing program commands.	16

LIST OF TABLES

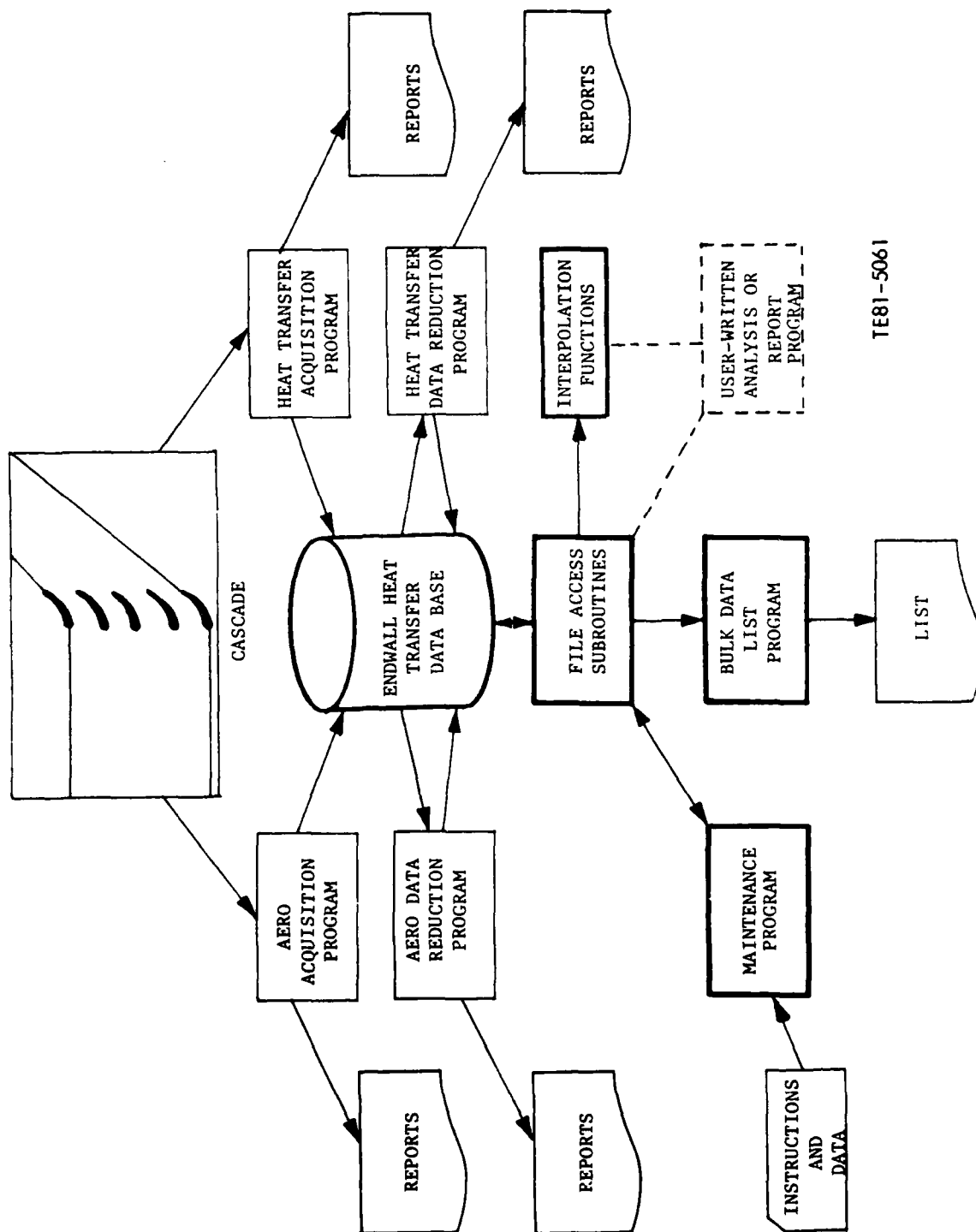
VOLUME III

<u>Table</u>	<u>Title</u>	<u>Page</u>
I	Subscripted variable definitions.	3
II	Data set GEOMETRY variable I.D. map	4
III	Data set RUNXXXXX variable I.D. map	6
IV	Sample input for maintenance program.	13
V	Sample output for maintenance program	14
VI	Sample input for listing program.	17
VII	Sample output for listing program	18
VIII	ANTRP input parameters.	27
IX	BNTRP input parameters.	27
X	Interpolation software demonstration program listing.	29
XI	Sample input for interpolation demonstration program.	30
XII	Sample output for interpolation demonstration program	31
XIII	Access subroutine functions	38
XIV	Printed output for DBLOAD	39
XV	DBSET/DBRUN options	40
XVI	Printed output for DBSAVE	42
XVII	General format for data base programs	43
XVIII	Control cards for creating a library of bank routines	43
XIX	Control cards for compilation/execution of data base programs .	44

1.0 INTRODUCTION

The purpose of the heat transfer data base system is to organize the massive data base generated under Contract F33615-77-C-2030 and to provide a convenient way of accessing the data through the computer. The data base system solves the problem of storing large volumes of printed data, since data are stored on magnetic tape. The data base is read from tape into a direct-access disk file that may be accessed in a number of ways using the software routines provided. The direct computerized access avoids the human errors associated with transferring large quantities of data from printed sheets to punched cards or other computer input formats.

Figure 1 is a schematic of the endwall heat transfer data base system at Detroit Diesel Allison. The figure highlights the parts of the data base delivered and described in this volume.



TE81-5061

Figure 1. Endwall heat transfer data base system.

2.0 SYSTEM OVERVIEW

2.1 DATA CONTENT

The contents of the data base include all raw and reduced data and geometry information used to generate the results reported in Volume II of this report. All temperatures are given in °F, all pressures in psia, and all dimensions (including relative coordinate locations) in inches. Velocities are given in ft/sec, and Reynolds numbers are based on true chord. Any other unique dimensions are stored in the list program and may be obtained from an index listing.

The listing program specifies the basis of several derived quantities in terms of subscripted variable names. Table I shows the subscript convention and defines the subscripted variable names. Definitions for the derived loss parameters also are included in Table I.

TABLE I. SUBSCRIPTED VARIABLE DEFINITIONS

Subscript convention: 1--measured inlet
2--measured exit
X--ideal exit
3--uniform mixed-out exit

<u>Variable</u>	<u>Definition</u>
PT1	Measured inlet total pressure
PS1	Average of measured inlet sidewall static pressures
TT1	Measured inlet total temperature
PTX	Ideal exit total pressure (= PT1)
PSX	Average of measured exit sidewall static pressures
TTX	Ideal exit total temperature (= TT1)
PT2	Exit cone probe total pressure
PS2	Exit cone probe static pressure
TT2	Exit probe total temperature
OMEGA2	$(PT1-PT2)/(PTX-PSX)$
DSOCP2	$\text{Ln} (TT2/TT1) - \left(\frac{\gamma-1}{\gamma} \right) \text{Ln} (PT2/PT1)$
EBAR2	$\frac{(PTX/PT2)^{\frac{\gamma-1}{\gamma}} - 1}{\frac{\gamma-1}{\gamma}}$
PT3	Calculated mixed-out total pressure
PS3	Calculated mixed-out static pressure
OMEGA3	$(PTX-PT3)/(PTX-PS3)$
DSCP3	$\text{Ln} (TTX/TT1) - \left(\frac{\gamma-1}{\gamma} \right) \text{Ln} (PT3/PT1)$
EBAR3	$\frac{(PTX/PT3)^{\frac{\gamma-1}{\gamma}} - 1}{\frac{\gamma-1}{\gamma}}$
	$(PTX/PS3)^{\frac{\gamma-1}{\gamma}} - 1.$

Most of the raw data items stored in the data base have been reviewed and edited to flag obvious bad readings due to instrumentation failures. The edited readings were flagged by replacing the bad data with the value -1. All thermocouple readings for the heat transfer endwall have been edited to include only those readings used as input to the finite element analysis program. Thermocouple data from the adiabatic endwall, the vane, and upstream total temperature rakes were also edited. Upstream total and static pressures, endwall static pressures, and exit static pressures were edited as well.

Data associated with the boundary layer pressure and temperature rakes were not edited. The data from these rakes were reliable for early runs, but indicated significant instrumentation failures as time went on. Since data from the traversing inlet probe have been supplied to provide boundary layer information, the boundary layer rake data were not edited. These data have been included in the data base, however, and should therefore be used with caution.

2.2 DATA ORGANIZATION

The data base consists of sets of data, and each set includes all data with a common identifier. The present data base consists of two types of data sets, but may be expanded as the user desires.

The first type of data set in the data base is the single data set GEOMETRY. This set includes relative coordinates for all endwall instrumentation and vane thermocouples. The data set also includes a vane profile description. Each item included in the set GEOMETRY has an assigned identification number that is used in accessing the data. Table II lists each GEOMETRY variable and its identification (I.D.) number.

TABLE II. DATA SET GEOMETRY VARIABLE I.D. MAP

ID NO.	DESCRIPTION	DIMENSIONS	
		PRIMAR.	SECONDARY
1	H.T. ENDWALL HOT SIDE T/C COORDINATES	2	53
2	H.T. ENDWALL COLD SIDE T/C COORDINATES	2	31
3	FINITE ELEMENT MODEL NODE COORDINATES	2	338
4	ADIABATIC ENDWALL T/C COORDINATES	2	90
5	ENDWALL STATIC PRESSURE COORDINATES	2	42
6	VANE T/C COORDINATES	2	113
7	ENDWALL BOUNDARY POINTS	3	85
8	VANE PROFILE COORDINATES	2	60
9	VANE TRUE CHORD	1	1
10	VANE SPACING	1	1
11	VANE LEADING EDGE RADIUS	1	1
12	VANE TRAILING EDGE RADIUS	1	1
13	VANE SETTING ANGLE	1	1
14	CASCADE THROAT HEIGHT	1	1

The second type of data set currently stored in the data base is made up of all data associated with a particular experimental run number. Forty-four such data sets, RUNXXXXX, are stored in the data base. Each variable stored for a RUNXXXXX set has an associated identification number. The general map of variables and associated identification numbers used for all RUNXXXXX data sets is shown in Table III. See Section 3 of Volume II of this report to determine the run conditions and data available for a given run number. A detailed index of the variables stored for each run number may also be generated using the program BLIST.

2.3 MANAGEMENT SOFTWARE

The management software provided with the data base includes two stand-alone programs for editing and listing data, two interpolation function subroutines, and 18 general access subroutines for data item retrieval and storage.

The stand-alone programs are MAINT and BLIST. These programs use the access subroutines to perform routine tasks. MAINT facilitates the interrogation of the data base with the option of editing, creating, or deleting data sets. BLIST provides for listing all or any subset of the data base.

The interpolation functions have been provided for the analyst to facilitate use of the data base. ANTRP provides a single value for any variable associated with the endwall region for a given (x,y). BNTRP returns a single value of any variable associated with the traversing exit probe for a given (z,y) point in the defined exit plane. The interpolation functions may be incorporated in any user-written program accessing the data base.

The file access subroutines are the building blocks for programs interacting with the data base. They are generalized subroutines for item storage and retrieval. The access subroutines are written in standard FORTRAN, and may be incorporated in any program requiring data base interaction.

TABLE III. DATA SET RUNXXXXX VARIABLE I.D. MAP

ID NO.	DESCRIPTION	DIMENSIONS	
		PRIMARY	SECONDARY
1	DATE: MO, DAY, YEAR	3	1
2	INLET TOTAL TEMPERATURE	1	1
3	IDEAL EXIT MACH NUMBER	1	1
4	INLET TOTAL PRESSURE	1	1
5	AVERAGE INLET STATIC PRESSURE	1	1
6	AVERAGE EXIT STATIC PRESSURE	1	1
7	INLET REYNOLDS NUMBER/ 10E6	1	1
8	IDEAL EXIT REYNOLDS NUMBER/ 10E6	1	1
9	AVERAGE TWALL/TCAS	1	1
10	T14 TRANSITION SECTION WALL TEMP.	1	1
11	CASCADE MASS FLOW RATE	1	1
12	INLET MACH NUMBER	1	1
13	EXPANSION RATIO	1	1
14	PRESSURE RATIO	1	1
15	INLET V/VCRTICAL	1	1
16	EXIT V/VCRTICAL	1	1
17			
18			
19			
20			
21			
22			
23			
24			
25	CORE RAKE PRESSURES	5	2
26	CORE RAKE TEMPERATURES	5	2
27	SIDEWALL STATIC PRESSURES - INLET	9	1
28	SIDEWALL STATIC PRESSURES - EXIT ROW1	9	1
29	SIDEWALL STATIC PRESSURES - EXIT ROW2	13	1
30	BOUNDARY LAYER RAKE PRESSURES	16	2
31	BOUNDARY LAYER RAKE TEMPERATURES	16	1
32	ENDWALL STATIC PRESSURES	42	1
33	VANE STATIC PRESSURES -PRESSURE SIDE	36	1
34	VANE STATIC PRESSURES -SUCTION SIDE	30	1
35			
36			
37	INLET TRAVERSE PROBE LENGTH	1	1
38	INLET TRAVERSE PROBE WIDTH	1	1
39	INLET TRAVERSE EXTERIOR ANGLE	1	1

ID NO.	DESCRIPTION	DIMENSIONS	
		PRIMARY	SECONDARY
40	INLET TRAVERSE DISTANCE FROM WALL	36	1
41	INLET TRAVERSE CORRECTED PRESSURE	36	1
42	INLET TRAVERSE TEMPERATURE	36	1
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60	EXIT PROBE SPANWISE POSITION	9	1
61	EXIT PROBE VERTICAL POSITION	25	9
62	EXIT PROBE REF. PRESSURE	25	9
63	EXIT PROBE TOTAL PRESSURE	25	9
64	EXIT PROBE STATIC PRESSURE 1	25	9
65	EXIT PROBE STATIC PRESSURE 2	25	9
66	EXIT PROBE STATIC PRESSURE 3	25	9
67	EXIT PROBE STATIC PRESSURE 4	25	9
68	EXIT PROBE TEMPERATURE	25	9
69	EXIT PROBE ANGLE	1	1
70			
71			
72			
73	GAS TEMPERATURE DURING ENDWALL READING	1	1
74	H.T. ENDWALL HOT SIDE TEMPERATURES	53	1
75	H.T. ENDWALL COLD SIDE TEMPERATURES	31	1
76	GAS TEMPERATURES DURING VANE T/C READING	1	1
77	VANE TEMPERATURES	113	1
78	ADIABATIC ENDWALL TEMPERATURE	90	1

TABLE III. (CONT)

ID NO.	DESCRIPTION	DIMENSIONS		ID NO.	DESCRIPTION	DIMENSIONS	
		PRIMARY	SECONDARY			PRIMARY	SECONDARY
79	INSULATION TEMPERATURE 1	1	1	118	EXIT AVERAGED BETA2	9	1
80	INSULATION TEMPERATURE 2	1	1	119	EXIT AVERAGED (PTX-PT)	9	1
81	GAGE BLOCK TEMPERATURE 1	1	1	120	EXIT AVERAGED (PTX-PT)/PTX	9	1
82	GAGE BLOCK TEMPERATURE 2	1	1	121	EXIT AVERAGED OMEGA	9	1
83				122	EXIT AVERAGED EBAR	9	1
84				123	EXIT AVERAGED DS/CP	9	1
85	INLET TRAVERSE MACH NUMBER	36	1	124	EXIT AVERAGED PS2/PS1	9	1
86	INLET TRAVERSE VELOCITY	36	1	125	EXIT AVERAGED V2	9	1
87				126	EXIT AVERAGED V2 AXIAL	9	1
88				127	EXIT AVERAGED V2 TANGENTIAL	9	1
89				128	EXIT AVERAGED V2 SPANWISE	9	1
90	EXIT TOTAL PRESSURE	25	9	129	EXIT AVERAGED T2 AT CONE PROBE	9	1
91	EXIT STATIC PRESSURE	25	9	130	EXIT MIXED OUT PT3	9	1
92	EXIT MACH NUMBER	25	9	131	EXIT MIXED OUT PS3	9	1
93	EXIT YAW ANGLE	25	9	132	EXIT MIXED OUT X/PAN RATIO	9	1
94	EXIT PITCH ANGLE	25	9	133	EXIT MIXED OT PS2/PS1	9	1
95	EXIT 2P (PASSAGE LOCATION)	25	9	134	EXIT MIXED OUT MACH NUMBER	9	1
96	EXIT V/V*	25	9	135	EXIT MIXED OUT BETA3	9	1
97	EXIT V/V* BASED ON PTX, PS/PROBE	25	9	136	EXIT MIXED OUT OMEGA3	9	1
98	EXIT BETA2	25	9	137	EXIT MIXED OUT EBAR3	9	1
99	EXIT OMEGA	25	9	138	EXIT MIXED OUT DS/CP	9	1
100	EXIT ERAR	25	9	139	EXIT MIXED OUT V/V*3	9	1
101	EXIT DS/CP	25	9	140	EXIT MIXED OUT V/V* BASED ON PTX, PS3	9	1
102	EXIT MASS FLUX	25	9	141	EXIT MIXED OUT V3	9	1
103	EXIT VELOCITY	25	9	142	EXIT MIXED OUT V3 AXIAL	9	1
104	EXIT AXIAL VELOCITY	25	9	143	EXIT MIXED OUT V3 TANGENTIAL	9	1
105	EXIT TANGENTIAL VELOCITY	25	9	145	EXIT AVERAGE MASS FLUX	1	1
106	EXIT SPANWISE VELOCITY	25	9	146	EXIT AVERAGE PT3	1	1
107	EXIT TOTAL TEMPERATURE AT CONE PROBE	25	9	147	EXIT AVERAGE TT3	1	1
108				148	EXIT AVERAGE MACH NUMBER	1	1
109	EXIT AVERAGED MASS FLOW	9	1	149	EXIT AVERAGE BETA3	1	1
110	EXIT AVERAGED TOTAL PRESSURE	9	1	150	EXIT AVERAGE OMEGA3	1	1
111	EXIT AVERAGED STATIC PRESSURE	9	1	151	EXIT AVERAGE EBAR3	1	1
112	EXIT AVERAGED MACH NUMBER	9	1	152			
113	EXIT AVERAGED YAW ANGLE	9	1	153			
114	EXIT AVERAGED PITCH ANGLE	9	1	154			
115	EXIT AVERAGED PT/PS	9	1	155			
116	EXIT AVERAGED V/VCRTICAL	9	1	156			
117	EXIT AVERAGED V/V* BASED ON PTX, PS/PROBE	9	1	157	H.T. ENDWALL NODAL TEMPERATURES	338	1
				158	H.T. ENDWALL NODAL HEAT FLUX	338	1
				159	H.T. ENDWALL NODAL STANTON NUMBER	338	1

3.0 USER'S MANUAL

3.1 SOFTWARE DESCRIPTIONS/USAGE INSTRUCTIONS

3.1.1 Main Programs

The main programs supplied with the data base are a maintenance program and a bulk data listing program. Both are written to read the data base from magnetic tape and create a random-access data bank disk file. Both programs read input data from punched cards and output information to a line printer in 133-character records. The general input deck structure is outlined in Section 3.2. The programs interact with the random-access data bank, processing a series of commands. The maintenance program creates a new data base magnetic tape file.

Program 1--Maintenance (MAINT)

The purpose of the maintenance program, MAINT, is to provide a means of making minor changes in the data base. It is not intended to be used for entering large volumes of data. The program generates a printed record of changes made as an audit trail, listing old and new values. The program also permits inquiry about the value of any data item without updating it.

The maintenance program begins by reading a sequential tape file of the data base and creating a random-access data bank disk file. The program then communicates only with the random-access data bank, processing a series of input commands sequentially. When the command list is exhausted, the data bank disk file is closed and copied to a new sequential tape file, forming the new data base.

Input to MAINT consists of a series of commands using key words that may be arranged to suit the user's needs. The format of the commands is shown in Figure 2. Each command line is 80 characters long. Key words are underlined, and when shown as beginning in column one, they must appear there. When blanks are underlined, they are an integral part of the key word.

All numbers are read in free format. Decimal points and exponential notations may be used, as in the FORTRAN language. Repeated values may be coded as N * #.##, where N is the number of occurrences. All free-format numbers must be separated by blanks or commas. Data may be continued on subsequent lines, and may begin on the line following the identification specification. No number field may be broken by a continuation.

The commands shown in Figure 2 can be separated into four major types. The mode command specifies the function to be performed, the set command specifies the data set to be accessed, the I.D. commands specify the data manipulation within the data set, and the end command signals the end of the maintenance session.

1	<u>INQUIRY</u>	
2	<u>UPDATE</u>	
3	<u>SET</u> NNNNNNNN	<u>CREATE</u> <u>DELETE</u> <u>CREATE</u> <u>DELETE</u>
4	<u>RUN</u> ###	
5	<u>ID</u> ## ##.##E±##	
6	<u>ID</u> ## (##) ###.##	
7	<u>ID</u> ## (*) ##.## ## ##.## .## . . .	
8	<u>ID</u> ### (##,##) ###	
9	<u>ID</u> ## (##,##) ### . . .	
10	<u>ID</u> ## (##,##) ## ## ## . . .	
11	<u>ID</u> ## (##,##) ##.##,##,## . . .	
12	<u>ID</u> ## <u>ID</u> ## <u>ID</u> ## (*) ### ## ## ## . . .	
13	<u>ID</u> ## <u>DELETE</u>	
14	<u>ID</u> ## <u>ID</u> ## . . . <u>DELETE</u>	
15	<u>END</u>	

TE81-5062

Figure 2. Maintenance program commands.

The mode command specifies the function to be performed by subsequent commands. In the UPDATE mode, the data base is changed as dictated by following commands. The program begins in the UPDATE mode. The INQUIRY mode will return the data stored under the identification following.

The mode may be changed at any point in the run, and the program remains in that mode until another mode command is encountered. In the INQUIRY mode, the program ignores any data encountered.

The second major type of command is used to specify the data set under which the following identification numbers are to be found or inserted. The commands are numbered 3 and 4 in Figure 2. Data stored in the data base are divided into data sets. All data for a given experimental run number are stored under a data set called RUNXXXXX, where the Xs represent the run number. Instrumentation coordinates and other geometry information are stored in a set called GEOMETRY. The SET command is used to enter the eight-character set name directly. The RUN command synthesizes the eight-character set name for the free-format run number given, and then performs the same action as the SET command. One of the option key words may be included (beyond column 12 on the command line) with either of these commands. DELETE causes the entire set, including all of its data, to be deleted from the data base. CREATE must be present on the set command line (beyond column 12) if a new data set is to be initiated. The default action of the SET command is designed to use a set if it exists, and to print an error message if it does not, unless CREATE is specified.

The third major type of command, the I.D. command, is shown on lines 5 through 14 of Figure 2. The I.D. command is used to update or inquire about the value of an individual data item within a data set, depending on the most recent mode specification. The item specified may be a scalar, an entire vector, an element of a vector, an entire matrix, an element of a matrix, a row of a matrix, or a column of a matrix.

When the program is in the UPDATE mode, the list of new data is entered according to the identification number and subscript specification, as shown. Since the list may be continued on subsequent lines, the program will not consider the command to be complete until it sees the beginning of the next command. If the program is running in the INQUIRY mode, any data list erroneously supplied will be ignored.

Commands numbered 5 through 11 in Figure 1 illustrate the various data manipulations possible using MAINT. Command 5 shows how a scalar would be entered in the UPDATE mode. In the INQUIRY mode, no data should be present on the command line. Command 6 shows how to refer to an element of a vector, with the subscript placed in parentheses, as shown. This command is normally used for vectors that already exist in the data base. If a new vector is being created, however, it will be initialized with maximum dimensions equal to the subscript, and the last element will be assigned the data supplied.

Command 7 shows how to update or interrogate an entire vector. The parentheses and asterisk are not really required for this case. If a new vector is being created, the list length determines the maximum dimensions it is assigned in the data base. When an existing vector is overstored, the length of

the list may not exceed the dimensions previously assigned to that vector. To increase the dimensions of the vector, the vector must first be deleted and then re-created. When the list length is less than the length of the existing vector, excess elements are not updated.

Commands 8 through 11 show the syntax for updating an element, a column, a row, or an entire matrix. The use of command 11 requires that the matrix must have been previously initialized, so that its primary dimensions may be obtained from the data base. Command 8 is used to initialize a new matrix with the maximum dimensions given. As in the case of a vector, the maximum dimensions may not be exceeded. The matrix must be deleted and re-created to expand its size. Commands 9 and 10 show how to update a row and column of a matrix, respectively. If the data list is shorter than the matrix row or column previously initialized, the update starts at the beginning of the row or column and proceeds until the list is exhausted. The remaining elements are not updated. When commands 8 through 11 are used in the INQUIRY mode, any data list is ignored.

Command 12 in Figure 1 illustrates a data base feature that facilitates entry of tabular data. The I.D.s specify the headings of the table and the data are input as ID1 (1), ID2 (1), ID3 (1), ID1 (2), ID2 (2), and so on. The dimensions must match for all I.D.s, and the length of the data list must be a multiple of the number of I.D.s specified. The subscript information specified in parentheses is not required for vectors, but must be supplied for matrices.

Commands 13 and 14 in Figure 1 show how to delete a variable or variables. Once dimensions have been set for a variable, they are permanently associated with the I.D. and may not be changed unless the I.D. is first deleted by means of one of these commands.

The final command is numbered 15 in Figure 1. The END command is used at the end of the command string to trigger the closing of the files and data base and the orderly termination of the maintenance session.

A sample input data set for MAINT is shown in Table IV. The output print generated for this sample maintenance session is shown in Table V.

The sample session illustrates several important rules for the maintenance program. In command 3, note that I.D. 1 must be deleted before its maximum dimensions can be altered. Command 4 shows that a vector size need not be specified for initialization. Command 7 shows the input convention for elements of a matrix. The primary index varies fastest. Commands 11, 12, and 13 illustrate the flexibility of free-format input.

In the inquiry mode, note the action resulting from commands 17 and 18. Command 17 calls out a matrix I.D., but only one column of the matrix is returned if (*,*) is not present on the command line. Commands 20 through 26 recall the data stored above and 23 illustrates the output for multiple I.D. lists supplied in the INQUIRY mode. Commands 27 through 31 demonstrate the use of the DELETE command and the error message printed when a deleted variable is called.

TABLE IV. SAMPLE INPUT FOR MAINTENANCE PROGRAM

```

SET EXAMPLE      CREATE
ID 1 1981
ID 1 5 26. 1731  DELETE
ID 1 (*) 7 4 1776
ID 2 (2,3) 1.
ID 2 (*) 11 21 12 22 13 23
ID 3 (4) 1.
ID 4 (4) 1.
ID 5 (4) 1.
ID 3 ID 4 ID 5 (*) 3 4      5      3      4 5
3 4 5
3 4 5
INQUIRY
RUN 122
ID 1
ID 2
ID 25
ID 25 (*,*)
SET EXAMPLE
ID 1
ID 2
ID 3 ID 4 ID 5
ID 3
ID 4
ID 5
UPDATE
ID 4 ID 5 DELETE
INQUIRY
ID 4
ID 5
END

```

Program 2--Bulk Data Listing (BLIST)

The purpose of the listing program, BLIST, is to provide a full or partial list of data base contents. The program provides for the listing of entire arrays or simply an index of variables stored for a given data set.

The BLIST software is comprised of a mainline, several "title" subroutines, and a generalized listing subroutine.

The mainline program first generates the random-access data bank from a sequential input tape file of the data base. It then proceeds to read and interpret the input data commands. The mainline processes the input commands sequentially and calls subroutines to generate the titles and variable lists specified. For each variable list, the code consists of a title subroutine call, followed by a listing subroutine call. When the END command is encountered, the mainline closes the random-access data bank. BLIST does not generate a new sequential tape file.

The "title" subroutines include subroutines GTITL and TITL1 through TITL10. GTITL retrieves the descriptive variable name corresponding to a specified GEOMETRY I.D. number. TITL1 through TITL10 contain the descriptive variable names for I.D. numbers of data sets of the type RUNXXXXX. If new variable I.D.s are added to the data base, additional title subroutines must be generated for BLIST.

TABLE V. SAMPLE OUTPUT FOR MAINTENANCE PROGRAM

OBANK LOAD COMPLETED

***SET EXAMPLE CREATE

***ID 1 1981

DATA SET EXAMPLE ID NUMBER 1 SIZE (1, 1)

ELEMENT	OLD VALUE	NEW VALUE
(1, 1)	.0	1981.00

***ID 1 1 DELETE

***ID 1 5 26. 1981

DATA SET EXAMPLE ID NUMBER 1 SIZE (3, 1)

ELEMENT	OLD VALUE	NEW VALUE	ELEMENT	OLD VALUE	NEW VALUE
(1, 1)	.0	5.00000	(2, 1)	.0	26.0000
(3, 1)	.0	1981.00			

***ID 1 (*) 7 4 1776

DATA SET EXAMPLE ID NUMBER 1 SIZE (3, 1)

ELEMENT	OLD VALUE	NEW VALUE	ELEMENT	OLD VALUE	NEW VALUE
(1, 1)	5.00000	7.00000	(2, 1)	26.0000	4.00000
(3, 1)	1981.00	1776.00			

***ID 2 (2,3) 1.

DATA SET EXAMPLE ID NUMBER 2 SIZE (2, 3)

ELEMENT	OLD VALUE	NEW VALUE
(2, 3)	.0	1.00000

***ID 2 (*,*) 11 21 12 22 13 23

DATA SET EXAMPLE ID NUMBER 2 SIZE (2, 3)

ELEMENT	OLD VALUE	NEW VALUE	ELEMENT	OLD VALUE	NEW VALUE
(1, 1)	.0	11.0000	(2, 1)	26.0000	21.0000
(1, 2)	1981.00	12.0000	(2, 2)	404820E-78	22.0000
(1, 3)	.001581E-78	13.0000	(2, 3)	1.00000	23.0000

***ID 3 (4) 1.

DATA SET EXAMPLE ID NUMBER 3 SIZE (4, 1)

ELEMENT	OLD VALUE	NEW VALUE
(4, 1)	.0	1.00000

***ID 4 (4) 1.

DATA SET EXAMPLE ID NUMBER 4 SIZE (4, 1)

ELEMENT	OLD VALUE	NEW VALUE
(4, 1)	.0	1.00000

***ID 5 (4) 1.

DATA SET EXAMPLE ID NUMBER 5 SIZE (4, 1)

ELEMENT	OLD VALUE	NEW VALUE
(4, 1)	.0	1.00000

***ID 3 ID 4 ID 5 (*) 3 4 5 3 4 5

***3 4 5

***3 4 5

DATA SET EXAMPLE ID NUMBER 3 SIZE (4, 1)

ELEMENT	OLD VALUE	NEW VALUE	ELEMENT	OLD VALUE	NEW VALUE
(1, 1)	.0	3.00000	(2, 1)	26.0000	3.00000
(3, 1)	1981.00	3.00000	(4, 1)	1.00000	3.00000

DATA SET EXAMPLE ID NUMBER 4 SIZE (4, 1)

ELEMENT	OLD VALUE	NEW VALUE	ELEMENT	OLD VALUE	NEW VALUE
(1, 1)	.0	4.00000	(2, 1)	26.0000	4.00000
(3, 1)	1981.00	4.00000	(4, 1)	1.00000	4.00000

DATA SET EXAMPLE ID NUMBER 5 SIZE (4, 1)

ELEMENT	OLD VALUE	NEW VALUE	ELEMENT	OLD VALUE	NEW VALUE
(1, 1)	.0	5.00000	(2, 1)	26.0000	5.00000
(3, 1)	1981.00	5.00000	(4, 1)	1.00000	5.00000

***INQUIRY

***RUN 122

***ID 1

DATA SET RUN 122 ID NUMBER 1 SIZE (3, 1)

ELEMENT	OLD VALUE	NEW VALUE	ELEMENT	OLD VALUE	NEW VALUE
(1, 1)	1.00000		(2, 1)	7.00000	
(2, 1)	80.0000				

***ID 2

TABLE V. (CONT)

```

DATA SET RUN 122 ID NUMBER 2 SIZE ( 1, 1)
ELEMENT OLD VALUE NEW VALUE
( 1, 1) 769.180
***ID 25

DATA SET RUN 122 ID NUMBER 25 SIZE ( 5, 2)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 59.3718 ( 2, 1) 59.8754
( 3, 1) 48.2730 ( 4, 1) 59.9444
( 5, 1) 59.9812
***ID 25 (*,*)

DATA SET RUN 122 ID NUMBER 25 SIZE ( 5, 2)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 59.3716 ( 2, 1) 59.8754
( 3, 1) 48.2730 ( 4, 1) 59.9444
( 5, 1) 59.9812 ( 2, 2) 59.8120
( 3, 2) 59.8146 ( 4, 2) 59.8518
( 5, 2) 59.9251
***SET EXAMPLE
***ID 1

DATA SET EXAMPLE ID NUMBER 1 SIZE ( 3, 1)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 7.00000 ( 2, 1) 4.00000
( 3, 1) 1776.00

DATA SET EXAMPLE ID NUMBER 2 SIZE ( 2, 3)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 11.0000 ( 2, 1) 21.0000
***ID 3 ID 4 ID 5

DATA SET EXAMPLE ID NUMBER 3 SIZE ( 4, 1)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 3.00000 ( 2, 1) 3.00000
( 3, 1) 3.00000 ( 4, 1) 3.00000

DATA SET EXAMPLE ID NUMBER 4 SIZE ( 4, 1)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 4.00000 ( 2, 1) 4.00000
( 3, 1) 4.00000 ( 4, 1) 4.00000

DATA SET EXAMPLE ID NUMBER 5 SIZE ( 4, 1)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 5.00000 ( 2, 1) 5.00000
( 3, 1) 5.00000 ( 4, 1) 5.00000
***ID 3

DATA SET EXAMPLE ID NUMBER 3 SIZE ( 4, 1)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 3.00000 ( 2, 1) 3.00000
( 3, 1) 3.00000 ( 4, 1) 3.00000
***ID 4

DATA SET EXAMPLE ID NUMBER 4 SIZE ( 4, 1)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 4.00000 ( 2, 1) 4.00000
( 3, 1) 4.00000 ( 4, 1) 4.00000
***ID 5

DATA SET EXAMPLE ID NUMBER 5 SIZE ( 4, 1)
ELEMENT OLD VALUE NEW VALUE ELEMENT OLD VALUE NEW VALUE
( 1, 1) 5.00000 ( 2, 1) 5.00000
( 3, 1) 5.00000 ( 4, 1) 5.00000
***UPDATE
***ID 4 ID 5 DELETE
***INQUIRY
***ID 4
UNABLE TO LOCATE 4 IN SET EXAMPLE
COMMAND ERROR
***ID 5
UNABLE TO LOCATE 5 IN SET EXAMPLE
COMMAND ERROR
***END
END OF UPDATES

```

The general listing subroutine, WRIT, prints the variable description obtained from the title subroutine and then retrieves and prints the data arrays if the program is in the ARRAY mode. If the program is in the INDEX mode, only the title line is printed. Subroutine WRIT is general and may be used to list new variables added to the data base, providing the title information is correctly supplied.

Input to BLIST consists of a series of commands using key words. The commands may be arranged to list all or any subset of the data base. The commands for BLIST are shown in Figure 3. Each command line is 80 characters long. Key words are underlined, and when shown as beginning in column one, they must appear there. When blanks are underlined, they are an integral part of the key word.

The commands shown in Figure 5 are of three major types. The mode command specifies the function to be performed by the program, and the set command specifies the data set or sets to be listed with optional I.D. specification. The end command signals the end of the run.

1	<u>INDEX</u>	
2	<u>ARRAY</u>	
3	<u>ALL</u>	
4	<u>RUN</u> ###	
5	<u>GEOMETRY</u>	
6	<u>RUN</u> ###	[<u>ID</u> ##]
7	<u>GEOMETRY</u>	[<u>ID</u> ##]
8	<u>END</u>	

TE81-5063

Figure 3. Listing program commands.

The mode command, INDEX, in Figure 3 will cause the program to list an index of variable names stored for the following data set names. In the ARRAY mode, variable names and their associated data arrays are listed. The program begins in the ARRAY mode and remains in that mode until another mode command is encountered. The program mode may be changed at any point in a listing series of commands.

The second major type of command is the set specification command, shown on lines 3, 4, and 5 of Figure 3. The ALL command will cause all data sets RUNXXXXX, to be listed according to the last mode specified. The RUN command will produce a list of all data associated with the specified run number. Geometry data, or an index of geometry data, will be listed if command 5 is present in the series of commands. Shown in brackets in commands 4 and 5 is the optional I.D. specification. If a list of one data array is desired, adding the I.D. number to the set specification command will produce the single array list. The I.D. option provides the flexibility for listing any subset of the data base.

Finally, the END command is used to terminate the listing session. The command triggers the orderly closing of files.

A sample input data set for BLIST is shown in Table VI. The resulting output listing appears in Table VII. Note the results of mode changes and the use of command 4 to generate special lists.

TABLE VI. SAMPLE INPUT FOR LISTING PROGRAM.

```
INDEX
ALL
GEOMETRY
ARRAY
RUN 96
RUN 122 IJ 3
RUN 122 IJ 5
GEOMETRY IJ 5
END
```

3.1.2 Interpolation Functions

Two interpolation function subroutines are provided to aid the analyst. The function ANTRP interpolates data associated with endwall instrumentation, and BNTRP interpolates data associated with the traversing exit probe. ANTRP and BNTRP interact with the data base, retrieving data as needed, and both use subroutine GNTRP to obtain an interpolated value. Subroutine GNTRP is a user-transparent routine that performs a weighted least-squares fit of the data and returns the single value for given coordinates.

Subroutines FCTNL, CHOLEP, and CBSBMP are matrix routines used by GNTRP to determine the curve-fit coefficients. They are also user-transparent.

TABLE VII. SAMPLE OUTPUT FOR LISTING PROGRAM

OBANK LOAD COMPLETED

***INDEX

***ALL

DATA	SET	T	RUN	91	ID	NUMBER	1	SIZE	(	3.	1)	DATE: MO, DAY, YEAR	DURING	ENDWALL	READING
DATA	SET	T	RUN	91	ID	NUMBER	73	SIZE	(	53.	1)	GAS TEMPERATURE	HOT	TEMPERATURE	
DATA	SET	T	RUN	91	ID	NUMBER	74	SIZE	(	31.	1)	H.1. ENDWALL	COLD	TEMPERATURE	
DATA	SET	T	RUN	91	ID	NUMBER	75	SIZE	(	113.	1)	GAS TEMPERATURE	DURING	VANE	T/C READING
DATA	SET	T	RUN	91	ID	NUMBER	76	SIZE	(	190.	1)	VANE TEMPERATURE			
DATA	SET	T	RUN	91	ID	NUMBER	77	SIZE	(	1.	1)	ADJABATIC	ENDWALL	TEMPERATURE	
DATA	SET	T	RUN	91	ID	NUMBER	78	SIZE	(	1.	1)	INSULATION	TEMPERATURE	1	
DATA	SET	T	RUN	91	ID	NUMBER	79	SIZE	(	1.	1)	INSULATION	TEMPERATURE	2	
DATA	SET	T	RUN	91	ID	NUMBER	80	SIZE	(	1.	1)	GAGE BLOCK	TEMPERATURE	1	
DATA	SET	T	RUN	91	ID	NUMBER	81	SIZE	(	1.	1)	GAGE BLOCK	TEMPERATURE	2	
DATA	SET	T	RUN	91	ID	NUMBER	82	SIZE	(	338.	1)	H.1. ENDWALL	MODE	HEAT FLUX	- BTU/HR/IN2
DATA	SET	T	RUN	91	ID	NUMBER	157	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	
DATA	SET	T	RUN	91	ID	NUMBER	158	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	
DATA	SET	T	RUN	91	ID	NUMBER	159	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	
DATA	SET	T	RUN	93	ID	NUMBER	1	SIZE	(	3.	1)	DATE: MO, DAY, YEAR	DURING	ENDWALL	READING
DATA	SET	T	RUN	93	ID	NUMBER	73	SIZE	(	53.	1)	GAS TEMPERATURE	HOT	TEMPERATURE	
DATA	SET	T	RUN	93	ID	NUMBER	74	SIZE	(	31.	1)	H.1. ENDWALL	COLD	TEMPERATURE	
DATA	SET	T	RUN	93	ID	NUMBER	75	SIZE	(	113.	1)	GAS TEMPERATURE	DURING	VANE	T/C READING
DATA	SET	T	RUN	93	ID	NUMBER	76	SIZE	(	190.	1)	VANE TEMPERATURE			
DATA	SET	T	RUN	93	ID	NUMBER	77	SIZE	(	1.	1)	ADJABATIC	ENDWALL	TEMPERATURE	
DATA	SET	T	RUN	93	ID	NUMBER	78	SIZE	(	1.	1)	INSULATION	TEMPERATURE	1	
DATA	SET	T	RUN	93	ID	NUMBER	79	SIZE	(	1.	1)	INSULATION	TEMPERATURE	2	
DATA	SET	T	RUN	93	ID	NUMBER	80	SIZE	(	1.	1)	GAGE BLOCK	TEMPERATURE	1	
DATA	SET	T	RUN	93	ID	NUMBER	81	SIZE	(	1.	1)	GAGE BLOCK	TEMPERATURE	2	
DATA	SET	T	RUN	93	ID	NUMBER	82	SIZE	(	338.	1)	H.1. ENDWALL	MODE	HEAT FLUX	- BTU/HR/IN2
DATA	SET	T	RUN	93	ID	NUMBER	157	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	
DATA	SET	T	RUN	93	ID	NUMBER	158	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	
DATA	SET	T	RUN	93	ID	NUMBER	159	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	
DATA	SET	T	RUN	94	ID	NUMBER	1	SIZE	(	3.	1)	DATE: MO, DAY, YEAR	DURING	ENDWALL	READING
DATA	SET	T	RUN	94	ID	NUMBER	73	SIZE	(	53.	1)	GAS TEMPERATURE	HOT	TEMPERATURE	
DATA	SET	T	RUN	94	ID	NUMBER	74	SIZE	(	31.	1)	H.1. ENDWALL	COLD	TEMPERATURE	
DATA	SET	T	RUN	94	ID	NUMBER	75	SIZE	(	113.	1)	GAS TEMPERATURE	DURING	VANE	T/C READING
DATA	SET	T	RUN	94	ID	NUMBER	76	SIZE	(	190.	1)	VANE TEMPERATURE			
DATA	SET	T	RUN	94	ID	NUMBER	77	SIZE	(	1.	1)	ADJABATIC	ENDWALL	TEMPERATURE	
DATA	SET	T	RUN	94	ID	NUMBER	78	SIZE	(	1.	1)	INSULATION	TEMPERATURE	1	
DATA	SET	T	RUN	94	ID	NUMBER	79	SIZE	(	1.	1)	INSULATION	TEMPERATURE	2	
DATA	SET	T	RUN	94	ID	NUMBER	80	SIZE	(	1.	1)	GAGE BLOCK	TEMPERATURE	1	
DATA	SET	T	RUN	94	ID	NUMBER	81	SIZE	(	1.	1)	GAGE BLOCK	TEMPERATURE	2	
DATA	SET	T	RUN	94	ID	NUMBER	82	SIZE	(	338.	1)	H.1. ENDWALL	MODE	HEAT FLUX	- BTU/HR/IN2
DATA	SET	T	RUN	94	ID	NUMBER	157	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	
DATA	SET	T	RUN	94	ID	NUMBER	158	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	
DATA	SET	T	RUN	94	ID	NUMBER	159	SIZE	(	338.	1)	H.1. ENDWALL	MODE	STATION NUMBER	

TABLE VII. (CONT)

[illegible]

TABLE VII. (CONT)

[illegible]

TABLE VII. (CONT)

***GEOMETRY												H.T. ENDWALL HOT SURFACE I/C COORDINATES												***																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
DATA SET	GEOMETRY	ID	NUMBER	SIZE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	

[illegible]H.T. ENDWALL NODE HEAT FLUX - BTU/HR/IN2

TABLE VII. (CONT)

[The page contains several horizontal bands of dense, repeating patterns resembling binary code or digital noise.]

TABLE VII. (CONT)

[illegible]

[illegible]

DATA SET RUN	122 ID NUMBER
ELEMENT	VALUE
1 1, 1)	.707661

DATA SET RUN	122 ID NUMBER
ELEMENT	VALUE
(1, 1)	58.4189

DATA SET GEOMETRY ID NUMBER	ELEMENT	VALUE
-----------------------------	---------	-------

ELEMENT	VALUE
1	750000E-01
2	665000
3	339500
4	339500
5	339500
6	339500
7	201000
8	332800
9	1385500
10	388200
11	388200
12	109100
13	49300
14	99600
15	29800
16	37600
17	39000
18	31800
19	31800
20	95000

IDEAL EXIT MACH NUMBER

AVERAGE INLET STATIC PRESSURE

ENDWALL STATIC PRESSURE TAP COORDINATES	
ELEMENT	VALUE

TABLE VII. (CONT)

* * * E N C

Function 1--Endwall Interpolation (ANTRP)

ANTRP is used to interpolate data associated with fixed endwall instrumentation. The formal input parameters for ANTRP are XIN, YIN, NRUN, IDX, and IDAT, as defined in Table VIII.

TABLE VIII. ANTRP INPUT PARAMETERS

<u>ANTRP Parameter</u>	<u>Definition</u>
XIN	Axial coordinate
YIN	Tangential coordinate
NRUN	Run number (data set)
IDX	Geometry I.D. of appropriate instrumentation coordinates
IDAT	Data array I.D. corresponding to the appropriate instrumentation

The function software will accept IDX values of 5 or less, corresponding to endwall instrumentation coordinates. ANTRP accepts IDAT values of 32, 74, 75, or 78 only, which correspond to endwall pressures, hot surface temperatures, cold surface temperatures, and adiabatic wall temperatures, respectively. ANTRP also checks the coordinates (XIN, YIN) to verify that they are within the valid endwall domain. The function returns a single interpolated value of the variable in C when the statement $C = \text{ANTRP}(XIN, YIN, NRUN, IDX, IDAT)$ is encountered.

Function 2--Exit Plane Interpolation (BNTRP)

The function BNTRP is used to interpolate raw and reduced data associated with the traversing exit probe. The function inputs are ZIN, YIN, NRUN, IDAT, as defined in Table IX.

TABLE IX. BNTRP INPUT PARAMETERS

<u>BNTRP Parameter</u>	<u>Definition</u>
ZIN	Radial coordinate
YIN	Tangential coordinate
NRUN	Run number (data set)
IDAT	Exit data array I.D. (array dimensioned (25, 9))

The function software accepts IDAT values of 90 through 107 only, corresponding to exit data arrays. BNTRP also checks the coordinates (ZIN, YIN) to verify that they are within the exit plane traversed for the run NRUN. The probe position coordinates are retrieved automatically by BNTRP. BNTRP returns a single interpolated value of the variable in C when the statement
C = BNTRP (ZIN,YIN,NRUN,IDAT) is encountered.

The listing of a program that demonstrates the use of ANTRP and BNTRP is given in Table X. The program is loaded with the data base access subroutines and interpolation subroutines, and reads cards specifying an interpolation function and input parameters until an END card is encountered. The program reads input parameters from the cards, synthesizes the correct call statement of ANTRP or BNTRP, and prints the resulting interpolated value for each data card. A sample input data set is shown in Table XI and the resulting output appears in Table XII. This program also uses the access subroutines DBDMNS, DBRUN, DBSET, and DBFCHC to retrieve the ANTRP input data and coordinate arrays, and displays them for convenience.

3.1.3 File Access Subroutines

The access subroutines are the building blocks of FORTRAN programs interacting with the data base. They read and write sequential magnetic tape files and create and access a random-access data bank disk file. Several routines interact with the random-access data bank, performing the specialized tasks required for item storage and retrieval. Table XIII summarizes tasks performed by the access subroutines. A description and usage instructions are given below for each subroutine, followed by comments on the general structure for data base programs in Section 3.1.4.

Subroutine DBLOAD

The calling convention for using DBLOAD is simply "CALL DBLOAD". DBLOAD is called to read the data base from a sequential tape file and generate a random-access data bank disk file. The subroutine begins with a call to DBSTRT, which initializes the master index for the random-access file on unit 29 and writes the first record of the file. Records are written to and read from the random-access data bank using the general routines DBPUT and DBGET, respectively. The subroutines DBSTRT, DBPUT, and DBGET are user-transparent. After the data bank is initialized, records are read from the sequential tape file using DBREAD, another user-transparent routine. The records are then stored in the random-access data bank, where a global directory and set directories map the location of each datum for future reference. Upon completion of the loading, DBDONE is called within DBLOAD to close the data bank and write its master index.

The subroutine DBLOAD expects to find a sequential tape file of the data base on unit 27. TAPE27 and TAPE29 must be specified in the program statement. The output resulting from a call to DBLOAD is shown in Table XIV. The number of data sets and the number of variables stored for each data set are listed. Upon successful completion of loading, the final message is printed.

TABLE X. INTERPOLATION SOFTWARE DEMONSTRATION PROGRAM LISTING

```

C      MAINLINE TO DEMONSTRATE USE OF FUNCTIONS ANTRP AND BNTRP
C      LOGICAL SCAN
      DIMENSION CARD(20),DATA(5),XCOORD(300),XDATA(150)
      DATA 'F/5/'
C
      CALL DELoad
      CALL DEOPEN
C
      1 READ(NR,2,END=99) CARD
      2 FORMAT(20A4)
      IF(SCAN(CARD,'ANTRP$')) GO TO 20
      IF(SCAN(CARD,'BNTRP$')) GO TO 30
      IF(SCAN(CARD,'END$')) GJ TJ 99
      IF(SCAN(CARD,'$')) GO TO 1
      WRITE(6,10)
      10 FORMAT(' COMMAND ERROR')
      GO TO 1
C
      ANTRP DATA
      L=1
      DO 25 I=1,5
      DATA(I)=FGTNUM(CARD,L,30,L,NF)
      IF(NF.EQ.0) GO TO 97
      25 CONTINUE
      XIN=DATA(1)
      YIN=DATA(2)
      IDX=DATA(3)
      NRUN=DATA(4)
      IDATA=DATA(5)
      CALL DERUN(NRUN,2)
      CALL DEOMNS(IDATA,MP,MS)
      CALL DEFCHC(IDATA,MP,MS,XDATA)
      CALL DESET('GEOMETRY',2)
      CALL DEFCHC(IDX,2,MP,XCOORD)
      WRITE(6,21)
      21 FORMAT(1H0,16HANTRP INPUT DATA,/,5X1HX,5X1HY,8X4HDATA)
      DO 22 I=1,MP
      K=2*I-1
      KP1=K+1
      22 WRITE(6,24) XCOORD(K),XCOORD(KP1),XDATA(I)
      24 FORMAT(2F7.2,F12.2)
      A=ANTRP(XIN,YIN,NRUN,IDX,IDATA)
      WRITE(6,26) XIN,YIN,NRUN,IDX,IDATA,A
      26 FORMAT(/,' ANTRP INPUT XIN,YIN,NRUN,IDX,IDATA=',2F7.2,3I5,/,
      1 ' OUTPUT VALUE=',F10.3)
      GO TO 1
C
      BNTRP DATA
      L=1
      DO 35 I=1,4
      DATA(I)=FGTNUM(CARD,L,80,L,NF)
      IF(NF.EQ.0) GO TO 97
      35 CONTINUE
      ZIN=DATA(1)
      YIN=DATA(2)
      NRUN=DATA(3)
      IDATA=DATA(4)
      B=BNTRP(ZIN,YIN,NRUN,IDATA)
      WRITE(6,36) ZIN,YIN,NRUN,IDATA,B
      36 FORMAT(/,' BNTRP INPUT ZIN,YIN,NRUN,IDATA=',2F7.2,2I5,/,
      1 ' OUTPUT VALUE=',F10.3)
      GO TO 1
      97 WRITE(6,98)
      98 FORMAT(' MISSING INPUT PARAMETER')
      GO TO 1
      99 CALL DEODNE
      CALL EXIT
      END

```

TABLE XI. SAMPLE INPUT FOR INTERPOLATION DEMONSTRATION PROGRAM

```

ANTRP 2.332 .5717 1 122 74
ANTRP 2.332 .5717 4 122 76
ANTRP 2.332 .5717 5 122 32
ANTRP 5.5 2.5933 1 122 74
ANTRP 5.5 2.5933 4 122 78
ANTRP 5.5 2.5933 5 122 32
ENTRP 1.200 12.0 122 90
ENC

```

Subroutine DBOPEN

DBOPEN opens an existing data bank on unit 29 by reading its first global directory record and reading its master index into core memory. Once the data bank has been opened using DBOPEN, data may be accessed using the subroutines DBSET, DBRUN, DBSTDM, DBDMNS, DBFCHC, DBFCHP, DBSTRC, and DBSTRP. The calling convention is simply "CALL DBOPEN." No printed output is generated by this routine.

Subroutine DBSET (NAME,IOPT)

Subroutine DBSET is used to access, create, or delete a data set in the random-access data bank. The subroutine calling convention is "CALL DBSET (NAME, IOPT)." NAME is the eight-character data set identifier, such as GEOMETRY or RUNXXXXX. IOPT is an integer corresponding to one of the desired action options outlined in Table XV.

Option 1 is useful for creating new data sets. Option 2 is used for accessing data sets only. Option 3 is used to ensure a unique new data set is being formed. For example, option 3 is used when the data bank is loaded using DBLOAD. Option 4 is used to purge data sets no longer needed.

Subroutine DBSET generates no printed output when it successfully performs the user's desired task. If option 2 is elected and the data set is not found, the message "DATA SET NAME NOT FOUND" is printed. If option 3 is elected and the data set is found, the message "DATA SET NAME ALREADY EXISTS" is printed. If the message "DATA BANK IS INCONSISTENT" is printed, an error occurred when the data bank was loaded. This message indicates a software problem in DBLOAD and should not appear during normal operation of the data base.

Subroutine DBRUN (NRUN,ICODE)

Subroutine DBRUN synthesizes the data set name RUNXXXXX and calls DBSET to access the data set. The calling convention is "CALL DBRUN (NRUN,ICODE)". NRUN is the experimental run number associated with a particular data set. ICODE is the user option associated with the DBSET call as outlined in Table XV.

TABLE XII. SAMPLE OUTPUT FOR INTERPOLATION DEMONSTRATION PROGRAM

CANIER INPUT DATA

X	Y	DATA
0.000	0.00	549.90
0.000	0.00	543.10
0.000	0.00	-1.00
0.000	0.00	538.80
0.000	0.00	538.70
0.000	0.23	-1.00
0.000	0.41	569.50
0.000	0.41	551.00
0.000	0.41	549.30
0.000	0.41	551.00
0.000	0.55	551.90
0.000	0.78	568.40
0.000	0.77	550.10
0.000	0.78	570.00
0.000	1.00	567.50
0.000	1.00	571.30
0.000	1.00	572.00
0.000	1.00	-1.00
0.000	1.35	588.50
0.000	1.38	588.20
0.000	1.41	588.90
0.000	1.41	591.00
0.000	1.60	605.70
0.000	1.60	602.00
0.000	1.60	-1.00
0.000	1.60	595.70
0.000	1.60	-1.00
0.000	1.60	-1.00
0.000	1.62	605.70
0.000	1.88	598.20
0.000	1.87	601.50
0.000	2.00	-1.00
0.000	2.00	607.40
0.000	2.00	598.50
0.000	2.28	612.50
0.000	2.25	-1.00
0.000	2.25	608.70
0.000	2.25	605.50
0.000	2.47	603.50
0.000	2.47	607.20
0.000	2.47	613.90
0.000	2.47	611.30
0.000	2.47	623.00
0.000	2.58	614.80
0.000	2.71	611.00
0.000	2.71	614.00
0.000	2.71	-1.00
0.000	2.97	-1.00
0.000	2.97	601.90
0.000	2.97	606.10
0.000	2.97	-1.00
0.000	2.97	646.60
0.000	2.97	657.80

ANIER INPUT XIN,YIN,NRUN,IDX, IDATA= 2.33 0.57 122 1 74
 OUTPUT VALJE= 557.549

TABLE XII. (CONT)

CANTER INPUT DATA

X	Y	DATA
5.82	3.01	722.70
5.18	3.01	718.80
4.75	3.01	714.90
4.29	3.01	717.40
3.84	3.01	717.20
3.40	3.01	718.10
2.92	3.01	-1.00
2.51	2.75	724.70
2.16	2.75	723.20
1.75	2.75	-1.00
1.29	2.75	722.70
0.84	2.75	723.30
0.40	2.75	718.90
0.00	2.75	720.80
-0.39	2.49	-1.00
-0.80	2.49	-1.00
-1.18	2.49	720.90
-1.54	2.49	-1.00
-1.88	2.49	724.00
-2.20	2.49	719.20
-2.50	2.49	729.70
-2.76	2.30	737.50
-3.00	2.30	734.50
-3.24	2.30	-1.00
-3.46	2.30	726.40
-3.67	2.23	-1.00
-3.86	2.23	747.80
-4.03	2.23	733.90
-4.20	2.23	720.10
-4.36	2.23	727.10
-4.50	2.23	718.70
-4.63	2.23	-1.00
-4.75	2.10	723.90
-4.86	2.10	-1.00
-4.96	2.10	722.10
-5.05	1.97	-1.00
-5.12	1.97	723.70
-5.19	1.97	721.90
-5.25	1.97	718.90
-5.30	1.97	718.80
-5.34	1.97	-1.00
-5.38	1.84	739.00
-5.42	1.71	-1.00
-5.45	1.71	722.30
-5.48	1.71	718.80
-5.50	1.71	-1.00
-5.52	1.71	-1.00
-5.54	1.45	-1.00
-5.56	1.45	718.90
-5.58	1.45	718.80
-5.60	1.45	726.00
-5.62	1.45	-1.00
-5.64	1.19	-1.00
-5.66	1.19	717.70
-5.68	1.19	715.60
-5.70	1.19	716.90

TABLE XII. (CONT)

0.93	1.19	-1.00
0.93	1.06	-1.00
2.41	0.93	709.00
2.00	0.93	718.20
1.73	0.93	-1.00
1.37	0.93	709.10
1.00	0.93	-1.00
0.66	0.93	713.00
2.24	0.80	723.40
2.40	0.87	-1.00
2.00	0.87	720.30
1.66	0.87	711.10
1.34	0.87	-1.00
0.64	0.87	-1.00
2.00	0.87	719.80
2.00	0.54	-1.00
2.00	0.41	-1.00
2.00	0.41	720.80
1.00	0.41	-1.00
1.00	0.41	707.50
1.00	0.41	-1.00
0.00	0.41	710.20
0.00	0.41	709.00
0.00	0.41	702.40
2.24	0.28	-1.00
2.24	0.28	724.10
2.24	0.15	737.90
2.24	0.15	734.00
1.00	0.15	720.20
1.00	0.15	700.00
1.00	0.15	705.50
0.00	0.15	704.70
-0.01	0.15	-1.00

ANTR INPUT AIN,YIN,IRON,IOA,IOATA= 2.35 0.57 122 4 7c
 OUTPUT VALUE= 723.772
 CANTRP INPUT DATA

X	Y	DATA
0.07	0.0	50.65
0.07	0.0	-1.00
1.00	0.0	58.29
1.00	0.0	57.97
2.00	0.0	58.53
0.27	0.41	58.98
0.20	0.41	58.73
1.00	0.41	58.04
1.00	0.41	57.22
2.00	0.41	56.51
0.09	0.91	58.77
1.00	0.91	58.01
1.49	0.91	57.15
1.90	0.91	54.71
2.30	0.91	50.37
1.30	1.41	58.73
1.00	1.41	55.24
1.99	1.41	52.81
2.30	1.41	49.35
2.62	1.41	44.04
1.93	1.76	52.76
2.27	1.76	-1.00
2.30	1.76	45.73
2.40	1.76	45.73

TABLE XII. (CONT)

0.73	1.70	41.20
0.73	2.10	44.01
0.73	2.10	42.77
0.73	2.10	42.56
0.73	2.10	42.57
0.73	2.10	42.53
0.73	2.47	44.02
0.73	2.47	43.07
0.73	2.47	42.79
0.73	2.47	43.17
0.73	2.47	44.52
0.73	2.97	41.47
0.73	2.97	41.34
0.73	2.97	43.37
0.73	2.97	43.23
0.73	2.97	43.22
0.73	2.97	43.23
0.73	2.97	43.02

ANTIP INPUT XIN, YIN, NEON, ICA, ICATA= 2.33 0.37 122 0 02
 CONTROL VALUE = 34.175
 OANTIP INPUT DATA

X	Y	DATA
0.00	0.00	547.90
0.00	0.00	543.10
1.77	0.00	-1.00
2.27	0.00	536.80
2.60	0.00	536.70
2.13	0.23	-1.00
0.23	0.41	569.50
0.36	0.41	553.00
1.69	0.41	549.30
1.43	0.41	551.00
1.52	0.55	552.90
1.18	0.78	566.40
1.30	0.77	560.10
1.33	0.78	570.00
0.31	1.00	567.50
1.24	1.00	570.30
1.60	1.00	572.00
1.23	1.00	-1.00
1.16	1.35	568.60
1.70	1.38	566.20
2.20	1.41	566.90
2.73	1.41	591.00
1.71	1.66	606.70
2.17	1.66	602.00
2.62	1.66	-1.00
3.08	1.66	595.70
2.09	1.86	-1.00
2.45	2.00	-1.00
3.00	1.82	605.70
3.55	1.66	596.20
2.53	1.87	601.50
3.04	2.00	-1.00
3.49	2.06	607.40
4.01	2.06	596.50
2.90	2.28	612.30
3.42	2.25	-1.00
3.59	2.25	608.70
4.40	2.25	605.80

TABLE XII. (CONT)

600.00	2.47	600.00
600.00	2.47	607.20
600.00	2.47	613.90
600.00	2.47	611.30
600.00	2.47	623.00
600.00	2.56	614.80
600.00	2.71	611.00
600.00	2.71	614.00
600.00	2.71	-1.00
600.00	2.97	-1.00
600.00	2.97	601.90
600.00	2.97	606.10
600.00	2.97	-1.00
600.00	2.97	646.60
600.00	2.97	657.80

ANTRP INPUT XIN,YIN,KNON,IDX,IBATA= 0.00 2.09 122 1 74
 CACTRPT VALUE= 600.420
 CARTRP INPUT DATA

X	Y	DATA
600.00	3.01	722.70
600.00	3.01	716.00
600.00	3.01	714.90
600.00	3.01	717.40
600.00	3.01	717.20
600.00	3.01	716.10
600.00	3.01	-1.00
600.00	2.75	724.70
600.00	2.75	723.20
600.00	2.75	-1.00
600.00	2.75	722.70
600.00	2.75	723.30
600.00	2.75	716.90
600.00	2.75	720.60
600.00	2.49	-1.00
600.00	2.49	-1.00
600.00	2.49	750.40
600.00	2.49	-1.00
600.00	2.49	721.60
600.00	2.49	724.60
600.00	2.49	719.20
600.00	2.49	726.70
600.00	2.36	737.30
600.00	2.36	734.50
600.00	2.36	-1.00
600.00	2.36	720.40
600.00	2.36	-1.00
600.00	2.23	747.60
600.00	2.23	735.90
600.00	2.23	720.10
600.00	2.23	727.10
600.00	2.23	716.70
600.00	2.23	-1.00
600.00	2.10	725.90
600.00	2.10	-1.00
600.00	2.10	722.10
600.00	1.97	-1.00
600.00	1.97	723.70
600.00	1.97	721.90
600.00	1.97	718.90
600.00	1.87	712.20

TABLE XII. (CONT)

2.21	1.97	-1.00
2.21	1.84	739.00
2.21	1.71	-1.00
2.21	1.71	722.30
2.21	1.71	716.80
2.21	1.71	-1.00
2.21	1.71	-1.00
2.21	1.45	-1.00
2.21	1.45	716.90
2.21	1.45	716.30
2.21	1.45	726.30
2.21	1.45	-1.00
2.21	1.19	-1.00
2.21	1.19	717.70
2.21	1.19	716.50
2.21	1.19	716.90
2.21	1.19	-1.00
2.21	0.93	-1.00
2.21	0.93	709.00
2.21	0.93	716.20
2.21	0.93	-1.00
2.21	0.93	709.10
2.21	0.93	-1.00
2.21	0.93	716.00
2.21	0.80	725.40
2.21	0.67	-1.00
2.21	0.67	726.30
2.21	0.67	711.10
2.21	0.67	-1.00
2.21	0.67	-1.00
2.21	0.67	715.50
2.21	0.54	-1.00
2.21	0.41	-1.00
2.21	0.41	726.80
2.21	0.41	-1.00
2.21	0.41	707.50
2.21	0.41	-1.00
2.21	0.41	710.20
2.21	0.41	709.00
2.21	0.41	702.40
2.21	0.28	-1.00
2.21	0.28	724.10
2.21	0.15	737.90
2.21	0.15	734.00
2.21	0.15	726.20
2.21	0.15	708.00
2.21	0.15	705.50
2.21	0.15	701.70
2.21	0.15	-1.00

ANTERP INPUT XIN,YIN,NRJN,IDX,IDATA= 5.50 2.59 122 4 7c.
 OUTPUT VALUE= 730.498
 GANTRP INPUT DATA...

TABLE XII. (CONT)

X	Y	DATA
0.07	0.0	58.63
0.67	0.0	-1.00
1.33	0.0	58.29
1.99	0.0	57.97
2.66	0.0	58.53
0.27	0.41	58.38
0.93	0.41	58.73
1.59	0.41	58.04
2.25	0.41	57.22
0.59	0.91	58.51
1.25	0.91	58.77
1.91	0.91	58.01
2.57	0.91	57.15
0.17	0.91	54.71
0.83	0.91	58.37
1.49	1.41	58.73
2.15	1.41	58.24
0.35	1.41	52.51
1.01	1.41	47.55
1.67	1.41	47.04
2.33	1.70	52.76
0.93	1.70	-1.00
1.59	1.70	43.75
2.25	1.70	42.83
0.59	2.10	41.26
1.25	2.10	44.01
1.91	2.10	42.77
2.57	2.10	42.56
0.17	2.10	42.37
0.83	2.10	42.53
1.49	2.47	44.02
2.15	2.47	43.07
0.35	2.47	42.79
1.01	2.47	43.17
1.67	2.47	44.52
2.33	2.97	43.47
0.93	2.97	43.64
1.59	2.97	43.37
2.25	2.97	43.23
0.59	2.97	43.22
1.25	2.97	43.23
1.91	2.97	43.02

ANTRP INPUT XIN,YIN,NRUN,IDA,IDATA= 5.50 2.59 122 5 32
 OUTPUT VALUE= 44.538
 BNTRP INPUT ZIN,YIN,NRUN,IDA= 1.20 12.00 122 50
 OUTPUT VALUE= 59.037

TABLE XIII. ACCESS SUBROUTINE FUNCTIONS

<u>Task</u>	<u>Subroutine</u>
o Load sequential tape file into random-access data bank disk file	DBLOAD
o Open existing random-access data bank	DBOPEN
o Access/create/delete a data set	DBSET
o Access/create/delete data set "RUNXXXXX"	DBRUN
o Reserve storage for an array	DBSTDM
o Retrieve dimensions of a stored array	DBDMNS
o Store/overstore an entire array	DBSTRC
o Retrieve an entire array	DBFCHC
o Store/overstore part of an array	DBSTRP
o Retrieve part of an array	DBFCHP
o Close random-access data bank	DBDONE
o Write data bank contents to sequential tape file	DBSAVE

As with the subroutine DBSET, there is no printed output for normal execution of subroutine DBRUN. User error messages for a given option are printed as generated in DBSET.

Subroutine DBSTDM (IV,MP,MS)

DBSTDM is used to reserve storage in the random-access data bank for a matrix dimensioned MP x MS. A call to DBSTDM must be preceded by a call to DBSET or DBRUN, which accesses a particular data set.

The calling convention for this subroutine is "CALL DBSTDM (IV,MP,MS)". The user-supplied arguments of DBSTDM are variable identification number (IV), matrix primary dimension (MP), and matrix secondary dimension (MS). The subroutine adds the variable I.D. to the set directory of the data set accessed, reserves storage records for a data array dimensioned MP by MS, and initializes the data array members to zero in anticipation of partial storage of the matrix. The subroutine normally generates no printed output. If the DBSTDM call is not preceded by a DBSET or DBRUN call, the message "NO SET CALLED BEFORE DBSTDM" is printed.

Subroutine DBDMNS (IV,MP,MS)

Subroutine DBDMNS retrieves the dimensions of a matrix stored in the random-access data bank. A call to DBDMNS must be preceded by a call to DBSET or DBRUN.

The subroutine call statement is "CALL DBDMNS (IV,MP,MS)," where IV is the user-supplied variable I.D. The subroutine returns the primary and secondary dimensions associated with IV in MP and MS, respectively. Printed output is generated only if an error occurs. If the DBDMNS call is not preceded by a DBSET or DBRUN call, the message "NO SET CALLED BEFORE DBDMNS" is printed. If the variable number IV is not present in the current set directory, the message "UNABLE TO LOCATE IV IN SET NAME" is printed.

TABLE XIV. PRINTED OUTPUT FOR DBLOAD

```

1  45 SETS BEING LOADED IN BANK
SET RUN 122 HAS 139 VARIABLES
SET RUN 116 HAS 121 VARIABLES
SET RUN 91 HAS 113 VARIABLES
SET RUN 96 HAS 21 VARIABLES
SET RUN 93 HAS 21 VARIABLES
SET RUN 94 HAS 43 VARIABLES
SET GEOMETRY HAS 14 VARIABLES
SET RUN 95 HAS 22 VARIABLES
SET RUN 98 HAS 43 VARIABLES
SET RUN 99 HAS 113 VARIABLES
SET RUN 105 HAS 95 VARIABLES
SET RUN 107 HAS 95 VARIABLES
SET RUN 108 HAS 95 VARIABLES
SET RUN 109 HAS 113 VARIABLES
SET RUN 111 HAS 113 VARIABLES
SET RUN 112 HAS 113 VARIABLES
SET RUN 113 HAS 113 VARIABLES
SET RUN 57 HAS 94 VARIABLES
SET RUN 61 HAS 94 VARIABLES
SET RUN 67 HAS 94 VARIABLES
SET RUN 114 HAS 113 VARIABLES
SET RUN 123 HAS 121 VARIABLES
SET RUN 125 HAS 10 VARIABLES
SET RUN 131 HAS 40 VARIABLES
SET RUN 132 HAS 51 VARIABLES
SET RUN 133 HAS 51 VARIABLES
SET RUN 149 HAS 51 VARIABLES
SET RUN 150 HAS 51 VARIABLES
SET RUN 165 HAS 51 VARIABLES
SET RUN 166 HAS 51 VARIABLES
SET RUN 168 HAS 51 VARIABLES
SET RUN 169 HAS 51 VARIABLES
SET RUN 170 HAS 51 VARIABLES
SET RUN 171 HAS 51 VARIABLES
SET RUN 172 HAS 51 VARIABLES
SET RUN 173 HAS 51 VARIABLES
SET RUN 174 HAS 51 VARIABLES
SET RUN 118 HAS 51 VARIABLES
SET RUN 124 HAS 121 VARIABLES
SET RUN 86 HAS 93 VARIABLES
SET RUN 87 HAS 93 VARIABLES
SET RUN 89 HAS 93 VARIABLES
SET RUN 176 HAS 9 VARIABLES
SET RUN 177 HAS 9 VARIABLES
SET RUN 179 HAS 9 VARIABLES
BANK LOAD COMPLETED

```

Subroutine DBSTRC (IV,MP,MS,ARRAY)

Subroutine DBSTRC is used to store or overstore in the data bank the entire matrix dimensioned MP by MS. The statement "CALL DBSTRC (IV,MP,MS,ARRAY)" must supply the arguments IV, MP, MS, and ARRAY. IV is the variable identification number (I.D.) to be found in the set directory or added to the set directory, and MP and MS are the primary and secondary dimensions of the data matrix, respectively. The data are stored in the vector ARRAY. ARRAY must contain MP x MS items or the storage will not be complete. The data should be arranged in ARRAY with the primary index varying fastest. In other words, elements of a matrix to be stored in the data bank under IV sized 3 by 2 should be ordered (1,1), (2,1), (3,1), (1,2), (2,2), (3,2) in ARRAY prior to the DBSTRC call.

TABLE XV. DBSET/DBRUN OPTIONS

<u>IOPT</u>	<u>Action</u>
1	Use data set if found, create if not found
2	Use data set if found, error if not found
3	Error if data set found, create if not found
4	Delete data set if found, return if not found

No printed output is generated by subroutine DBSTRC normally. If the call was not preceded by a DBSET or DBRUN call, an error message is printed.

Subroutine DBFCHC (IV,MP,MS,ARRAY)

Subroutine DBFCHC retrieves an entire matrix from the data bank. The user-supplied arguments are IV, MP, and MS, which are the variable I.D. and matrix primary and secondary dimensions, respectively. The subroutine returns the data in ARRAY, with elements ordered with the primary index varying fastest.

For a normal fetch of a complete matrix, no print is generated. If the user-supplied primary and secondary dimensions do not match the dimensions stored in the data bank for IV, the message "DIMENSIONS DON'T MATCH" is printed. If the variable I.D. (IV) is not present in the current set directory, "UNABLE TO LOCATE IV IN SET NAME" is printed. If the DBFCHC call was not preceded by a DBSET or DBRUN call, "NO SET CALLED BEFORE DBFCHC" is printed.

Subroutine DBSTRP (IV,IBP,IEP,IBS,IES,ARRAY,IPI)

Subroutine DBSTRP is used to store or overstore a part of a data matrix in the data bank. The calling sequence arguments IV, IBP, IEP, IBS, IES, ARRAY, and IPI are supplied by the user. IV is the variable identification number. IBP and IEP are the beginning and ending primary indices for the data to be stored. IBS and IES are the beginning and ending secondary indices. The data must be present in core in the vector ARRAY. The primary index IPI of the dimensioned vector ARRAY must also be supplied.

DBSTRP generates no print for a successful partial store of a vector. If the variable IV has not been initialized by a DBSTRC or DBSTDM call, the message "UNABLE TO LOCATE IV IN SET NAME" is printed. If the DBSTRP call was not preceded by a call to DBSET or DBRUN, the message "NO SET CALLED BEFORE DBSTRP" is printed.

Subroutine DBFCHP (IV,IBP,IEP,IBS,IES,ARRAY,IPI)

Subroutine DBFCHP is used to retrieve a part of a matrix stored in the data bank. The arguments IV, IBP, IEP, IBS, IES, ARRAY, and IPI are defined as for subroutine DBFCHP described above. The part of the matrix requested is returned in ARRAY, which has the primary dimension IPI.

Subroutine DBFCHP generates no print for normal execution. If the variable IV has not been previously stored, the message "UNABLE TO LOCATE IV IN SET NAME" is printed. If the DBFCHP call was not preceded by a DBSET or DBRUN call, an error message is printed.

Subroutine DBDONE

Subroutine DBDONE is used to close the random-access data bank file on unit 29. The routine writes the total number of records in the bank on the first record of the file and copies the master index onto the end of the file.

Subroutine DBSAVE

Subroutine DBSAVE is called when the random-access data bank has been edited and a new sequential tape file of the data base is to be created. The subroutine reads each record of each data set, one data set at a time, and writes the data to unit 28 using DBSTOR. DBSTOR is a user-transparent routine that packs the data into 80-character records and writes the records to unit 28 as they are filled.

A call to DBSAVE requires no input information. The subroutine assumes that a consistent data bank exists on unit 29 and that it may write to unit 28. Both TAPE28 and TAPE29 must be present in the program statement. The printed output generated by DBSAVE is similar to that of DBLOAD and is shown in Table XVI. The total number of data sets in the data bank is printed, and each set stored is listed with the number of variables in it specified.

3.1.4 User-Written Analysis/Report Programs

All FORTRAN programs interacting with the data base follow a simple format, which can be seen by referring to Table XVII or a listing of the maintenance program. First, the program statement should designate all file units used by the program. Then the data base must be loaded into a random-access disk file by calling DBLOAD. If the data bank exists as a random-access disk file designated unit 29, the DBLOAD call may be omitted. The random-access file is opened by calling DBOPEN, which retrieves the first global directory record.

Once the bank has been opened, it may be accessed by means of the subroutines described in Section 3.1.3 and the interpolation functions described in Section 3.1.2. To enable the print messages associated with the access subroutines, the print flag IERPRT must be set to 1 in the FORTRAN code and the following common block must be present in the program:

```
COMMON/BUFR/A(20), LREC, IA, IB, IAS, IERROR, IERPRT.
```

A program interacting with the data base should always be terminated with a call to DBDONE to preserve the master index for the disk file. If a new sequential file of the data base needs to be written, the DBDONE call should be followed by a DBSAVE call.

TABLE XVI. PRINTED OUTPUT FOR DBSAVE

```

END OF UPDATES
1 45 SETS BEING STORED FROM BANK
SET RUN 122 HAS 139 VARIABLES
SET RUN 115 HAS 121 VARIABLES
SET RUN 91 HAS 113 VARIABLES
SET RUN 96 HAS 21 VARIABLES
SET RUN 93 HAS 21 VARIABLES
SET RUN 94 HAS 43 VARIABLES
SET GEOMETRY HAS 14 VARIABLES
SET RUN 95 HAS 22 VARIABLES
SET RUN 98 HAS 43 VARIABLES
SET RUN 99 HAS 113 VARIABLES
SET RUN 105 HAS 95 VARIABLES
SET RUN 107 HAS 95 VARIABLES
SET RUN 103 HAS 95 VARIABLES
SET RUN 109 HAS 113 VARIABLES
SET RUN 111 HAS 113 VARIABLES
SET RUN 112 HAS 113 VARIABLES
SET RUN 113 HAS 113 VARIABLES
SET RUN 57 HAS 94 VARIABLES
SET RUN 61 HAS 94 VARIABLES
SET RUN 67 HAS 94 VARIABLES
SET RUN 114 HAS 113 VARIABLES
SET RUN 123 HAS 121 VARIABLES
SET RUN 125 HAS 113 VARIABLES
SET RUN 131 HAS 43 VARIABLES
SET RUN 132 HAS 51 VARIABLES
SET RUN 133 HAS 51 VARIABLES
SET RUN 149 HAS 51 VARIABLES
SET RUN 150 HAS 51 VARIABLES
SET RUN 165 HAS 51 VARIABLES
SET RUN 166 HAS 51 VARIABLES
SET RUN 163 HAS 51 VARIABLES
SET RUN 164 HAS 51 VARIABLES
SET RUN 170 HAS 51 VARIABLES
SET RUN 171 HAS 51 VARIABLES
SET RUN 172 HAS 51 VARIABLES
SET RUN 173 HAS 51 VARIABLES
SET RUN 174 HAS 51 VARIABLES
SET RUN 119 HAS 51 VARIABLES
SET RUN 124 HAS 121 VARIABLES
SET RUN 85 HAS 93 VARIABLES
SET RUN 97 HAS 93 VARIABLES
SET RUN 89 HAS 93 VARIABLES
SET RUN 176 HAS 9 VARIABLES
SET RUN 177 HAS 9 VARIABLES
SET RUN 179 HAS 9 VARIABLES
OBANK STORAGE COMPLETED

```

3.2 GENERAL OPERATING PROCEDURES

The access subroutines, interpolation software, and utility subroutines SCAN, COMPAR, FGNUM, and GETNUM should all be loaded into a permanent library. Sample control cards used to generate a library named BANK are shown in Table XVIII. The last subroutine in the source file should be DBDONE. If the source code is stored as a disk file, named FILE, the control cards should be preceded by an appropriate control card to access the file, and the FTN control card should be changed to read FTN,I=FILE,R=2. In such a case the source deck is omitted.

TABLE XVII. GENERAL FORMAT FOR DATA BASE PROGRAMS

Col. 7

PROGRAM NAME(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE27,TAPE28,TAPE29)

COMMON/BUFR/A(40),LREC,IA,IB,IAS,IERROR,IERPRT

CALL DBLOAD

CALL DBOPEN

IERPRT = 1

[
FORTRAN code
]

CALL DBDONE

CALL DBSAVE

END

TABLE XVIII. CONTROL CARDS FOR CREATING A LIBRARY OF BANK ROUTINES

[JOB CARD]

FTN,R=2.

REWIND, LGO.

REQUEST,BANK,*PF.

EDITLIB.

CATALOG,BANK,RP=999.

7/8/9 [eor]

[
Source deck
]

7/8/9 [eor]

LIBRARY(BANK,NEW)

ADD (*+DBDONE,LGO)

FINISH.

ENDRUN.

6/7/8/9 [eof]

The general control card format for running MAINT, BLIST, or other data base FORTRAN programs is shown in Table XIX. Again, if the program source code is stored as a disk file called SFILE, an appropriate control card must precede the FTN card and the FTN card should be written FTN,I=SFILE,R=2. The source deck would then be omitted.

TABLE XIX. CONTROL CARDS FOR COMPILATION/EXECUTION OF DATA BASE PROGRAMS

[JOBCARD (INCLUDES TAPE SPECIFICATION)]

FTN,R.

ATTACH,BANK.

LIBRARY,BANK.

VSN,TAPE27=nnn,TAPE28=mmm.

REQUEST,TAPE27,PE,NORING.

REQUEST,TAPE28,PE,RING.

LGO.

7/8/9 [eor]

[Source deck]

7/8/9 [eor]

[Commands/data]

6/7/8/9 [eof]

The program is loaded with the data base library BANK, and serial numbers for tapes 27 and 28 are given. Unit 27 is specified NORING to guard against writing over the input data base. The source deck is followed by a sequence of commands for the programs MAINT and BLIST or input data for other programs.

To implement the data base system on an IBM 370 computer, several software changes must be made. First, the file access subroutines and utility subroutines listed in the Appendix must replace the CDC subroutines with the same names. Also, the error messages printed by the subroutines DBDMNS, DBFCHC,

DBSTRP, and DBFCHP must be changed to print the data set name correctly. The CDC versions of the routines store the entire set name as NA(1). The IBM versions store the set name in A(1) and A(2). Thus, write statements that display the set name must write A(1) and A(2), rather than NA(1). The formats should correspondingly be changed from A10 to 2A4 to display the set name.

The matrix routines used by the interpolation software contain DOUBLE PRECISION statements, which must be changed to REAL * 8 for equivalent IBM function.

Finally, the main programs MAINT and BLIST must be changed. The program statements must be removed, and the IF EOF statements must be replaced with the IBM option END = NNN for read statements.

APPENDIX

REPLACEMENT DATA BANK ROUTINES FOR
IBM SYSTEM OPERATION

```

C
C      SUBROUTINE DBLOAD
C      LOAD THE BANK FROM SEQUENTIAL FILE
C      DIMENSION B(40),NA(40),C(40),NC(40),NAME(2),D(1000)
C      COMMON /BUFR/ A(40),LREC,IA,IB,IAS,IERROR,IERPRT
C      EQUIVALENCE (A(1),B(1),NA(1)),(C(1),NC(1))
C      CALL DBSTRT
C      FIND OUT HOW MANY SETS ARE IN THE BANK
C      CALL DBREAD(1,NSETS)
C      WRITE(6,40) NSETS
40  FORMAT('1',15,' SETS BEING LOADED IN BANK')
C      LOOP THROUGH THE SETS
C      IF (NSETS.EQ.0) GO TO 550
C      DO 500 ISET=1,NSETS
C      CALL DBREAD(2,NAME)
C      CALL DBREAD(1,NVAR)
C      WRITE(6,50) NAME,NVAR
50  FORMAT('1 SET ',2A4,' HAS',15,' VARIABLES')
C      CALL DBSET(NAME,1)
C      DO 400 IVAR=1,NVAR
C      CALL DBREAD(1,IDNO)
C      CALL DBREAD(1,IPI)
C      CALL DBREAD(1,ISI)
C      ISIZE=IPI*ISI
C      IF (ISIZE.GT.1000) GO TO 150
C      CALL DBREAD(ISIZE,D)
C      CALL DBSTRC(IDNO,IPI,ISI,D)
C      GO TO 400
150 CALL DBSTRM(IDNO,IPI,ISI)
C      DO 200 JSI=1,ISI
C      IF (IPI.GT.1000) GO TO 170
C      CALL DBREAD(IPI,D)
C      CALL DBSTRP(IDNO,1,IPI,JSI,JSI,D,1000)
C      GO TO 200
170 DO 190 JPI=1,IPI,1000
C      ILIM=MINO(JPI+999,IPI)
C      CALL DBREAD(ILIM+1-JPI,D)
C      CALL DBSTRP(IDNO,JPI,ILIM,JSI,JSI,D,1000)
190 CONTINUE
200 CONTINUE
400 CONTINUE
500 CONTINUE
550 CONTINUE
C      CLOSE OUT BUFFER
C      CALL DBDONE
C      WRITE(6,600)
600  FORMAT('0BANK LOAD COMPLETED')
C      RETURN
C      END

C
C
C
C      SUBROUTINE DBSTRT
C      CALL DBSTRT TO INITIALIZE A NEW BANK
C      DIMENSION B(40),NA(40)
C      COMMON /BUFR/ A(40),LREC,IA,IB,IAS,IERROR,IERPRT
C      EQUIVALENCE (A(1),B(1),NA(1))
C      DEFINE FILE 29(15000,40,U,IDUM)
C      IERROR=0
C      IERPRT=1
C      DO 10 I=1,40
10  NA(I)=0
C      LREC=1
C      NA(38)=LREC
C      CALL DBPUT(1,A)
C      IA=0

```

```

      IB=0
      RETURN
      END
C
C
C
C
      SUBROUTINE DBOPEN
C      CALL DBOPEN TO OPEN AN EXISTING BANK
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1))
      DEFINE FILE 29(15000, 40,U,ICUM)
      IERROR=0
      IERPRT=1
      IA=0
      IB=0
      IAS=0
      CALL DBGET(1,A)
      LREC=NA( 39)
      RETURN
      END
C
C
C
C
      SUBROUTINE DBSET(NAME,IOPT)
      CALL SET TO DEFINE A DATA SET
C
C
C
C
C
C
C
C
      IOPT VALUES AND MEANING
      1      USE      IF FOUND.      CREATE IF NOT FOUND.
      2      USE      IF FOUND.      ERROR IF NOT FOUND.
      3      ERROR    IF FOUND.      CREATE IF NOT FOUND.
      4      DELETE   IF FOUND.      RETURN IF NOT FOUND.
C
      DIMENSION NAME(2)
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1))
      IERROR=0
      IA=0
      IB=0
      ID=1
      10 CALL DBGET(ID,A)
      IDS=ID
      DO 100 I=1, 31,5
      IF (NA(I).EQ.0) GO TO 150
      IF (NA(I).NE.NAME(1)) GO TO 100
      IF (NA(I+1).EQ.NAME(2)) GO TO 200
      100 CONTINUE
      ID=NA( 40)
      IF (ID.NE.0) GO TO 10
C      ADD A NEW DIRECTORY ENTRY
      LREC=LREC+1
      NA( 40)=LREC
      CALL DBPUT(IDS,A)
      ID=LREC
      DO 90 I=1, 40
      90 NA(I)=0
      CALL DBPUT(ID,A)
      GO TO 10
      150 CONTINUE
      IF (IOPT.EQ.2) GO TO 1000
      IF (IOPT.EQ.4) GO TO 250
      NA(1)=NAME(1)
      NA(I+1)=NAME(2)
      LREC=LREC+1

```

```

      NA(I+4)=LREC
      CALL DBPUT(10,A)
      IA=LREC
      IAS=IA
      NA(1)=NAME(1)
      NA(2)=NAME(2)
155  DO 160 I=5, 40
160  NA(I)=0
      NA(40)=IA
      CALL DBPUT(1A,A)
      GO TO 250
200  IF (IOPT.E2.4) GO TO 300
      IA=NA(I+4)
      IAS=IA
      CALL DBGET(1A,A)
      IF (NA(1).NE.NAME(1)) GO TO 1050
      IF (NA(2).NE.NAME(2)) GO TO 1050
      IF (IOPT.E2.3) GO TO 1100
      IF (IOPT.E2.4) GO TO 155
250  RETURN
300  NA(I)=-1
      CALL DBPUT(10,A)
      IA=0
      GO TO 250
1000 IF (IERPRT.NE.0) WRITE (6,1010) NAME
1010 FORMAT ('**DATA SET ',2A4,' NOT FOUND*')
      IERROR=1
      RETURN
1050 IF (IERPRT.NE.0) WRITE (6,1060)
1060 FORMAT ('***DATA BANK IS INCONSISTANT*')
      IERROR=2
      CALL EXIT
1100 IF (IERPRT.NE.0) WRITE (6,1110) NAME
1110 FORMAT ('**DATA SET ',2A4,' ALREADY EXISTS*')
      IERROR=3
      IA=0
      RETURN
      END

```

C
C
C
C
C

```

C      SUBROUTINE DBDOONE
      CLOSE A BANK BEFORE END OF JOB OR OVERLAY
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1))
      IERROR=0
      IA=0
      CALL DBGET(1,A)
      NA(38)=LREC
      CALL DBPUT(1,A)
      RETURN
      END

```

C
C
C

```

C      SUBROUTINE DBDUMP
      DUMP TO PRINT THE CONTENTS OF THE BANK
      DIMENSION F(4)
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),NA(1),B(1))
      IB=1

```

```

WRITE (6,123)
123 FORMAT ('DUMP OF DATA BANK')
CALL DBGET(1,A)
LREC=NA(38)
LREC=MAX0(LREC,LREC)
DO 1000 IR=1,LREC
WRITE (6,124) IR
124 FORMAT ('RECORD',I5)
CALL DBGET(IR,A)
DO 200 I=1,40,4
J=1
IE=I+3
JE=J+3
CALL MVC(F,16,A('))
DO 100 K=1,16
IF (CLC(F,K,'A',1,1)*CLC(F,K,'I',1,1).LE.0.) GO TO 100
IF (CLC(F,K,'J',1,1)*CLC(F,K,'R',1,1).LE.0.) GO TO 100
IF (CLC(F,K,'S',1,1)*CLC(F,K,'Z',1,1).LE.0.) GO TO 100
IF (CLC(F,K,'O',1,1)*CLC(F,K,'9',1,1).LE.0.) GO TO 100
IF (CLC(F,K,' ',1,1).EC.C.) GO TO 100
50 CALL MVC5(F,K,' ',1,1)
100 CONTINUE
WRITE (6,125) I,(NA(K),K=I,IE),F,(A(K),K=J,JE)
125 FORMAT (I4,4I8,3X,4A4,3X,4G14.7)
200 CONTINUE
WRITE(1,124) IR
1000 CONTINUE
RETURN
END

C
C
C
C
SUBROUTINE DBGET(IREC,BUF)
C GENERALIZED DIRECT ACCESS DISK READ FOR DATABANK
DIMENSION BUF(40)
READ (29*IREC) BUF
RETURN
END

C
C
C
SUBROUTINE DBPUT(IREC,BUF)
C GENERALIZED DIRECT ACCESS DISK WRITE FOR DATABANK
DIMENSION BUF(40)
WRITE (29*IREC) BUF
RETURN
END

C
C
C
SUBROUTINE DBRUN(NRUN,ICODE)
C SYNthesize A DATASET NAME FOR A RUN NUMBER "RUN---NNN"
DIMENSION NAME(2)
DATA NAME /'RUN ', ' '
CALL INCORE(NAME(2),4)
WRITE(99,99) NRUN
99 FORMAT (I4)
CALL DBSET(NAME,ICODE)
RETURN
END

C
C
C
SUBROUTINE DBSAVE
C STORE THE BANK ON A SEQUENTIAL FILE

```

```

DIMENSION B(40),NA(40),C(40),NC(40),NAME(2),D(1000)
COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IEMPRT
EQUIVALENCE (A(1),B(1),NA(1)),(C(1),NC(1))
REWIND 29
CALL DBOPEN
C FIND OUT HOW MANY SETS ARE IN THE BANK
NSETS=0
ID=1
10 DO 100 I=1,31,5
CALL DBGET(ID,C)
IF (NC(I).EQ.0) GO TO 150
IF (NC(I).EQ.-1) GO TO 100
C SEE IF IT HAS ANY VARIABLES
IC=NC(I+4)
CALL DBGET(IC,C)
IF (NC(5).EQ.0) GO TO 100
NSETS=NSETS+1
100 CONTINUE
CALL DBGET(ID,C)
IF (NC(40).EQ.0) GO TO 150
ID=NC(40)
GO TO 10
150 WRITE(6,160) NSETS
160 FORMAT('1',I5,' SETS BEING STORED FROM BANK')
C CALL DBSTOR(1,NSETS)
C
C LOOP THROUGH THE SETS
IF (NSETS.EQ.0) GO TO 550
ID=1
180 DO 500 I=1,31,5
CALL DBGET(ID,C)
IF (NC(I).EQ.0) GO TO 550
IF (NC(I).EQ.-1) GO TO 500
NAME(1)=NC(I)
NAME(2)=NC(I+1)
C COUNT THE VARIABLES
NVAR=0
IC=NC(I+4)
ICS=IC
220 CALL DBGET(IC,C)
DO 240 J=5,33,4
IF (NC(J).EQ.0) GO TO 250
IF (NC(J+1)*NC(J+2).LE.0) GO TO 240
NVAR=NVAR+1
240 CONTINUE
IC=NC(40)
IF (IC.NE.ICS) GO TO 220
250 IF (NVAR.EQ.0) GO TO 500
WRITE(6,260) NAME,NVAR
260 FORMAT('1 SET ',2A4,' HAS',I5,' VARIABLES')
C CALL DBSTOR(2,NAME)
CALL DBSTOR(1,NVAR)
C
C LOOP THROUGH AGAIN STORING VARIABLES
CALL DBSET(NAME,2)
IC=ICS
300 CALL DBGET(IC,C)
DO 400 J=5,33,4
IF (NC(J).EQ.0) GO TO 410
IDND=NC(J)
IPI=NC(J+1)
ISI=NC(J+2)
IF (IPI*ISI.LE.0) GO TO 400

```

```

C      CALL DBSTOR(1,IDNO)
      CALL DBSTOR(1,IPI)
      CALL DBSTOR(1,ISI)
C
      ISIZE=IPI*ISI
      IF (ISIZE.GT.1000) GO TO 310
      CALL DBFCHC(IDNO,IPI,ISI,D)
C
      CALL DBSTOR(ISIZE,D)
C
      GO TO 400
310  DO 390 JSI=1,ISI
      IF (IPI.GT.1000) GO TO 320
      CALL DBFCHP(IDNO,1,IPI,JSI,JSI,D,1000)
C
      CALL DBSTOR(IPI,D)
C
      GO TO 390
320  DO 350 JPI=1,IPI,1000
      ILIM=MIN0(JPI+999,IPI)
      CALL DBFCHP(IDNO,JPI,ILIM,JSI,JSI,D,1000)
C
      ISZ=ILIM+1-JPI
      CALL DBSTOR(ISZ,D)
C
350  CONTINUE
390  CONTINUE
400  CONTINUE
      IC=NC(40)
      IF (IC.NE.IC5) GO TO 300
C
410  CONTINUE
      END OF VARIABLE LOOP
C
500  CONTINUE
      CALL DBGET(ID,C)
      IF (NC(40).EQ.0) GO TO 550
      ID=NC(40)
      GO TO 180
C
      END OF SET LOOP
C
550  CONTINUE
      CLOSE OUT BUFFER
      CALL DBSTOR(0,0)
      WRITE(6,600)
600  FORMAT ('OBANK STORAGE COMPLETED')
      CALL DBDONE
      REWIND 28
      RETURN
      END

```

```

C
C
C
C      SUBROUTINE DBSTOR(N,D)
      MACHINE DEPENDENT ROUTINE FOR OUTPUT TO SEQUENTIAL FILE
      DIMENSION D(N),CARD(20)
      DATA IC /0/,IOUT /28/
      IF (N.EQ.0) GO TO 100
C
      COPY D INTO CARD
      ID=0
10   ID=ID+1
      IC=IC+1
      CARD(IC)=D(ID)
      IF (IC.LT.20) GO TO 50
      WRITE(IOUT,20) CARD
20  FORMAT (20A4)

```

```

      IC=0
50  IF (ID.EQ.N) RETURN
      GO TO 10
100 IF (IC.EQ.0) RETURN
      ICP=IC+1
      GO 110 I=ICP,20
110 CARD(I)=0
      WRITE(IOUT,20) CARD
      IC=0
      RETURN
      END

```

```

C
C
C SUBROUTINE DBREAD(N,D)
C   MACHINE DEPENDENT ROUTINE FOR INPUT FROM SEQUENTIAL FILE
      DIMENSION D(N),CARD(20)
      DATA IC /20/,IIN /27/
      ID=0
      IF (IC.NE.20) GO TO 50
10  READ(IIN,20) CARD
20  FORMAT (20A4)
      IC=0
50  ID=ID+1
      IC=IC+1
      D(ID)=CARD(IC)
      IF (ID.EQ.N) RETURN
      IF (IC.EQ.20) GO TO 10
      GO TO 50
      END

```

```

C
C
C
C
C FUNCTION FGNUM(CARD,NSTART,NEND,K,NCHAR)
      DIMENSION CARD(1)
      FGNUM=0.
      CALL GETNUM(CARD,NSTART,NEND,XDUM,K,NCHAR)
      IF (NCHAR.NE.0) FGNUM=XDUM
      RETURN
      END

```

```

C
C
C SUBROUTINE GETNUM(CARD,NSTART,NEND,XNUM,K,NCHAR)
      DIMENSION CARD(1)
      DIMENSION FIELD(5)
      REAL MINUS
      DATA BLANK/' ',Z/'Z',DEC/'.',PLUS1/'+',PLUS2/'+',MINUS1/'-',MINUS2/'-',EQUAL1/'=',EQUAL2/'=',
1  PLUS2/'+',MINUS1/'-',EQUAL1/'=',EQUAL2/'=',
2  COMMA/',',E/'E',D/'D'
      NCHAR=0
      XNUM=0.0
      KI=NEND+1
      IF (NEND.LT.NSTART) GO TO 300
      N1=NSTART
      N2=NEND-1
      I=NSTART-1
      K=I+1
      IF (CLC(CARD,K,Z,1,1).GT.0.0.OR.CLC(CARD,K,DEC,1,1).EQ.0.0.OR.
1  CLC(CARD,K,PLUS1,1,1).EQ.0.0.OR.CLC(CARD,K,PLUS2,1,1).EQ.0.0
2  OR. CLC(CARD,K,MINUS1,1,1).EQ.0.0) GO TO 100
      IF (N1.GT.N2) GO TO 300
      DO 50 I=N1,N2
      K=I+1
      IF ((CLC(CARD,I,BLANK,1,1).EQ.0.0.OR.CLC(CARD,I,EQUAL1,1,1).EQ.0.0

```

```

1. OR. CLC(CARD,I,EQUAL2,1,1).EQ.0.0. OR. CLC(CARD,I,CUMMA,1,1).EQ.0.0)
2. AND.
1 (CLC(CARD,K,Z,1,1).GT.0.0. OR. CLC(CARD,K,DEC,1,1).EQ.0.0. OR.
2 CLC(CARD,K,PLUS1,1,1).EQ.0.0. OR. CLC(CARD,K,PLUS2,1,1).EQ.0.0)
3. OR. CLC(CARD,K,MINUS,1,1).EQ.0.0) GO TO 100
50 CONTINUE
GO TO 300
100 CONTINUE
IDEC = 0
IF (CLC(CARD,K,DEC,1,1).EQ.0.0) IDEC = 1
IF (K.EQ.NEND) GO TO 200
KPI = K+1
DO 150 I=KPI,NEND
KL=I
IF (CLC(CARD,I,Z,1,1).GT.0.0. OR. (CLC(CARD,I,DEC,1,1).EQ.0.0. AND.
1 IDEC.EQ.0)) GO TO 125
GO TO 200
125 CONTINUE
IF (CLC(CARD,I,DEC,1,1).EQ.0.0) IDEC = 1
150 CONTINUE
KL=KL+1
200 CONTINUE
NCHAR = MINO(20,KL-K)
DO 250 I=1,5
FIELD(I) = BLANK
250 CONTINUE
CALL MVC5(FIELD,21-NCHAR,CARD,K,NCHAR)
CALL INCORE(FIELD,20)
READ(5,1000) XNUM
IF (KL.GE.NEND) GO TO 300
N1=KL
N2=NEND-1
DO 400 I=N1,N2
IF (CLC(CARD,I,BLANK,1,1).NE.0.0) GO TO 410
400 CONTINUE
GO TO 300
410 IF (CLC(CARD,I,E,1,1).NE.0.0. AND. CLC(CARD,I,D,1,1).NE.0.0)
X GO TO 300
IPI=I+1
DO 420 K=IPI,NEND
IF (CLC(CARD,K,BLANK,1,1).NE.0.0) GO TO 430
420 CONTINUE
GO TO 300
430 IF (CLC(CARD,K,Z,1,1).LE.0.0. AND. CLC(CARD,K,PLUS1,1,1).NE.0.0. AND.
X CLC(CARD,K,PLUS2,1,1).NE.0.0. AND. CLC(CARD,K,MINUS,1,1).NE.0.0)
X GO TO 300
KL=K+1
IF (K.EQ.NEND) GO TO 440
KPI=K+1
DO 450 I=KPI,NEND
KL=I
IF (CLC(CARD,I,Z,1,1).LE.0.0) GO TO 440
450 CONTINUE
KL=KL+1
440 CONTINUE
NCHAR1=MINO(20,KL-K)
DO 460 I=1,5
FIELD(I)=BLANK
460 CONTINUE
CALL MVC5(FIELD,21-NCHAR1,CARD,KL -NCHAR1,NCHAR1)
CALL INCORE(FIELD,20)
READ(5,1001) IEXP
XNUM=XNUM*10.**IEXP
NCHAR=NCHAR+NCHAR1+1
300 K=KL

```

```

      RETURN
1000 FORMAT(F20.0)
1001 FORMAT(I20)
      END

```

C
C

```

      LOGICAL FUNCTION SCAN(CARD,WORD)
      LOGICAL *1 CARD(1),WORD(1),DECLAR
      LOGICAL COMPAR
      DATA DOLAR /'$/
      SCAN=.FALSE.
      DO 10 LEN=1,80
      IF (CLC(WORD,LEN+1,DECLAR,1,1).EQ.0.) GO TO 20
10 CONTINUE
      RETURN
20 SCAN=COMPAR(CARD,1,80,LEN,WORD,J)
      RETURN
      END

```

C
C

```

      LOGICAL FUNCTION COMPAR(C,I,L,N,CHAR,J)
      LOGICAL *1 C(1),CHAR(1),BLK
      DATA BLK /' '/
      NUM=N
      DO 5 K=1,N
      IF (CLC(CHAR,K,BLK,1,1).EQ.0.) NUM=NUM-1
5 CONTINUE
      IL=L-1
      IF (I.GT.IL) GO TO 10
      DO 15 K=1,IL
      IF (CLC(C,K,BLK,1,1).EQ.0.) GO TO 15
      K1=K
      K2=1
      IL1=IL-1
20 IF (K2.GT.N) GO TO 25
      IF (CLC(CHAR,K2,BLK,1,1).NE.0.) GO TO 30
      K2=K2+1
      GO TO 20
30 IL1=IL1+1
40 IF (K1.GT.IL1) GO TO 15
      IF (CLC(C,K1,BLK,1,1).NE.0.) GO TO 45
      K1=K1+1
      GO TO 40
45 IF (CLC(C,K1,CHAR,K2,1).NE.0.) GO TO 15
      K1=K1+1
      K2=K2+1
      GO TO 20
15 CONTINUE
      GO TO 10
25 J=K1
      COMPAR=.TRUE.
      RETURN
10 J=1
      COMPAR=.FALSE.
      RETURN
      END

```