AD845521

COAST GUARD

SHIP DESIGN AND MAINTENANCE COMPUTER PROGRAM

ICEBREAKER PROPULSION SYSTEM SIMULATION (DIESEL - ELECTRIC)

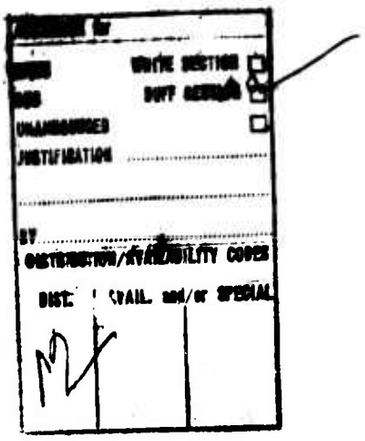
MAY 1968

DDC FILE COPY

DDC
RECEIVED
JAN 8 1969
B

U.S. COAST GUARD HEADQUARTERS
OFFICE OF ENGINEERING
NAVAL ENGINEERING DIVISION

PROGRAM NO. ENE-11



TCD New source of AD 845 520;
CG, WDC, Off of Eng (404 342).
Lower echelons x-ref'd to
keep reports together.

Allan Kuhn
9 Jan 68

DEPARTMENT OF TRANSPORTATION

U.S. Coast Guard Headquarters
Washington, D. C. 90591

SHIP DESIGN AND MAINTENANCE COMPUTER PROGRAM

Title: ICEBREAKER PROPULSION SYSTEM SIMULATION (DIESEL-ELECTRIC)

APPROVAL

The contents of this documentation are in accordance with good engineering design and computer programming practices.

TITLE	BRANCH/PROJECT	SIGNATURE	DATE
Project Officer/Engineer	ENE-4A	J. W. Lovin	27 May '68
Computer Aided Design Coordinator	ENE-4A	J. W. Lovin	27 May '68
Chief	ENE-4A	R. B. Schwinghamer	27 May '68

This documentation is correct and complete and the program is coordinated with related work in other branches.

	SIGNATURE	DATE
Coordinator Computer Application - Naval Engineering	R. B. Schwinghamer	27 May '68

LIMITATION OF LIABILITY

The U. S. Government will incur no liability resulting from use of this computer program by organizations or persons other than the U. S. Coast Guard.

This document is subject to special export controls and transmittal to foreign governments or foreign nationals may be made only with prior approval of the Commandant (Eg), U. S. Coast Guard.

(See form 173)

Computer Documentation

TABLE OF CONTENTS

Section	Description	Page
PART I	IDENTIFICATION	1
1.	Title	1
2.	Brief Description	1
3a.	Author	1
b.	Date	1
c.	Documentation	1
4.	Language	1
5.	Machine	1
6.	Security Classification	1
7.	Estimated Running Time	1
PART II	PURPOSE AND METHOD	2
1.	Purpose	2
2.	Method and Theory	2
	a. Method	2
	b. Theory	2
	c. References	3
PART III	RESTRICTIONS	4
1.	General Restrictions	4
2.	Tapes Required	4
3.	Nonstandard Hardware	4
4.	Maximum Array Sizes	4

Section	Description	Page
PART IV	NONSTANDARD OPERATING INSTRUCTIONS	5
1.	Special Operating Instructions	5
	a. General	5
	b. Documentation Procedures	5
	c. Problem Solving	6
	d. Program Alterations	8
	e. Plotter Output	8
2.	Error Corrections	9
3.	Restart Instructions	9
4.	Program Changes	9
PART V	DATA PREPARATION	13
1.	Input Data	13
2.	Output Data from Sample Run	89
APPENDIX		101
1.	Symbol Table	102
2.	System Flow Charts	112
3.	Mainline Programs and Subroutine Flow Charts	116
4.	Mainline Programs and Subroutine Documentation	151
5.	Program Listings	201

PART I - IDENTIFICATION

1. Title: Icebreaker Propulsion System Simulation (Diesel-Electric).

2. Brief Description: This program solves numerically the set of simultaneous non-linear differential equations which describe the dynamic behavior of the complete propulsion system (including vessel dynamics) of a diesel-electric icebreaker during maneuvering. The program also contains a means of subjecting the propeller to ice impact loadings of the user's making. At the option of the user, input data can readily be printed for documentation purposes and output data can be plotted.

3a. Author: Jack W. Lewis, Lieutenant, U.S. Coast Guard

b. Date: May 1968

c. Documentation: Prepared by Computer Usage Development Corporation, Washington Scientific Office.

4. Language: FORTRAN IV (IBM 1130 version)

5. Machine: IBM 1130 with disk (plotter desirable but not necessary).

6. Security Classification: UNCLASSIFIED

7. Estimated Running Time: Will vary directly with length of maneuver and time step employed. Using time steps of .08 seconds will require approximately 7 minutes computation time for 15 seconds of ship time. After initial propulsion plant transients have disappeared, generally within 10 to 20 seconds after the start of a maneuver, the time step can be increased to .15 to .20 seconds thus reducing computation time.

PART II - PURPOSE AND METHOD

1. Purpose: The purpose of this program is to simulate transient conditions within a diesel electric propulsion system for an icebreaker. The program will simulate various conditions to which an icebreaker may be subject, including crash reversals and accelerations under free route conditions, ice ramming, and subjection of propellers to ice encounters. Use of the program will permit designers to realistically study the effect of the transient conditions on design parameters and to optimize design selection of equipment for new icebreakers as required by the U. S. Coast Guard.

2. Method and Theory:

   a. Method: The construction of mathematical models for complete ship propulsion systems involves writing and solving a large number of coupled non-linear differential equations. This computer model for solving these equations describes as accuarately as possible the complete dynamics of all components of the icebreaker's propulsion system. Numerical integration techniques are used to simultaneously solve this complete set of equations and to present the resulting data as a time history of key variables both on a printer and graphic plotter.

   b. Theory: The theoretical approach used in developing the model was to derive the equations and program them for computer solution. The computer solution for the USCGC GLACIER was compared to sea trial data for this ship. Complete theory and results of initial test runs is presented in references 1, 2 and 3 below.

c. References:

ICEBREAKER PROPULSION SYSTEM SIMULATION, U.S. Coast Guard Publication, 1967, by Lt. J. Lewis, U.S. Coast Guard, F. Scoville, and LCDR E.J. Lecourt, U.S. Coast Guard.

1) Phase I - Simulation of USCGC GLACIER.

2) Phase II - Dynamics of AC-DC Icebreaker Propulsion System.

3) Phase III - Propulsion System Behavior During Propeller - Ice Interactions.

PART III - RESTRICTIONS

1. General Restrictions: See other restrictions.

2. Tapes Required: None

3. Nonstandard Hardware: A Calcomp plotter is desirable but not necessary to run a simulation. If a plotter is not available, the GRAPH mainline routine and its four subroutines -GRID, DATA, VAPLO, and PRINT - must not be compiled nor any attempts be made to execute the GRAPH routine (data switch 2 must therefore remain OFF).

4. Maximum Array Sizes: The only size restriction of any concern is the limit of 600 records that can be stored on the disk during a run. A disk write interval time step must therefore be such as to not cause more than 600 records to be written on the disk.

PART IV - NONSTANDARD OPERATING INSTRUCTIONS

1. Special Operating Instructions:

a. General: This program has been written in a manner which places the engineer in the computation loop. This type of operation was choosen in lieu of batch processing because of 1) the experimental nature of the problem frequently requires decision making during runs and 2) the IBM 1130 computer is quite adaptable to this type of operation. Considerable use is made of the keyboard console for the desicion making process.

b. Documentation Procedures: The sole purpose of the mainline program GENIS is documentation of each run. Although it is somewhat expensive in machine time to swap mainline programs in and out of core storage, the advantages of being able to document each run overides.

Documentation is handled both in a decision making process at the keyboard console and by GENIS itself. In other words, certain input data and headers will automatically be printed as output wheras other input may or may not be printed depending on the console selector switches turned on. On the very first run switch 9 should be turned on in order that the serial number disk file can be initialized. This switch should then be left in the off position until it is desired to reinitialize the serial number file. Printed instructions will appear on the keyboard console to guide the user through each run.

c. Problem Solving: The primary problem involved in the solution of differential equations by numerical methods is the selection of the time step. If the time step is too large the numerical integration method may lose the solution and "blow up." If the step is too small solution time can be prohibitive. As a general rule it is best to make several runs at first with different time steps to compare accuracy and solution time. During the first parts of any maneuver (usually the first 10 to 20 seconds) experience with this program indicates time steps around .06 to .08 seconds to be about right.

It is possible to interupt the solution of a problem at any time by turning on console selector switch zero. In addition to stopping the process of integration, turning this switch on calls in subroutine FINIS which allows the user to select a number of options concerning what he wants to do next. For example if it is desired to change the time step, switches 3 and 5 would be turned on (make sure switch zero is off) and the push button marked "START" depressed. A message "( ) TIME STEP" will be typed at which time the user types in the new value of the time step. The key marked "EOF" (end of field) is depressed and the program continues the problem solution using the updated value of the time step.

Experience has shown that the time constant associated with the armature circuit governs the size of the time step. Generally speaking, after the armature circuit current stops fluctuating greatly (usually after 10 to 20 seconds of a maneuver) the time constant of this circuit can be increased without introduction of errors. Increasing the time constant allows larger time steps and hence decreases solution

time. Increasing this time constant can be readily accomplished by increasing the value of the armature circuit inductance (XLA). A value of .008 henries allowed a solution to be accomplished with time steps of .20 seconds in one case with no noticable difference in the solution obtained by not changing the inductance (.0002 henories) and keeping the time step at .08 seconds.

It is a simple programming problem to change subroutine FINIS to include additional options desired by the user . Switches 8 through 15 are still available for this purpose.

Subroutine CNTRL was written with the idea that the user could add a library of maneuvers and data changes through the use of the variable ICNTR. For example a "Flank Power Ahead to Flank Power Astern" maneuver could be called using ICNTR equal to 2 and setting predetermined initial conditions into the program in addition to the ordered bridge controller position.

It is often easier to obtain the necessary initial conditions for a particular maneuver by using the simulation. For example, suppose it is desired to conduct a "Flank Power Ahead to Flank Power Astern" maneuver. The initial conditions at "Flank Power Ahead" are needed. These could be calculated roughly by hand, fed into the simulation and then by keeping the simulation at "Flank Power Ahead" allow the simulator to find the correct initial conditions. Once found, a new set of initial condition cards can be created by using switch 1 option of FINIS.

d. Program Alterations: Altering the program has already been briefly mentioned above and will be further expanded upon here. An attempt was made to place components that may possibly be changed by a user into separate subroutines. This should facilitate changing components from one design to another. For example to change diesel engines, subroutine QDEV would have to be changed. If idle speeds or max speeds are changed it would also be necessary to make changes in subroutine CON2 which is the bridge controller command generator subroutine.

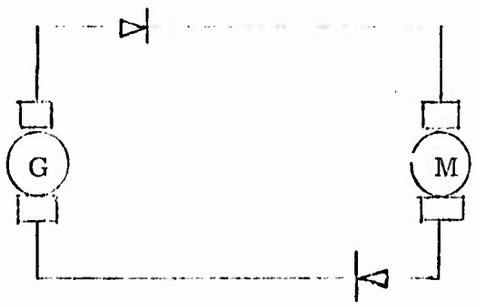
It is hoped that the program contains sufficient comments so that a user with the aid of this documentation can make major or minor changes to the program to solve his specific problem

e. Plotter Output: Mainline program GRAPH is called into operation by turning on selector switch 2 after a program interrupt. Plotting and labeling the solution data is the purpose of this program. Before the data is plotted the user must supply the program with three essential pieces of information, i.e., 1) which variables the user wishes to be plotted 2) whether or not a grid and labeling is to be constructed and 3) any titling information. Data on variables to be plotted can be entered in any order; however, leave no spaces and insure the data is right justified. Drawing the grid requires time; therefore, during parametric investigation it is often best to first draw one grid and then use this for the remaining problems. Titling information may be omitted by depressing the key "EOF" three times. After the data is plotted mainline program GENIS is reloaded automatically.

2. Error Corrections: None

3. Restart Instructions: None

4. Program Changes: Each of the subroutines are written in modular form and in FORTRAN. This is done to allow the user options of changing the routines to simulate special situations. As an example of the type of changes that might be made let us suppose the armature circuit looked as follows:



This circuit would only allow unidirectional current flow. Following is a modified subroutine CALVR which would simulate this situation - i. e. if EC > EG then curvature circuit = 0. 0.

```
      SUBROUTINE CALVR                                                  CALV0030
C*****************      MODIFIED VERSION     ***************************
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    CALV0040
C                                                                       CALV0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             CALV0060
     1 XJMP,XJDG,FIMAX,RA,RG,CL,XK1,QBRG,CSHFR,CWM,PCMS,XLA,            CALV0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          CALV0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           CALV0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   CALV0100
      COMMON DT,CT,T,TEND,TIMSC                                         CALV0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  CALV0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      CALV0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TCALV0140
     1 ENDO                                                             CALV0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,CALV0160
     1QY,QERM,AIHAT,QP,QPI,QI,TP,XPRQI,R,VAR,XKW,CVK                    CALV0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZCP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLACALV0180
     1SF,AMPTR,XGEN                                                     CALV0190
      COMMON CFY,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,EMIRV,TAUA,TAUSCALV0200
      COMMON B(64),BB(64),E(32),EE(32)                                  CALV0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  CALV0220
      COMMON IELO,JELO,KELO,LELO,MELO                                   CALV0230
      COMMON SPARE(5) ISPAR(5)                                          CALV0240
      COMMON CAD,DELTA,RV                                               CALV0250
      COMMON RBA(25),RBMX,BLOC1,DBB12,BLOC2,DBB21                       CALV0260
      COMMON EQ,EK,DCT,DCR,DSH,CICE,TICE                                CALV0270
      COMMON CK1,CK2,CK3                                                CALV0280
C                                                                       CALV0290
      EQUIVALENCE (XNGR,SPARE(2)),(XNGRD,SPARE(3)),(FMI,SPARE(1))      CALV0300
      EQUIVALENCE(IPLUG,ISPAR(2))                                       CALV0310
      EQUIVALENCE (FLOW,SPARE(5))                                       CALV0320
C                                                                       CALV0330
C                          COMPUTE ACTUAL AND TIME RATE OF BRIDGE       CALV0340
C                          LEVER POSITION                               CALV0350
      CALL CON1                                                         CALV0360
C                          COMPUTE ORDERED GENERATOR FIELD VOLTAGE,     CALV0370
C                          DIESEL SPEED, AND DIESEL SPEED RATE          CALV0380
      CALL CON2                                                         CALV0390
C                                                                       CALV0400
C                                                                       CALV0410
C                          CALCULATE DIESEL ERROR SPEED AND TIME RATE   CALV0420
      XNE=XNGR-XNPM                                                     CALV0430
      XNEDO=XNGRD-XNPMD                                                 CALV0440
C                          CALCULATE DEVELOPED DIESEL TORQUE            CALV0450
      CALL QDEV                                                         CALV0460
C                                                                       CALV0470
C                          ADJUST DEVELOPED DIESEL TORQUE FOR           CALV0480
C                          FRICTION TORQUE                              CALV0490
      QDE=QDE-(QBRG+CWM*XNPM)*XGEN                                      CALV0500
      AMPTR=TRNFG*FGI/2.-TRNSF*AI/(2.*XGEN)                             CALV0510
```

```
C                                                                       CALV0520
C                          CALCULATE GENERATOR FLUX, INDUCTANCE, AND    CALV0530
C                          TIME CONSTANT                                CALV0540
      CALL FLUXG                                                        CALV0550
      EG=60.*CFG*XNPM                                                   CALV0560
C                          CALCULATE MOTOR FIELD FLUX, INDUCTANCE, AND  CALV0570
C                          TIME CONSTANT                                CALV0580
      CALL FLUXM                                                        CALV0590
      EC=60.*CFM*XNP                                                    CALV0600
      IF(EG-EC) 400,250,250
  400 AI=0.0
      EB=EC
      QG=0.0
      QM=0.0
      GO TO 251
  250 QG=7.04*CFG*AI
      QM=7.04*CFM*AI
C                          CALCULATE MOTOR FIELD CURRENT                CALV0630
      E0=EG-(RG/XGEN)*AI                                                CALV0640
C                          CHECK FOR REVERSE DIRECTION                  CALV0650
  251 IF(THETA) 5,6,6
    5 GO TO(7,8),IPLUG                                                  CALV0670
C                          SAVE INITIAL TIME AND MOTOR FIELD CURRENT    CALV0680
    7 TTN=T                                                             CALV0690
      FMIIC=FMI                                                         CALV0700
      IPLUG=2                                                           CALV0710
    8 ABLE=1./EXP((T-TTN)/TAUA)                                         CALV0720
      BAKER=1./EXP((T-TTN)/TAUB)                                        CALV0730
      FMI=FMIIC+FMIRV/(TAUA-TAUB)*(TAUA*(1.-ABLE)-TAUB*(1.-BAKER))      CALV0740
      GO TO 34                                                          CALV0750
    6 IF(EB-C1) 10,10,20                                                CALV0760
   20 AIHAT=XK1*(EB-C1)                                                 CALV0770
      FMI=FIMAX-FIP                                                     CALV0780
      GO TO 30                                                          CALV0790
   10 AIHAT=0.                                                          CALV0800
   30 IF (FMI-84.0) 31,32,32                                            CALV0810
   31 FMI=84.0                                                          CALV0820
   32 IF (FMI-FIMAX) 34,34,33                                           CALV0830
   33 FMI=FIMAX                                                         CALV0840
   34 QFRM=CSHFR*ABS(XNP)*XNP                                           CALV0850
C                          CALCULATE SHIP'S RESISTANCE, WAKE AND THRUST CALV0860
C                          DEDUCTIONS                                   CALV0870
      CALL HULL                                                         CALV0880
C                          CALCULATE ADVANCE VELOCITY OF PROPELLER      CALV0890
      VA=WDED*VS                                                        CALV0900
C                          CALCULATE IDEAL PROPELLER TORQUE AND         CALV0910
C                          THRUST PER PROPELLER                         CALV0920
      CALL PROP                                                         CALV0930
C                          CALCULATE TOTAL PROPELLER THRUST             CALV0940
      QP=QPI                                                            CALV0950
      TP=TDED*XPROI*XKK                                                 CALV0960
C                          CHECK IF SHIP HAS ENCOUNTERED ICE AND SET    CALV0970
C                          ICE TORQUE TO ZERO IF NOT                    CALV0980
      IF(CT-TICE)50,51,51                                               CALV0990
```

```
   50   QI=0.0                                                    CALV1000
        GO TO 52                                                  CALV1010
C                                                                 CALV1020
   51 CALL QICE          CALCULATE ICE TORQUE OF PROPELLER        CALV1030
C                                                                 CALV1040
   52   CALL STATE       CALCULATE DERIVATIVES                    CALV1050
        RETURN                                                    CALV1060
        END                                                       CALV1070
```

PART V- DATA PREPARATION

1. Input Data: The following indicates in detail how each data card is prepared and from where the data is gathered or calulated.

   Each page or section of the report discusses individual data cards, preceded by an example data card prepared for the simulation discussed in references 1, 2 and 3.

PROPELLER DATA

IZB DAR PDR D EO EK DCT

Data Card 1

FORMAT I5, 5x, 6F10.4

ELEMENTS

IZB = Number of Propeller Blades (3 for USCGC GLACIER, 4 for M-13 hull)

DAR= Propeller developed area ratio (non dimensional)

$$DAR = \frac{EAR}{1.067 - 0.229 \times P/D}$$

where DAR = Developed area ratio

EAR = Expanded area ratio

P/D = Propeller Pitch=Diameter ratio.

One or the other (DAR or EAR) will be available from propeller characteristics.

PDR = Propeller pitch/diameter ratio (P/D or H/D) - Non dimensional available from propeller characteristics.

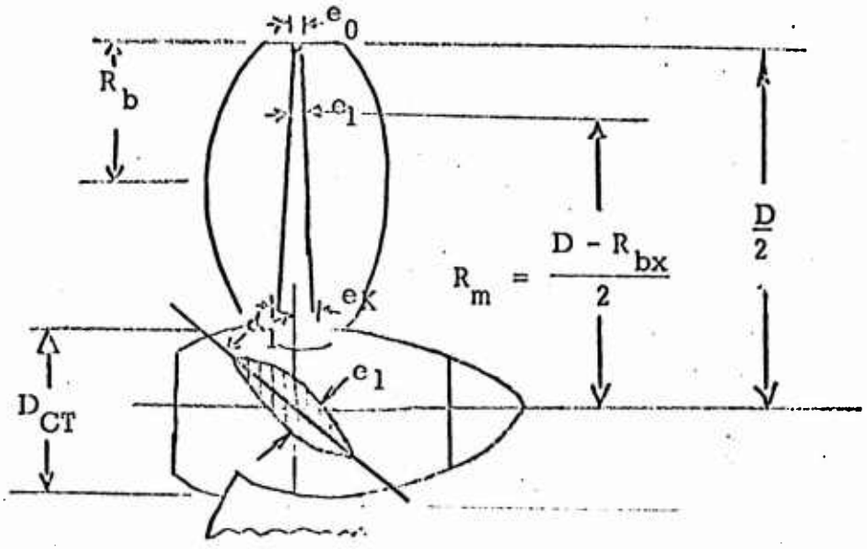
D = Diameter of propeller from tip to tip (Feet).

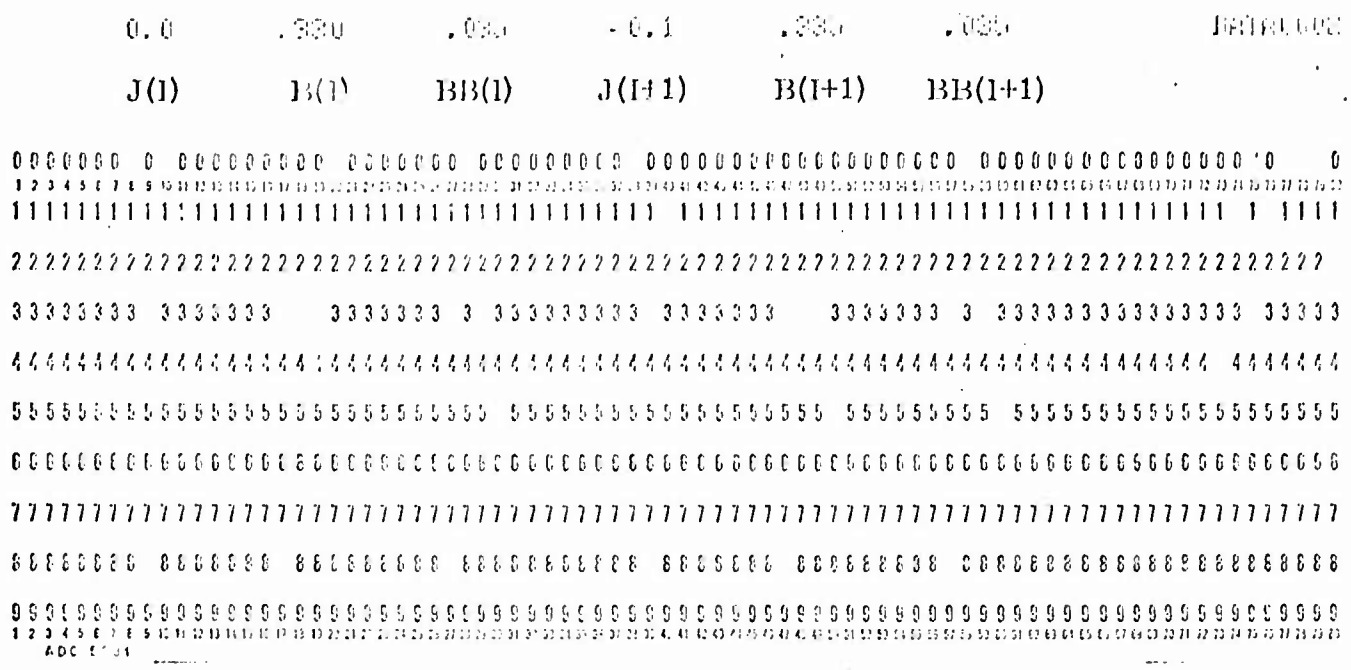
EO* = Thickness of propeller blade at tip (Feet).

* See figure following the next page for dimensional locations on propeller.

EK = Thickness of propeller blade at root (Feet)

DCT = Diameter of hub (Feet).


e_0
R_b
e_1
$R_m = \frac{D - R_{bx}}{2}$
$\frac{D}{2}$
e_K
e_1
D_{CT}



Data Cards 2 - 33

FORMAT 2 (10x, 2E10.4)

ELEMENTS

J = Non dimensional speed

$$J = \frac{V_a}{n_p D}$$

where V_a = speed of advance in feet/second

n_p = propeller speed in revolutions per second

D = as previously defined.

B = Thrust coefficient (non-dimensional) = K_T

$$K_T = \frac{T}{\rho (n_p)^2 D^4}$$

where ρ = mass density of water $\left(\frac{\text{lbs-sec}^2}{\text{ft}^4}\right)$

T = Thrust in lbs.

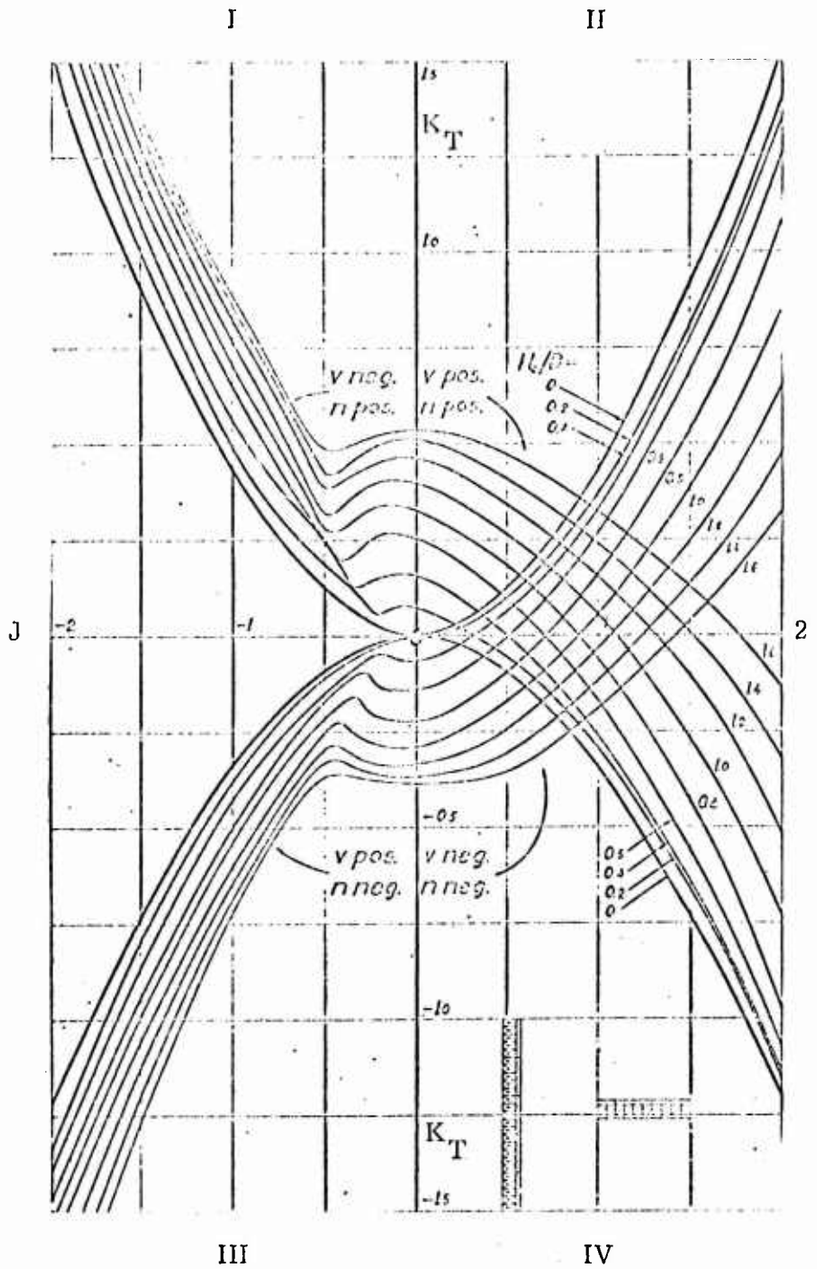
BB = Torque coefficient (non dimensional) = K_Q

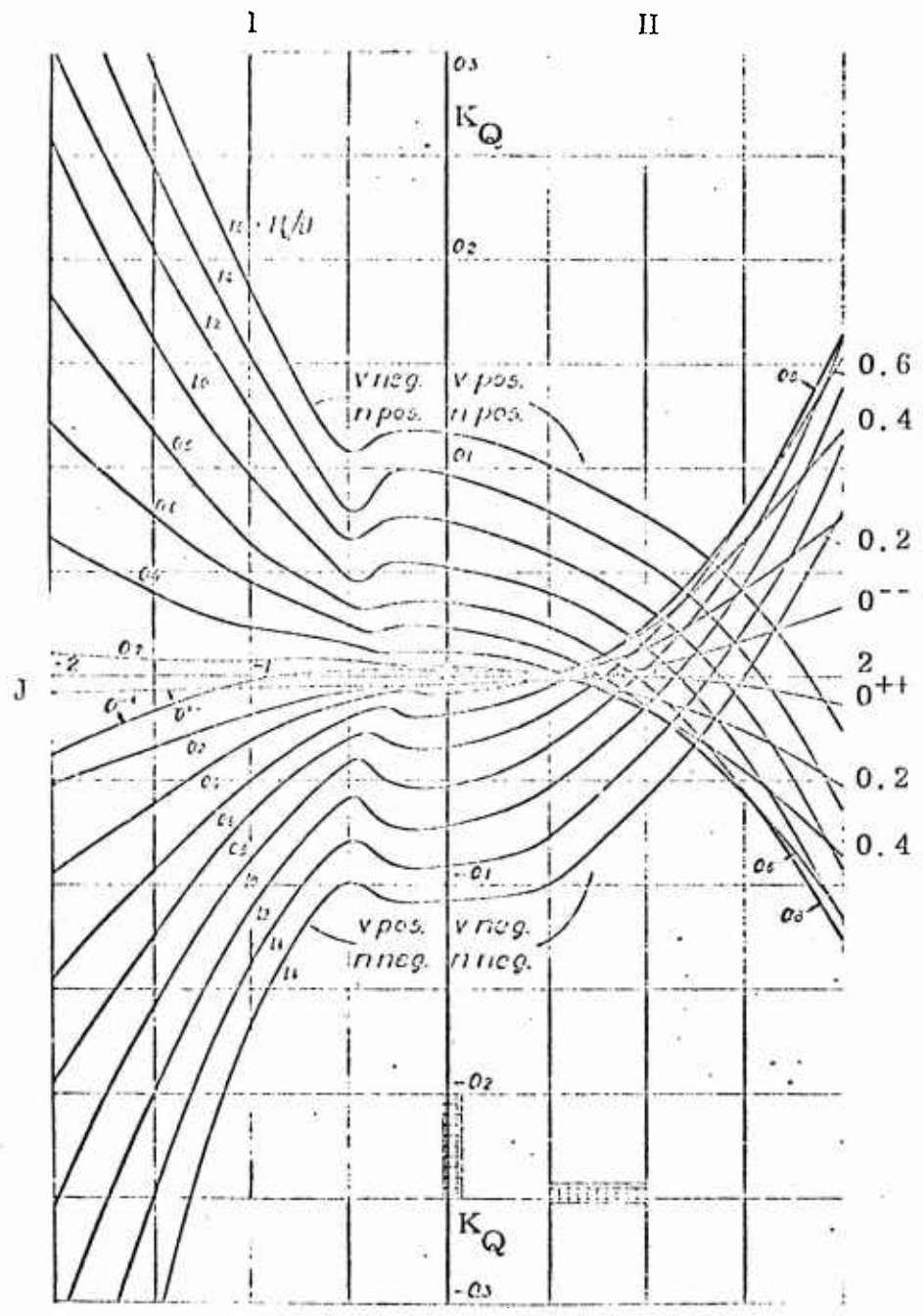
$$K_Q = \frac{Q_p}{\rho (n_p)^2 D^5}$$

Q_p = Torque in foot - pounds

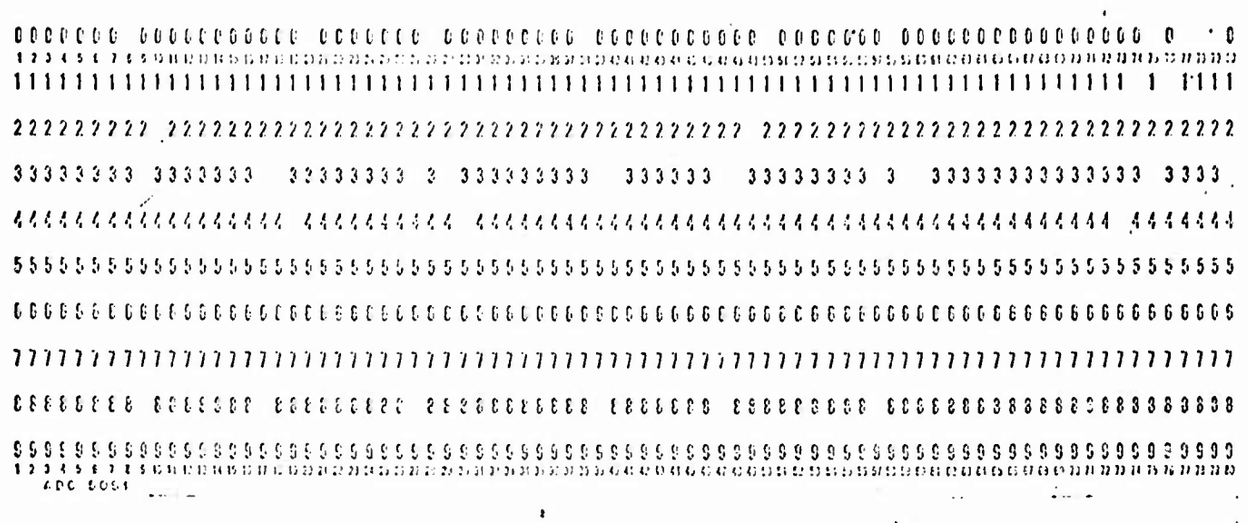
These elements are read from standard propeller curves (Miniovich or Nordstrom) in the order of quadrants I, II, III, IV as shown on the following figures. Note that the J spacing must be as shown on the example cards as the J value is not read, it is assumed by the program.

Note: Written description continues on page 33 .

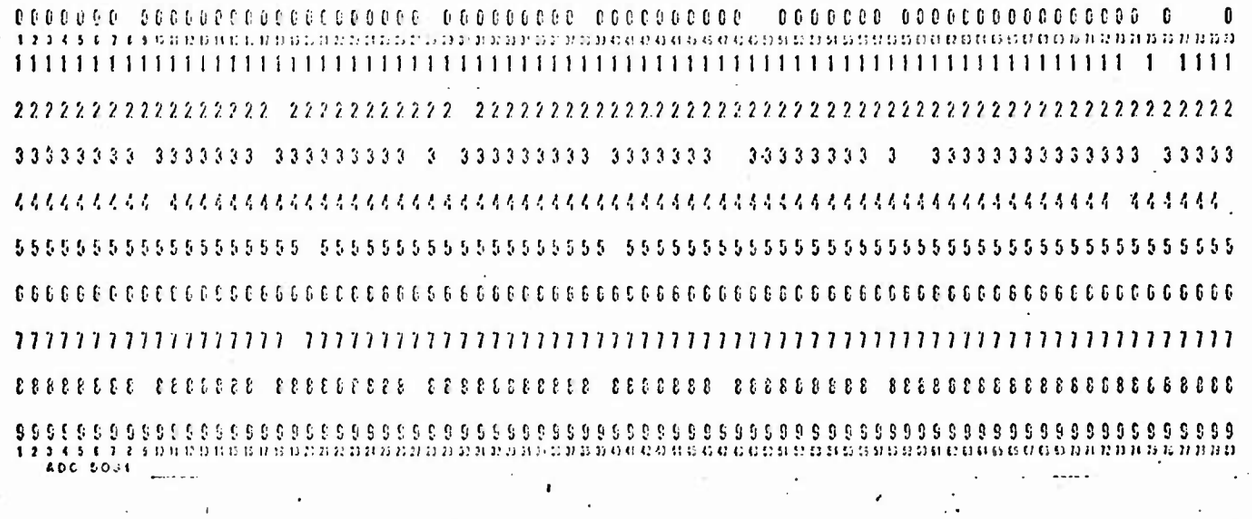

I
II
III
IV
J
2
K_T
K_T
v neg.
n pos.
v pos.
n pos.
v pos.
n neg.
v neg.
n neg.
-2
-1
-0.5
-1.0
-1.5


I
II
III
IV
K_Q
K_Q
J
v neg.
n pos.
v pos.
n pos.
v pos.
n neg.
v neg.
n neg.
0.6
0.4
0.2
0--
2
0++
0.2
0.4

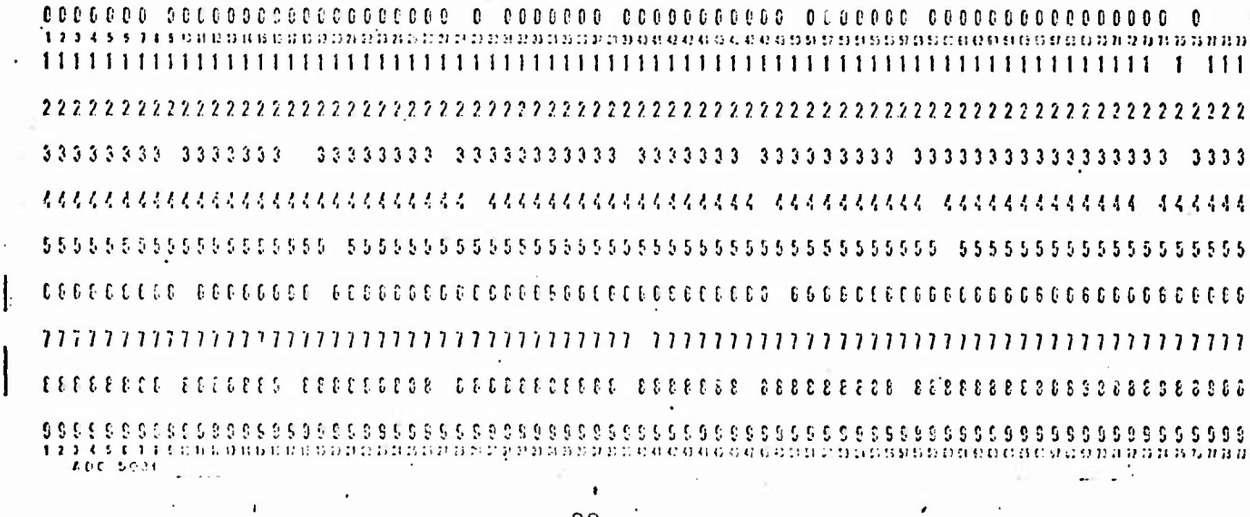
-0.2 .340 .034 -0.3 .320 .038 DATA0003



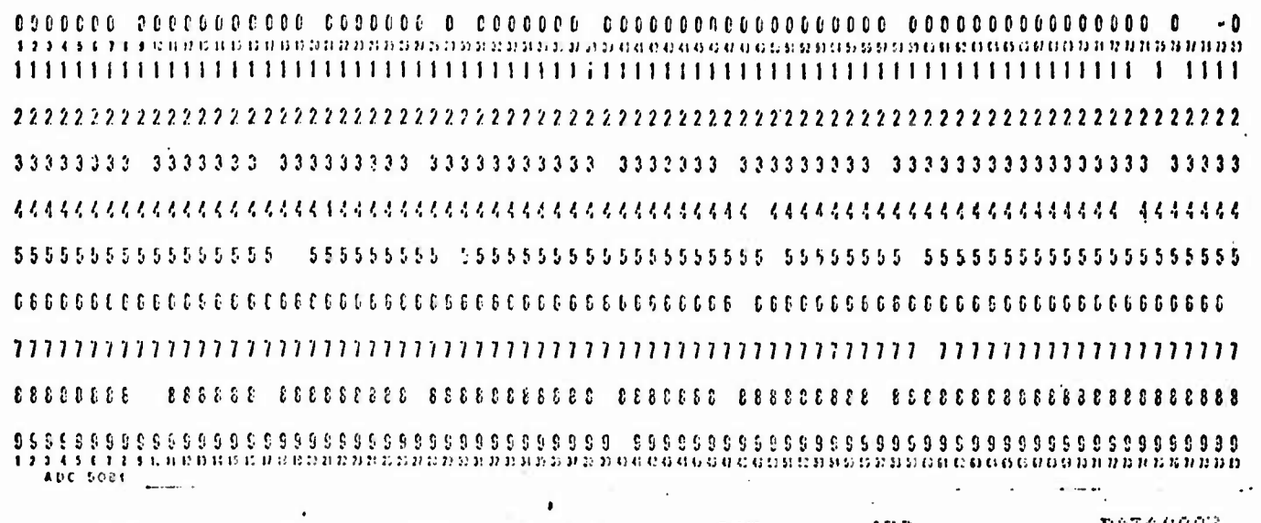
-0.4 .270 .032 -0.5 .300 .033 DATA0004



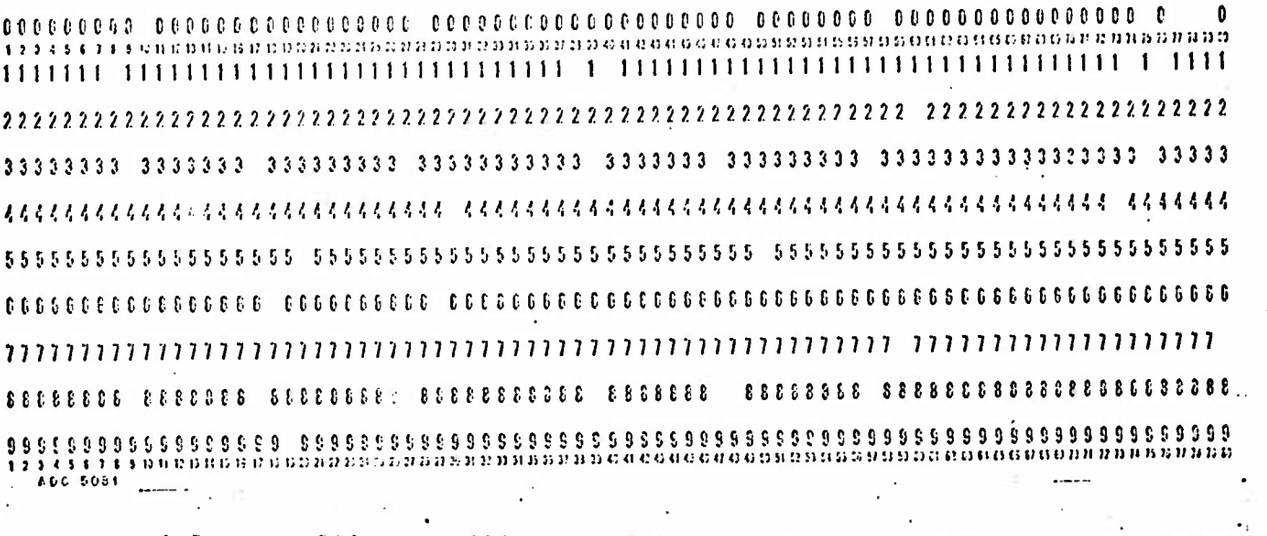
-0.6 .365 .040 -0.7 .460 .045 DATA0005



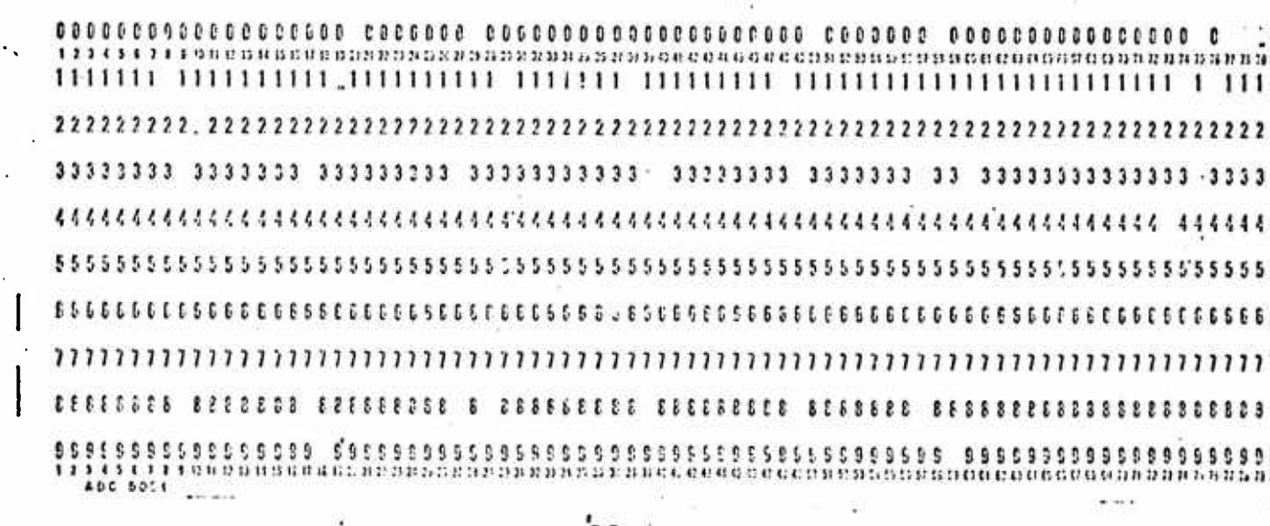
-0.8 .550 .050 -0.9 .645 .057 DATA0006



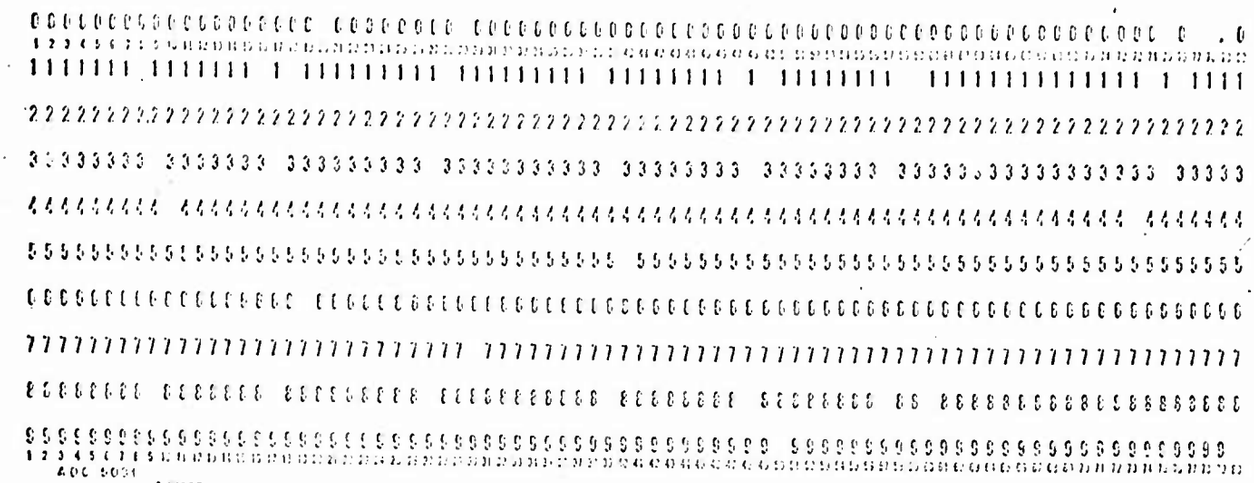
-1.0 .695 .064 -1.1 .805 .072 DATA0007



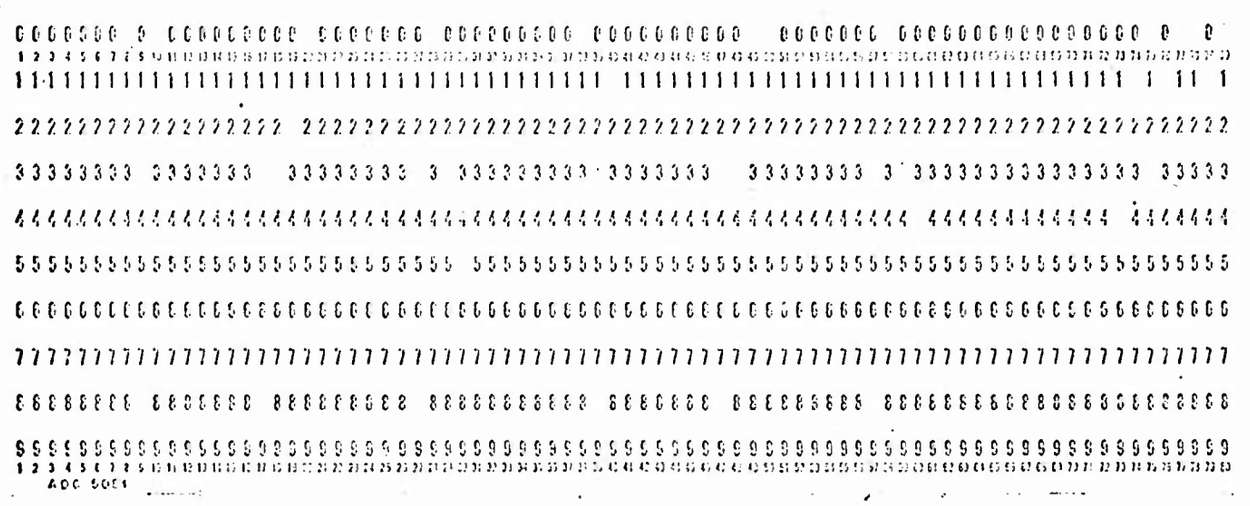
-1.2 .910 .081 -1.3 1.0 .093 DATA0008



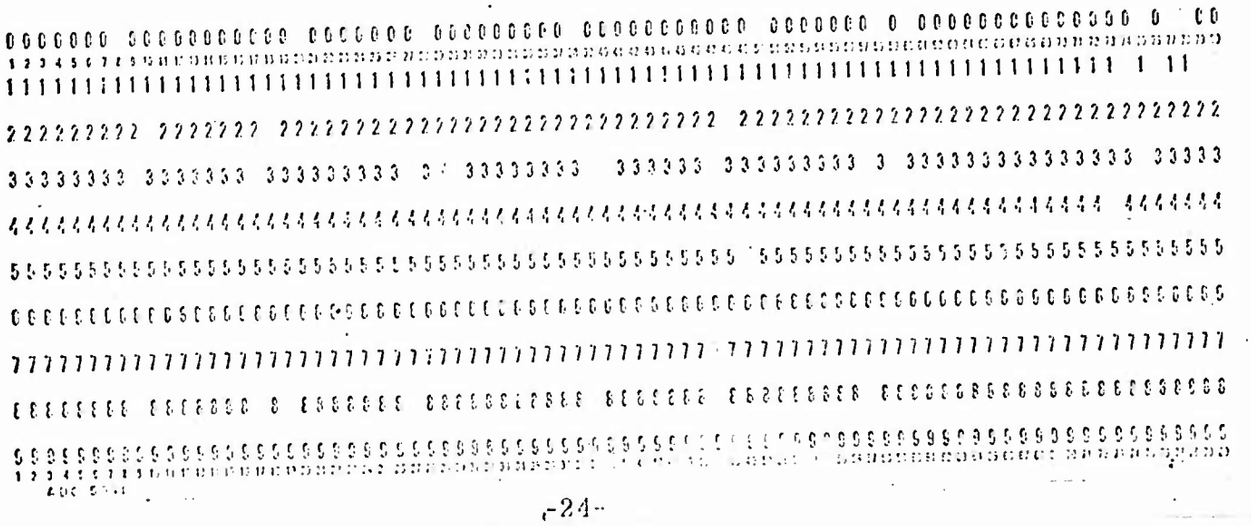
-1.4 1.100 .107 -1.5 1.19 .110 DATA00...



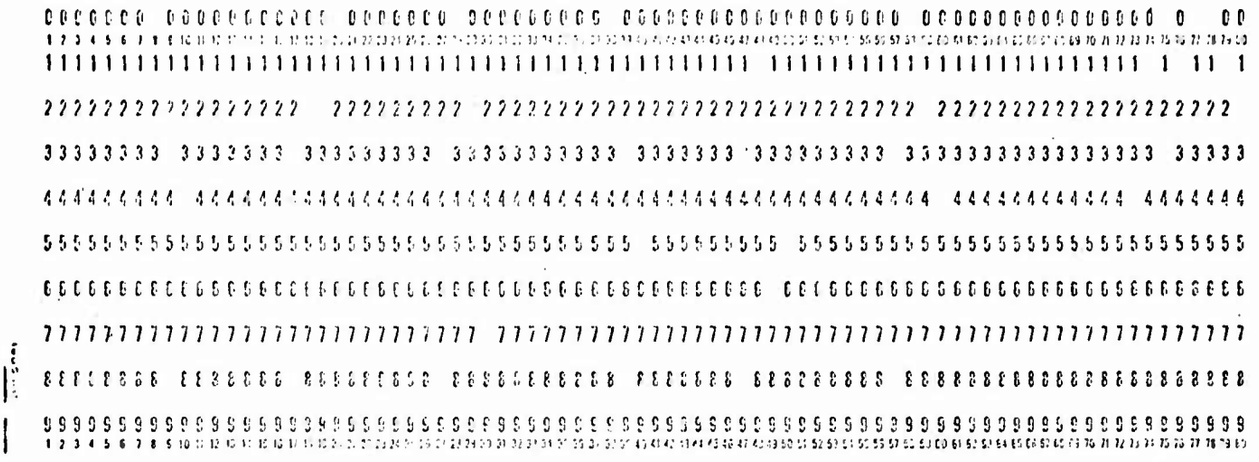
0.0 .320 .035 0.1 .300 .034 DATA0010



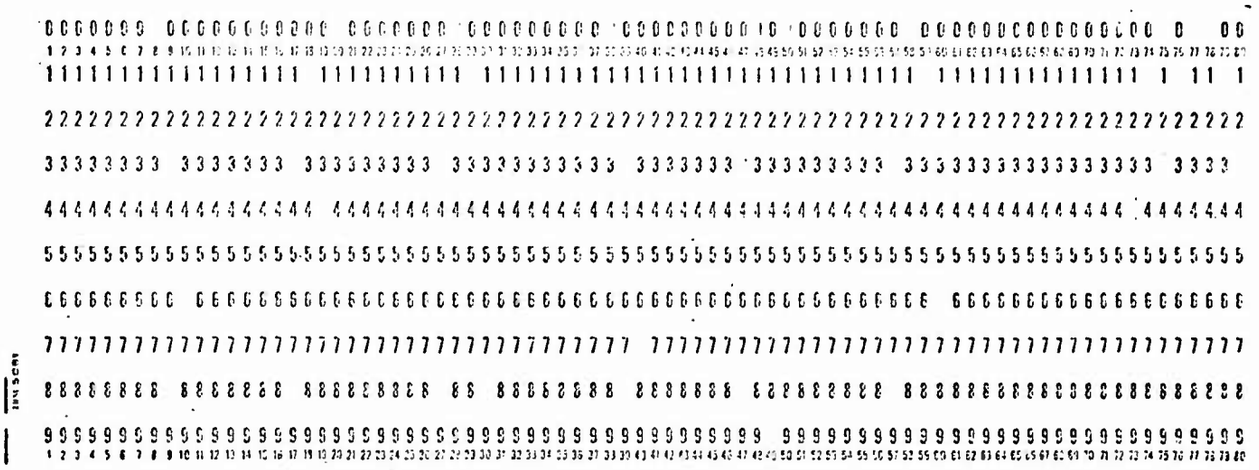
0.2 .280 .033 0.3 .250 .030 DATA0011



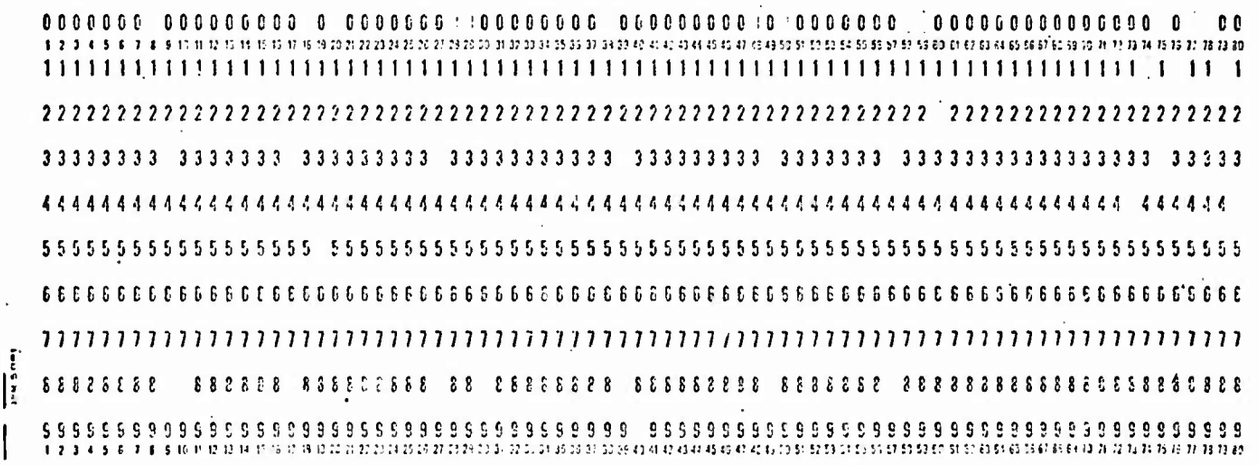
0.4 .220 .027 0.5 .105 .024 DATA0012



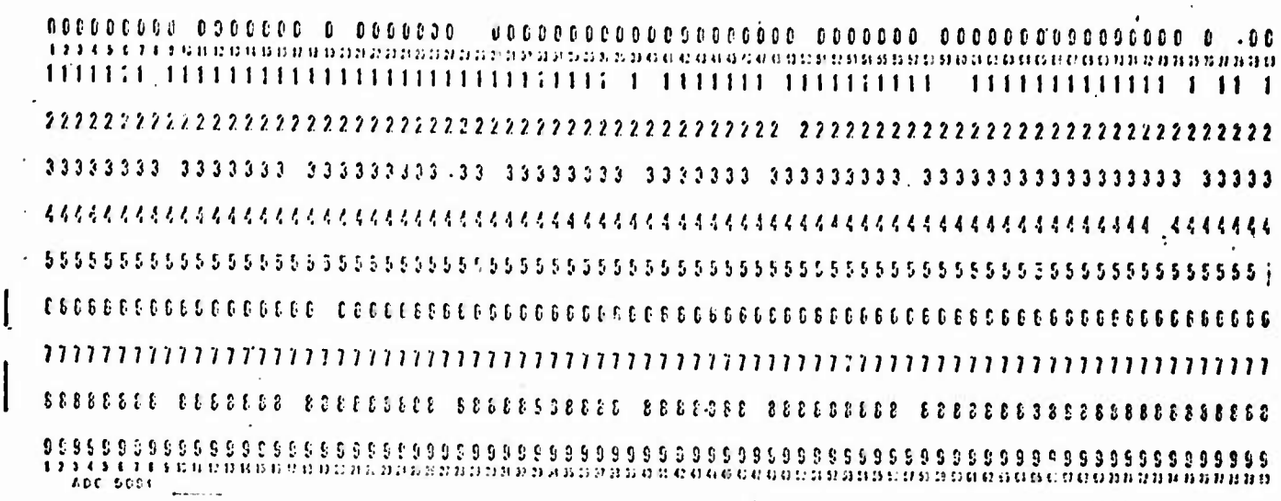
0.6 .140 .018 0.7 .090 .016 DATA0013



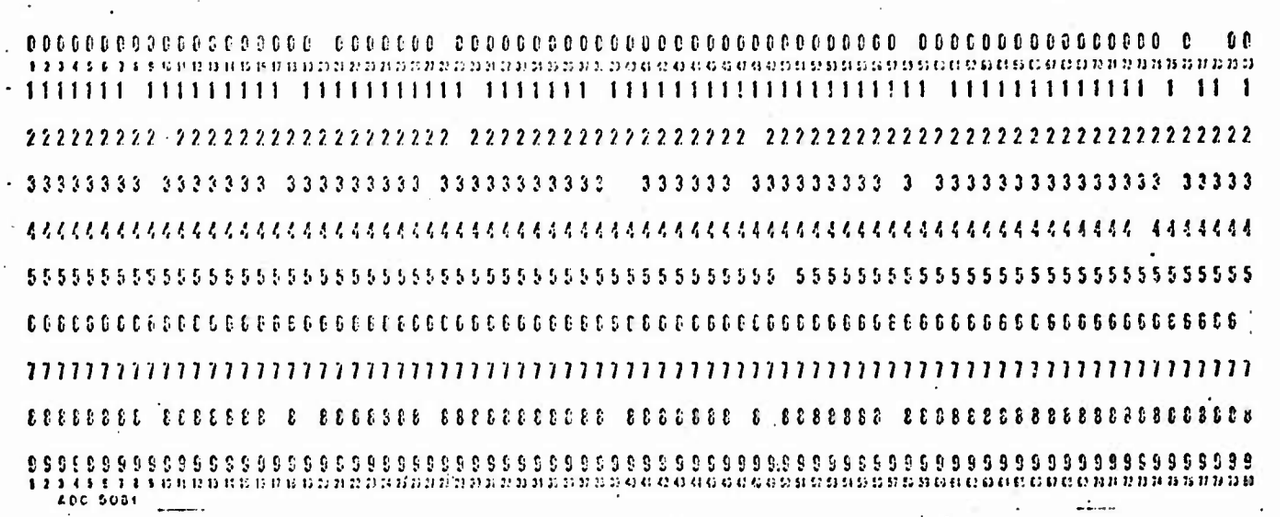
0.8 .050 .008 0.9 0.0 .002 DATA0014



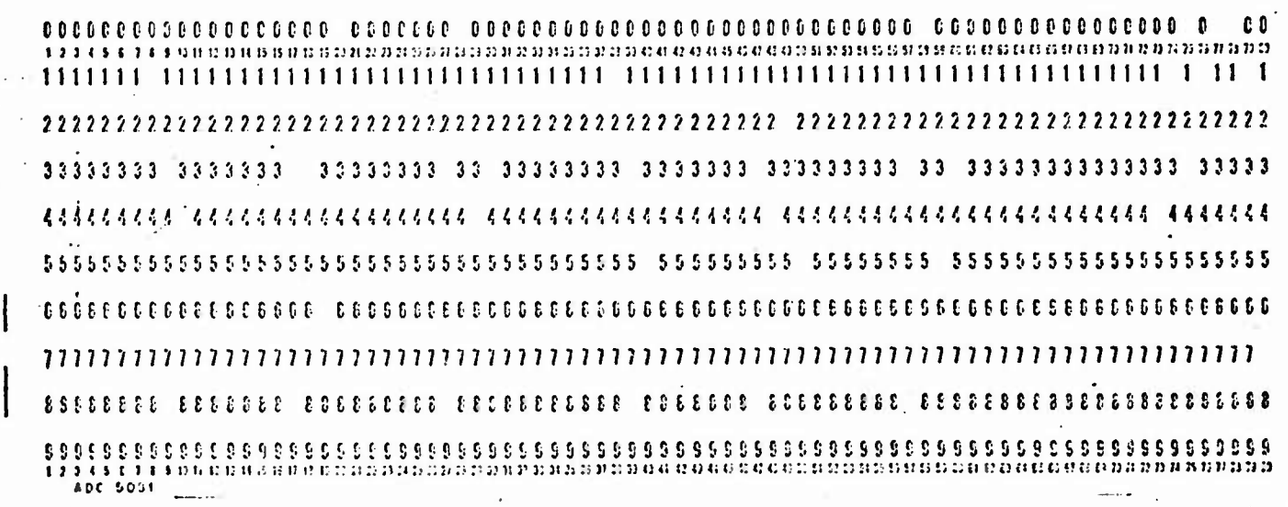
1.0 -.060 -.003 1.1 -.120 -.011 DATA0015



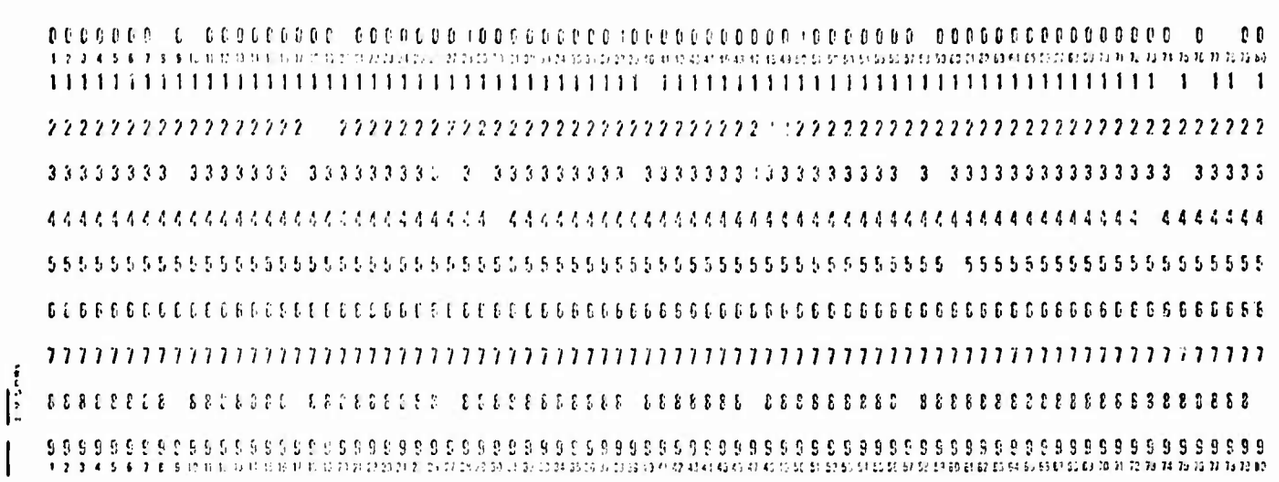
1.2 -.180 -.021 1.3 -.285 -.031 DATA0016



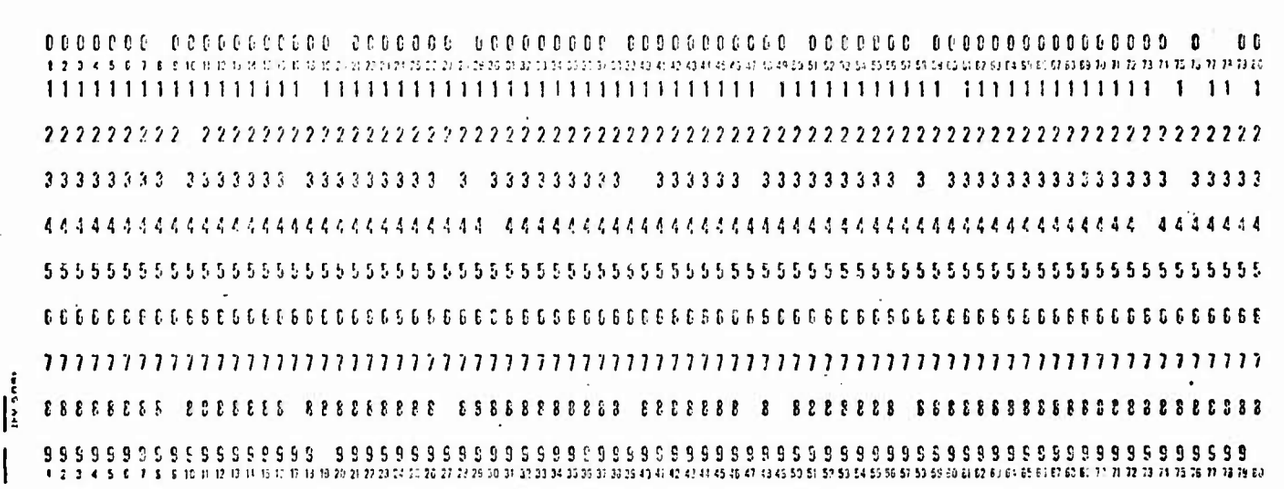
1.4 -.360 -.043 1.5 -.425 -.053 DATA0017



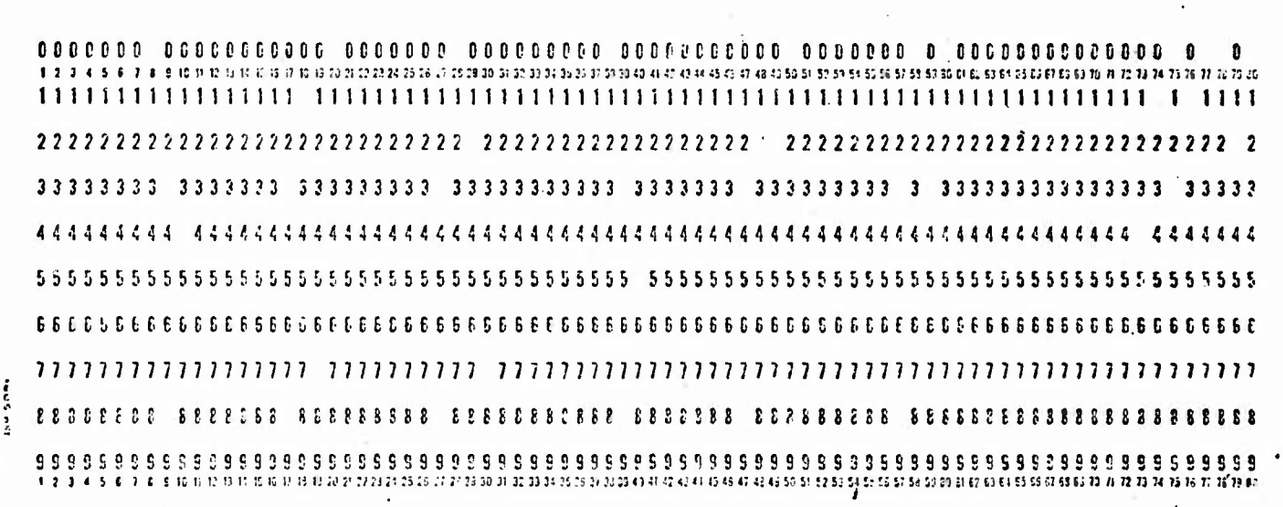
```
0.0      -.220      -.034      -0.1      -.280      -.035      DATA0018
```



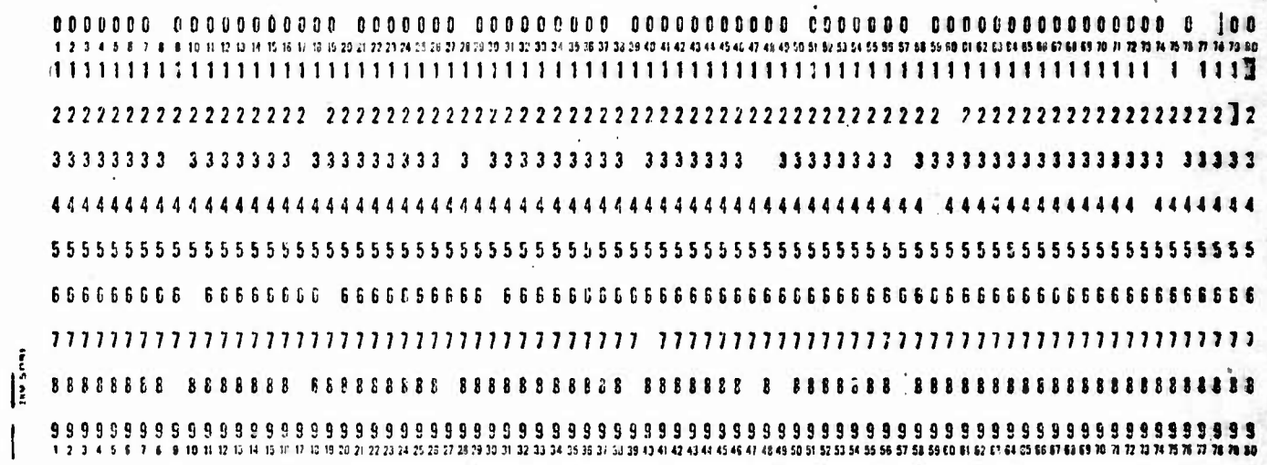
```
-0.2      -.190      -.034      -0.3      -.180      -.031      DATA0019
```



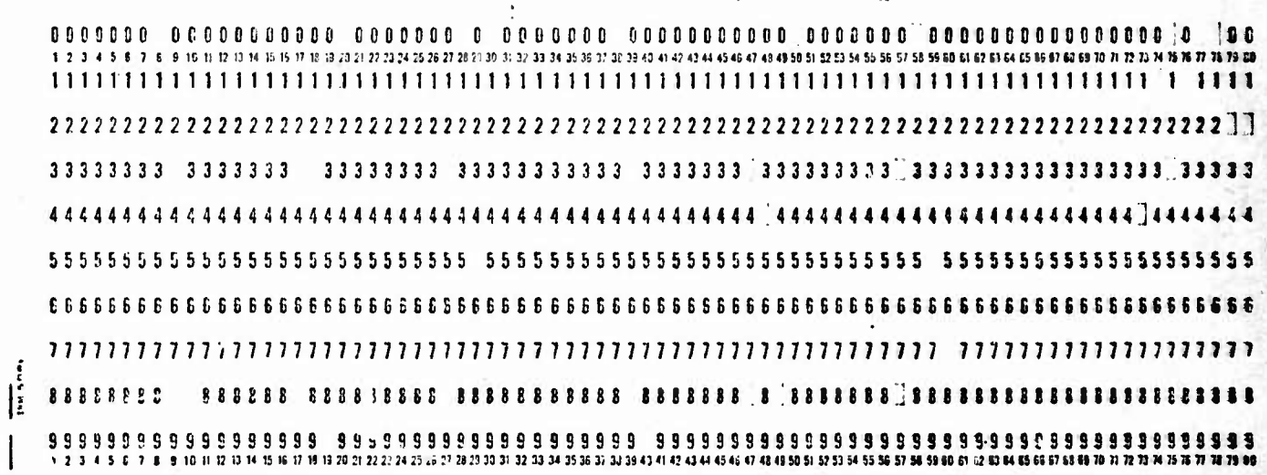
```
-0.4      -.170      -.027      -0.5      -.220      -.030      DATA0020
```



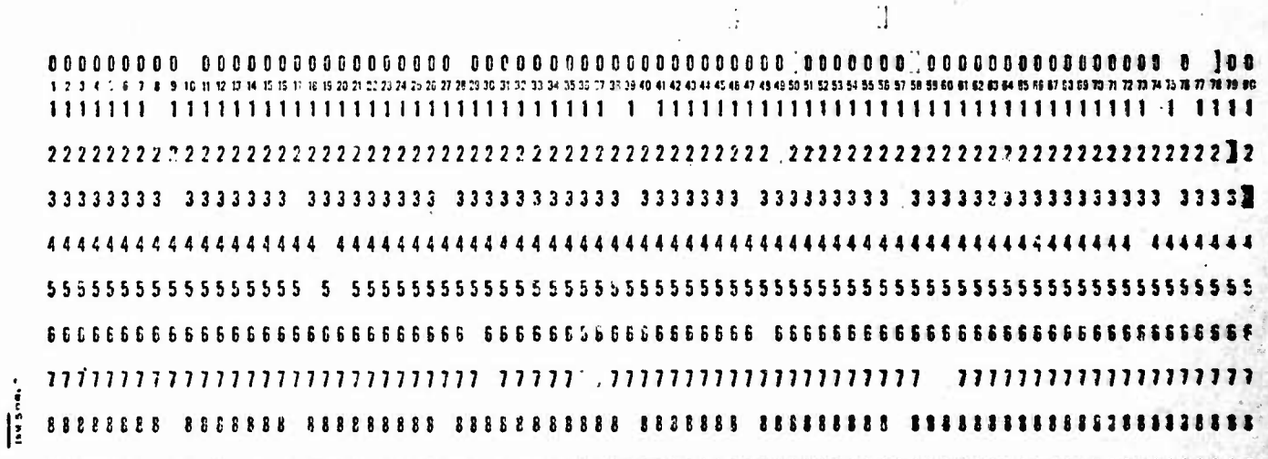
-0.6 -.260 -.036 -0.7 -.380 -.042 DATA0021



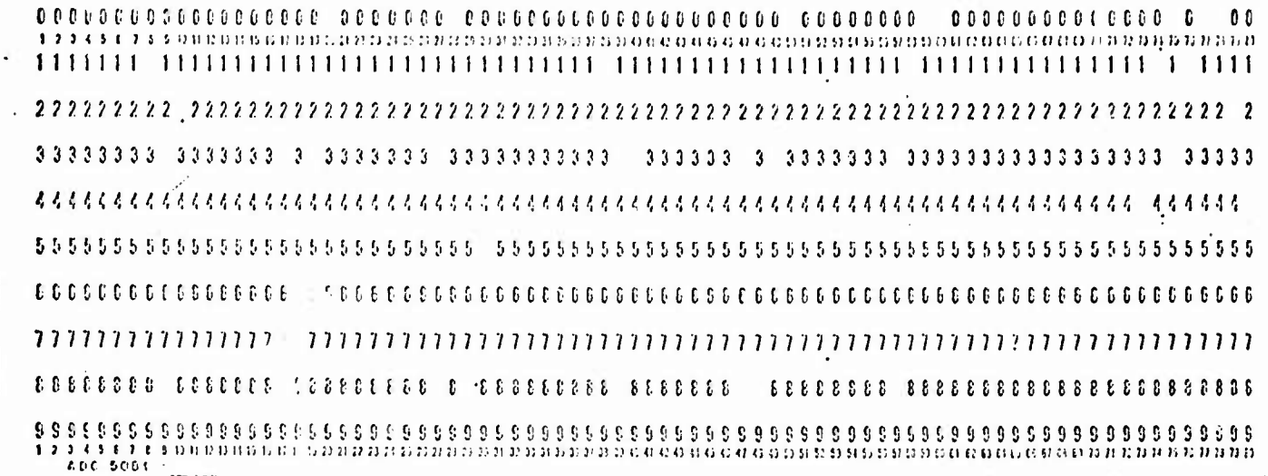
-0.8 -.390 -.050 -0.9 -.480 -.057 DATA0022



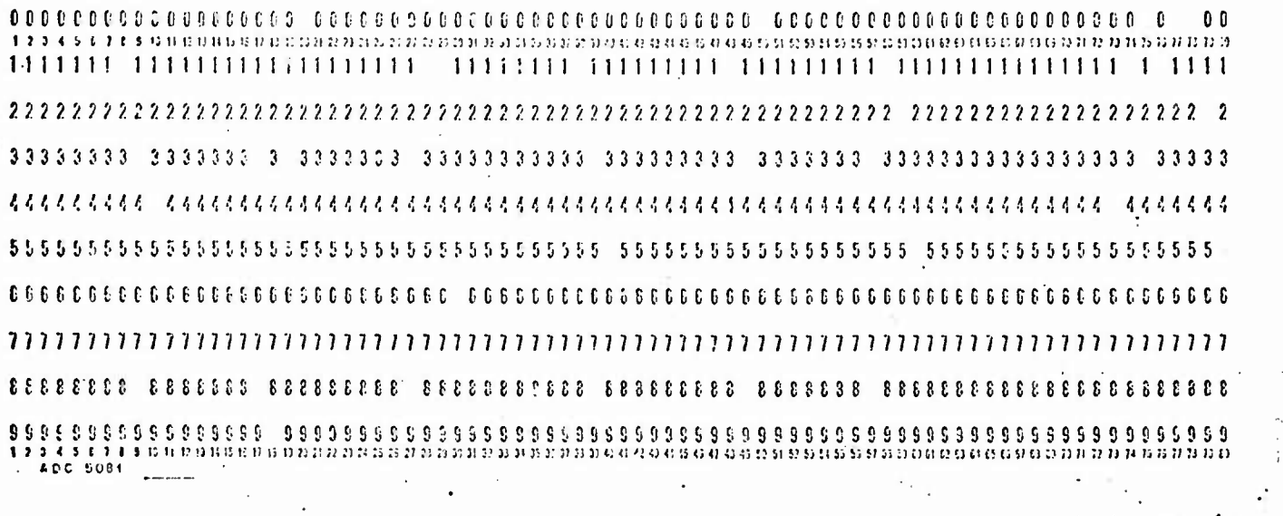
-1.0 -.545 -.067 -1.1 -.620 -.077 DATA0023



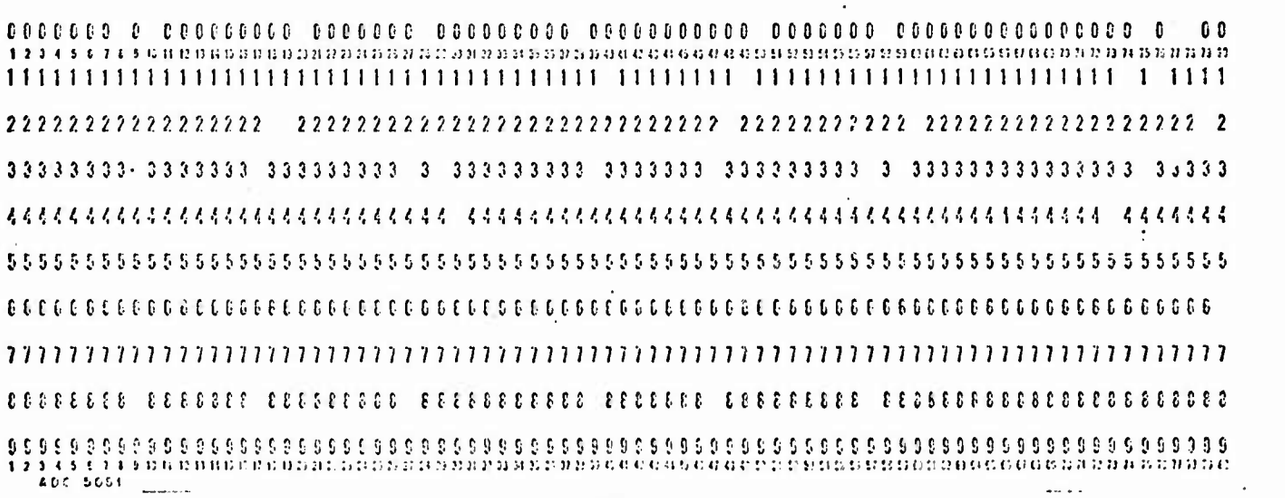
-1.2 -.730 -.025 -1.3 -.930 -.100 DATA0024



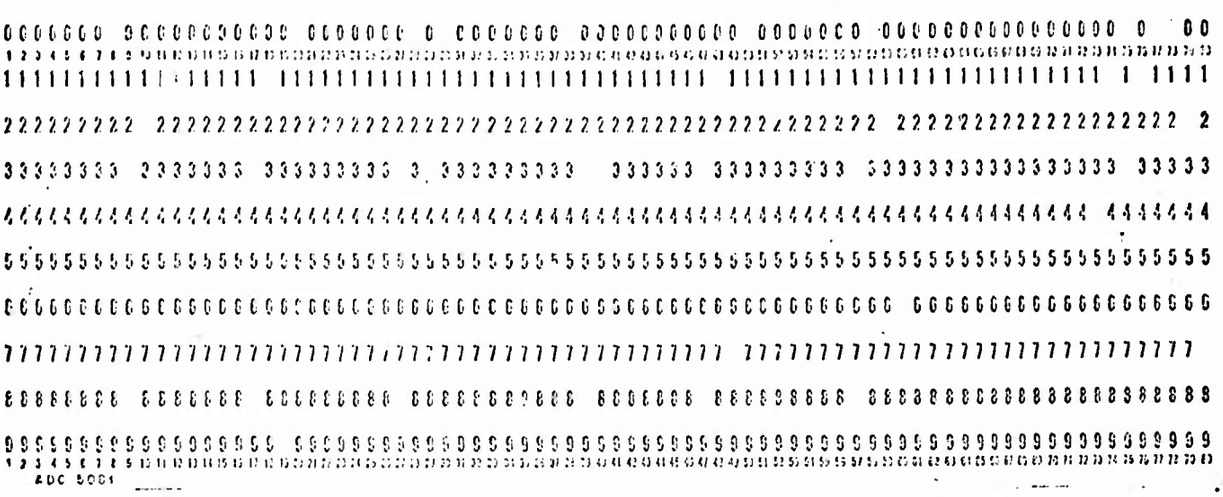
-1.4 -.930 -.116 -1.5 -1.0 -.195 DATA0025



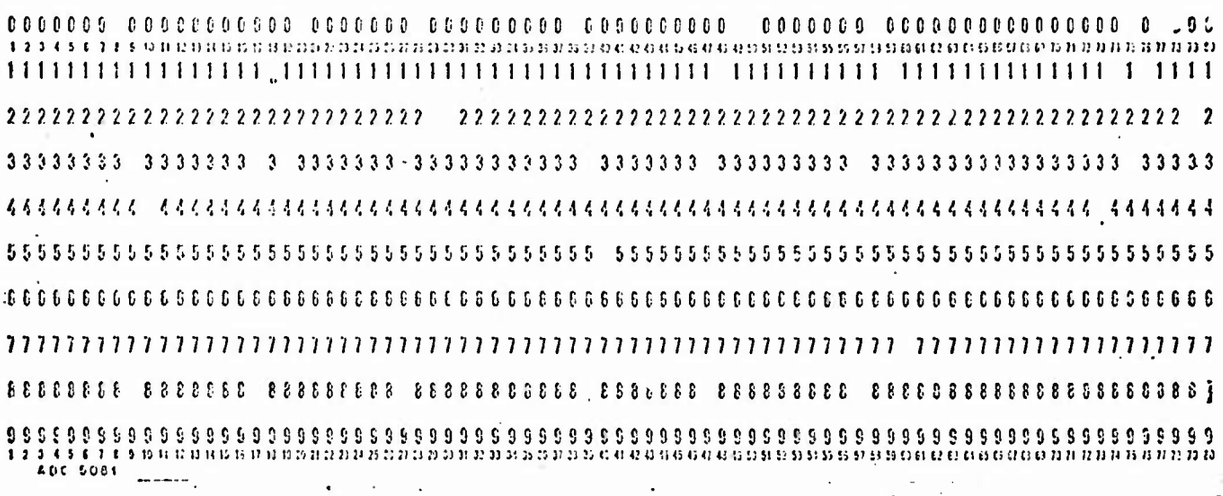
0.0 -.250 -.034 0.1 -.210 -.032 DATA0026



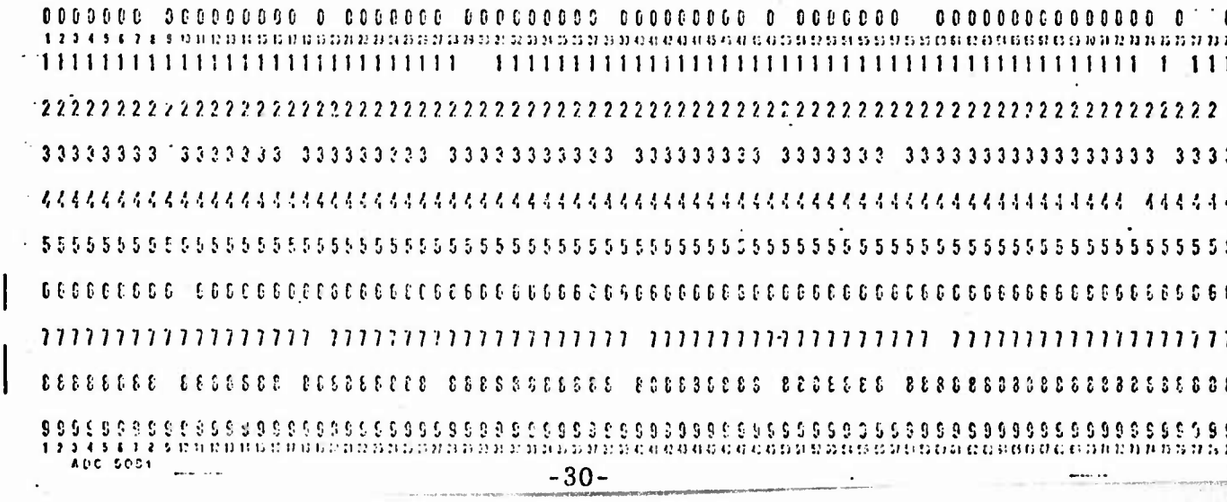
0.2 -.190 -.030 0.3 -.170 -.028 DATA0027



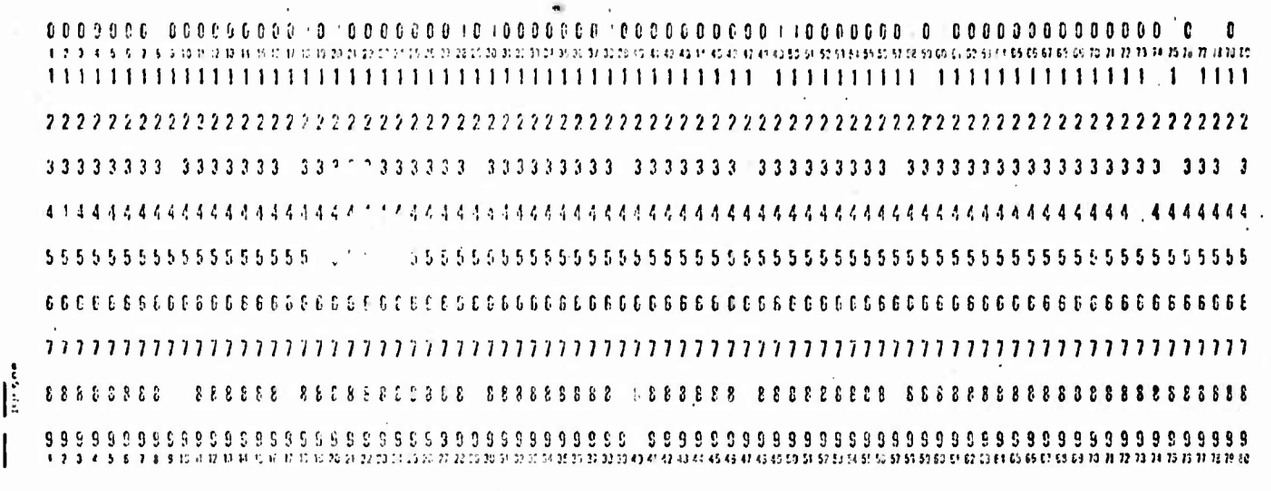
0.4 -.130 -.022 0.5 -.100 -.017 DATA0028



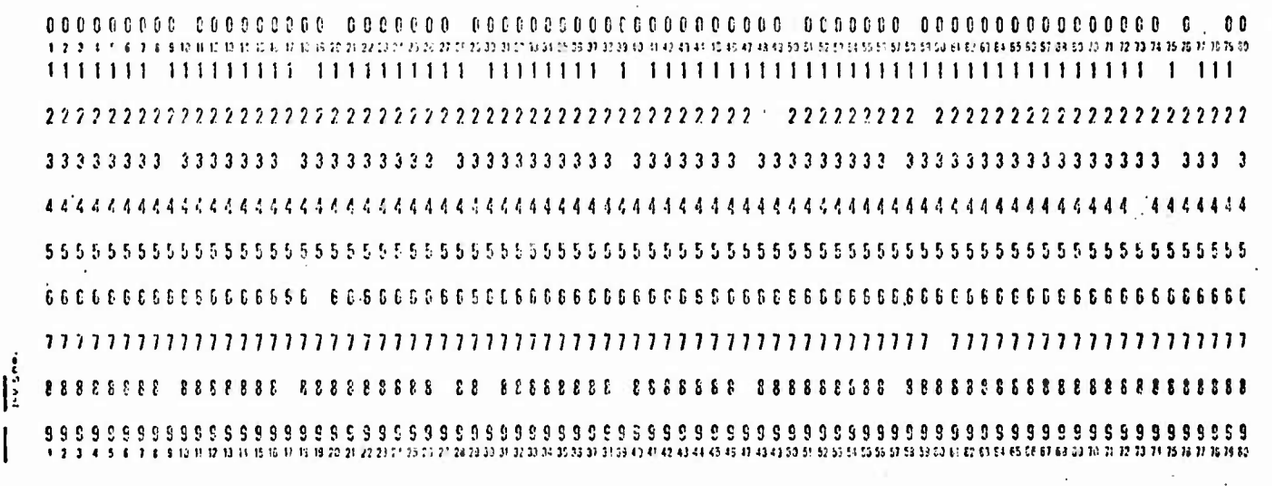
0.6 -.070 -.013 0.7 0.0 -.007 DATA0029



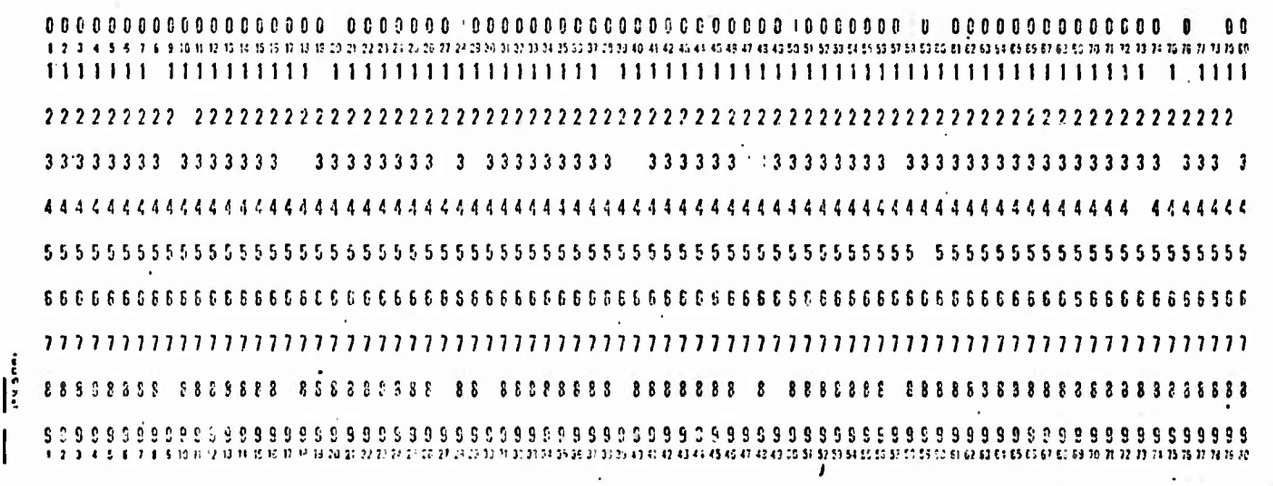
```
0.8       .050        0.0        0.9        .100       .010                DATA0030
```



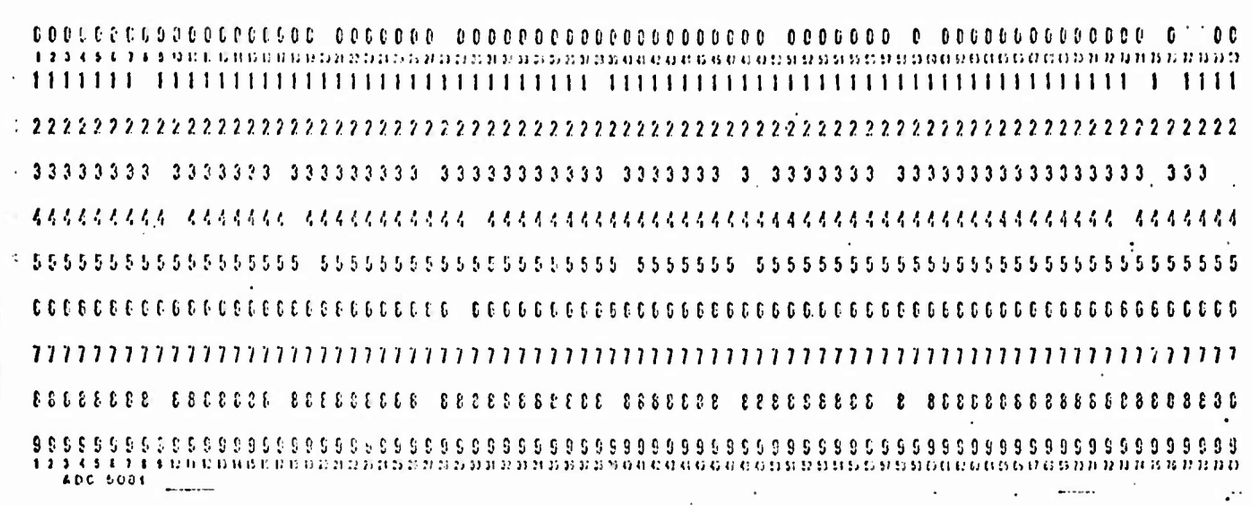
```
1.0       .160        .018       1.1        .220       .027                DATA0031
```



```
1.2       .310        .038       1.3        .380       .050                DATA0032
```

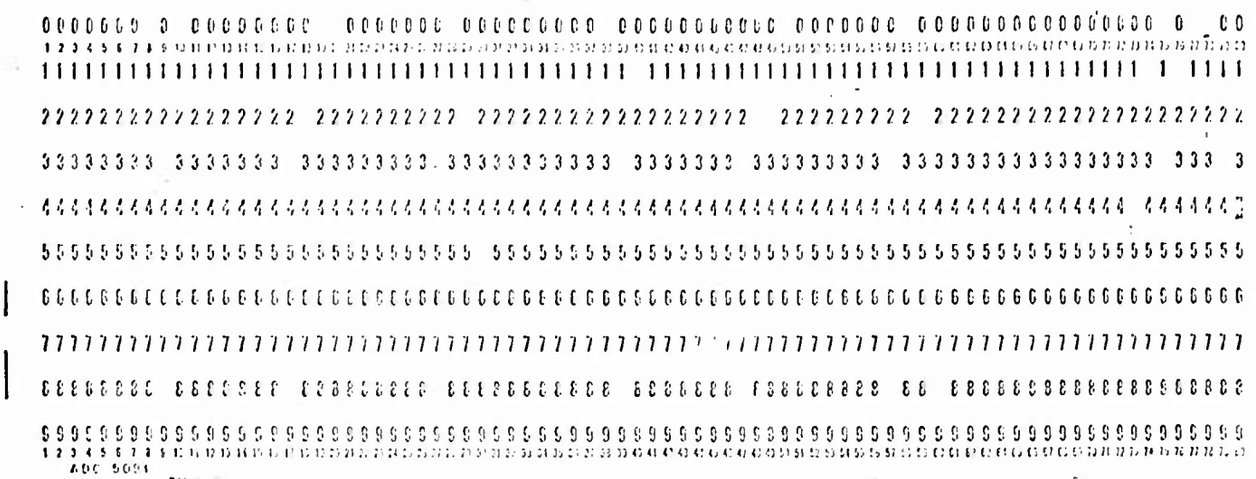


1.4 .450 .004 1.5 .530 .030 DATE0035



ADC 5001

0.0	.200	.0[illegible]	- 0.1	.220	.0[illegible]	[illegible]
1/J(1)	E(1)	EE(1)	1/J(1+1)	E(1+1)	EE(1+1)	



Data Cards 34-49

FORMAT 2 (10x, 2F10.4)

ELEMENTS

1/J = Inverse non dimensional speed. Note that as J goes to 1.5 it is only necessary to carry 1/J to 1/1.5 ≈ .7 for coverage of full range.

E = Modified Thrust coefficient (non-dimensional) = C_T

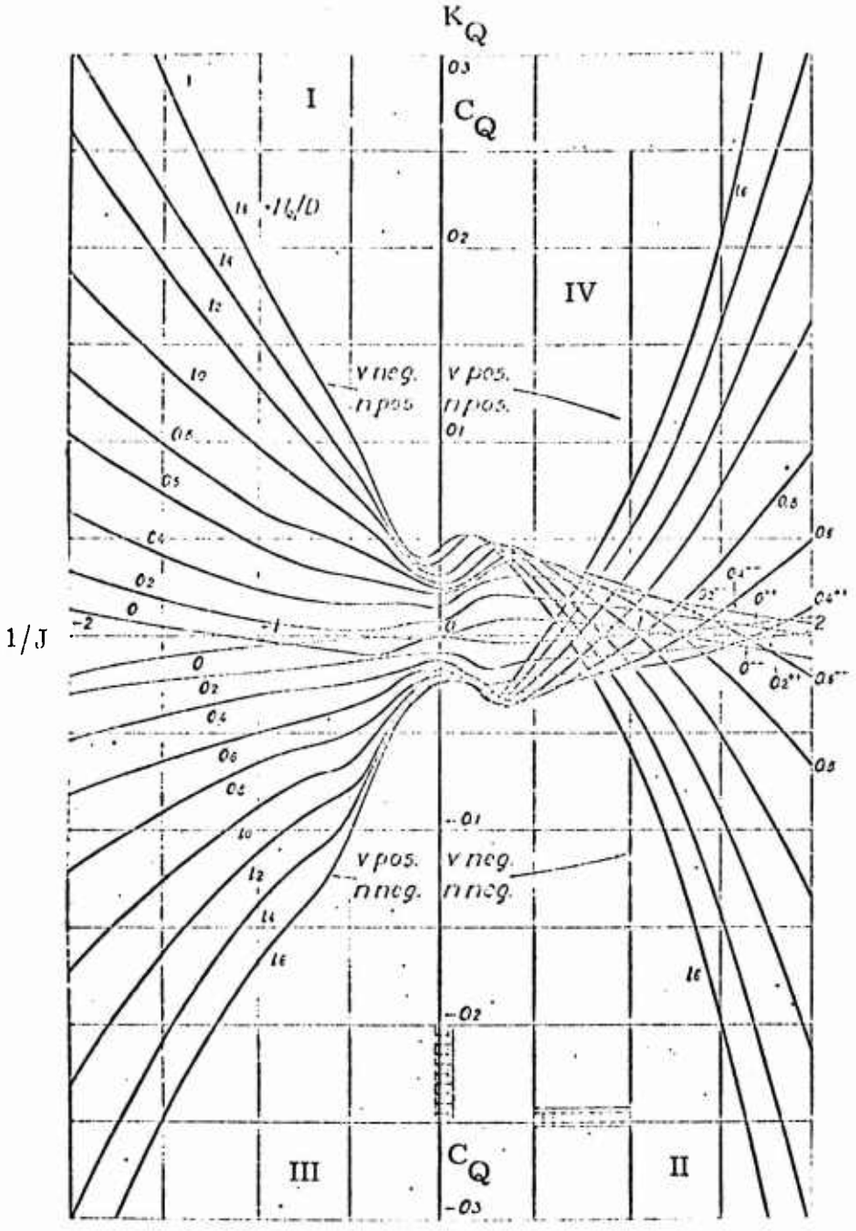
$$C_T = \frac{T}{\rho V_a^2 D^2}$$

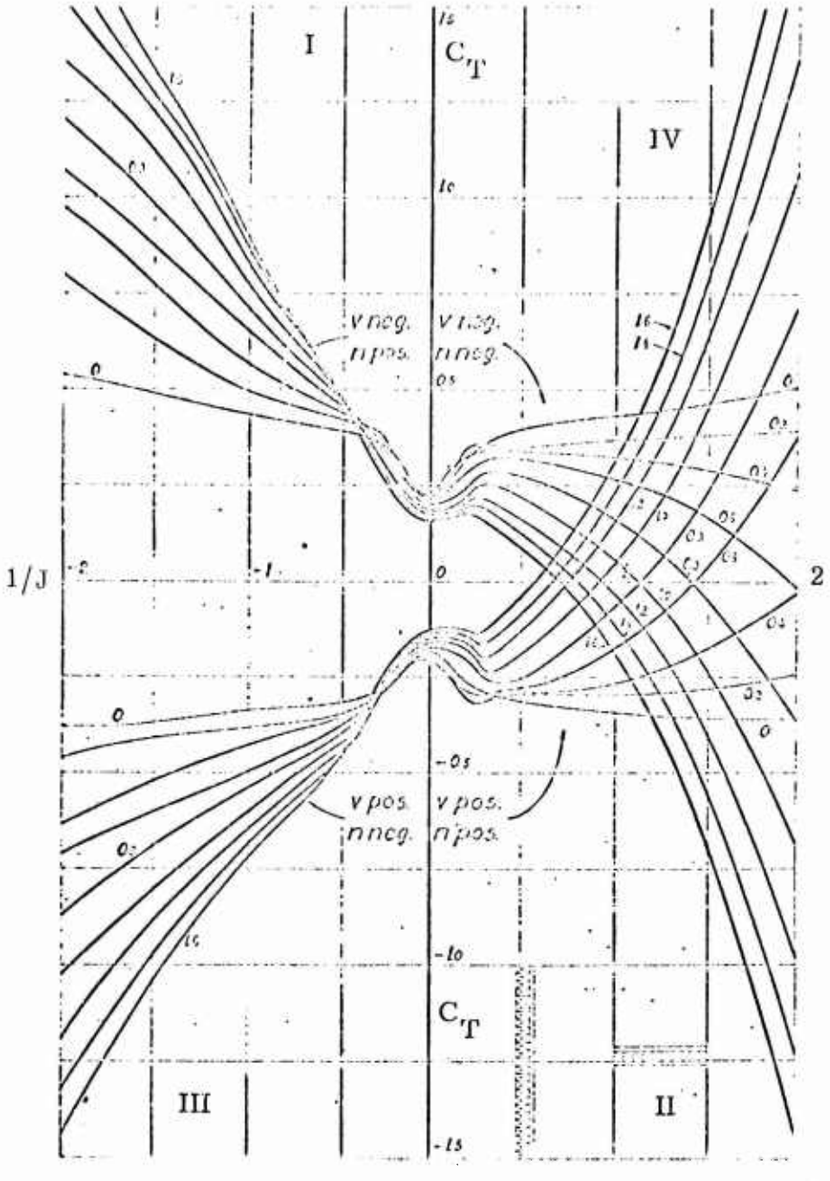
EE= Modified Torque coefficient (non-dimensional) = C_Q

$$C_Q = \frac{Q_p}{\rho V_a^2 D^3}$$

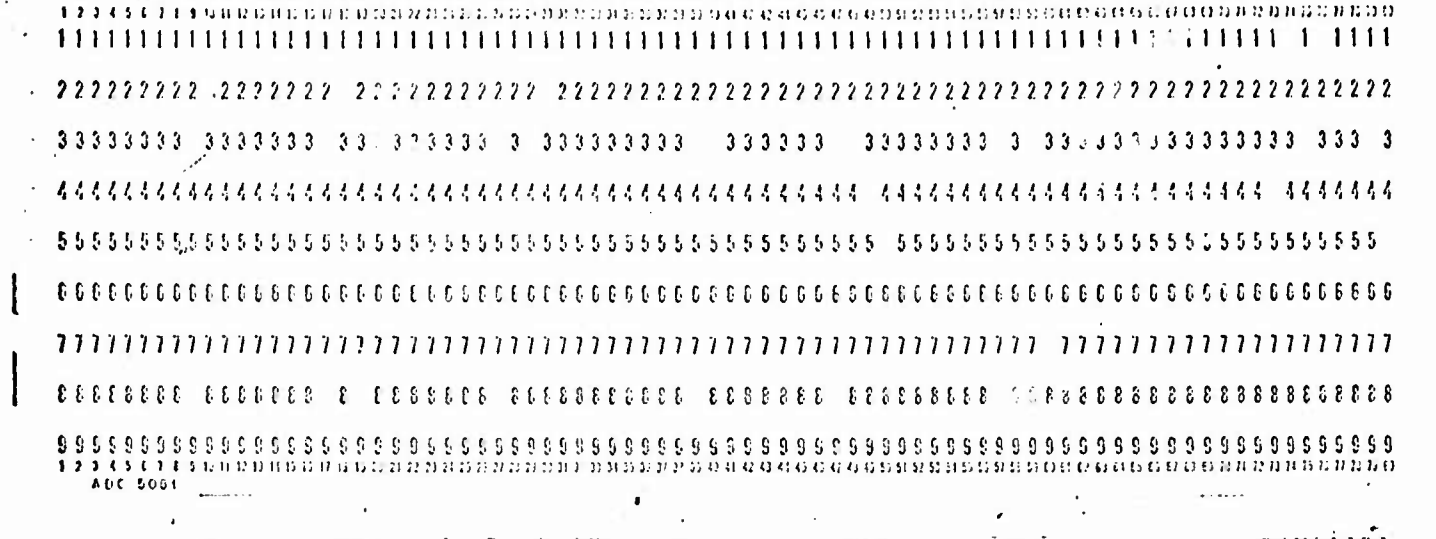
These elements are read from standard propeller curves (Miniovich or Nordstrom) in the order of quadrants I, II, III, IV as shown on the following figures. Note that the 1/J spacing must be as shown on the example cards as the 1/J value is not read, it is assumed by the program.

Note: Written description continues on page 42.

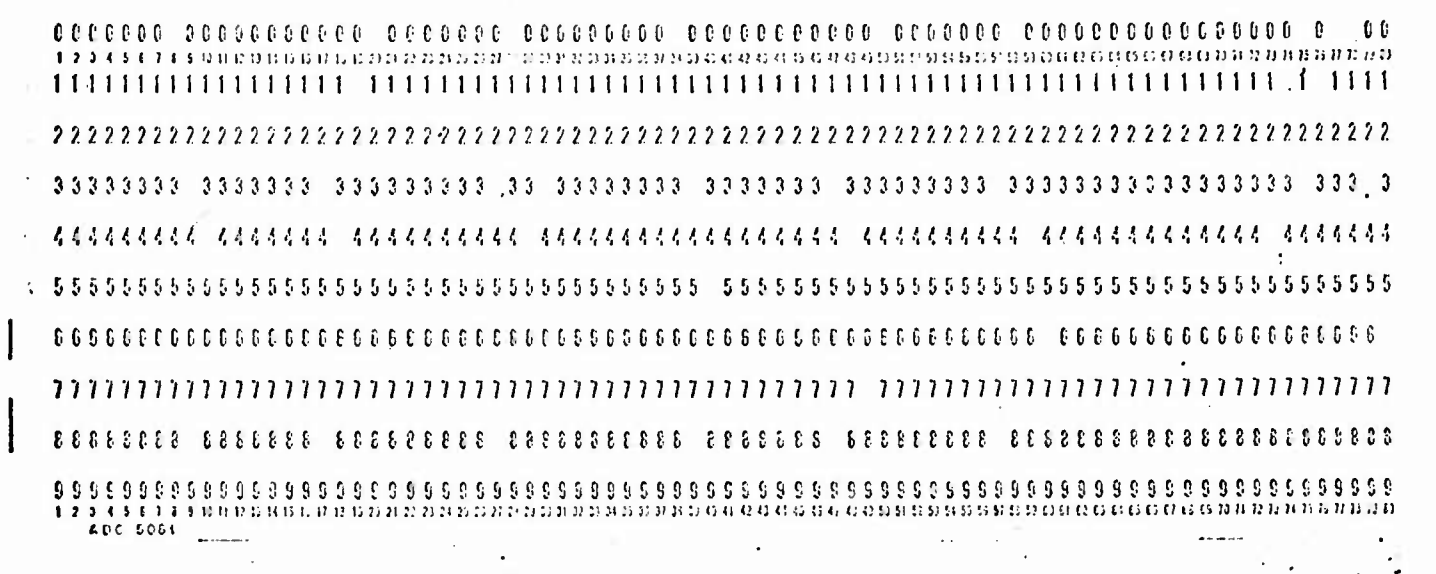

K_Q
I
C_Q
IV
III
C_Q
II
$1/J$
$-1/J$
v neg.
n pos.
v pos.
n pos.
v pos.
n neg.
v neg.
n neg.


I
IV
III
II
C_T
C_T
v neg.
n pos.
v neg.
n neg.
v pos.
n neg.
v pos.
n pos.
1/J
2
0
-1

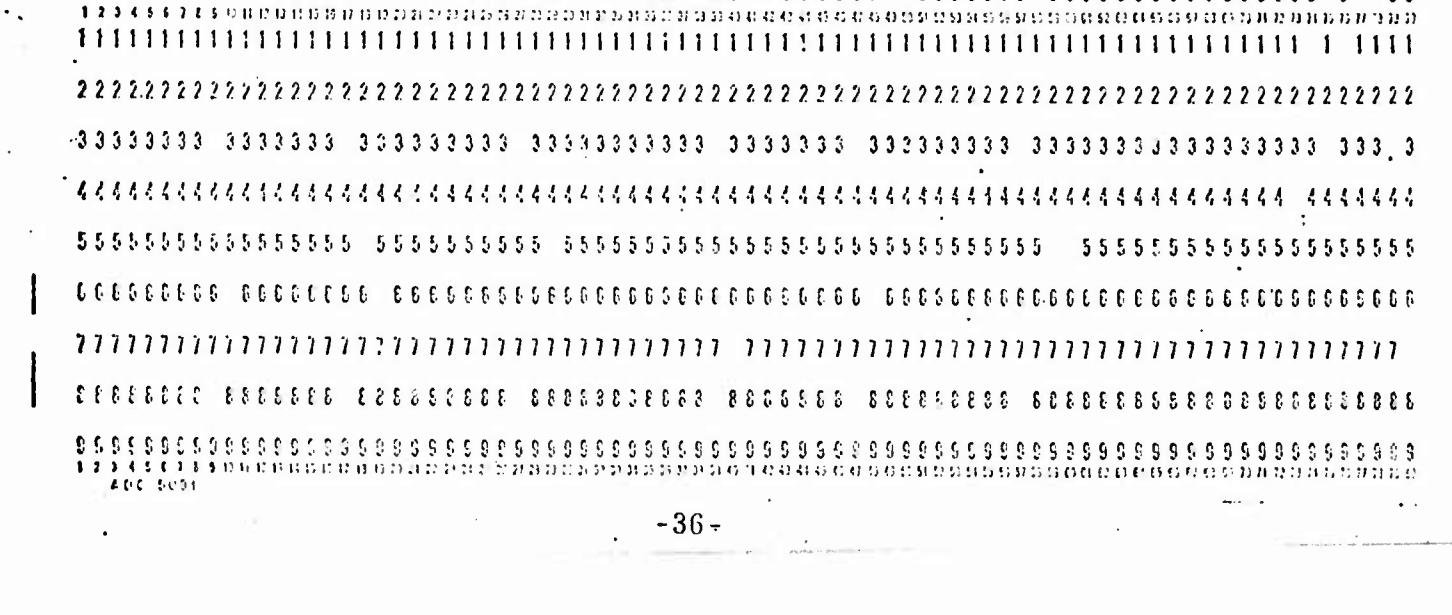
-0.2 .980 .032 -0.3 .345 .027 DATA0035



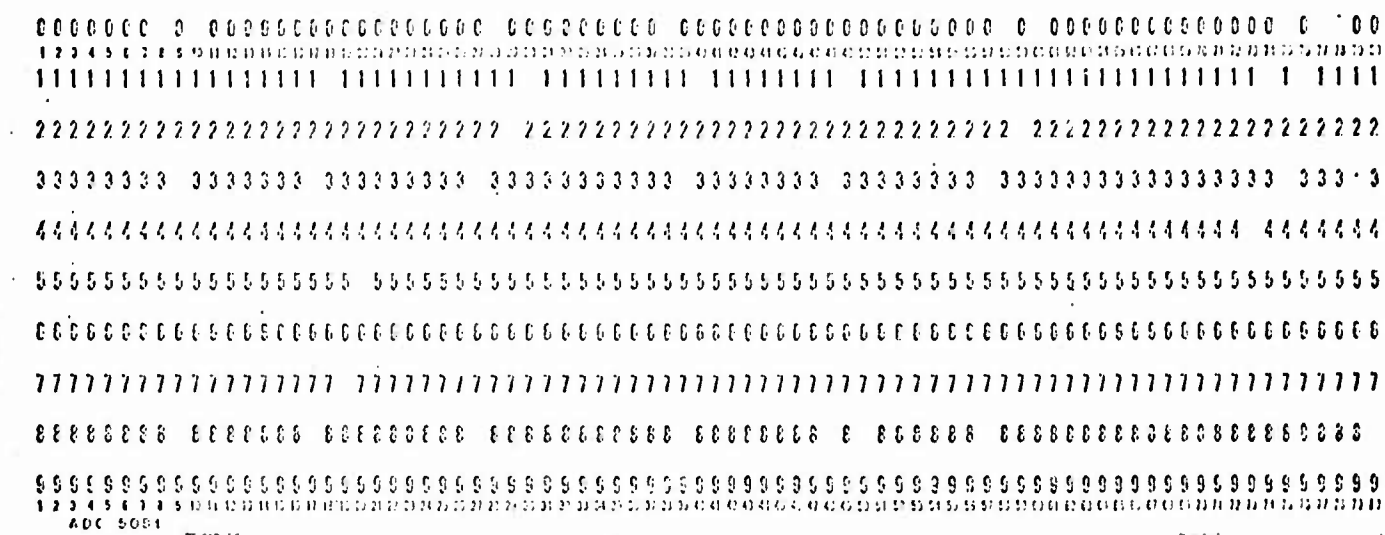
-0.4 .410 .048 -0.5 .470 .046 DATA0036



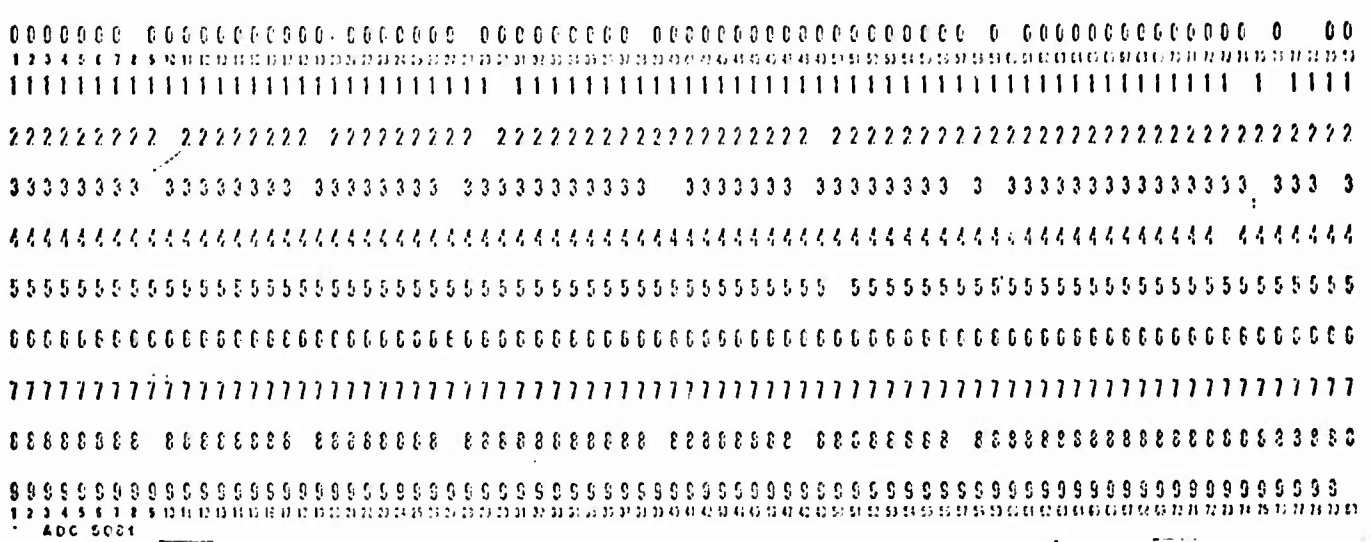
-0.6 .560 .050 -0.7 .600 .055 DATA0037



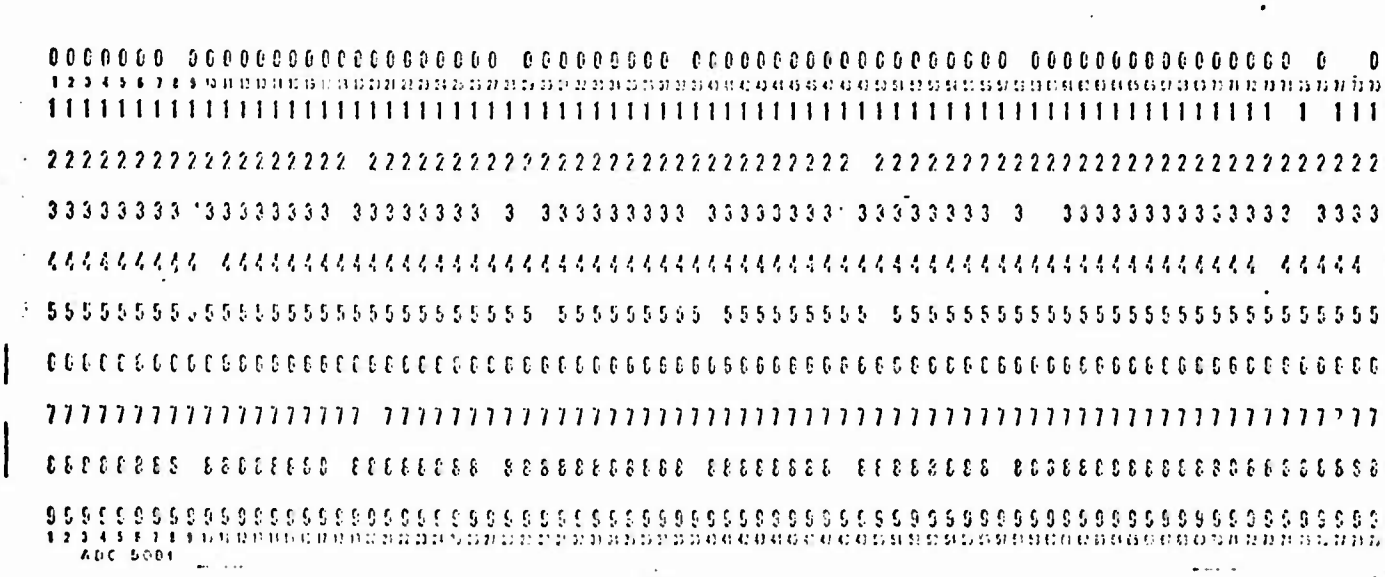
0.0 -.175 -.021 0.1 -.18 -.020 DATA0035



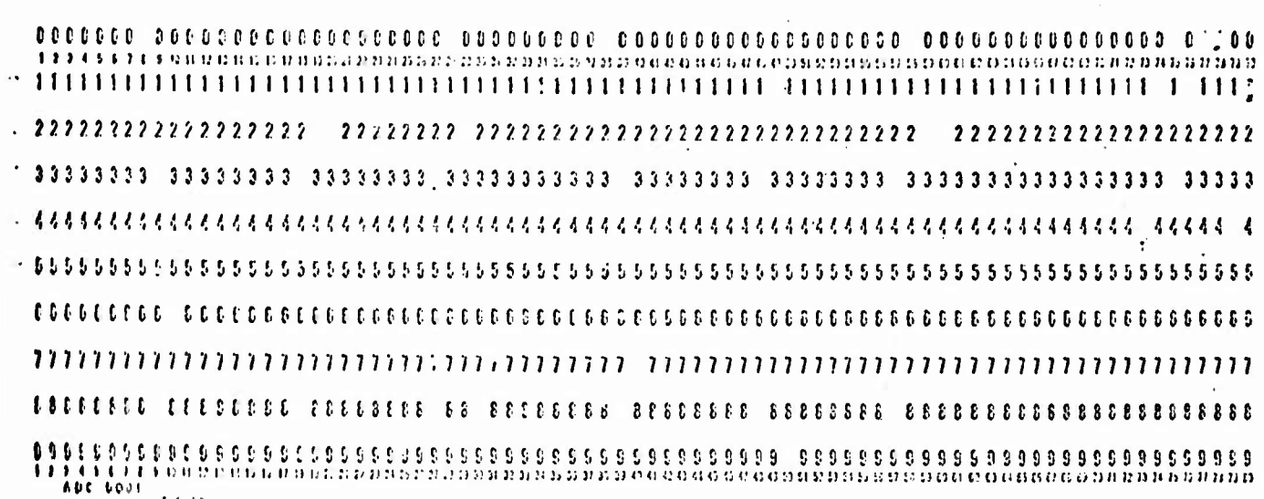
0.2 -.20 -.021 0.3 -.23 -.030 DATA0036



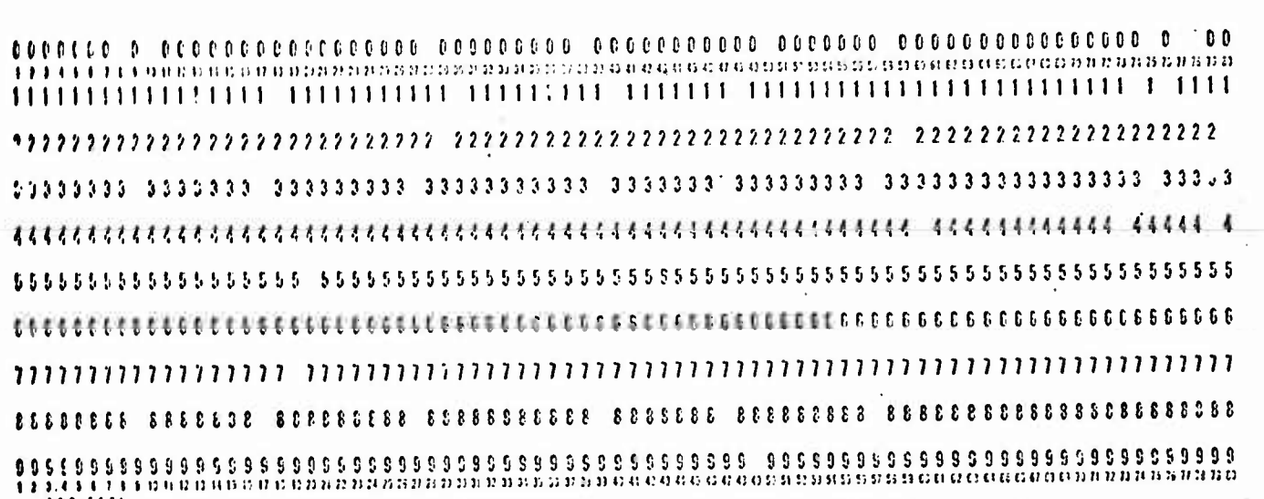
0.4 -.27 -.035 0.5 -.25 -.033 DATA0040



0.6 -.22 -.028 0.7 -.19 -.022 DATA0041



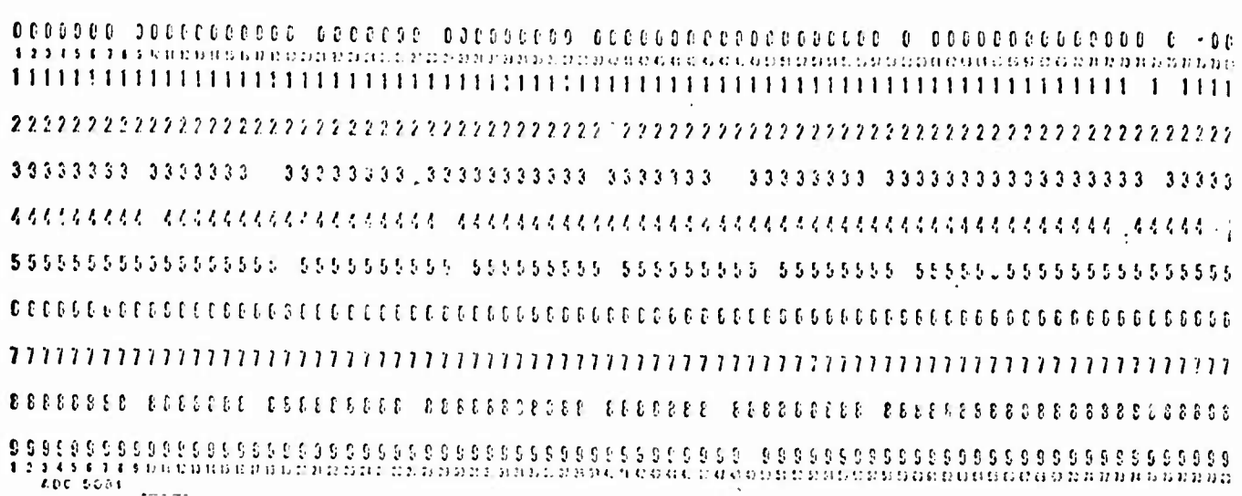
0. -.17 -.023 -0.1 -.190 -.024 DATA0042



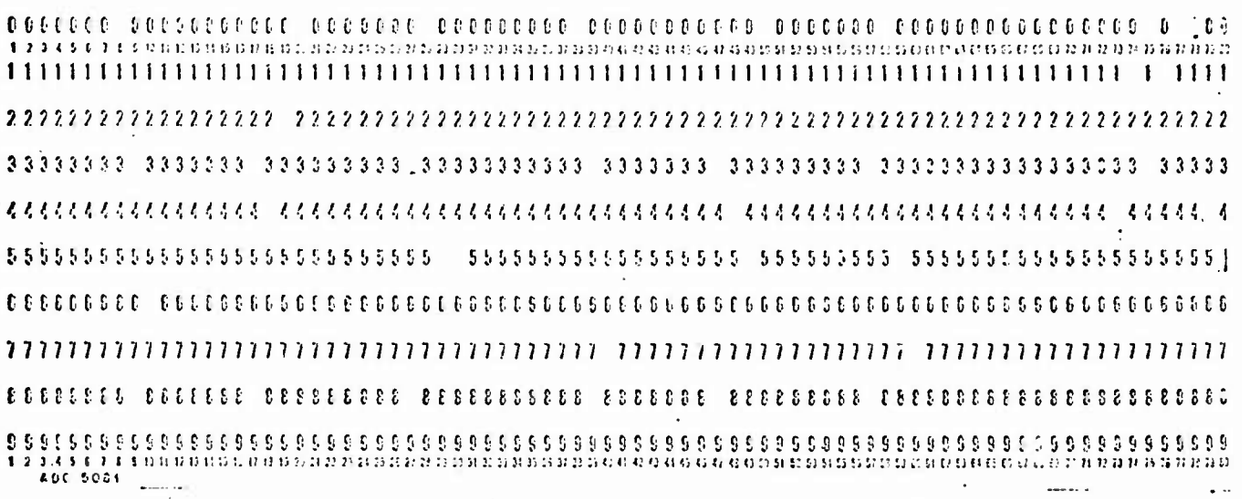
-0.2 -.230 -.029 -0.3 -.290 -.038 DATA0043



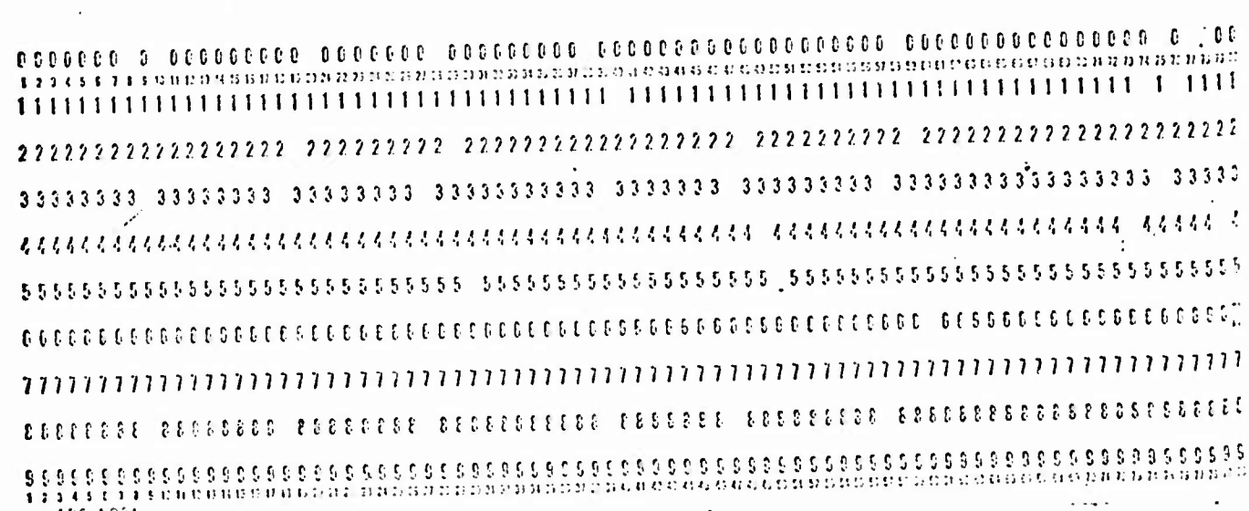
-0.4 -.340 -.045 -0.5 -.385 -.050 DATA0044



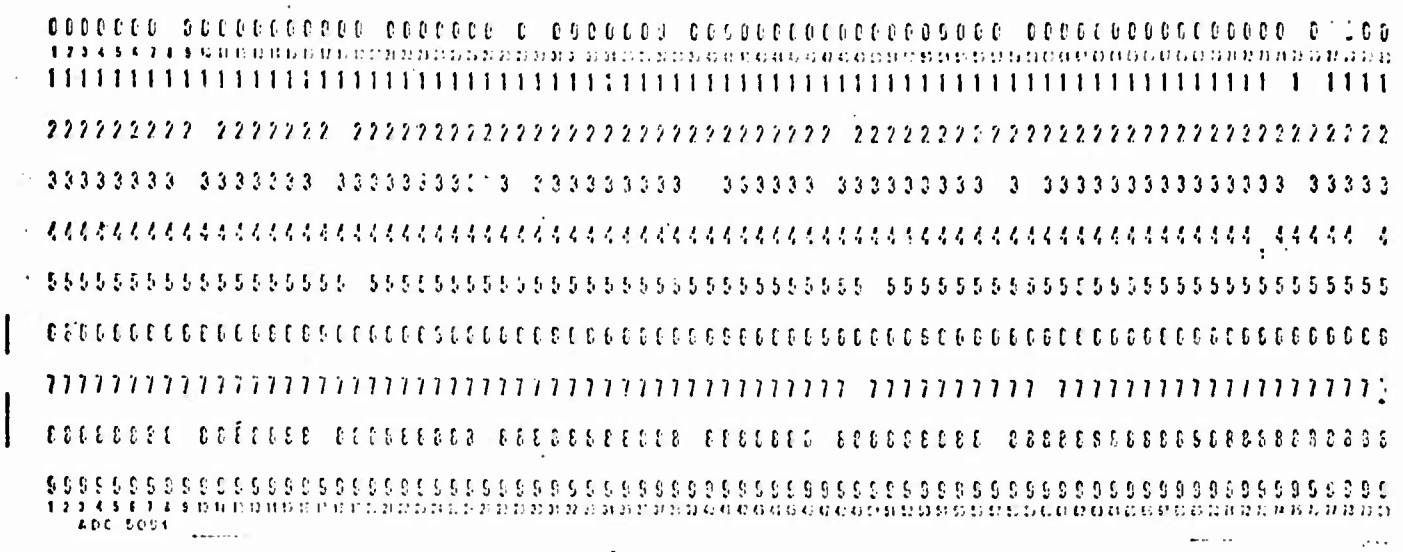
-0.6 -.420 -.055 -0.7 -.450 -.057 DATA0045



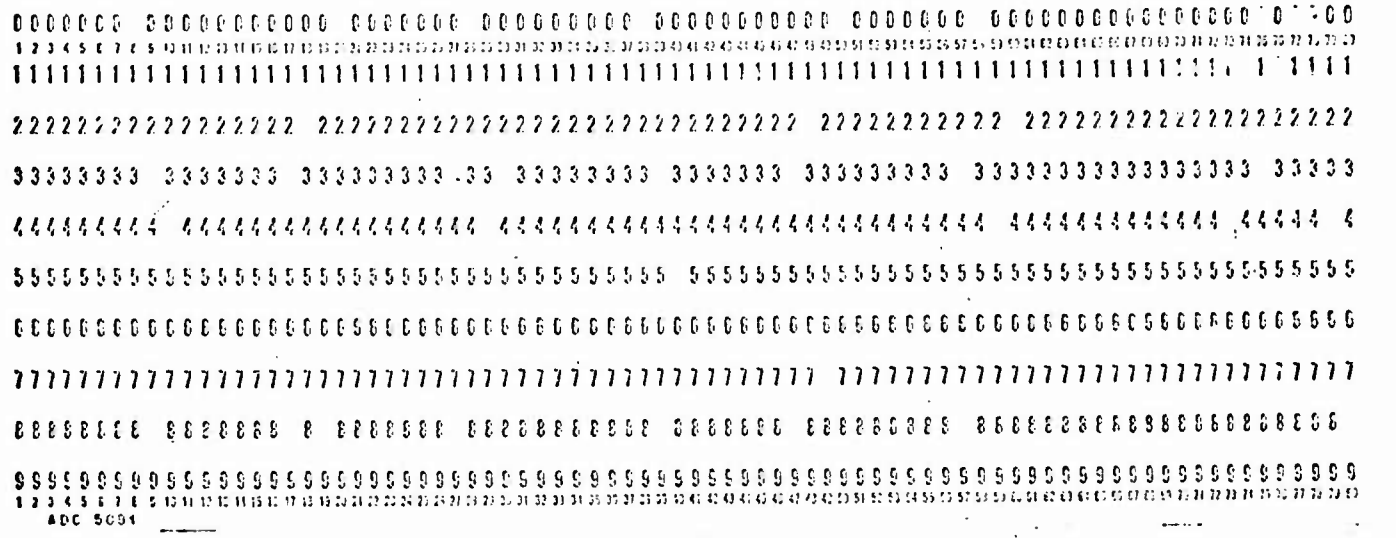
0.0 .20 .025 0.1 .245 .026 DATA0046



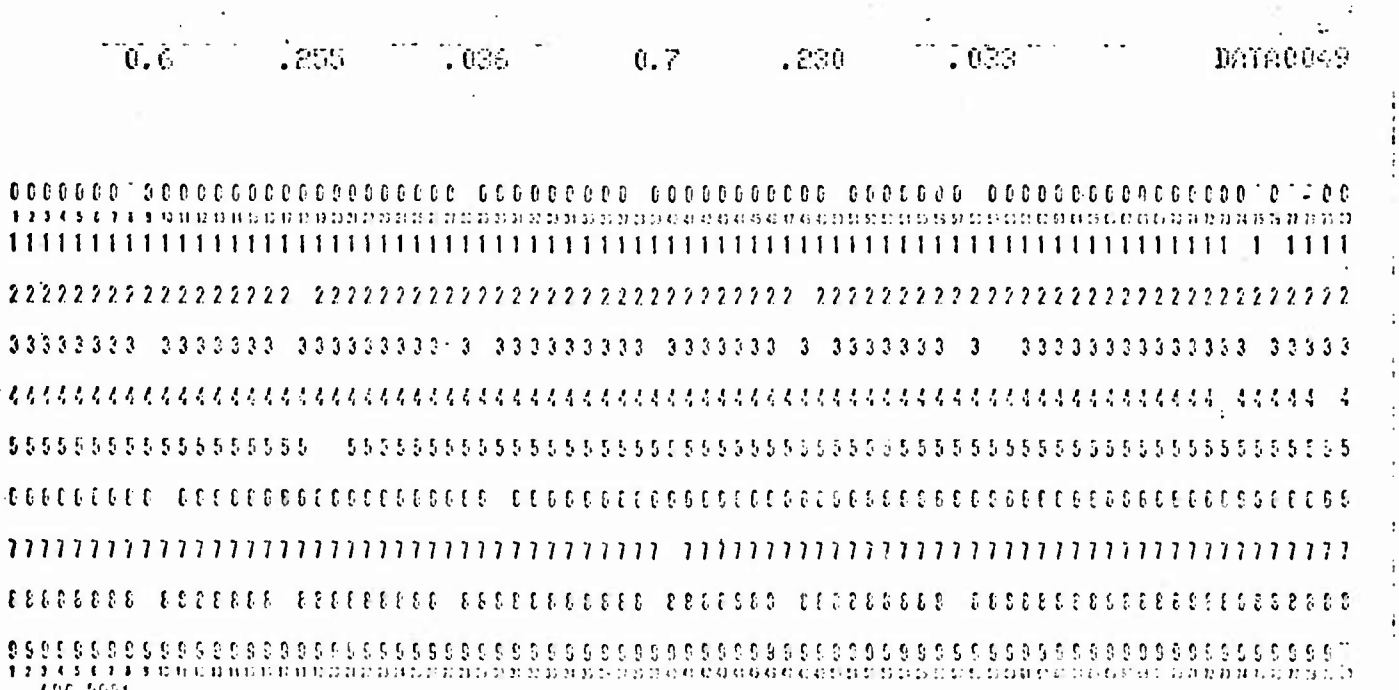
0.2 .250 .030 0.3 .275 .037 DATA0045



0.4 .280 .043 0.5 .270 .042 DATA0047



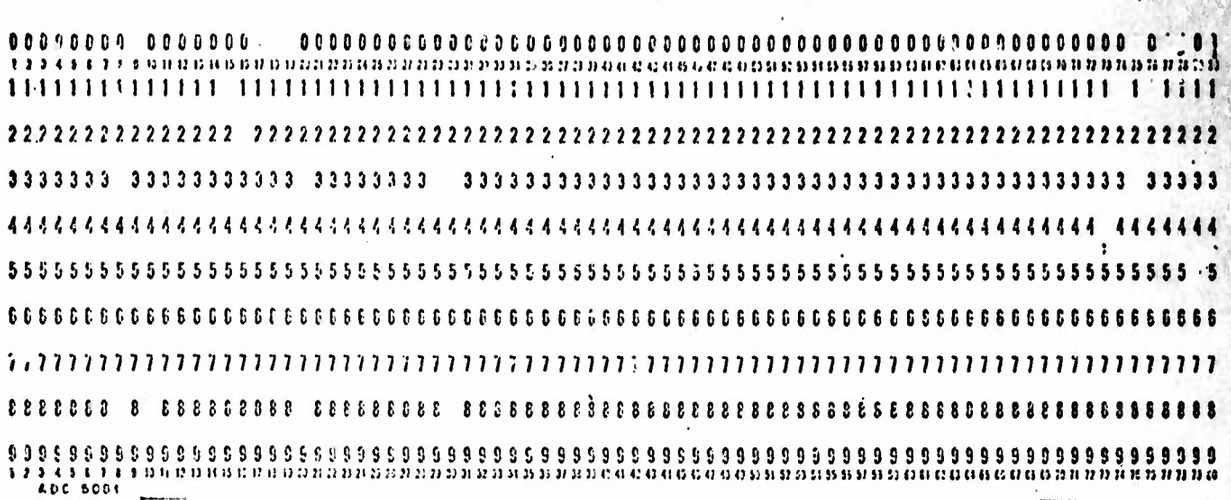
0.6 .255 .036 0.7 .230 .033 DATA0049



HULL DATA

.08 12000. 3. DATA0050

CAD DELTA XKK



Data Card 50

FORMAT 3F10.4

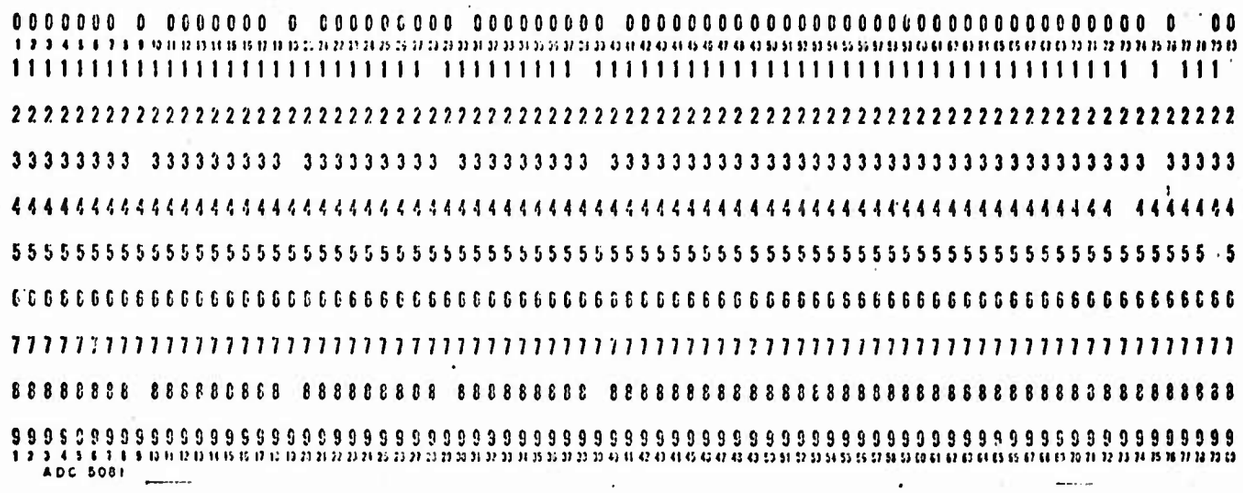
ELEMENTS

CAD = Added mass coefficient (non-dimensional)
Use 8%.

DELTA = Ship displacement (long tons)

XKK = Number of propellers.

```
      0.0       0.0       1.0       1.0                                    DATA0051
      VK        EHP(I)    WADED(I)  THDED(I)
```



Data Cards 51-70

FORMAT 10x, 3F10.4

ELEMENTS

VK = Speed of ship (knots)

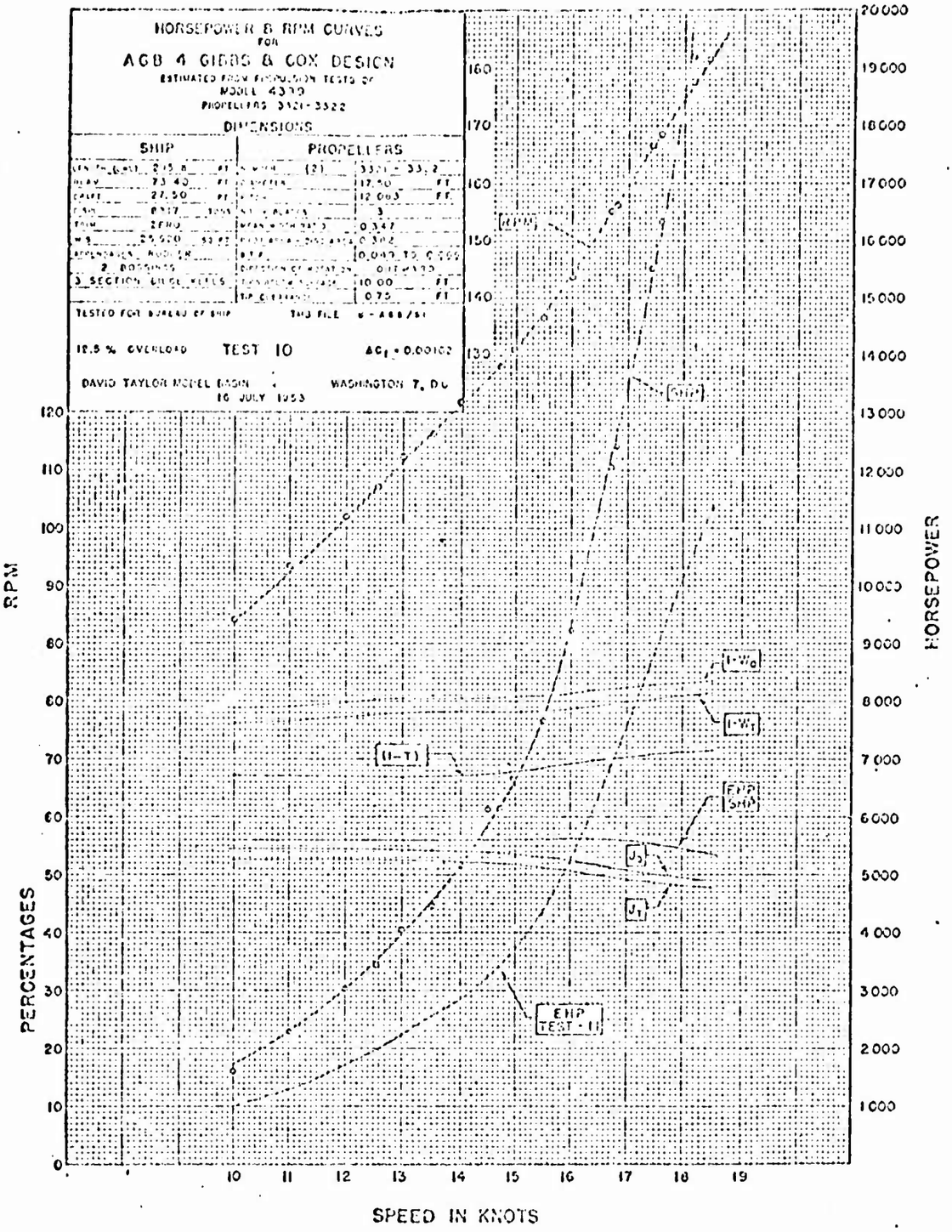
EHP = Effective horsepower (horsepower)

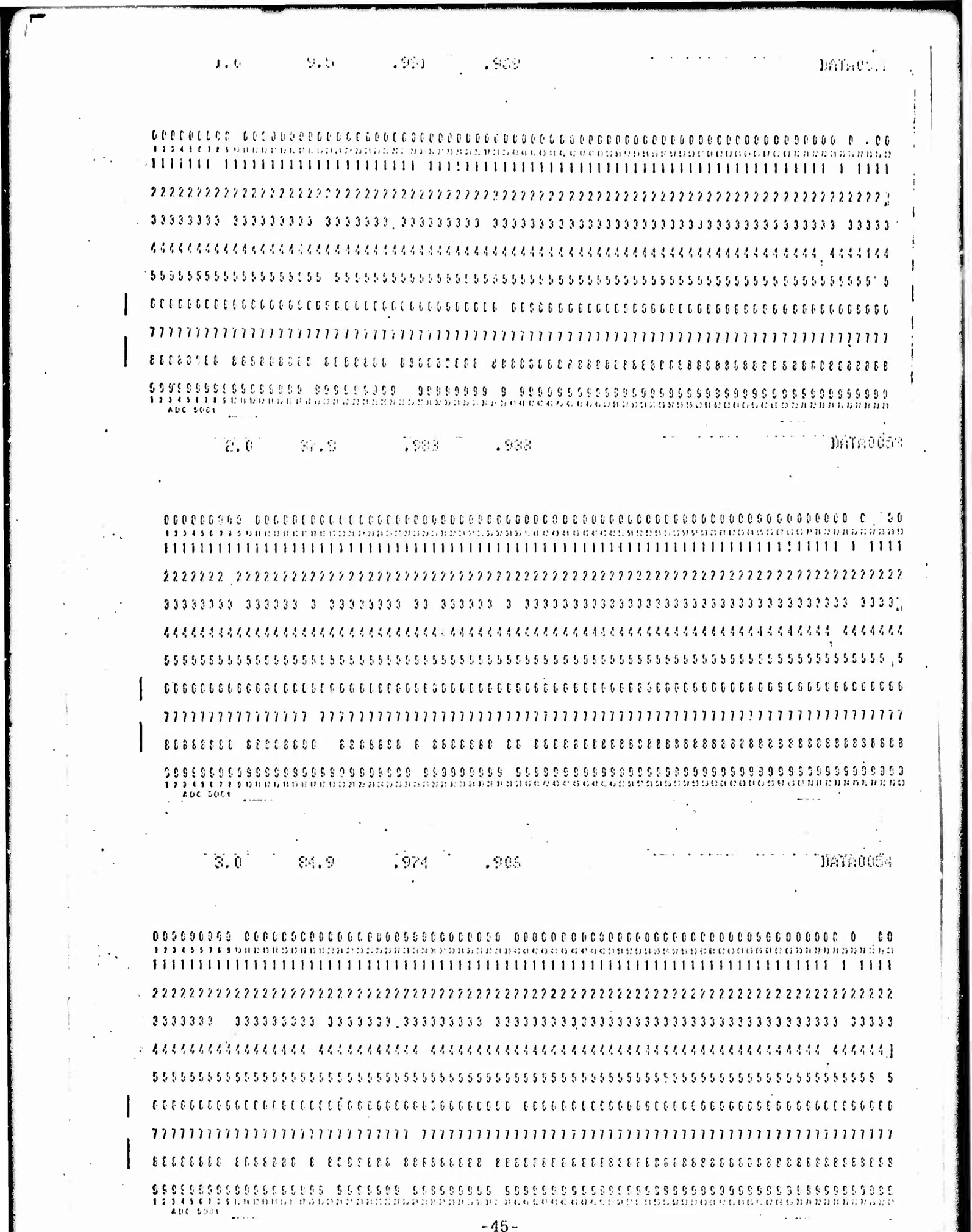
WADED Wake deduction ($1-W_T$) (non dimensional)

THDED = Thrust deduction (1-T) (non-dimensional)

Each of the elements are read from the ship curves such as those following. VK spacing must be as given as it is assumed by the program, not read. Values for EHP, 1-w, and 1-t not given on the curve can be found from straight line measurements back to origin for 1-t and 1-w and a square law for EHP.

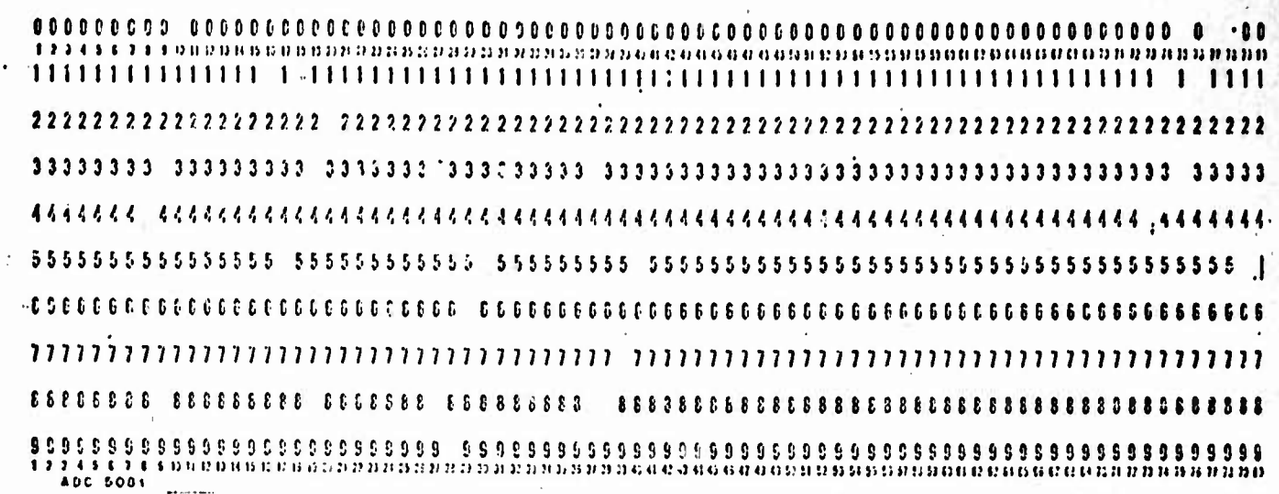
Note: Written description continues on page 53.


HORSEPOWER & RPM CURVES
FOR
AGB 4 GIBBS & COX DESIGN
ESTIMATED FROM PROPULSION TESTS OF
MODEL 4339
PROPELLERS 3321-3322
DIMENSIONS
SHIP
PROPELLERS
2 BOSSINGS
3 SECTION SHAFTING
TESTED FOR BUREAU OF SHIP
TMB FILE
12.5 % OVERLOAD
TEST 10
DAVID TAYLOR MODEL BASIN
WASHINGTON 7, D.C.
16 JULY 1953
RPM
PERCENTAGES
HORSEPOWER
SPEED IN KNOTS
(RPM)
(1-T)
(1-Wa)
(1-Wt)
J3
J1
EHP TEST - 11

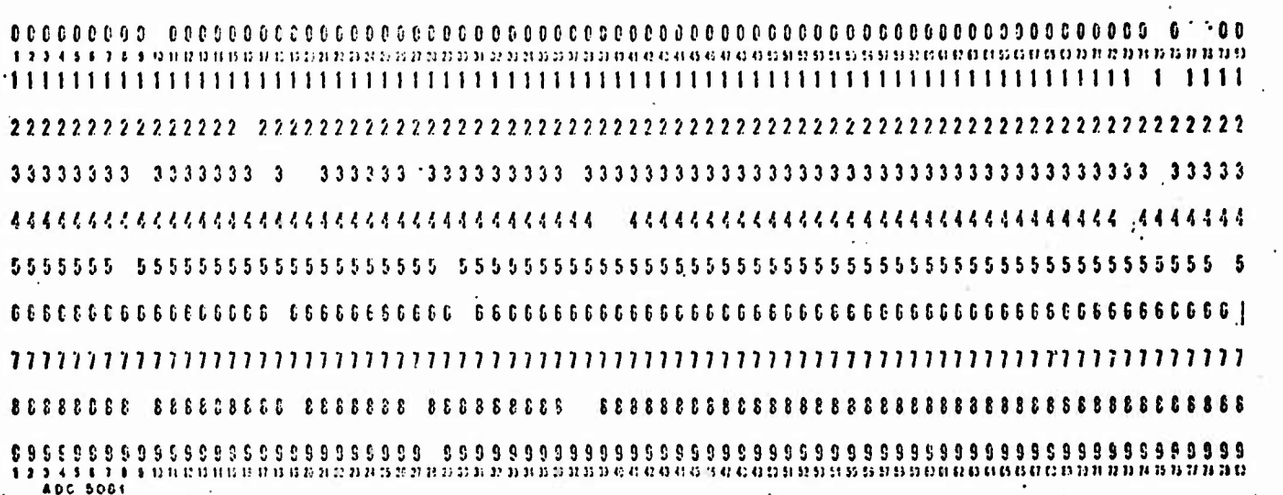


-45-

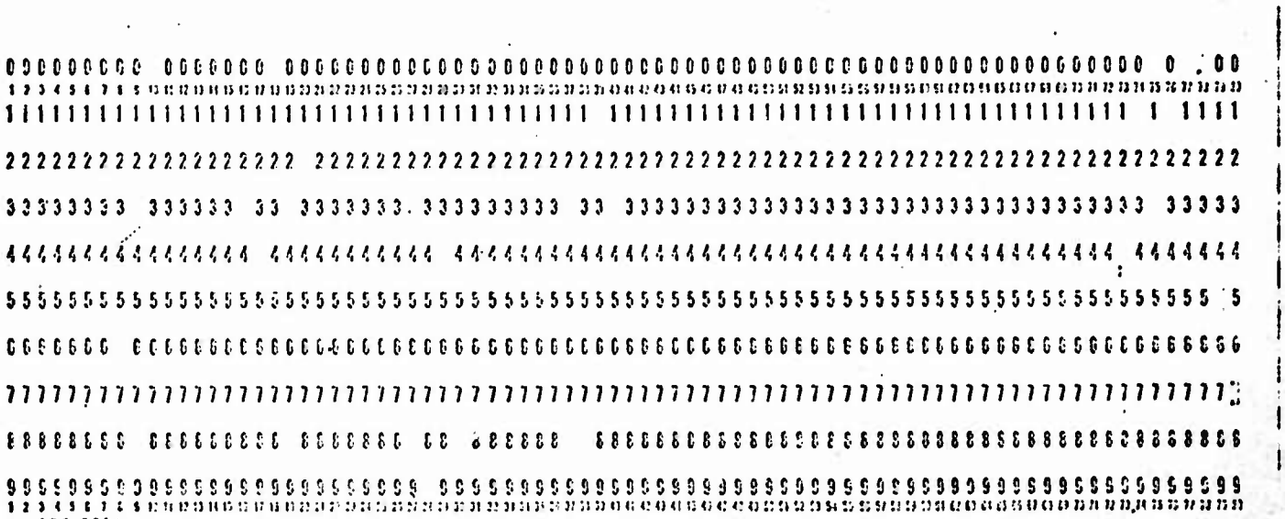
4.0 151.2 .965 .875 DATA0055

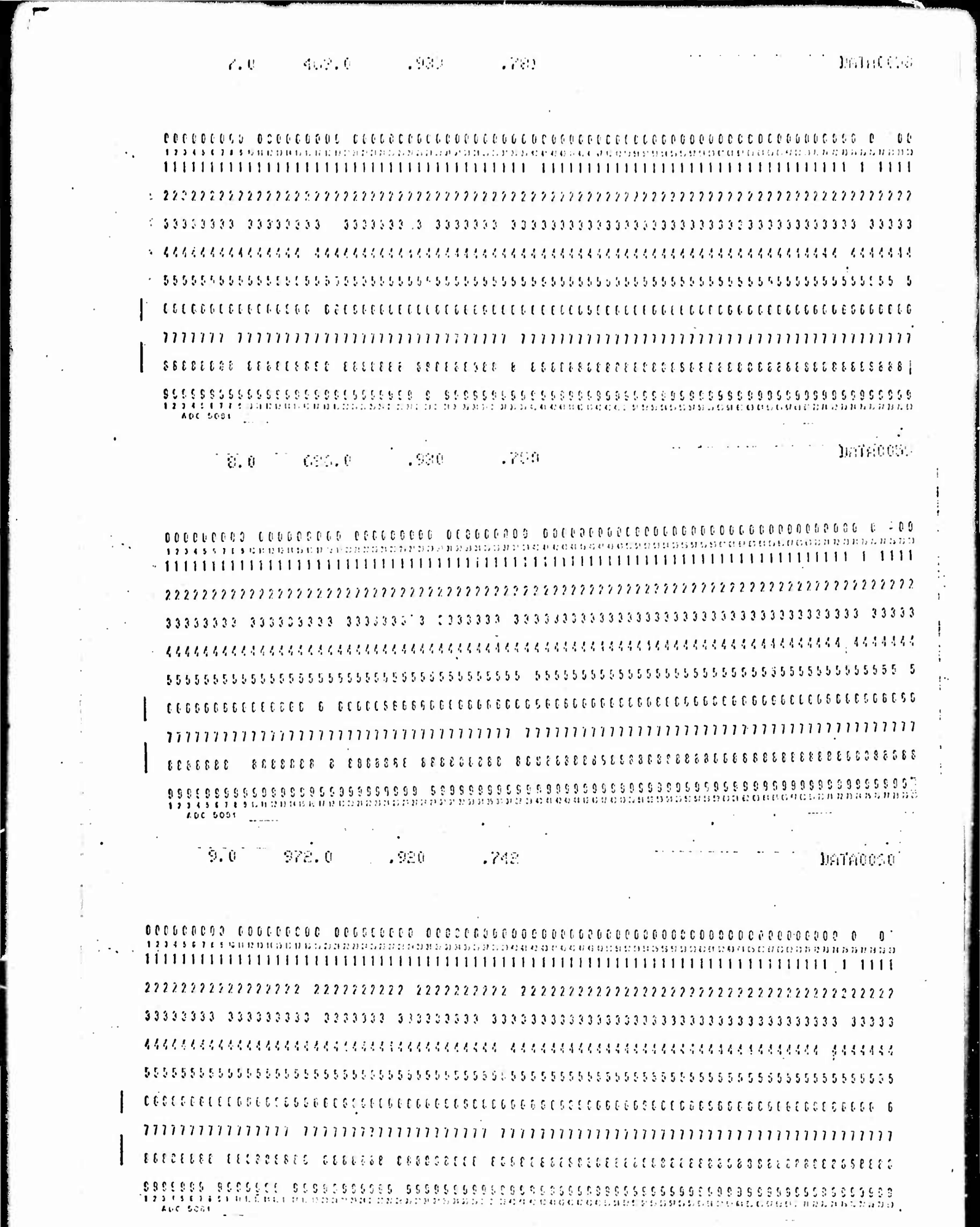


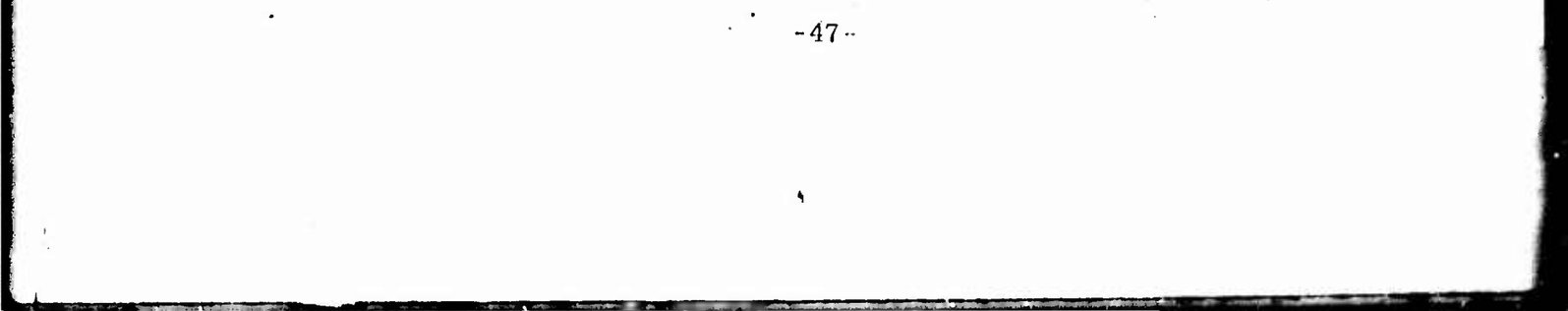
5.0 236.3 .956 .844 DATA0056



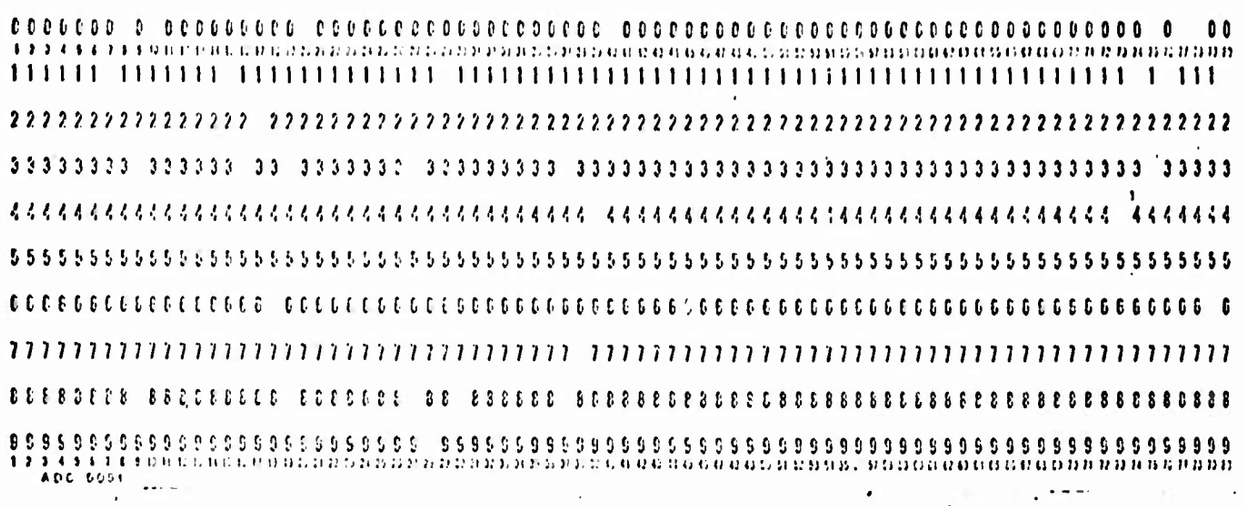
6.0 340.2 .948 .813 DATA0057



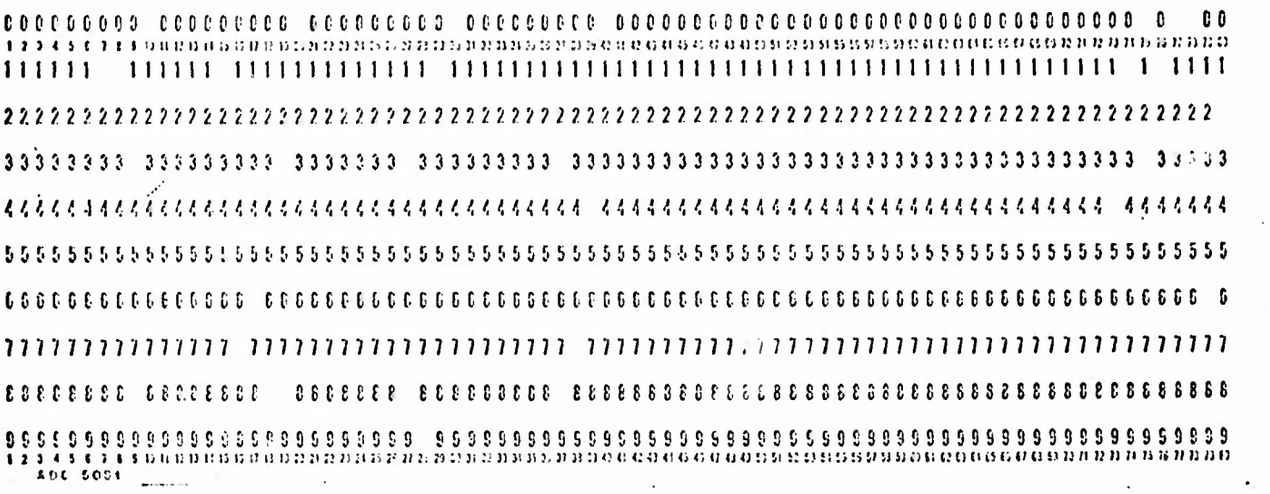

ADC 5001



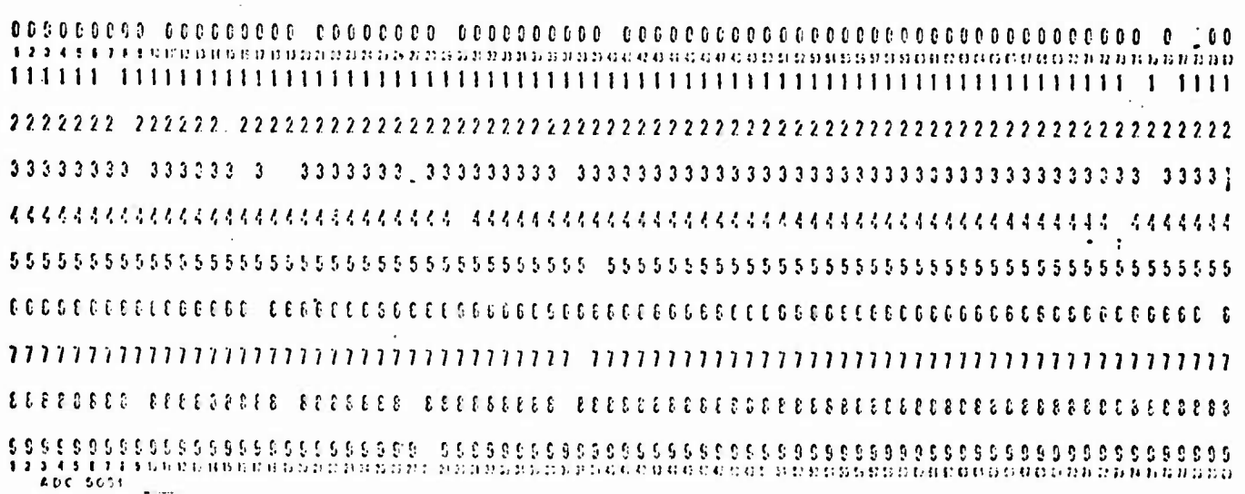
10.0 1500.0 .910 .740 DATA0061



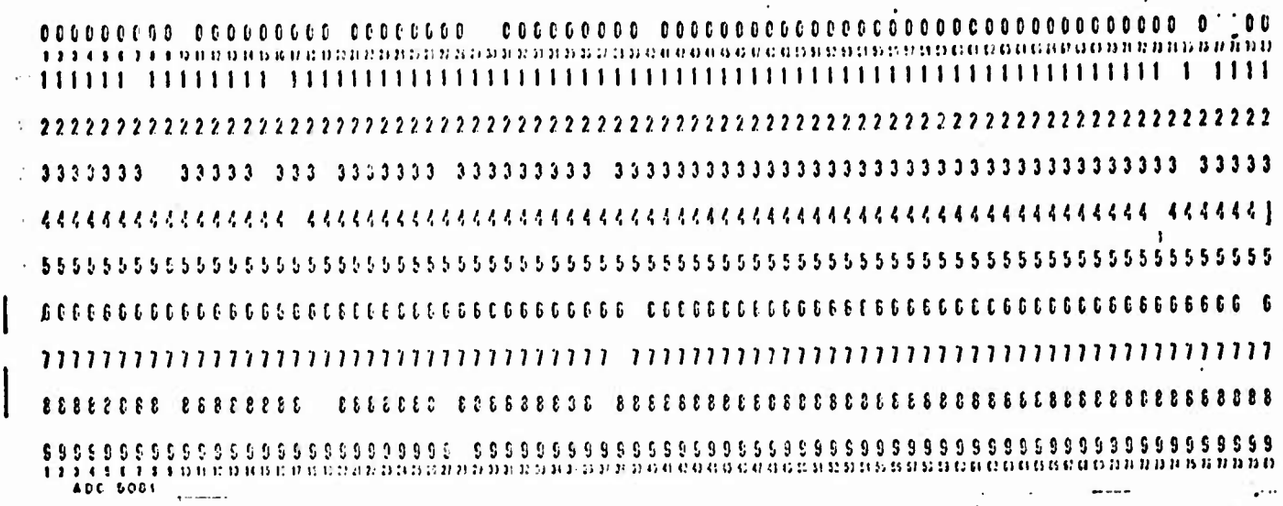
11.0 1708.0 .910 .740 DATA0062



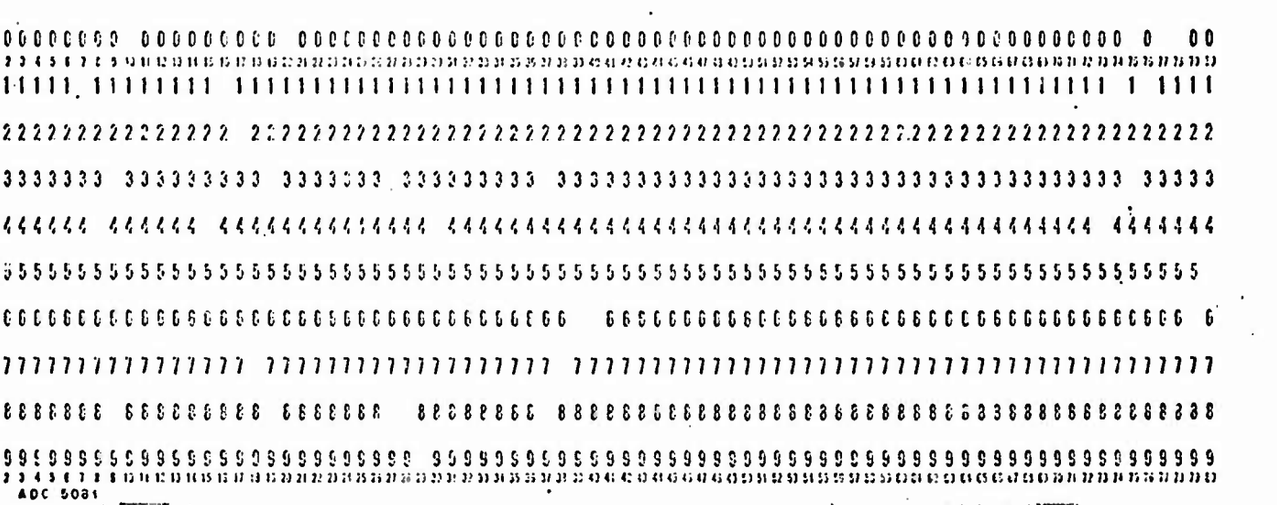
12.0 2363.0 .904 .750 DATA0063



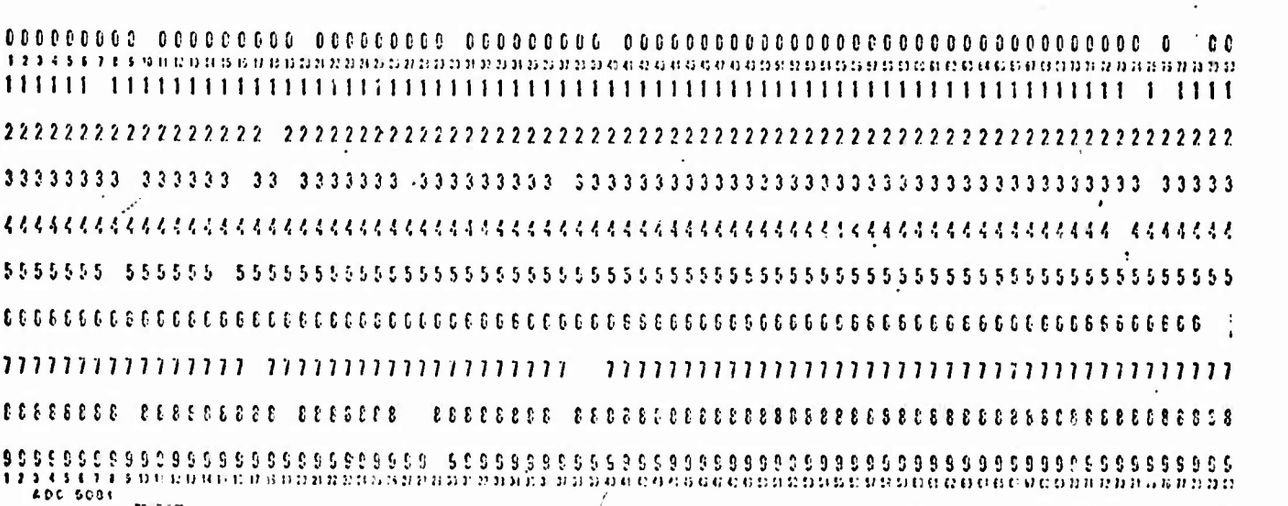
13.0 3148.0 .900 .760 DATA0064



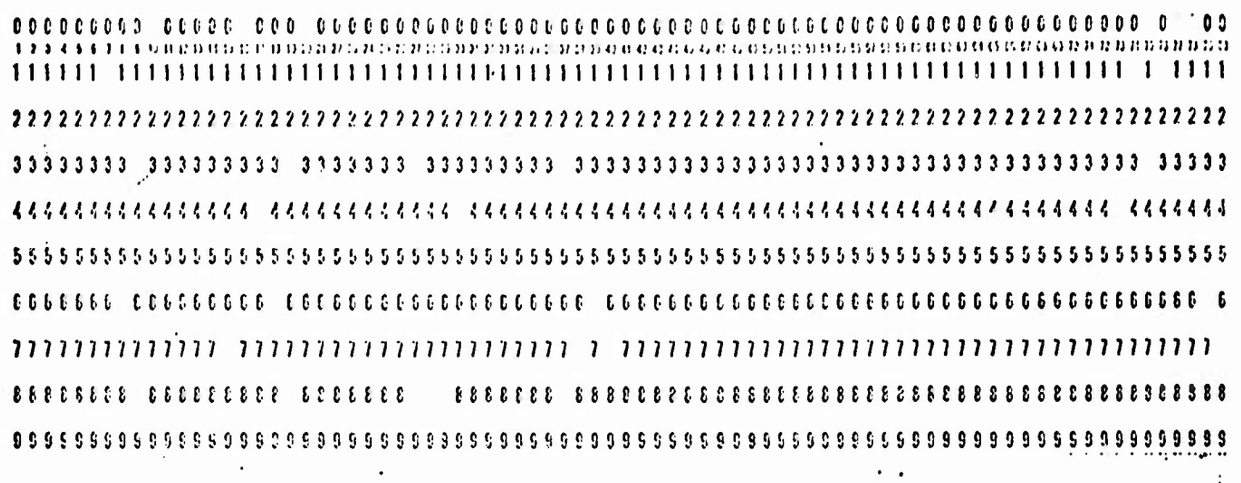
14.0 4127.0 .894 .766 DATA0065



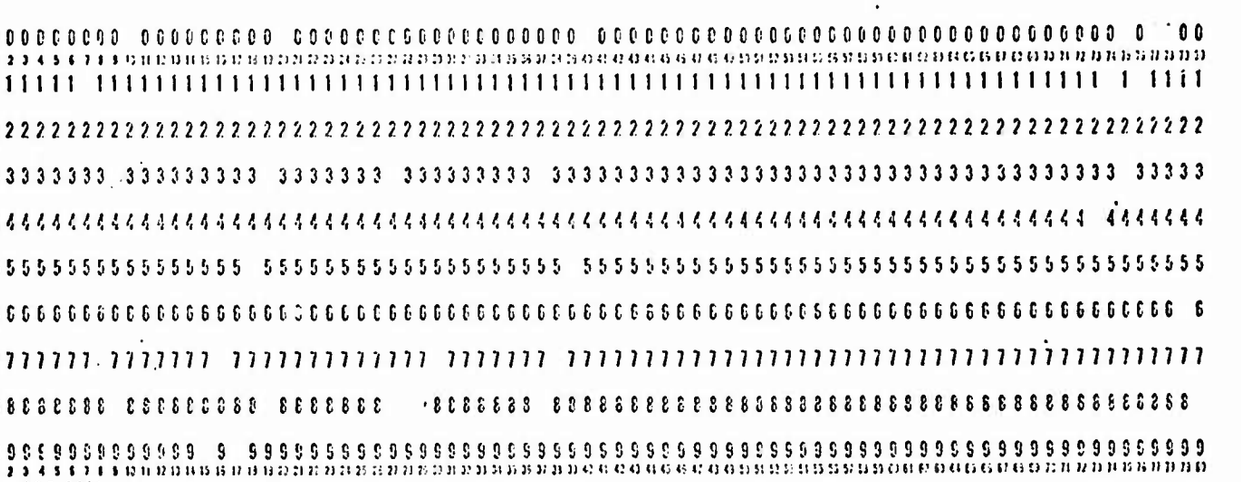
15.0 5872.0 .890 .770 DATA0066



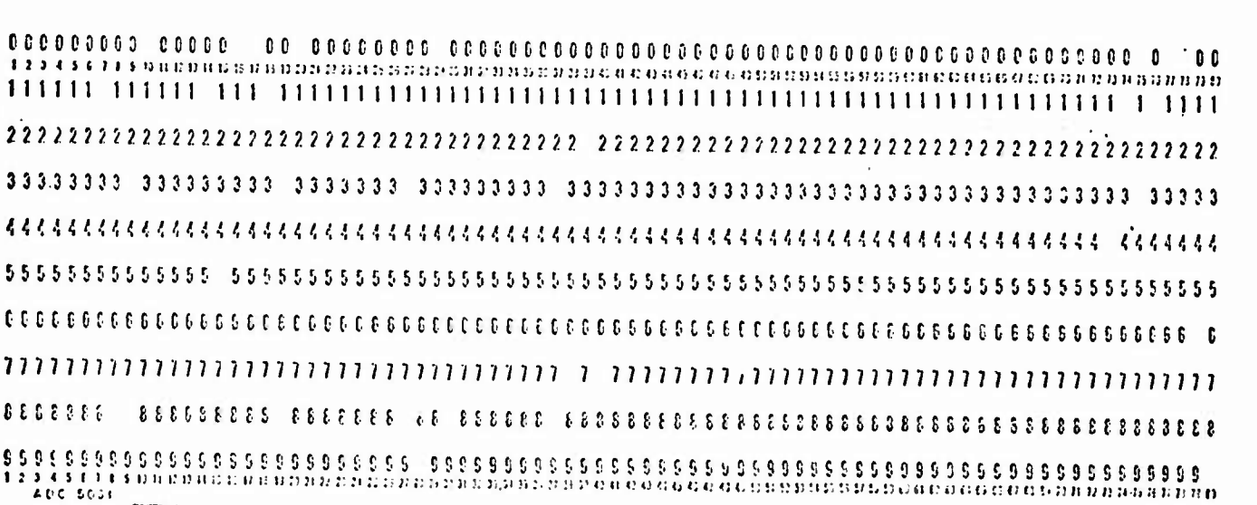
18.0 7046.0 .884 .767 DATA0067



17.0 9795.0 .887 .750 DATA0068



18.0 15001.0 .908 .727 DATA0069



```
19.0   25763.0      .908        .787                                    DATA0070

000000000 000000 00 00000000 0000000000000000000000000000000000000000000000000 0  0
123456789012345678901234567890123456789012345678901234567890123456789012345678901234
111111 1111111 1111111111111111111111111111111111111111111111111111111111111 1 1111

2222222222222 2 22222222222222222222222 22222222222222222222222222222222222222

33333333 333333333 3333333 333333333 33333333333333333333333333333333333 33333

44444444444444444444444444444444444444444444444444444444444444444444444444 4444444

55555555555555555555555555555555555555555555555555555555555555555555555555555555

66666666666666666666666666666666666666666666666666666666666666666666666666666666

77777777777777777777777777777777777 7 7777777777777777777777777777777777777 7

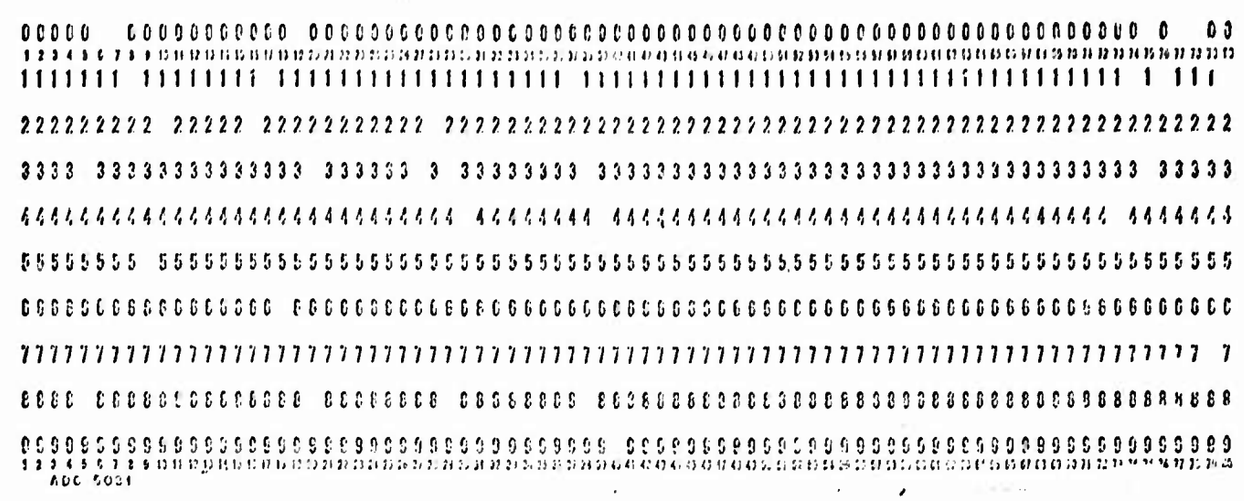
88888888 88888888 88888888 88 88888888 88888888 88888888

99999999 999999999 999999999 999999999 999999999
123456789012345678901234567890123456789012345678901234567890123456789012345678901234
  ADC 9001
```

ELECTRICAL DATA

.0015 2160. 32.4 1.1 DATA071

RM TRNFM ZMP RFM



Data Card 71

FORMAT 4F10.6

ELEMENTS

Each of these elements is found from data on the equipment such as from the certification data shown for the GLACIER. Note that values are on a per shaft basis.

RM = Resistance of motor armature circuit including the compensating windings. (ohms)

TRNFM= Turns of motor excitation or shunt field. For example on the GLACIER there is 90 turns per pole, 12 poles in series, two series paths

2 x 12 x 90 = 2160

ZMP = Z_m' = flux conversion factor (volts/weber-rpm) from

$$e_m = \frac{P_1 \phi Z_s}{P_2 \cdot 60}$$

p_1 = number of poles

p_2 = number of parallel armature paths

ϕ = flux per pole

Z = number of armature force conductors

s = armature speed in rpm

$$Z'_m = \frac{P_1 Z}{P_2 \cdot 60} \quad \text{and em} = e_m = Z'_m \Phi s$$

data for this is read from the technical manual or design sheets for the equipment.

RFM = Resistance of motor shunt field (ohms)

2. [illegible]

(a) Manufactured by Westinghouse Electric Corporation.

(b) Propulsion Generator for Dual-Drive.

(c) Number of Generators - 10 per Vessel.

(d) Contract No. W.S. Corp., NS-41000.

(e) Contract No. Navy - NObs 7956.

(f) Shop Order Number - 179717, 522 & 699.

(g) 1 Navy - U.S. Navy Destroyer 4000.

3. [illegible]

The generator will conform to Military Specifications MIL-G 3151A and below.

4. [illegible]

Title	Mfg's No.	No. of This Fig No.
Outline & Shaft	1-378-1935	364-D-M-347846
[illegible]	1-378-1935	364-D-M-347848
[illegible]	1-378-1937	364-D-M-347852
Winding Diagram & Field Data	1-378-1938	364-D-M-3493475
Commutating Plan & Tools	1-378-1939	364-D-M-3464826
Air Gauge	Service Engineer Dwg. E-852707	364-D-M-3477825

5. [illegible]

	Continuous	1 Hour	1 Min.
(a) KW	1350	1700	1700
(b) Volts	437	500	700
(c) Amperes	3608	3390	2520
(d) RPM	720	720	720
(e) Temp. Rise	70°C.	85°C.	65°C.

(f) Temp. Rises - Windings will not exceed rises values stated when supplied with 40°C. air out of the cooler with 65°F. sea water.

6. [illegible]

(a) Generator Complete - 29,450 lbs. (With Bedplate Dwg).

(b) Complete Armature - 4,540 lbs.

(c) Field Coils Each (Approx.)
Shunt - 247 lbs., Commutating - 95 lbs.

7. [illegible] - (Rotating Rectifier)

(a) 1350 KW - 96

(b) 1700 KW - 93.75

8. [illegible]

(a) Material - Rolled Steel.

(b) Split - Along Horizontal Centerline.

9. [illegible]

(a) Number - 6.

(b) Nominal Air Gap - 0.188 Inches.

10. [illegible]

(a) Number - 6.

(b) Nominal Air Gap - 0.375

11. [illegible]

(a) Type - Sleeve.

(b) Material - Bearing Shells are cast steel lined with babbitt metal - Westinghouse Alloy #75.

(c) Lubrication - Force lubricated from Engine lubrication System.

(d) Diameter - 5.

(e) Length - 5.

(f) Grade of Oil Recommended - Navy Symbol Oil 2190T.

12. [illegible]

(a) Refer to "Winding Diagram and Field Data" Drawing.

(b) Turns per Pole - 145.

(c) Connection - 2 Parallel Circuits.

(d) Resistance at 75°C. - 1.5 Ohms.

(e) Removable Coil - Yes.

13. [illegible]

(a) Refer to "Winding Diagram and Field Data" Drawing.

(b) Number of Turns in Each Pole Face - 7.

(c) Number of Turns per Pole of Commutating Field - 5.

(d) Connection - All in Series.

(e) Resistance at 75°C. - 0.001315 Ohms.

14. [illegible]

(a) Refer to "Winding Diagram and Field Data" Drawing.

(b) Number of Turns per Pole - 3.

(c) Resistance at 75°C. - 0.00095 Ohms.

(d) Connection - 2 Parallel Circuits.

15. [illegible]

(a) Diameter - 32 Inches. Length of Core - 19 Inches.

(b) Bonding - Refer to "Winding Diagram and Field Data" Drawing.

(c) Method of Securing Laminations to Shaft - The armature punchings are assembled on the armature spider with 0.000 to 0.002 inch interference and held together with end plates keyed to the spider.

(d) Number of Slots - 69.

(e) Conductors per Slot - 4.

(f) Number of Parallel Paths - 6.

(g) Resistance at 75°C. - 0.00595 Ohms.

16. [illegible]

(a) Diameter - 25 Inches.

(b) Type - Two Ring - Refer to "Brushrig Data and Assembly" Drawing.

(c) Mounting Depth - 1/2 Inch per Side.

(d) End Play Tolerance - 1/16 Inch.

(e) Insulation to Ground - 1/32 Inch Mica (2 Layers - 1/32 Inch half overlap).

17. [illegible]

(a) Number of Brush Arms - 6.

(b) Number of Brushes per Arm - 5 x 2 (Double Brushholders)

(c) Type of Brushes - Carbon
Length 2-1/2, Width 1-1/4, Thickness 1/2

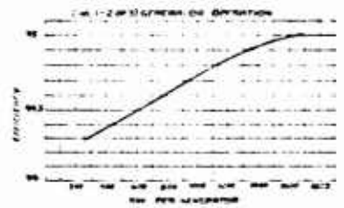
(d) Grade Name of Brush - National Carbon Co. - Grade 3345.

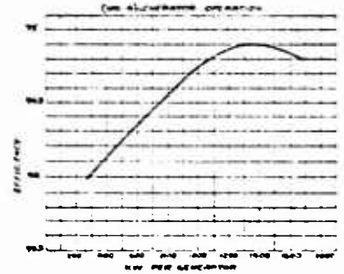
(e) Duty Grade Letter - "A".

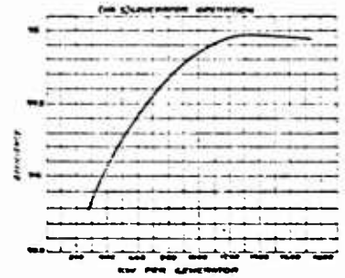
(f) Angle to Radial - 25°.

(g) Type of Pigtail Connection - The shunt of 13,500 MCM stranded cable is [illegible] to a terminal block [illegible] to the brushholder and this ring is connected to the brush with the same stranded cable. [illegible] Each shunt of this group has three [illegible] of the same size as is used for the shunt connection. The shunt length is 7-1/2 inches.

EFFICIENCY vs KW

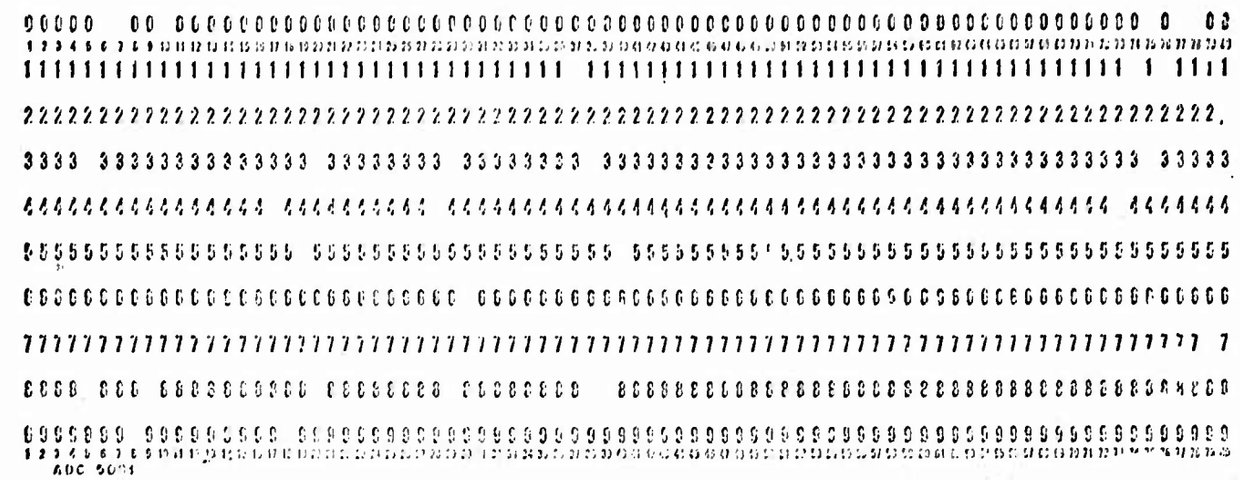






.00950 195. 1.5 1.85 DATA0072

RG TRNFG ZGP RFG



Data Card 72

FORMAT 4F10.6

ELEMENTS

Each of these elements is found from data on the equipment such as from the certification data shown for the GLACIER. Note that values are on a per shaft basis.

RG = Resistance of generator armature circuit including series field and commutating windings.

TRNFG = Turns of generator shunt field winding/pole.

ZGP = Z'_g = flux conversion factor (volts/weber-rpm)from

$$e_g = \frac{P_1 \phi Z s}{P_2 \cdot 60}$$

$$Z'_g = \frac{P_1 Z}{P_{260}} \quad \text{and} \quad e_g = Z'_g \phi s$$

data for this is read from the technical manual or design sheets for the equipment.

RFG = resistance of generator shunt field (ohms).

1. IDENTIFICATION

(a) Manufactured by Westinghouse Electric Corporation.

(b) Propulsion motor for Ice Breaker.

(c) Number of motors - 2 per Vessel.

(d) Contract No. U.S. Corp. - NObs-1108.

(e) Contract No. Navy - NObs 5548.

(f) Shop Order No. 17-0-513.

(g) 1 Ship - U.S. Navy Ice Breaker ABM.

2. APPLICABLE SPECIFICATIONS

The motor described hereunder will conform to Military Specifications NAVSHIPS 250-660-10 dated 1 July 1949 EXCEPTIONS: A - The motor furnished on the thrust end of the machine will be marked THAT END IN THE DIRECTION IN WHICH SHAFT ROTATES WHEN VIEWED FROM THE OUTSIDE OF THE SHAFT. B - THE SHAFT EXTENSION OF THE MOTOR WILL BE CONSTRUCTED TO AN ANGLE OF 45° MEASURED FROM VERTICAL.

3. STANDARD RATINGS

Title	RPM's, Hr.	No. of Ships Furn Rtg
Outline & Shaft	1-20-1146	36,500'-W-2177,179
Cylindrical Roller Thrust Bearing	1-20-1913	AGO-0-36,500'-W-1,493,195
Wiring Diagram & Field Data	1-20-1918	36,500'-W-2,197 249
Mounting Plan & Field Data	1-20-1919	36,500'-W-2,494,497
Speed-Torque & Temp.	1-20-1750	36,500'-W-2,171 249
AIR COOLED	(ambient temperature cannot exceed 40°C)	

4. RATED

(a) 525/500 Volts.

(b) 600/5700 Amps.

(c) 8,50/10,500 HP.

(d) 110/120/175 RPM.

(e) Temperature - Windings will not exceed 70°C, while supplied with 40°C. air out of cooler with 65°F. sea water and 65°C. after 4 hours at 10,500 HP.

5. MECH. CLASSIFICATION

(a) Degree of Enclosure - Enclosed at the commutator end, dripproof at the rear end, water tight up to ventilating opening in rear end bell.

(b) Class of Insulation - Class B.

(c) Method of Cooling - Forced ventilated by motor driven ventilating fans, double-inlet air-to-water coolers, and filter in the air discharge opening at commutator end, located in top of front end bell.

(d) Service Application - Ice Breaker propulsion.

(e) Ambient Temperature of Enclosure - 40°C.

(f) Type of Winding (Field) - Form wound.

(g) Manufacturer's Type - CF.

6. CURRENT - CHARACTERISTICS

(a) 8,150 HP, 137 Volts, 5000 Amps., 102 RPM - 94.5%.

(b) 10,500 HP, 500 Volts, 5300 Amps., 120 RPM - 94.5%.

7. WEIGHTS

(a) Motor, Complete - 813,500 lbs.

(b) Armature, Complete - 95,700 lbs.

(c) Field Coils Each (Approx.)
Steel - 1,50 lbs., Commutating - 130 lbs., Compensating - 90 lbs.

(d) Non-Magnetic Material - Total Wt. - 15,500 lbs.

(e) Magnetic Material - Total Wt. - 189,500 lbs.

8. SHAFT DATA

(a) Material - Solid Steel.

(b) Spline on horizontal centerline.

9. MAIN POLES

(a) Number of Poles - 24.

(b) Nominal Air Gap - 0.500 in.

10. COMMUTATING POLES

(a) Number of Poles - 24.

(b) Nominal Air Gap - 0.562 in.

11. FRAMES

(a) Type - Sheets.

(b) Material - Bearing shields are cast steel lined with babbitt metal, Westinghouse Alloy #25.

(c) Lubrication - Forced lubrication from a separate motor driven pump.

(d) Diameter - 97 in.

(e) Length - 13 in.

(f) Grade of Oil Recommended - Navy Symbol Oil 2190.

12. MAIN FIELD WINDING

(a) Refer to "Winding Diagram and Field Data", Drawing No. 1-20-1913.

(b) Turns per Pole - 34.

(c) Connection - 12 Coils in Series, 2 Coils in parallel.

(d) Resistance - 0.1 ohms at 75°C.

(e) Coils Removable - Yes.

13. COMMUTATING WINDING

(a) Refer to "Winding Diagram and Field Data", Drawing No. 1-20-1913.

(b) Turns per pole - 6.

(c) Connection - 12 coils in series, 2 coils in parallel.

(d) Resistance - 0.00079 ohms at 75°C.

14. COMPENSATING FIELD WINDING

(a) Refer to "Winding Diagram and Field Data", Drawing No. 1-20-1913.

(b) Turns per pole - 8.

(c) Connection - 12 coils in series, 2 coils in parallel.

(d) Resistance - 0.0082 ohms at 75°C.

15. ARMATURE

(a) Diameter - 111 inches, Length of core - 17 inches.

(b) Bonding - Refer to "Winding Diagram & Field Data" Drawing No. 1-20-1913.

(c) Method of Securing Laminations to Shaft - Armature Punchings are expanded and held to armature spider with two dovetails per laminations. The laminations are held together with end plates keyed to the spider.

(d) Number of Slots - 288.

(e) Conductors per Slot - 6.

(f) Resistance at 75°C. - 0.00020 ohms.

(g) Number of parallel paths - 24.

16. COMMUTATOR

(a) Diameter - 77-3/8 inches.

(b) Type - CF.

(c) Wearing Depth - 5/8 inch per side.

(d) Bar Mica Undercut - 1/32 inches.

(e) Insulation to Ground - 1/16 inch mica (2 layers - 1/32 inch half overlap).

17. BRUSHES

(a) Number of Brush Arms - 24.

(b) Number of Brushes per Arm - 24 (

(c) Size of Brushes - Length 2-1/2 in., Width 1-3/4 in., Thickness 1/2 in.

(d) Trade Name of Brushes - National Carbon Co., Grade AB-45.

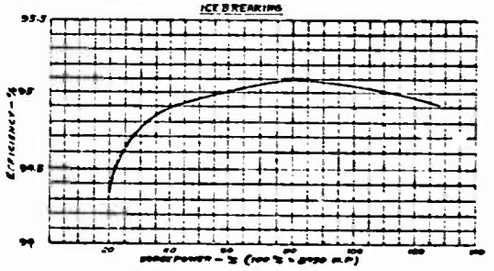
(e) Duty Grade Letter - "D".

(f) Angle to Radial - 25°.

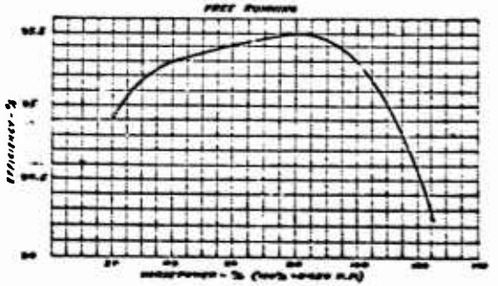
(g) Type of Pigtail Connection - The shunt of 113-0.007 copper cable is attached to a copper plate clip by pressure and this clip is riveted to the brush with two rivets which hold the contact plate. Both ends of the shunt are tinned to an open end spade clip for bolting to the brushholder. The shunt length is 3-1/2 inches.

EFFICIENCY vs. HORSEPOWER

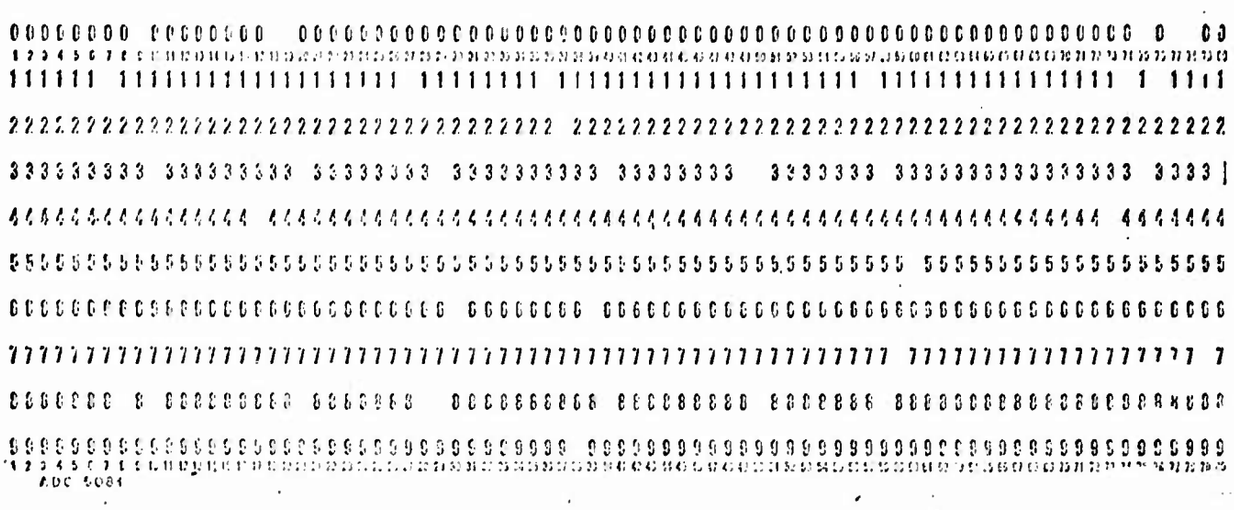
ICE BREAKING



FREE RUNNING



130.	40%.	15.5	1250.	3.	1.75	
FIMAX	C1	XK1	CSHFR	TRNSF	RX	



Data Card 73

FORMAT 6F10.4

ELEMENTS

FIMAX = Strong field (bollard conditions) value of motor field current (amps)

C1 = offset of motor controller curve (volts)(see following figure)

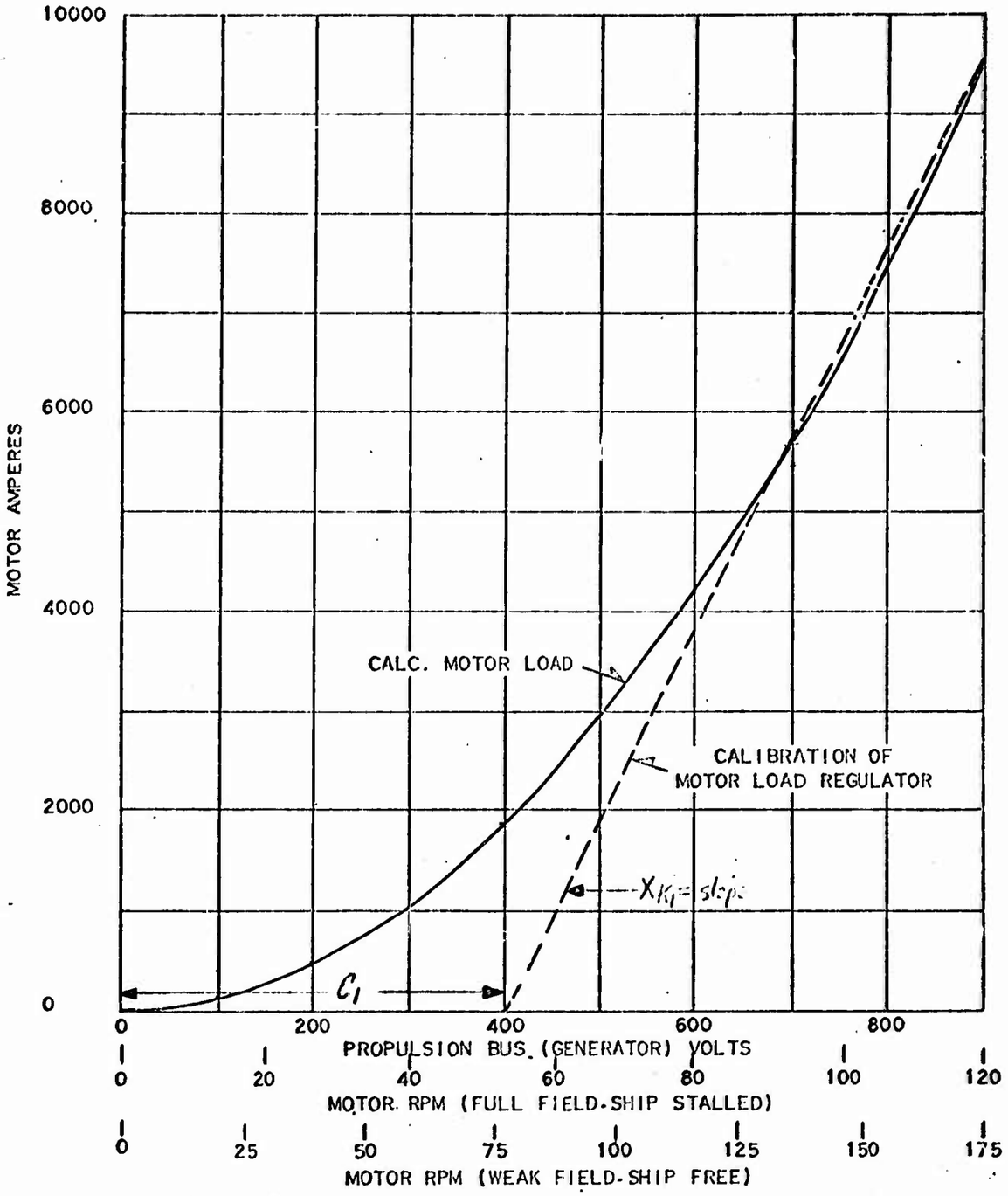
XK1 = slope of motor controller curve (amps/volt) (see following figure)

CSHFR = shaft frictional coefficient (foot-lb-sec^2)

$$Q_{rost} = C_{SHFR}\,(n_p)\,(\,|n_p|\,)$$

TRNSF = generator series field turns/pole

RX = resistance (ohms) of the variable resistor in the generator field circuit.



· *Approximate Motor Load Currents*

0.0 0.0 0.0

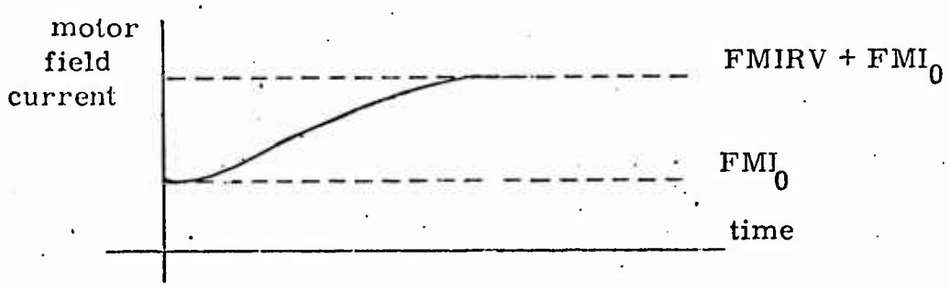
FMIRV TAUA TAUB

Data Card 74

FORMAT 3F10.4

ELEMENTS

On the GLACIER, the motor field is plugged in the reverse direction during a crash back approximately as follows:

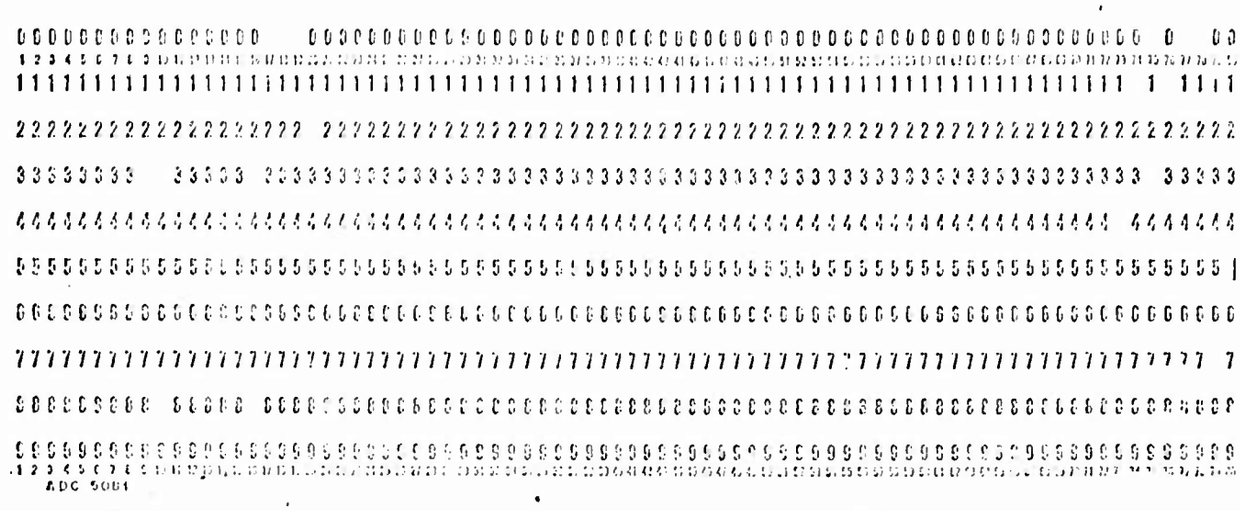


FMIRV = value in amperes per above sketch

TAUA, TAUB = exponent values to fit the following equation to the above sketch.

$$FMI = FMI_0 + \frac{FMIRV}{TAUA - TAUB} (TAUA\, e^{-t/TAUA} - TAUB\, e^{-t/TAUB})$$

XGEN XLA



Data card 75

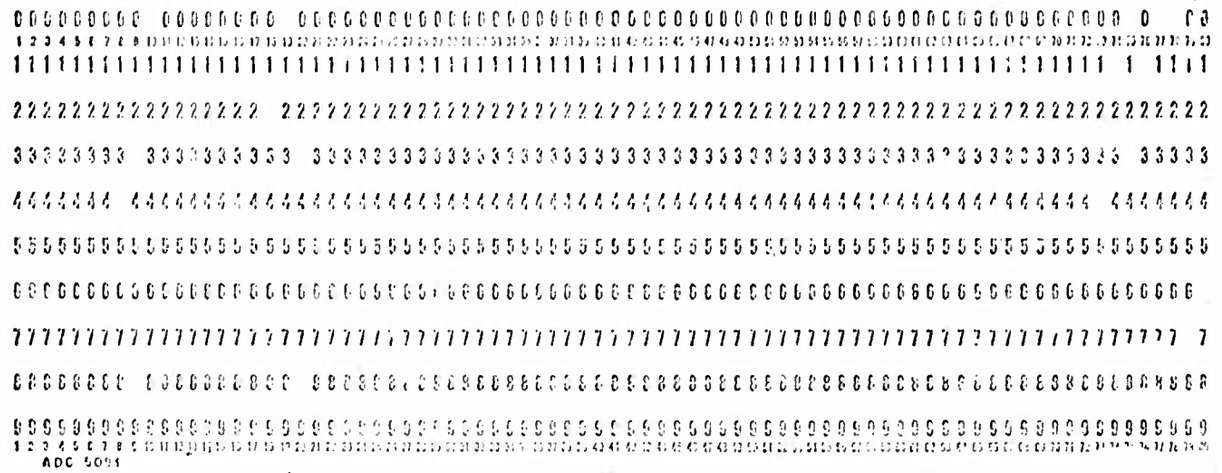
FORMAT F10.4,F10.7

ELEMENTS

XGEN = number of diesel generator sets

XLA = total inductance of armature circuit for one shaft (henries)

SPGEN SPMOT



Data Card 76

FORMAT 2F10.4

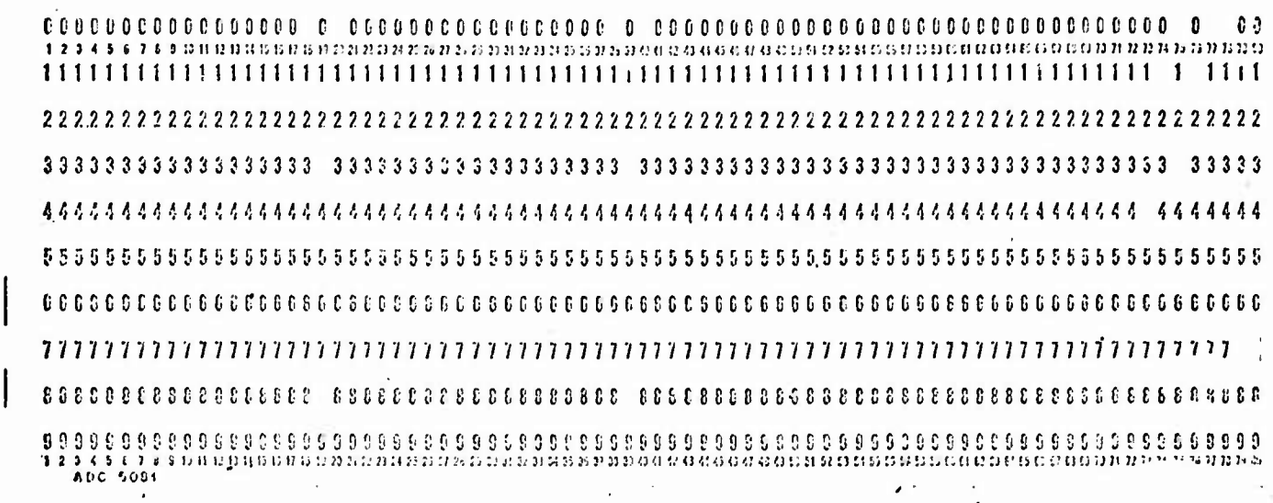
ELEMENTS

SPGEN = abscissan spacing between points on the generator flux curves (amperes)

SPMOT = abscissan spacing between points on the motor flux curves (amperes)

0.0 0.0 DATA0077

CFGA(I) CFMA(I)



Data Cards 77-89

FORMAT 2(10x, F10.4)

ELEMENTS

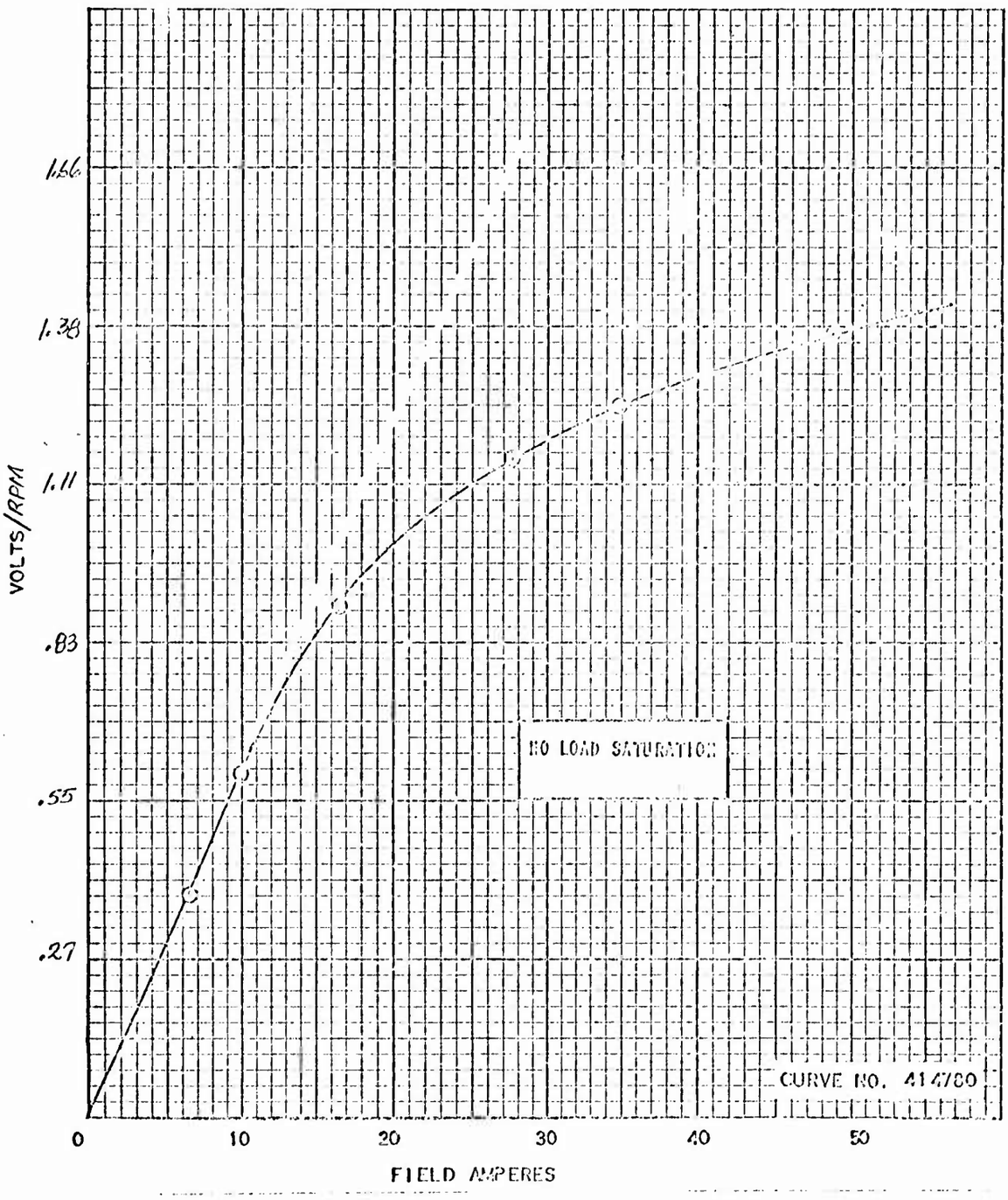
The elements are derived from the generator and motor flux curves (see following figures)

CFGA = constant times the generator field magnetic flux (volts/rpm)

CFMA = constant times the motor field magnetic flux (volts/rmp)

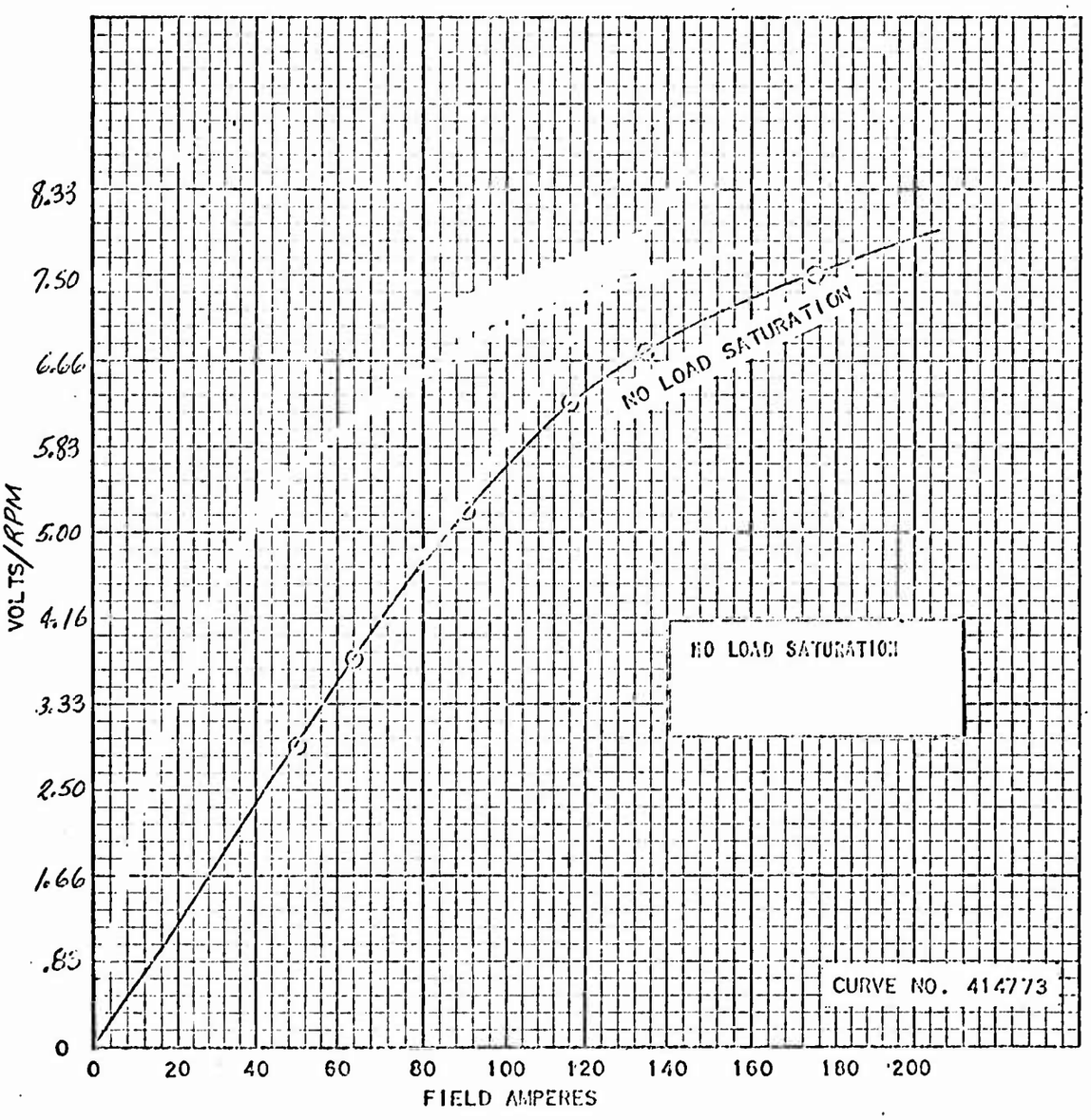
Note: Written description continues on page 71 .

FRAME NO. CC-219.5 TYPE QH D.C. GENERATOR
1340 KW - 837 VOLTS - 1600 AMP
6 POLES 720 RPM
U.S. NAVY CONTRACT NObs 2884



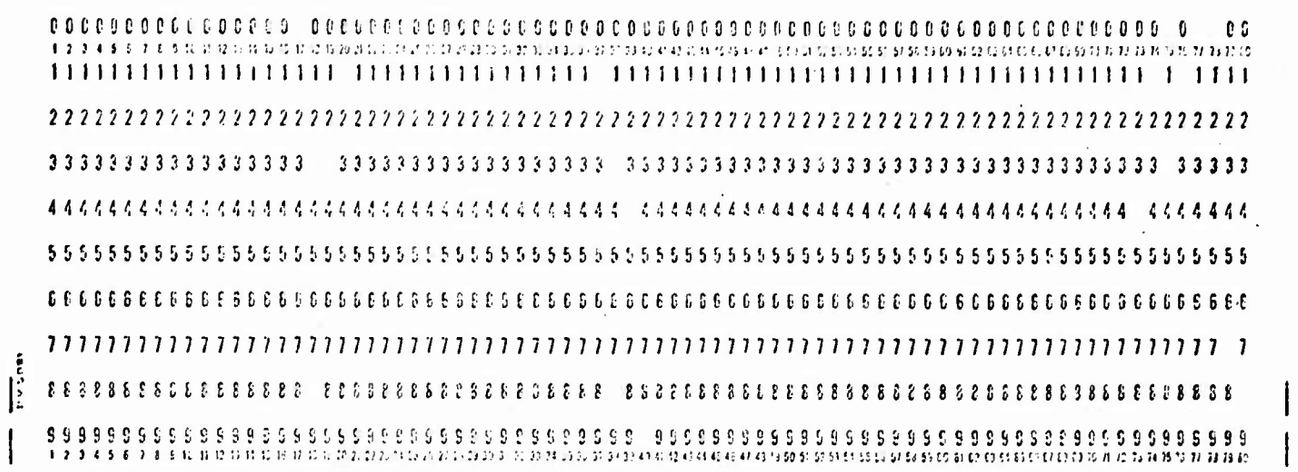
Saturation Curves of Propulsion Generator

FRAME NO. LM-429.7 - TYPE QV D.C. MOTOR
8450/10500 H.P. - 837/900 VOLTS - 8000/9300 AMPS
24 POLES - 112/175 RPM
U.S. NAVY CONTRACT NObs 2981

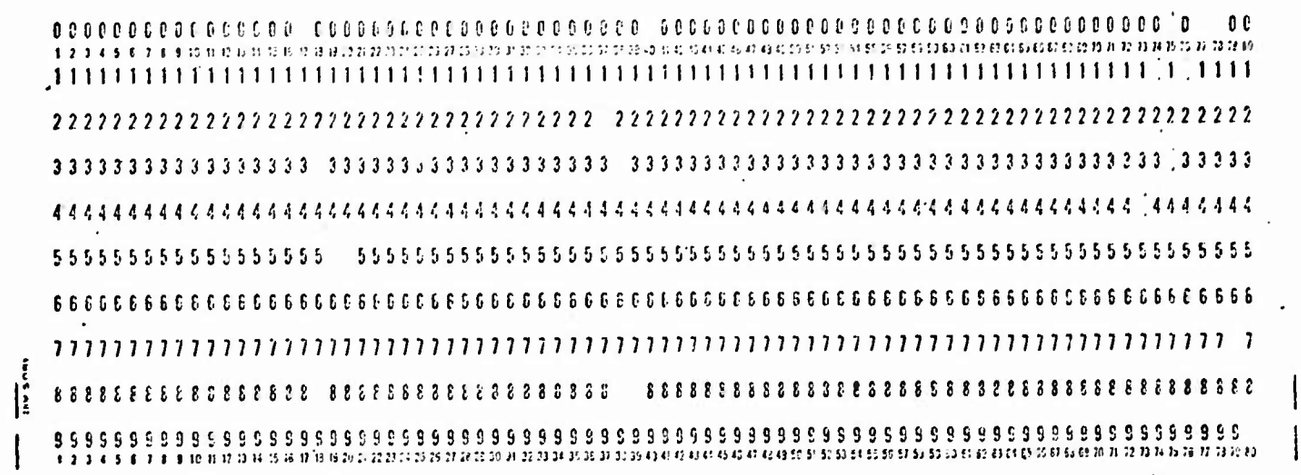


Propulsion Motor Saturation Curve

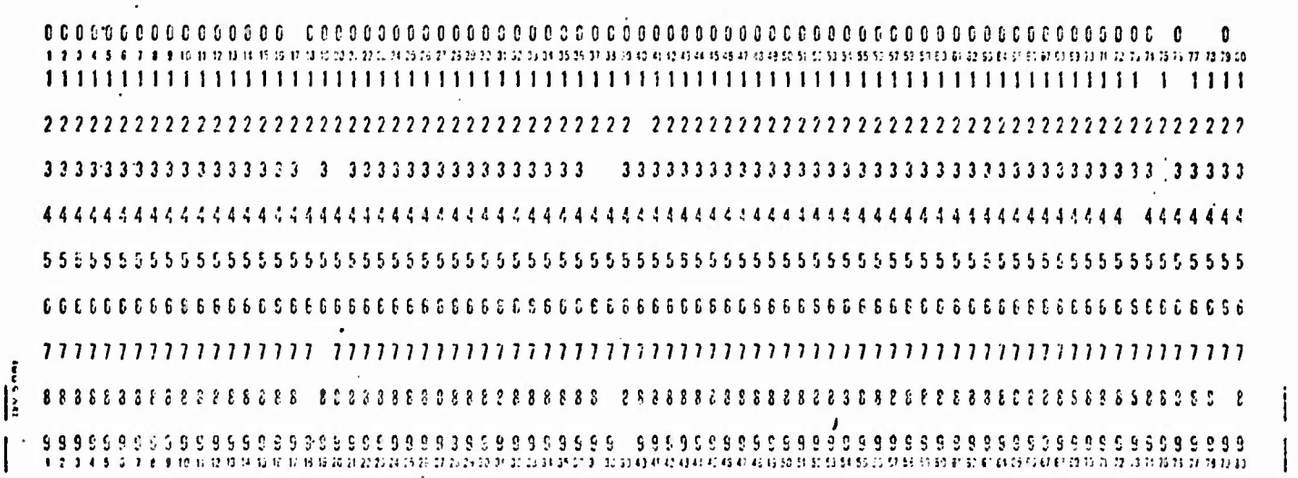
0.31 1.49 DATA0078



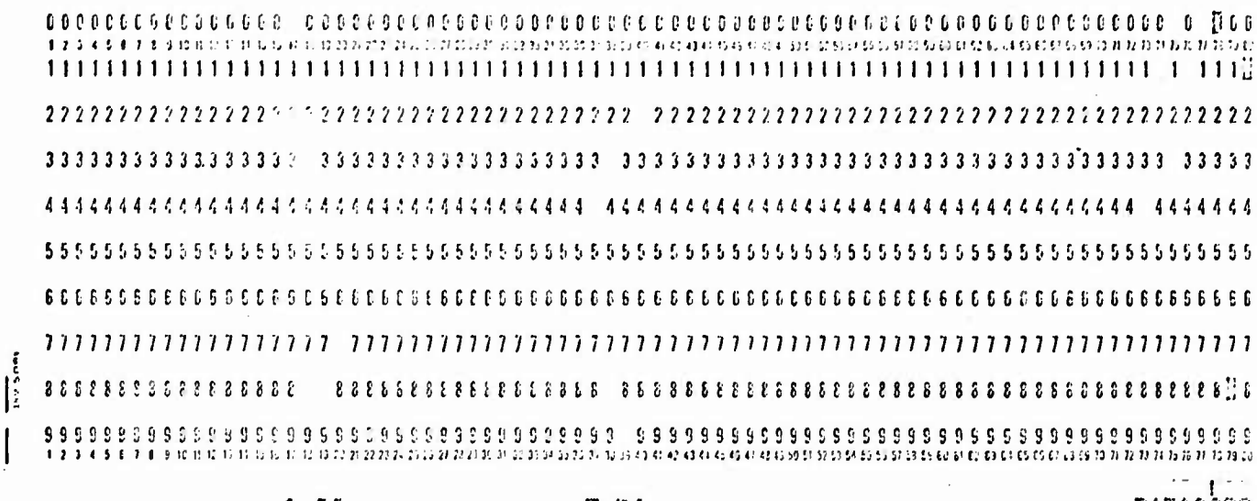
0.56 2.80 DATA0079



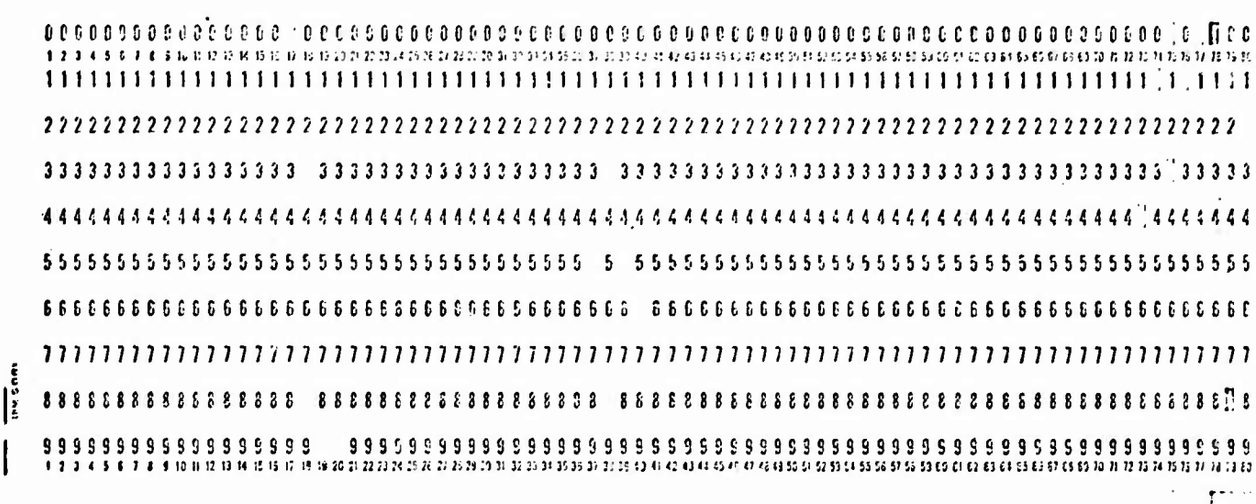
0.73 3.92 DATA0080



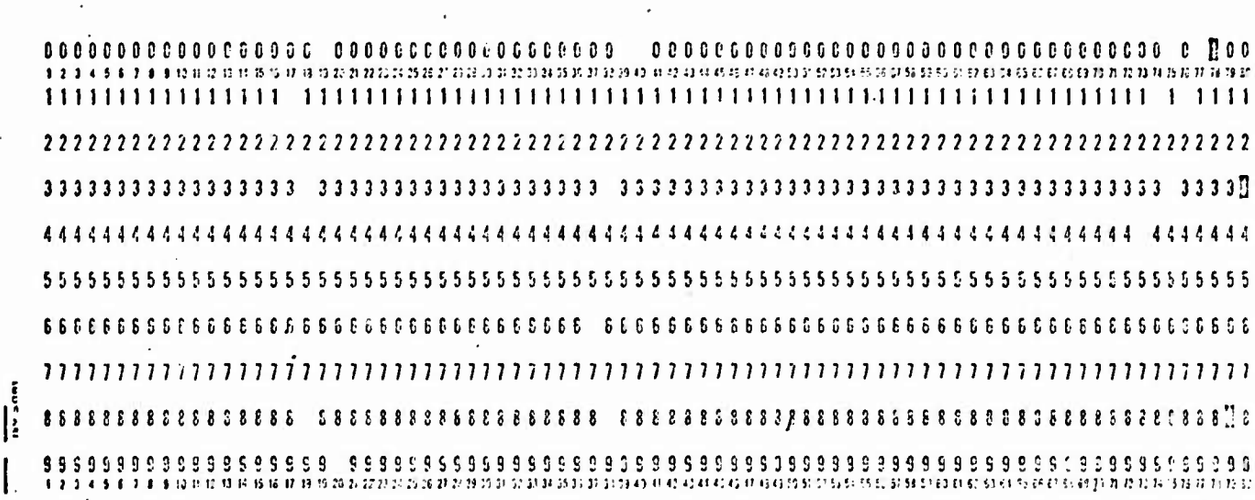
0.87 4.92 DATA0081



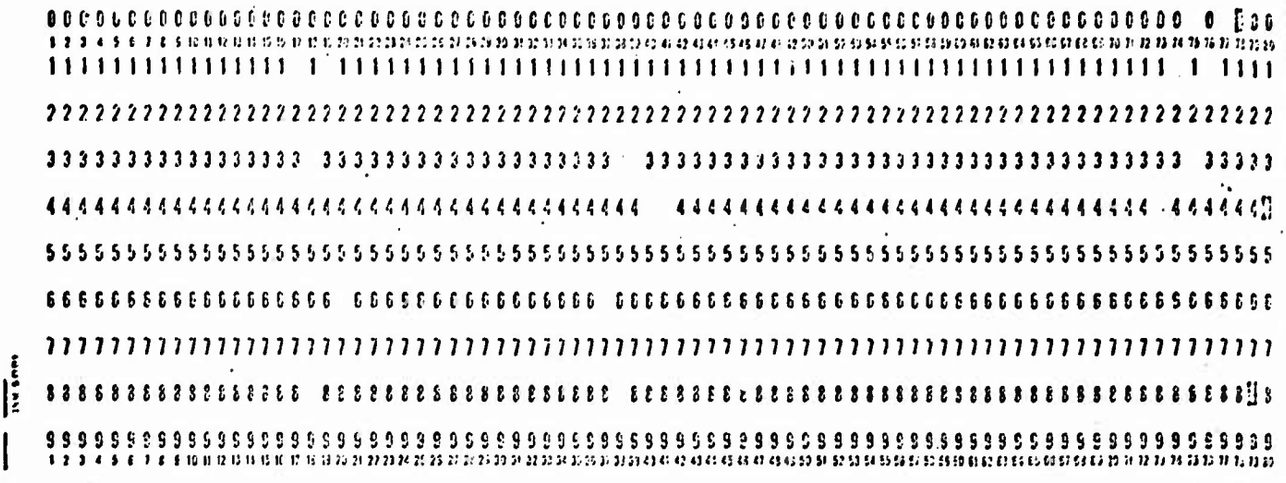
0.99 5.56 DATA0082



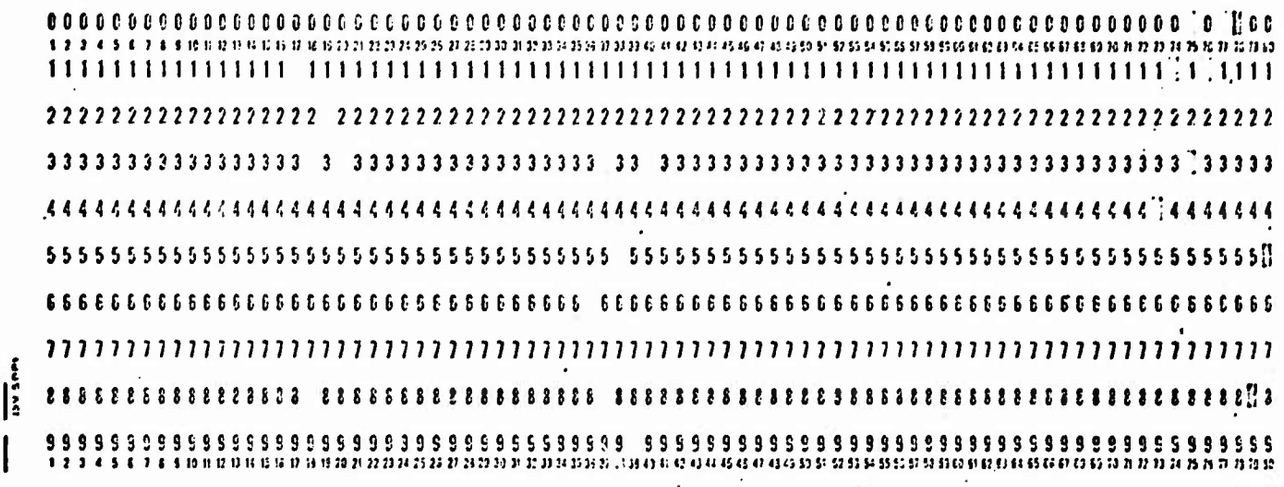
1.09 6.00 DATA0083



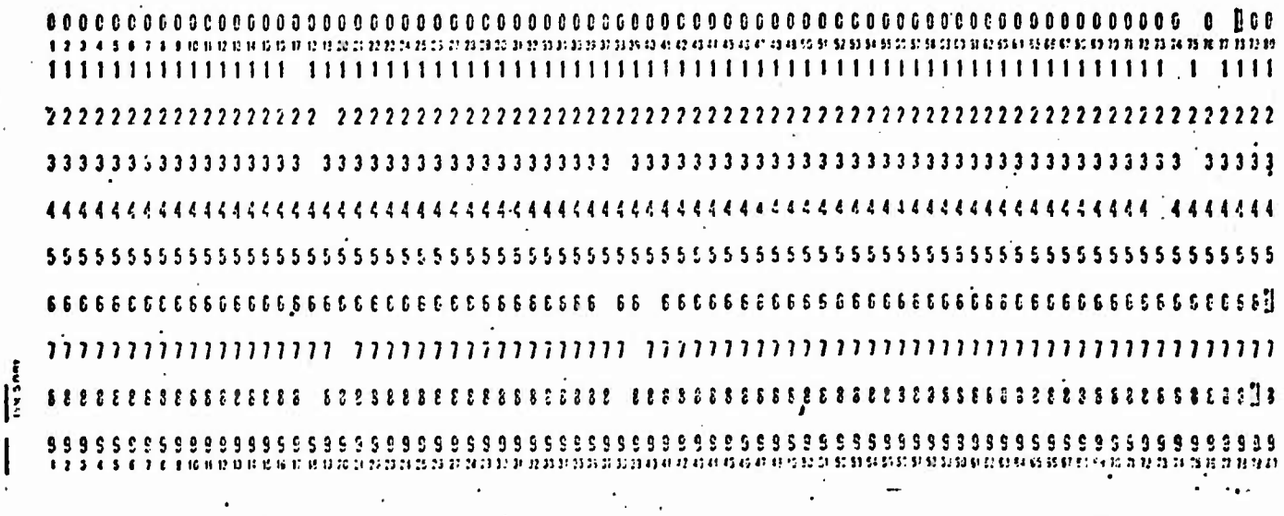
1.16 6.344 DATA0084



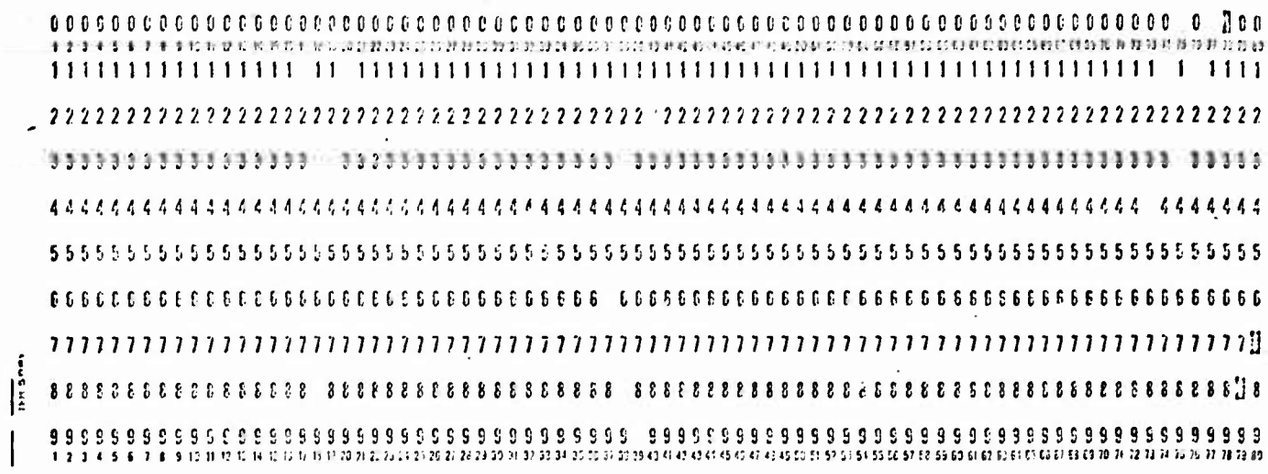
1.23 6.593 DATA0085



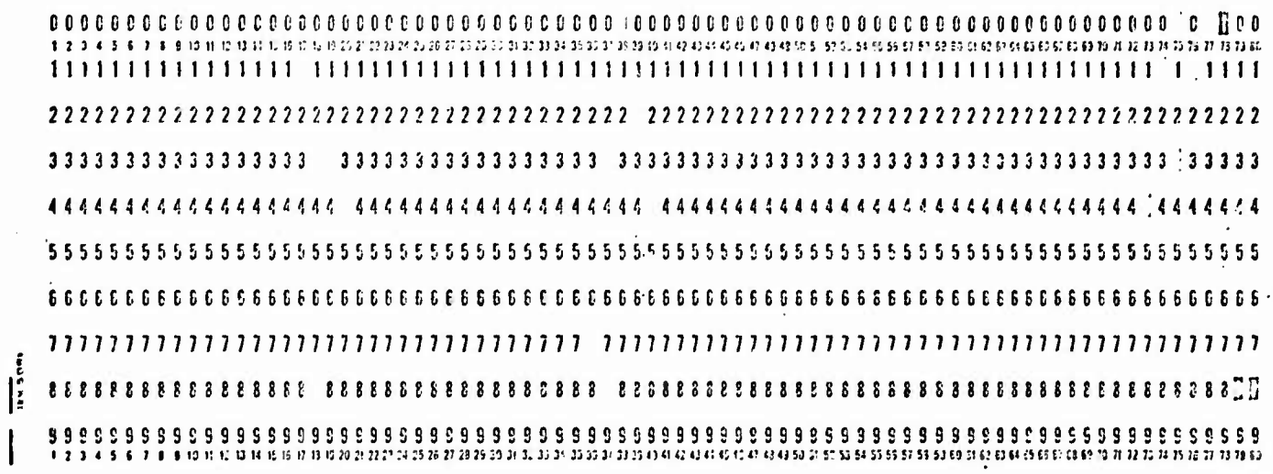
1.27 6.76 DATA0086



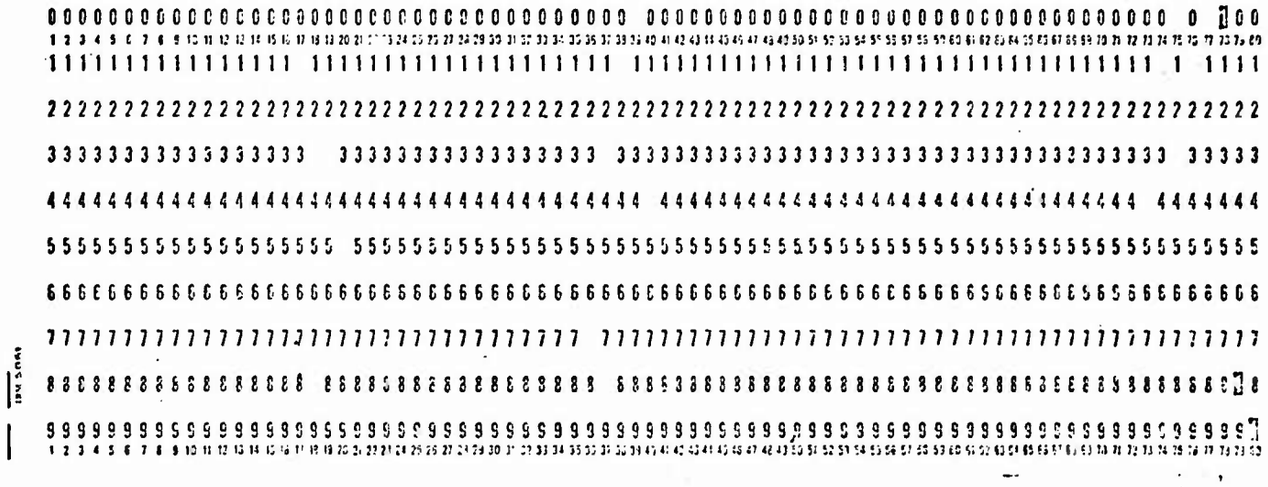
1.31 6.98 DATA0087



1.34 7.024 DATA0088



1.35 7.104 DATA0089



DIESEL DATA AND SYSTEM INERTIAS

```
 7.2      .0632       1.1        0.0     20650.      2380.       44.4  DATA0090
ALP1     ALP2        ALP3       ALPD     XK6       QBRG        CWM
```

Data Card 90

FORMAT 7F10.4

ELEMENTS

ALP1 = α_1 buffer spring scale *

ALP_2 = α_2 governor to engine constant*

ALP_3 = α_3 needle value adjustment*

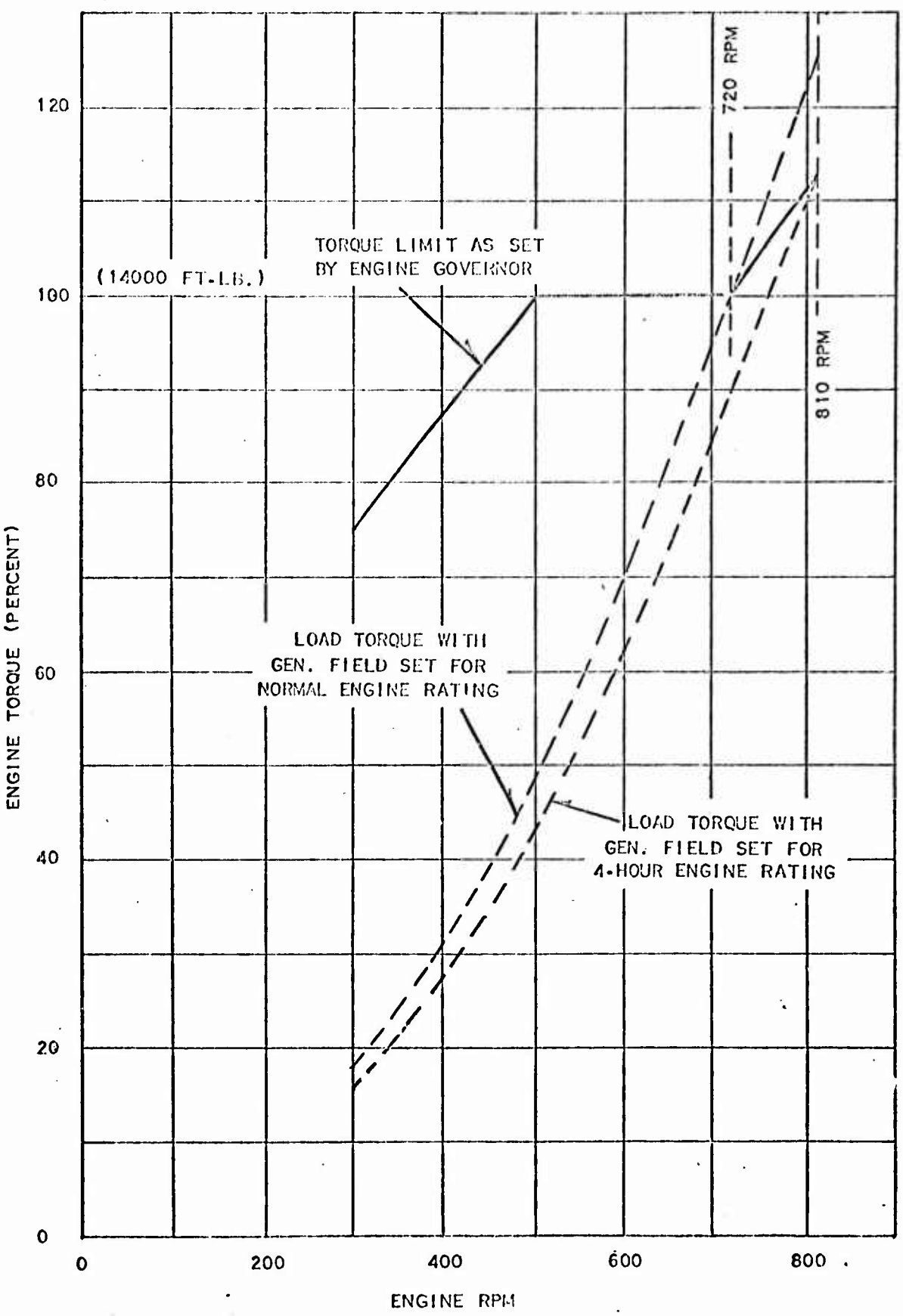
ALPD = α_D = speed droop coefficient*

XK6 = $\frac{\text{rated diesel load}}{\text{rated diesel stroke}} \left(\frac{\text{foot-lbs}}{\text{inch}}\right)$

The rated diesel load is read from curves such as the attached. The rated diesel stroke is obtained from the manufacturer.

---

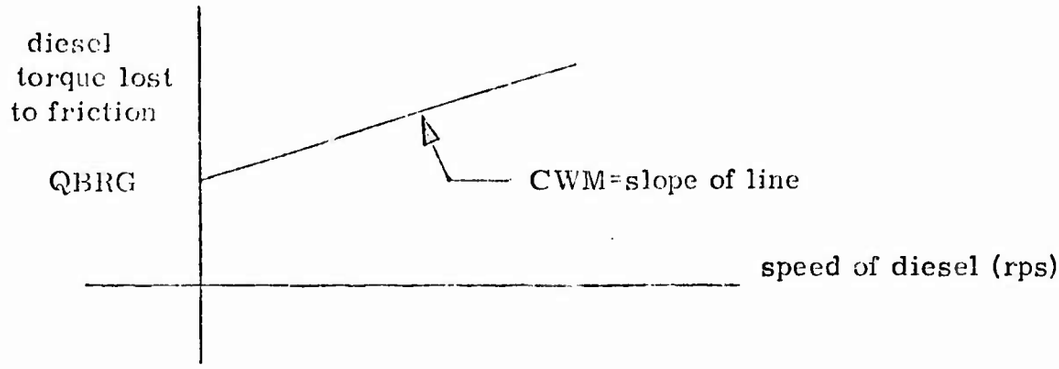
* See reference 1 for discussion of diesel governor data.

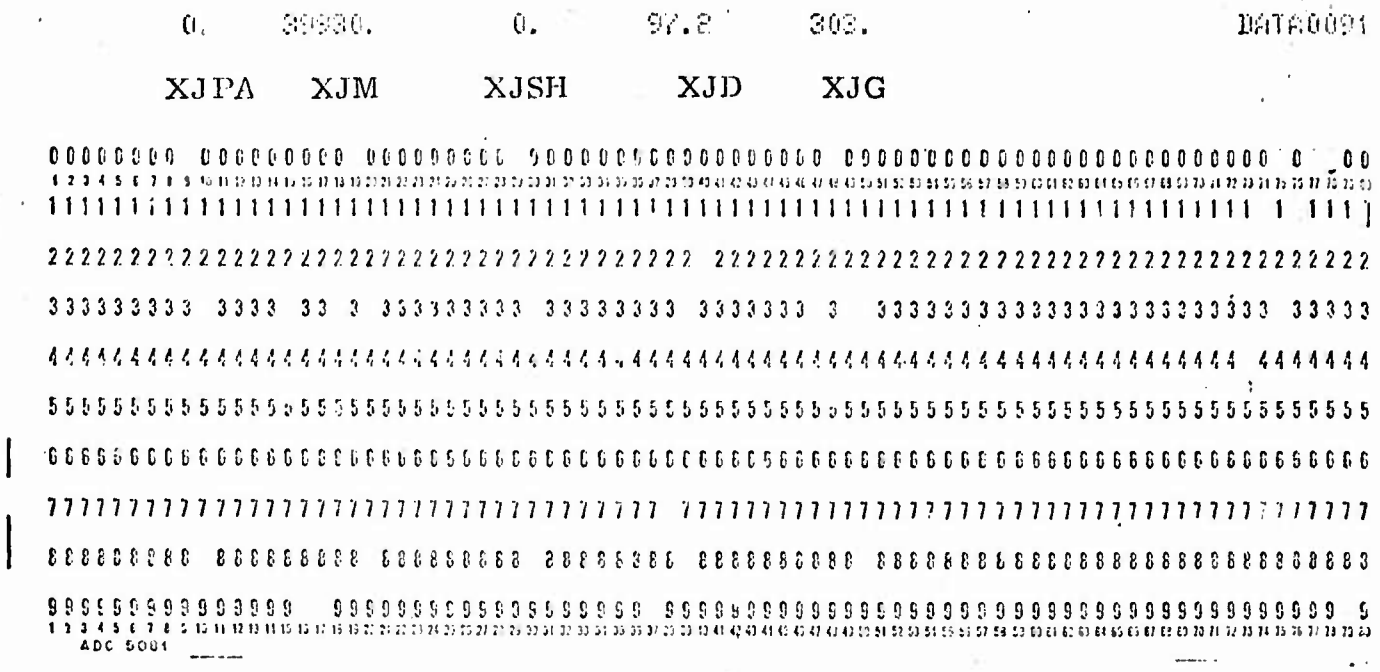


Engine Speed - Torque Characteristics

QBRG (foot-lbs) = see sketch below

CWM (foot-16-sec) = see sketch below





Data Card 91

FORMAT (7F10.4)

ELEMENTS

XJPA = Polar moment of inertia of propeller in air ($ft\text{-}lb\text{-}sec^2$).

If given a value of zero on data card it will be calculated as per subroutine POLM I. Otherwise, given value will be used.

XJM = Polar moment of inertia of one motor =

$$\frac{wk^2}{g} \quad (ft - lb\text{-}sec^2)$$

where g = value of gravity

wk^2 = given inertia factor for motor

XJSH = Propeller shaft inertia ($ft\text{-}lb\text{-}sec^2$). Usually negligible.

XJD = Polar moment of inertia of one diesel ($ft - lb - sec^2$). from manufacturers data.

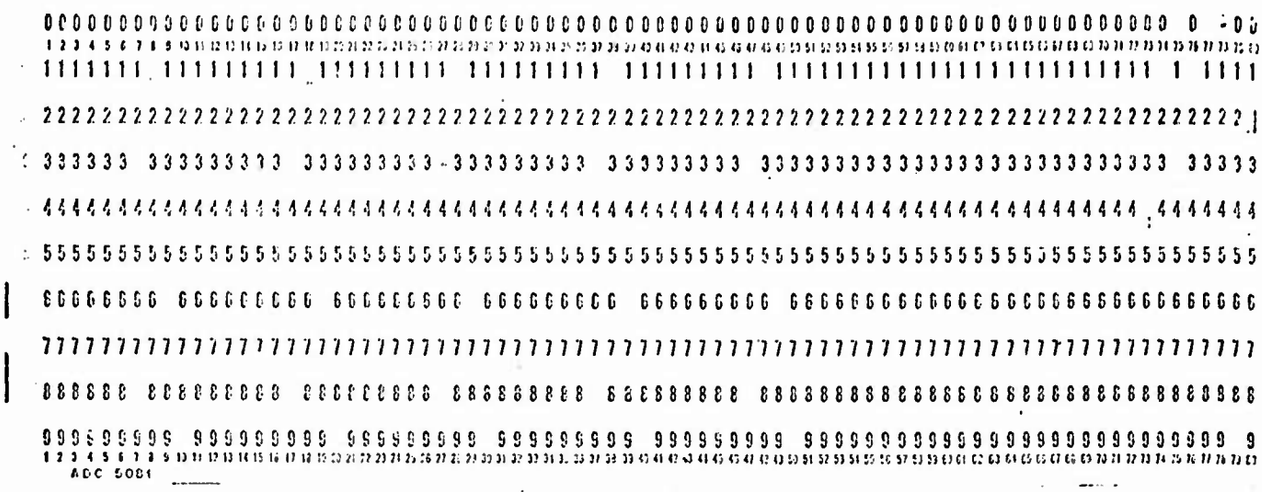
XJG = Polar moment of inertia of one generator =

$$\frac{wk^2}{g} \quad (ft\text{-}lb\text{-}sec^2)$$

ICE VALUES

.169 .169 .169 .169 .169 DATA0092

RBA(1) RBA(I+1) RBA(I+2) RBA(I+3) RBA(I+4)



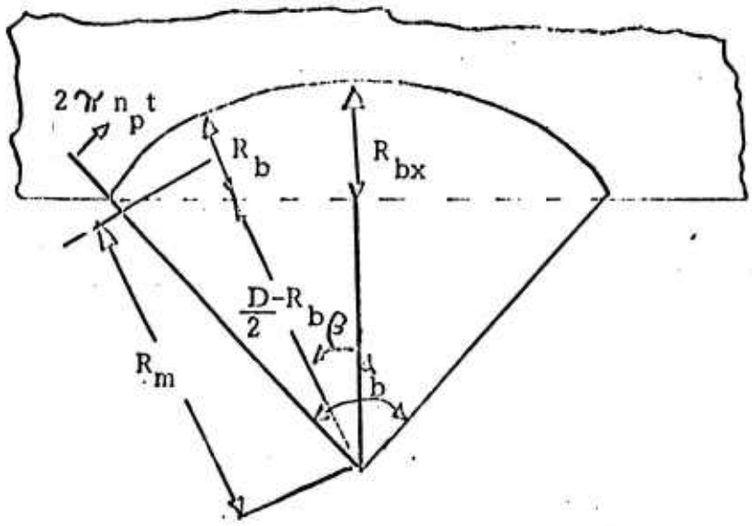
ADC 5081

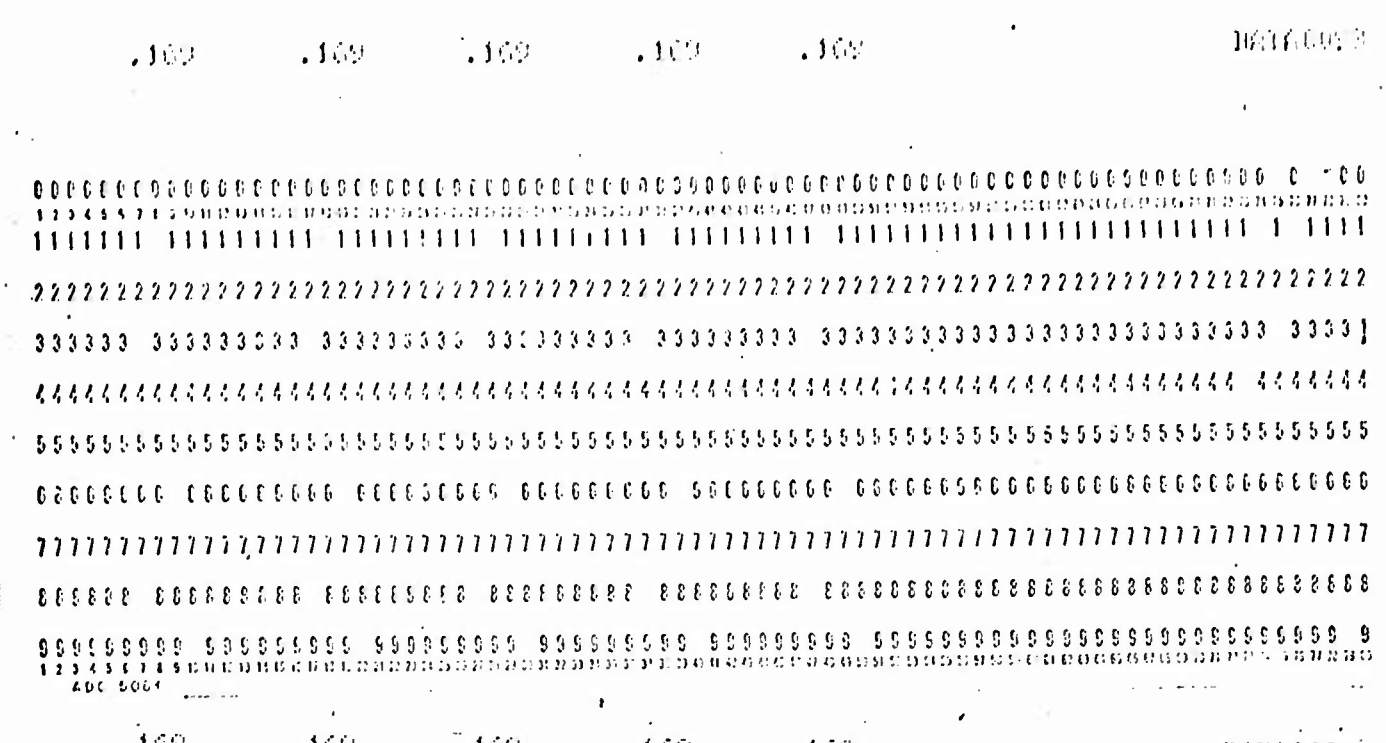
Data Cards 92-96

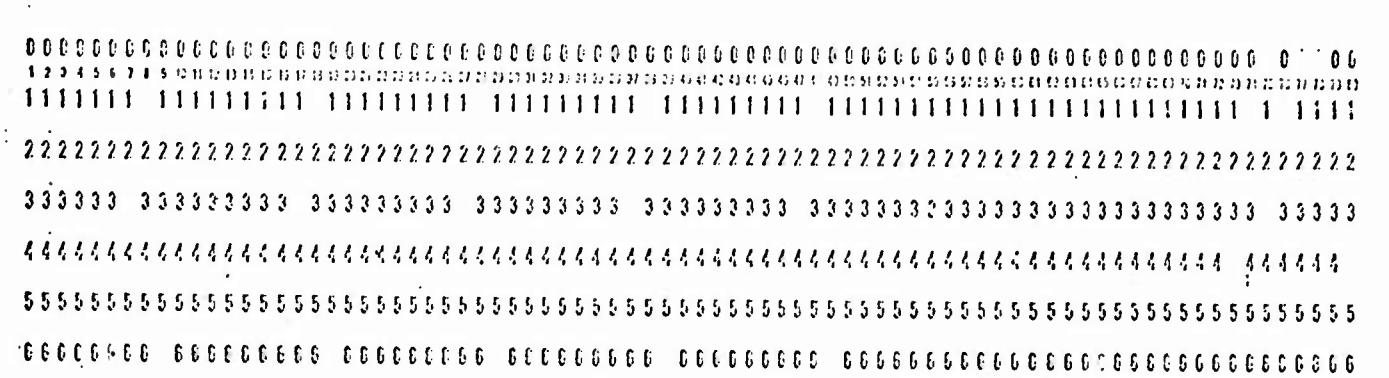
FORMAT 5F10.3

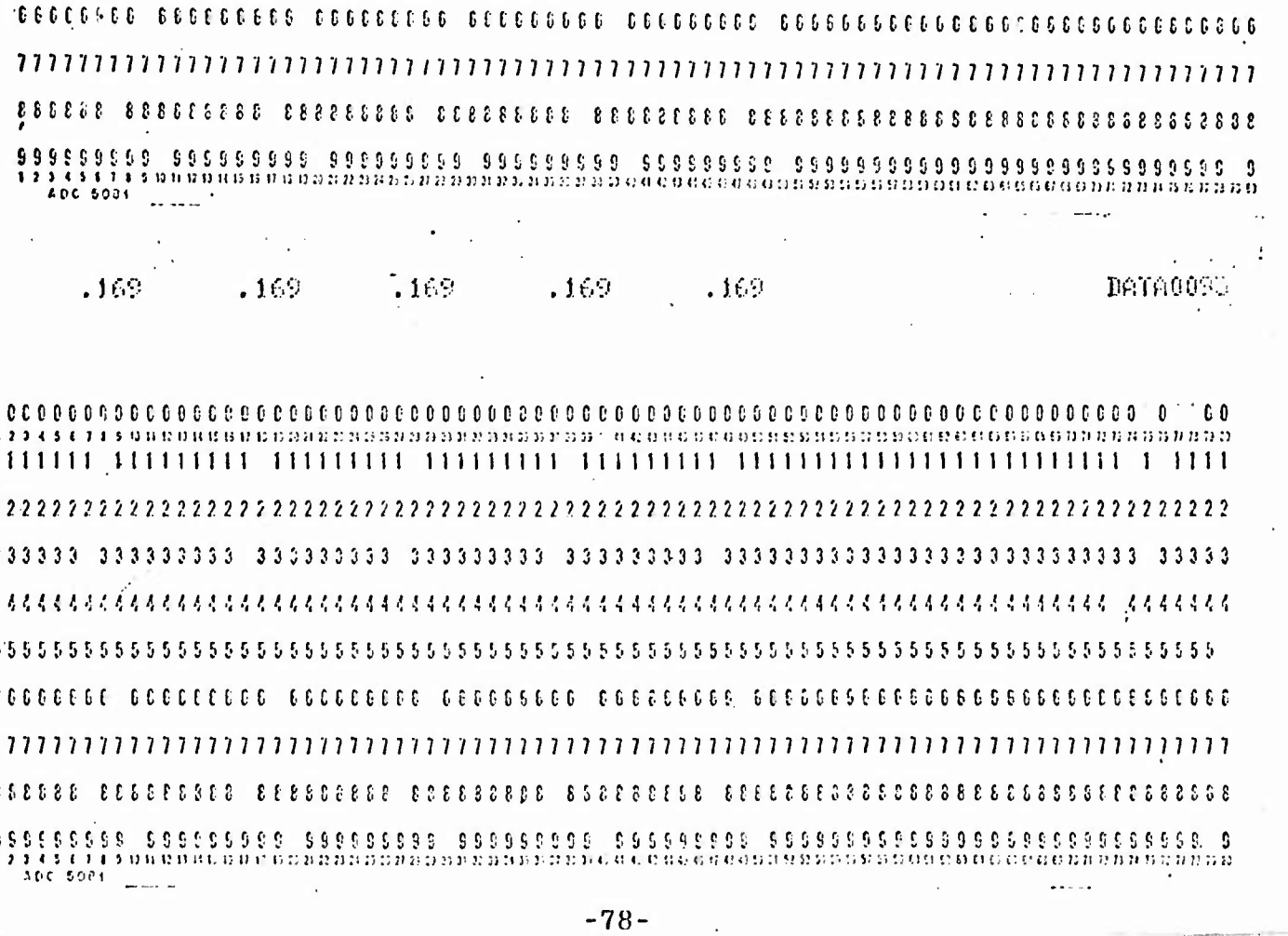
ELEMENTS

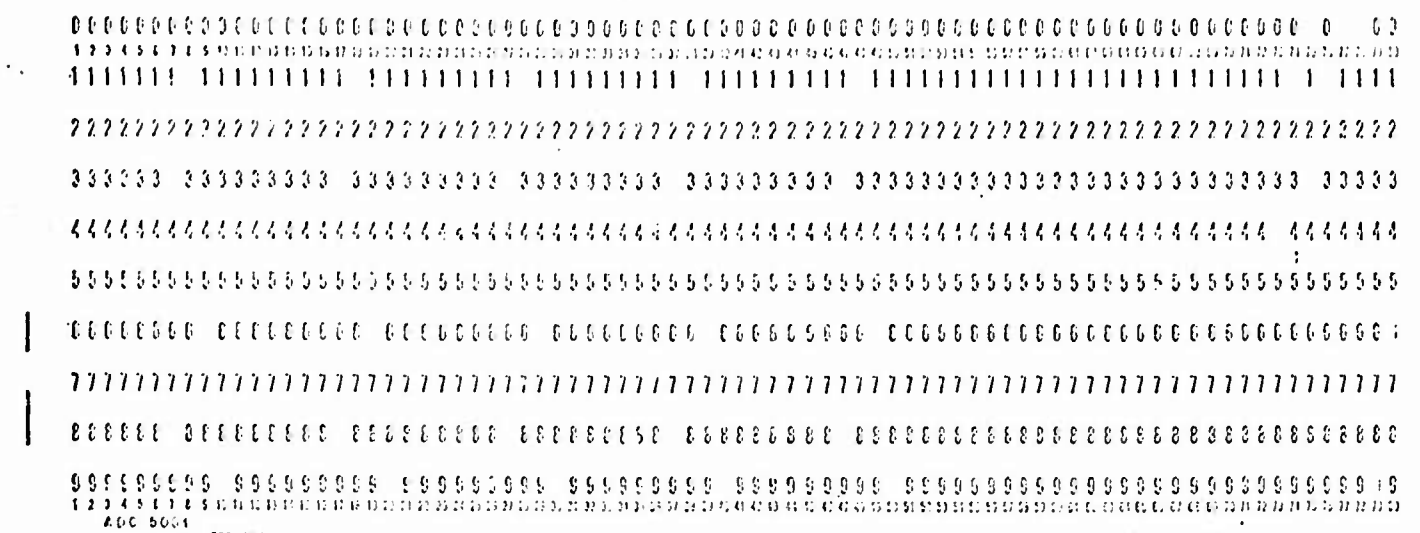
RBA = $\frac{R_{bx}}{D}$ = depth of cut factor for propeller encountering ice. Values are non dimentional for each successive propeller blade. See following figure.


2πn_p t
R_b
R_bx
(D-R_b)/2
β
R_m
b


.169
.169
.169
.169
.169
DATA0093
ADC 5001


.169
.169
.169
.169
.169
DATA0094
ADC 5001


.169
.169
.169
.169
.169
DATA0095
ADC 5001



```
1000.      5.      30.      5.     4000.  53400.      13320.  DATA0097
BLOC1      DBB12   BLOC2    DBB21  CICE   DCR         DSH
```

Data Card 97

FORMAT 7F10.3

ELEMENTS

BLOC1, DBB12, BLOC2, DBB21 (feet)

BLOC1 - Length of ice block 1
DBB12 - Distance between ice blocks 1 and 2
BLOC2 - Length of ice block 2
DBB21 - Distance between ice blocks 2 and 1.

CICE= increased resistance of hull in ice $\left(\frac{\text{lb-sec}^2}{\text{ft}^2}\right)$

$$R_{ice} = CICE \cdot V_s^2$$

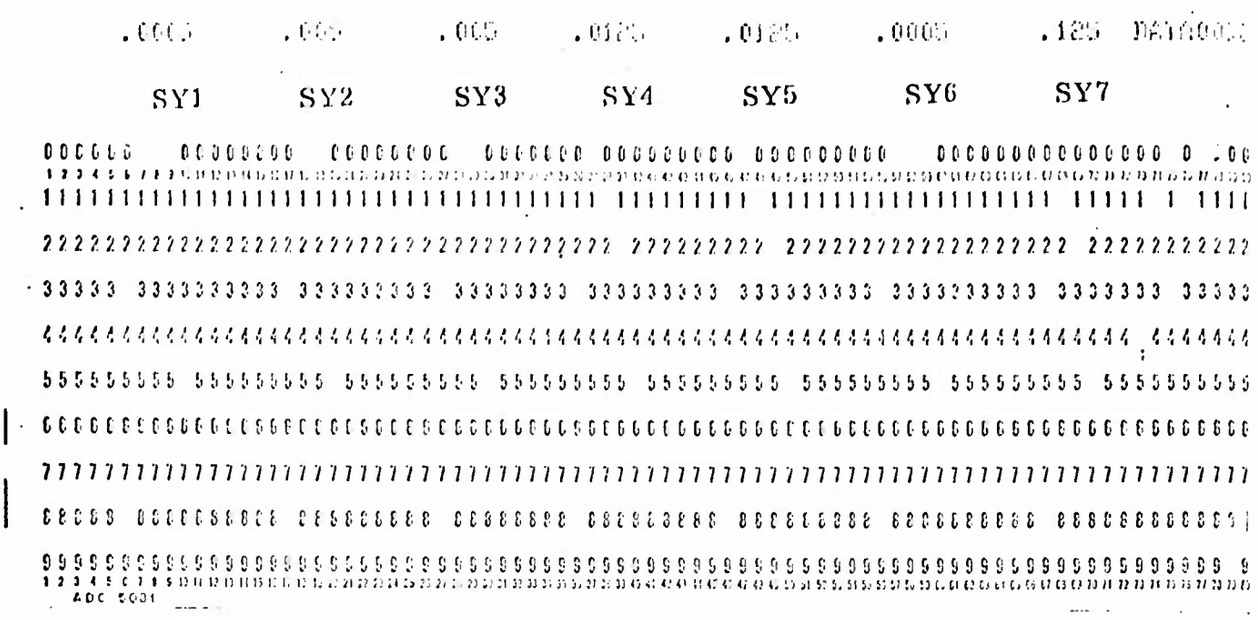
where R_{ice} = hull resistance in ice

DCR= crushing failure strength of ice (lb/ft^2)*

DSH = shearing failure strength of ice (lb/ft^2)*

*See reference 3

INITIAL VALUES AND SCALE FACTORS



Data Card 98

FORMAT 7F10.7

ELEMENTS

Each of these elements are plotter scale factors - the actual number of inches along the Y-axis that corresponds to one (1) unit of the particular variable in question. The description of each plotter scale factor is as follows:

SY1 = armature circuit current (amps)

SY2 = bus volts

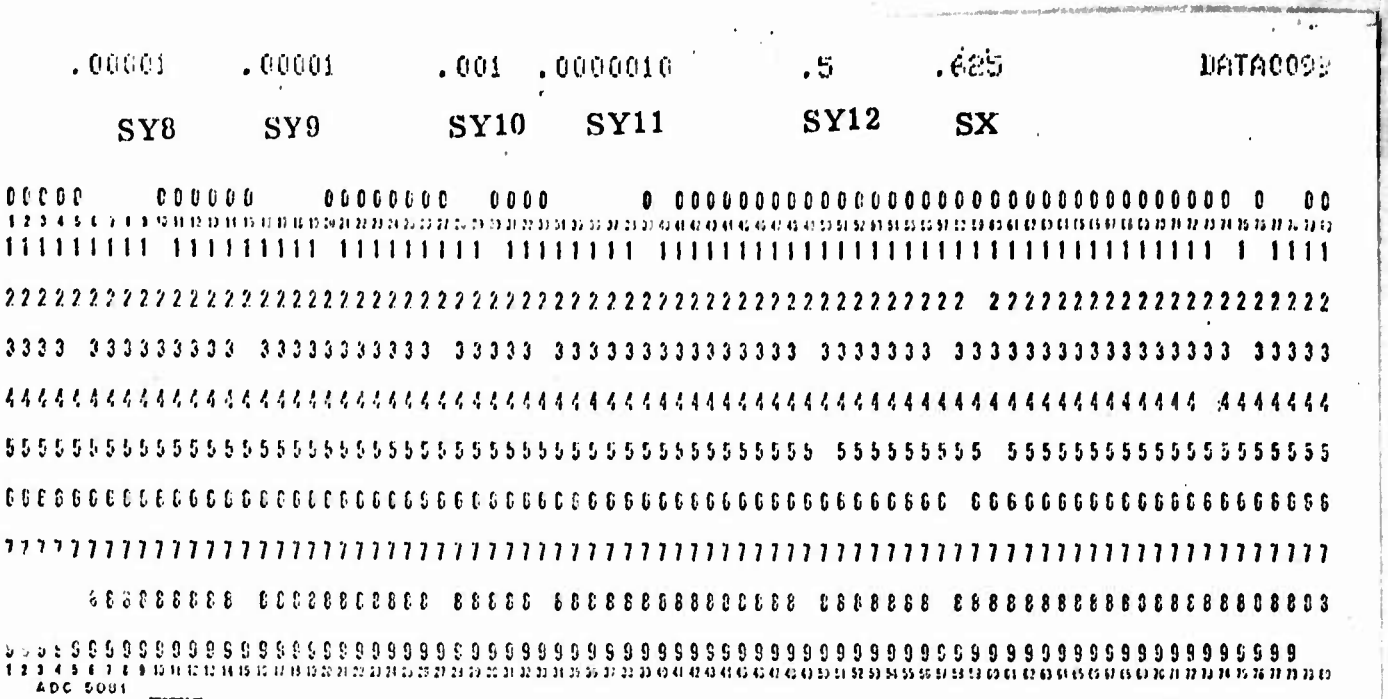
SY3 = diesel speed (RPM)

SY4 = motor field current (amps)

SY5 = propeller speed (RPM)

SY6 = power (KW)

SY7 = vessel speed (KTS)



Data Card 99

FORMAT 6F10.7

ELEMENTS

Each of these elements are plotter scale factors - the actual number of inches along the axis in question (Y-axis for SY8, SY9, SY10, SY11 and SY12; X-axis for SX) that corresponds to one (1) unit of the variable in question. The description of each plotter scale factor is as follows:

SY8= propeller torque (ft-lbs)

SY9 = propeller thrust (lbs)

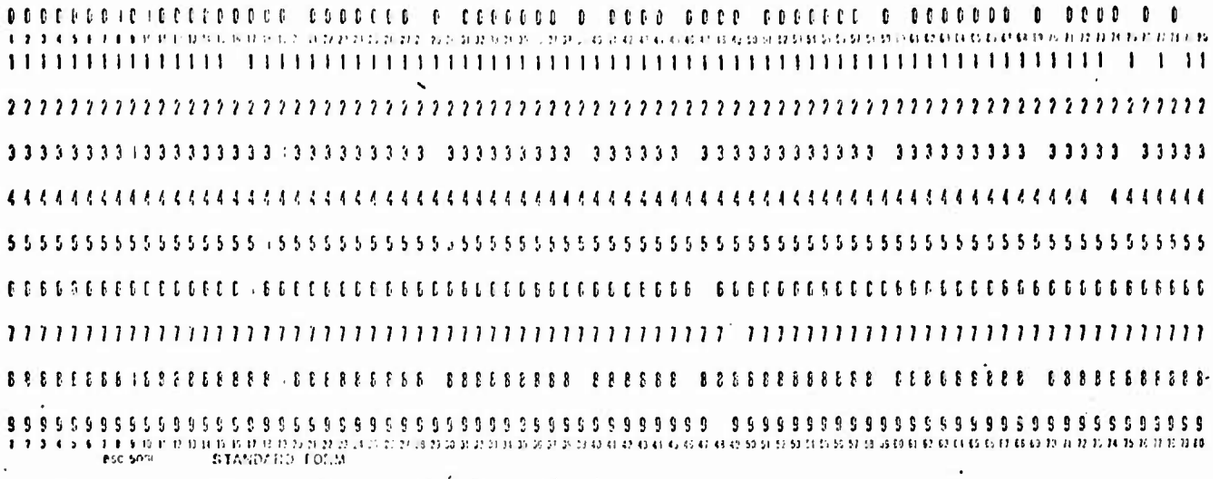
SY10 = distance (feet)

SY11 = Ice torque (ft-lbs)

SY12 = (not used)

SX = Time (seconds)

0.0	165.0	0.0	0.0	0.0579	0.0	0.0
T	THEIC	XNPMD	Y	Z	RV	TEMP



Data Card 100

FORMAT 7F10.4

ELEMENTS

T = time program begins

THEIC = initial position of bridge controller in degrees between + 165 and -165.

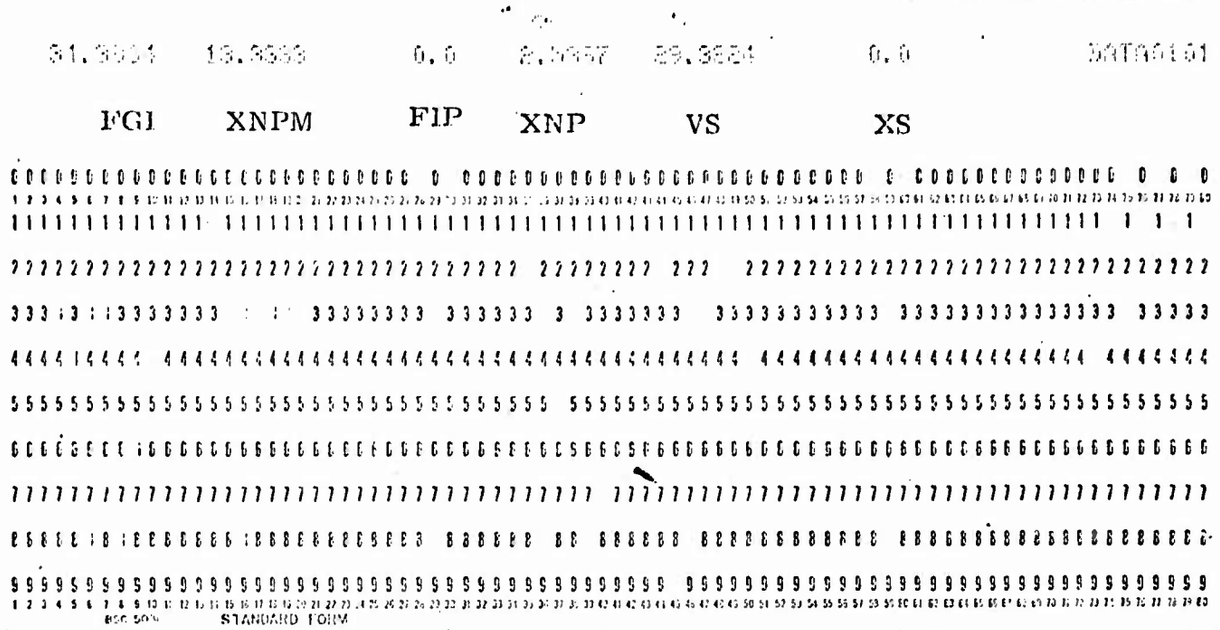
XNPMD = initial rate of change of diesel speed (normally zero) (rev/sec^2)

Y = initial rate of change of diesel rack position(rack velocity) (in/sec)

Z = initial rack position (inches) corresponding to initial ship horsepower for maneuver.

RV = reserve state variables

TEMP = reserve state variables.



Data Card 101

FORMAT 6F10.4

ELEMENTS

FGI = initial generator field current (amps)

XNPM = initial diesel speed (rps)

FIP = field adjusting current * (amps)

XNP = initial shaft speed (rps)

VS = initial ship speed (feet/sec)

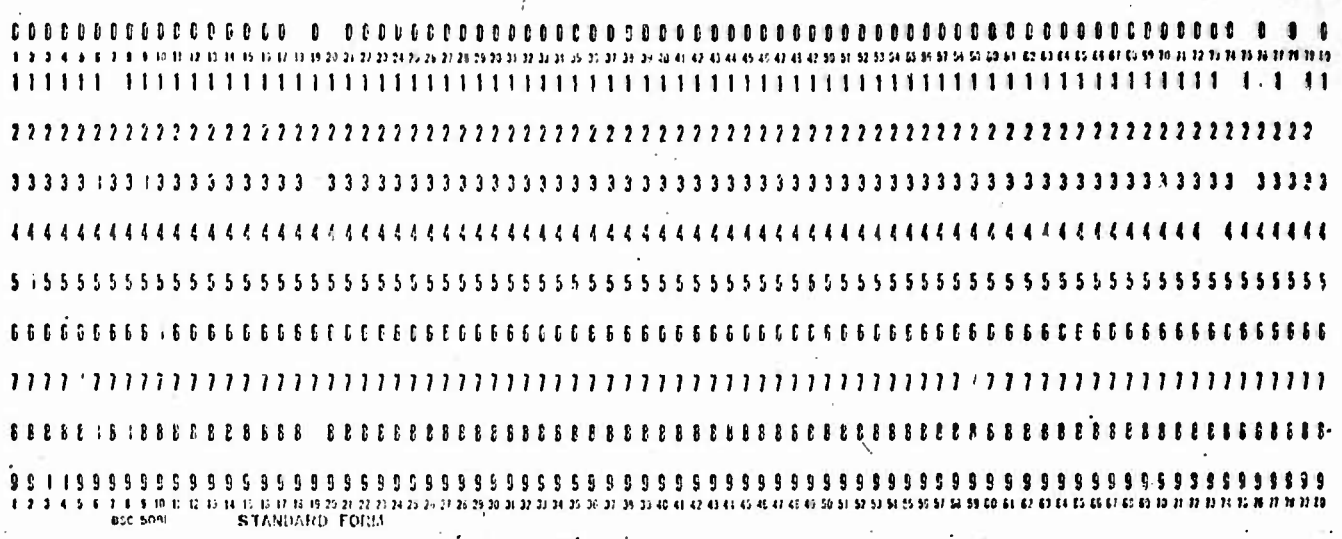
XS = initial position of ship (feet)

---

* See reference 1

5997.1686 0.0 DATA0102

A1 AIDO



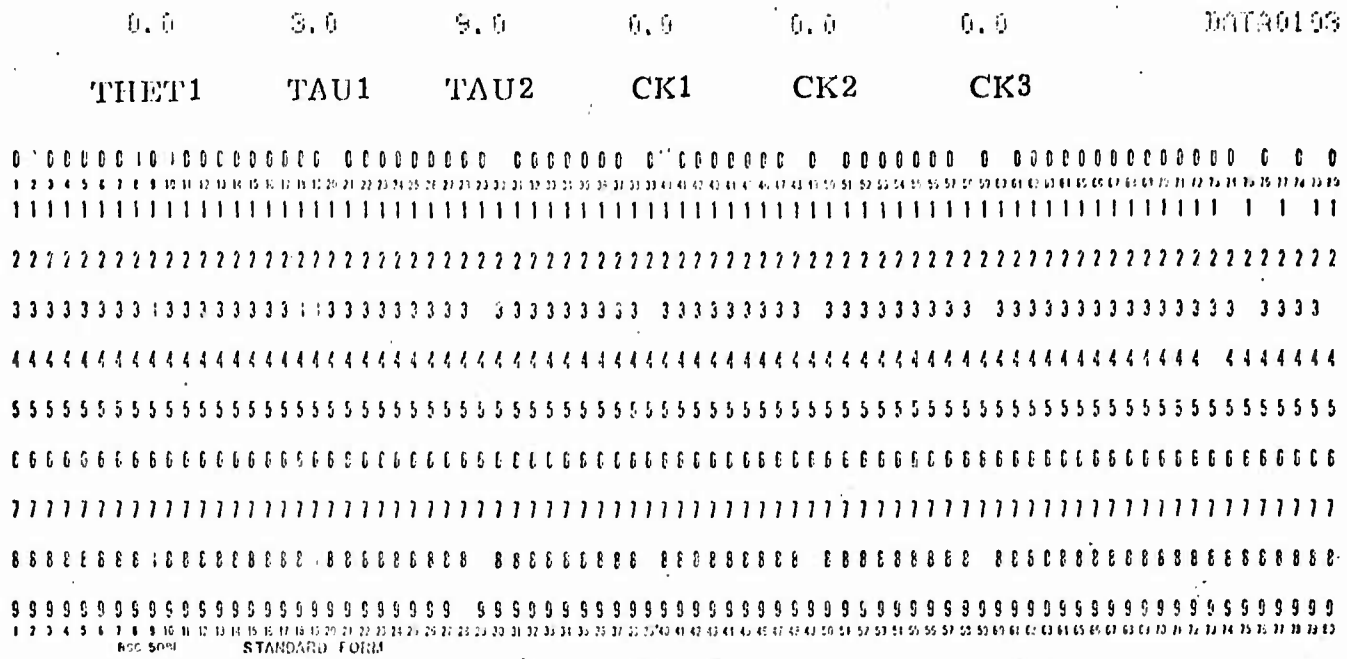
Data Card 102

FORMAT 2F10.4

ELEMENTS

AI = initial armature current (amps)

AIDO = initial armature current derivative (amps/sec)



Data Card 103

FORMAT 6F10.4

ELEMENTS

THET1 - second position of bridge controller (degrees) reached in TAU1 seconds having started at THEIC.

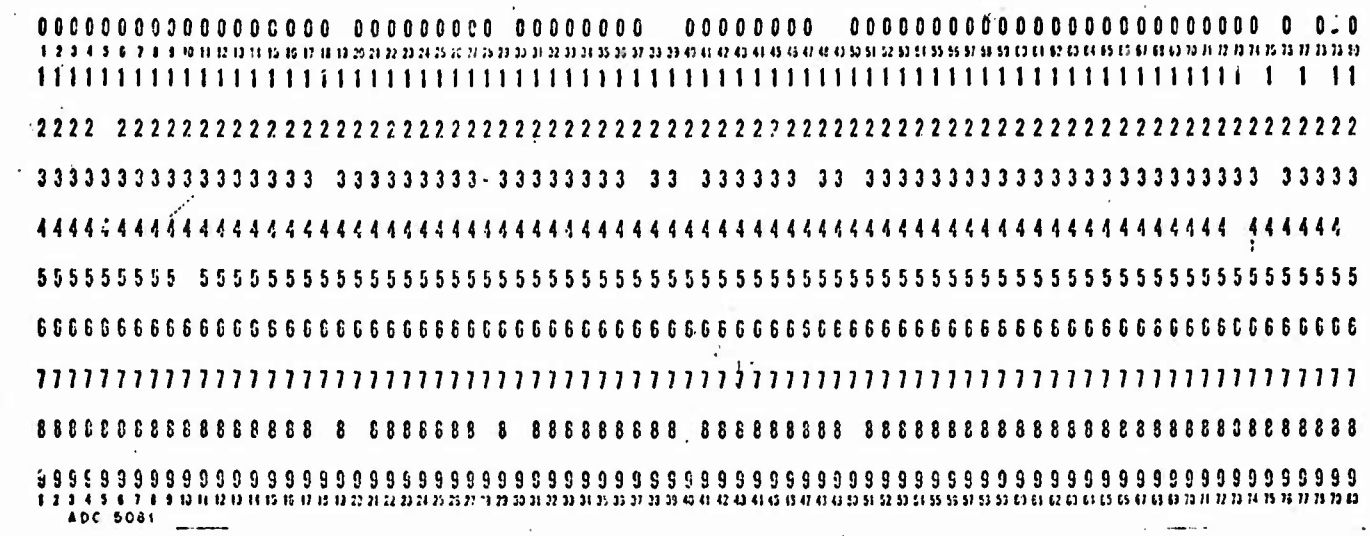
TAU1 = time (seconds) for bridge controller to reach THET1 from THEIC.

TAU2 = time (seconds) for bridge controller to go from THET1 to THET0

CK1, CK2, CK3 = spare variables

```
    2    5        .08        .08       300.       300.                    DATA0104

  ICNTR  NON    DT     TIMSC      TICE     TEND
```



Data Card 104

FORMAT 215, 4F10.4

ELEMENTS

ICNTR = variable defining type of manuever (see subroutine CNTRL)

NON = program prints output every NON times DT seconds

DT = time step (seconds) for integration

TIMSC = time increment (seconds) for writing solution on disc.
TIMSC $\geq$ DT and if TIMSC $>$ DT then TIMSC = Integer times DT

TICE = time (seconds) when start hitting ice

TEND = time (seconds) when program ends.

SAMPLE RUN

LISTING OF INPUT DECK

```
// JOB
// XEQ GENIS    1
*LOCALEXEC,CNTRL,HEUN,FINIS
    4                .55          .8         16.0         .125        1.082        5.12  DATA0001
      0.0            .330         .035       -0.1         .335         .035               DATA0002
     -0.2            .340         .034       -0.3         .320         .033               DATA0003
     -0.4            .275         .032       -0.5         .300         .033               DATA0004
     -0.6            .365         .040       -0.7         .460         .045               DATA0005
     -0.8            .550         .050       -0.9         .645         .057               DATA0006
     -1.0            .695         .064       -1.1         .805         .072               DATA0007
     -1.2            .910         .081       -1.3         1.0          .093               DATA0008
     -1.4           1.160         .107       -1.5        1.19          .118               DATA0009
      0.0            .220         .035        0.1         .300         .034               DATA0010
      0.2            .280         .033        0.3         .250         .030               DATA0011
      0.4            .220         .027        0.5         .165         .024               DATA0012
      0.6            .140         .018        0.7         .090         .016               DATA0013
      0.8            .050         .008        0.9          0.0         .002               DATA0014
      1.0           -.060        -.003        1.1         -.120        -.011               DATA0015
      1.2           -.180        -.021        1.3         -.285        -.031               DATA0016
      1.4           -.360        -.043        1.5         -.425        -.053               DATA0017
      0.0           -.220        -.034       -0.1         -.220        -.035               DATA0018
     -0.2           -.190        -.034       -0.3         -.180        -.031               DATA0019
     -0.4           -.170        -.027       -0.5         -.220        -.030               DATA0020
     -0.6           -.260        -.036       -0.7         -.390        -.042               CATA0021
     -0.8           -.390        -.050       -0.9         -.480        -.057               DATA0022
     -1.0           -.545        -.067       -1.1         -.620        -.077               DATA0023
     -1.2           -.730        -.085       -1.3         -.830        -.100               DATA0024
     -1.4           -.930        -.116       -1.5        -1.0          -.125               DATA0025
      0.0           -.220        -.034        0.1         -.210        -.032               DATA0026
      0.2           -.190        -.030        0.3         -.170        -.026               DATA0027
      0.4           -.130        -.022        0.5         -.100        -.017               DATA0028
      0.6           -.070        -.011        0.7          0.0         -.007               DATA0029
      0.8            .050         0.0         0.9          .100         .010               DATA0030
      1.0            .160         .015        1.1          .220         .027               DATA0031
      1.2            .310         .038        1.3          .280         .050               DATA0032
      1.4            .450         .064        1.5          .530         .080               DATA0033
      0.0            .200         .025       -0.1          .220         .028               DATA0034
     -0.2            .280         .032       -0.3          .345         .037               DATA0035
     -0.4            .410         .043       -0.5          .470         .046               DATA0036
     -0.6            .560         .050       -0.7          .600         .055               DATA0037
      0.0           -.175        -.021        0.1         -.18          -.020               DATA0038
      0.2           -.20         -.021        0.3         -.25          -.030               DATA0039
      0.4           -.27         -.035        0.5         -.25          -.033               DATA0040
      0.6           -.22         -.028        0.7         -.19          -.022               DATA0041
      0.0           -.175        -.021       -0.1         -.190         -.024               DATA0042
     -0.2           -.230        -.029       -0.3         -.290         -.038               DATA0043
     -0.4           -.350        -.045       -0.5         -.395         -.050               DATA0044
```

-0.6	-.420	-.055	-0.7	-.450	-.057		DATA0045
0.0	.20	.025	0.1	.245	.026		DATA0046
0.2	.250	.030	0.3	.275	.037		DATA0047
0.4	.280	.043	0.5	.270	.042		DATA0048
0.4	.255	.036	0.7	.230	.033		DATA0049
.38	12000.	3.					DATA0050
0.0	0.0	1.0	1.0				DATA0051
1.0	9.5	.991	.969				DATA0052
2.0	37.8	.983	.938				DATA0053
3.0	84.9	.974	.906				DATA0054
4.0	151.2	.965	.875				DATA0055
5.0	236.3	.956	.844				DATA0056
6.0	340.2	.948	.813				DATA0057
7.0	463.0	.939	.781				DATA0058
8.0	605.0	.930	.750				DATA0059
9.0	972.0	.920	.742				DATA0060
10.0	1326.0	.918	.740				DATA0061
11.0	1768.0	.910	.740				DATA0062
12.0	2363.0	.904	.750				DATA0063
13.0	3148.0	.900	.760				DATA0064
14.0	4127.0	.894	.766				DATA0065
15.0	5372.0	.890	.770				DATA0066
16.0	7046.0	.884	.767				DATA0067
17.0	9795.0	.887	.750				DATA0068
18.0	15001.0	.908	.727				DATA0069
19.0	21209.0	.908	.727				DATA0070
.00152	2160.	32.4	1.49				DATA0071
.00980	495.	4.6	1.85				DATA0072
34.	10000.	0.0	1296.	3.	1.75		DATA0073
0.0	0.0	0.0					DATA0074
3.	.0002						DATA0075
1000.	20.						DATA0076
	0.0		5.64				DATA0077
	.230		5.64				DATA0078
	.478		5.64				DATA0079
	.722		5.64				DATA0080
	.925		5.64				DATA0081
	1.059		5.64				DATA0082
	1.150		5.64				DATA0083
	1.220		5.64				DATA0084
	1.260		5.64				DATA0085
	1.294		5.64				DATA0086
	1.325		5.64				DATA0087
	1.345		5.64				DATA0088
	1.366		5.64				DATA0089
7.2	.0632	1.1	0.0	26650.	2380.	44.4	DATA0090
0.	39930.	0.	97.2	303.			DATA0091
.169	.169	.169	.169	.169			DATA0092
.169	.169	.169	.169	.169			DATA0093
.169	.169	.169	.169	.169			DATA0094
.169	.169	.169	.169	.169			DATA0095
.169	.169	.169	.169	.169			DATA0096
1000.	5.	30.	5.	4000.	53400.	13320.	DATA0097
.0005	.005	.005	.0125	.0125	.0005	.125	DATA0098

```
     .00001      .00001      .001    .0000010        .5       .625              DATA0099
       0.0         165.0       0.0         0.0   0.5970        0.0       0.0   DATA0100
   34.3864     13.3333         0.0      2.5387  29.3224        0.0             DATA0101
5997.1934         0.0                                                         DATA0102
       0.0         3.0         9.0         0.0     0.0         0.0             DATA0103
     2   5         .08         .08        300.    300.                         DATA0104
```

LISTING OF OUTPUT ON CONSOLE TYPEWRITER

FOR SAMPLE RUN

```
// XEQ GENIS    1
*LOCALEXEC,CNTRL,REUN,FINIS
TYPE IN CONCLUSION OF PHRASE'SIMULATION OF'USING 80 CHARACTERS OR LESS
TEST SHIP FOR DOCUMENTATION - - CRASH REVERSAL OF A 12000 TON SHIP
TURN ON APPROPRIATE SWITCHES
     ICE DATA- 10
     PROP DATA- 11
     HULL DATA- 12
     ELECTRICAL DATA- 13
     GOVERNOR AND BRIDGE CONTROLLER DATA- 14
     INERTIAS- 15


TURN ON SELECTOR SWITCHES AND PRESS START
TURN ON SWITCH --
          1 - TO PUNCH NEW INITIAL CONDITION CARDS
          2 - TO PLOT DATA
          3 - TO CONTINUE SOLUTION OR CURRENT PROBLEM - SWITCH OFF STARTS NEW PROBLEM
          4 - TO SUPPRESS FUTURE TYPING OF THESE INSTRUCTIONS
          5 - TO CHANGE TIME STEP
          6 - TO CHANGE PRINT INTERVAL
          7 - TO CHANGE XLA


TURN ON SELECTOR SWITCHES AND PRESS START
AI--1,EB--2,XNPH--3,FHI--4,XNP--5,XKH--6,CVK--7,QPI--8, XPROI--9,CXS--10,QI--11,TEMP
-12

WHICH VARIABLES TO BE PLOTTED (10 ONLY) -
(  )(  )(  )(  )(  )(  )(  )(  )(  )(  )
  1   2   3   4   5   6   7   8   9  10
DO YOU WANT A GRID (0 - NO,1 - YES)
(  )
 1
SIMULATION OF TEST SHIP FOR DOCUMENTATION
CRASH REVERSAL OF A 12000 TON SHIP
```

LISTING OF OUTPUT ON PRINTER

AND PLOTTER

UNITED STATES COAST GUARD

ICEBREAKER PROPULSION SYSTEM SIMULATOR

SIMULATION OF TEST SHIP FOR DOCUMENTATION - - CRASH REVERSAL OF A 12000 TON SHIP
SERIAL NUMBER-- 1.

SELECTED INPUT DATA

INITIAL CONDITIONS

T	THEIC	XNPMD	Y	Z	RV	TEMP	
0.0000	165.0000	0.0000	0.0000	0.6970	0.[illegible]	0.0000	
FGI	XNPM	FIP	XNP	VS		AI	AIDC
34.3064	13.3333	0.0000	2.5387	29.322[illegible]		[illegible].1845	0.0000

CONTROL DATA

ICNTR= 2 NON = 5 DT = 0.0800 TICE= 300.0000 TEND = 300.0000 TIMSC= 0.0800
THETI= 0.0000 TAU1 = 3.0000 TAU2= 9.0000
SY1 = 0.0005000 SY2 = 0.0050000 SY3 = 0.0050000 SY4 = 0.0125000 SY5 = 0.0125000
SY6 = 0.0005000 SY7 = 0.1250000 SY8 = 0.0000100 SY9 = 0.0000100 SY10= 0.0010000
SY11 = 0.0000010 SY12 = 0.5000001 SX = 0.6250001

PROP DATA

IZR= 4 DAR= 0.5500 EO= 0.1250 OCT= 5.1200
PDR= 0.8000 D = 16.0000 EK= 1.0820

B	BB	E	EE
0.3300	0.025[illegible]	0.2000	0.0250
0.3350	0.0350	0.2200	0.0280
0.3400	0.0340	0.2800	0.0320
0.3200	0.0330	0.3450	0.0370
0.2750	0.0320	0.4100	0.0430
0.3000	0.0330	0.4700	0.0460
0.3650	0.0400	0.5600	0.0500
0.4600	0.0450	0.6000	0.0550
0.5500	0.0500	-0.1750	-0.0210
0.6450	0.0570	-0.1800	-0.0200
0.6950	0.0640	-0.2000	-0.0210
0.8050	0.0720	-0.2500	-0.0300
0.9100	0.0810	-0.2700	-0.0350
1.0000	0.0930	-0.2500	-0.0330
1.1600	0.1070	-0.2200	-0.0280
1.1900	0.1180	-0.1900	-0.0220
0.8200	0.0350	-0.1750	-0.0210
0.3000	0.0340	-0.1900	-0.0240
0.2800	0.0330	-0.2300	-0.0290
0.2500	0.0300	-0.2900	-0.0380
0.2200	0.0270	-0.3500	-0.0450
0.1650	0.0240	-0.3950	-0.0500
0.1400	0.0180	-0.4200	-0.0550
0.0900	0.0160	-0.4500	-0.0570
0.0500	0.0080	0.2000	0.0250
0.0000	0.0020	0.2450	0.0260

-0.0600	-0.0030	0.2500	0.0300
-0.1200	-0.0110	0.2750	0.0370
-0.1800	-0.0210	0.2800	0.0430
-0.2850	-0.0310	0.2700	0.0420
-0.3600	-0.0430	0.2550	0.0360
-0.4250	-0.0530	0.2300	0.0330
-0.2200	-0.0340		
-0.2200	-0.0350		
-0.1900	-0.0340		
-0.1600	-0.0310		
-0.1700	-0.0270		
-0.2200	-0.0300		
-0.2600	-0.0360		
-0.3300	-0.0420		
-0.3900	-0.0500		
-0.4600	-0.0570		
-0.5400	-0.0670		
-0.6200	-0.0770		
-0.7300	-0.0850		
-0.8300	-0.1000		
-0.9300	-0.1160		
-1.0000	-0.1250		
-0.2200	-0.0340		
-0.2100	-0.0320		
-0.1900	-0.0300		
-0.1700	-0.0260		
-0.1300	-0.0220		
-0.1000	-0.0170		
-0.0700	-0.0110		
0.0000	-0.0070		
0.0500	0.0000		
0.1000	0.0100		
0.1600	0.0180		
0.2200	0.0270		
0.3100	0.0380		
0.3800	0.0500		
0.4500	0.0640		
0.5300	0.0800		

HULL DATA

CAD= 0.0800 DELTA= 12000.0 XKK= 3.

VK	EHP	WADED	THDED
0	0.0000	1.0000	1.0000
1	9.5000	0.9910	0.9690
2	37.8000	0.9830	0.9380
3	84.9000	0.9740	0.9060
4	151.2000	0.9650	0.8750
5	236.3000	0.9560	0.8440
6	340.2000	0.9480	0.8130
7	463.0000	0.9390	0.7810
8	686.0001	0.9300	0.7500
9	972.0001	0.9200	0.7420

10	1326.0002	0.9180	0.7400
11	1768.0002	0.9100	0.7400
12	2363.0004	0.9040	0.7500
13	3148.0004	0.9000	0.7600
14	4127.0009	0.8940	0.7660
15	5372.0009	0.8900	0.7700
16	7046.0009	0.8840	0.7670
17	9795.0019	0.8870	0.7500
18	15001.0019	0.9080	0.7270
19	21209.0039	0.9080	0.7270

ELECTRICAL DATA

RM = 0.001520 RG = 0.009800 TRNFG= 495.0 ZGP = 4.60
RFG = 1.850000 TRNFM= 2160.00 ZMP = 32.40 RFM = 1.490000
FIMAX= 84.0000 C1 =10000.0019 XK1 = 0.0000
CSMFR= 1296.0002 TAUB = 0.0000 FMIRV= 0.0000 TAUA= 0.0000
TRNSF= 3.0000 RX = 1.7500 XGEN = 3.0000 XLA = 0.0002 SPGEN= 1000.0001 SPMOT= 20.0000

CFGA	CFMA
0.0000	5.6400
0.2300	5.6400
0.4780	5.6400
0.7220	5.6400
0.9250	5.6400
1.0590	5.6400
1.1500	5.6400
1.2200	5.6400
1.2600	5.6400
1.2940	5.6400
1.3250	5.6400
1.3450	5.6400
1.3660	5.6400

GOVERNOR AND DIESEL DATA

ALP1 = 7.2000 ALP2= 0.0632 ALP3= 1.1000 ALPD= 0.0000
XK6 = 26650.00 GBRG= 2380.0004 CWM = 44.4000

SYSTEM INERTIAS

XJPA= 13312.00 XJM= 39930.00 XJSH= 0.00
XJD = 97.20 XJG= 303.00 XJPE= 2001.24

ICE DATA

BLOC1= 1000.000 DBB12= 5.000 BLOC2= 30.000 DBB21= 5.000
RBA

```
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
          0.169
DCR= 53400.007 DSH= 13320.001 CICE=  4000.000
CK1=     0.000 CK2=     0.000 CK3 =     0.000
```

OUTPUT DATA

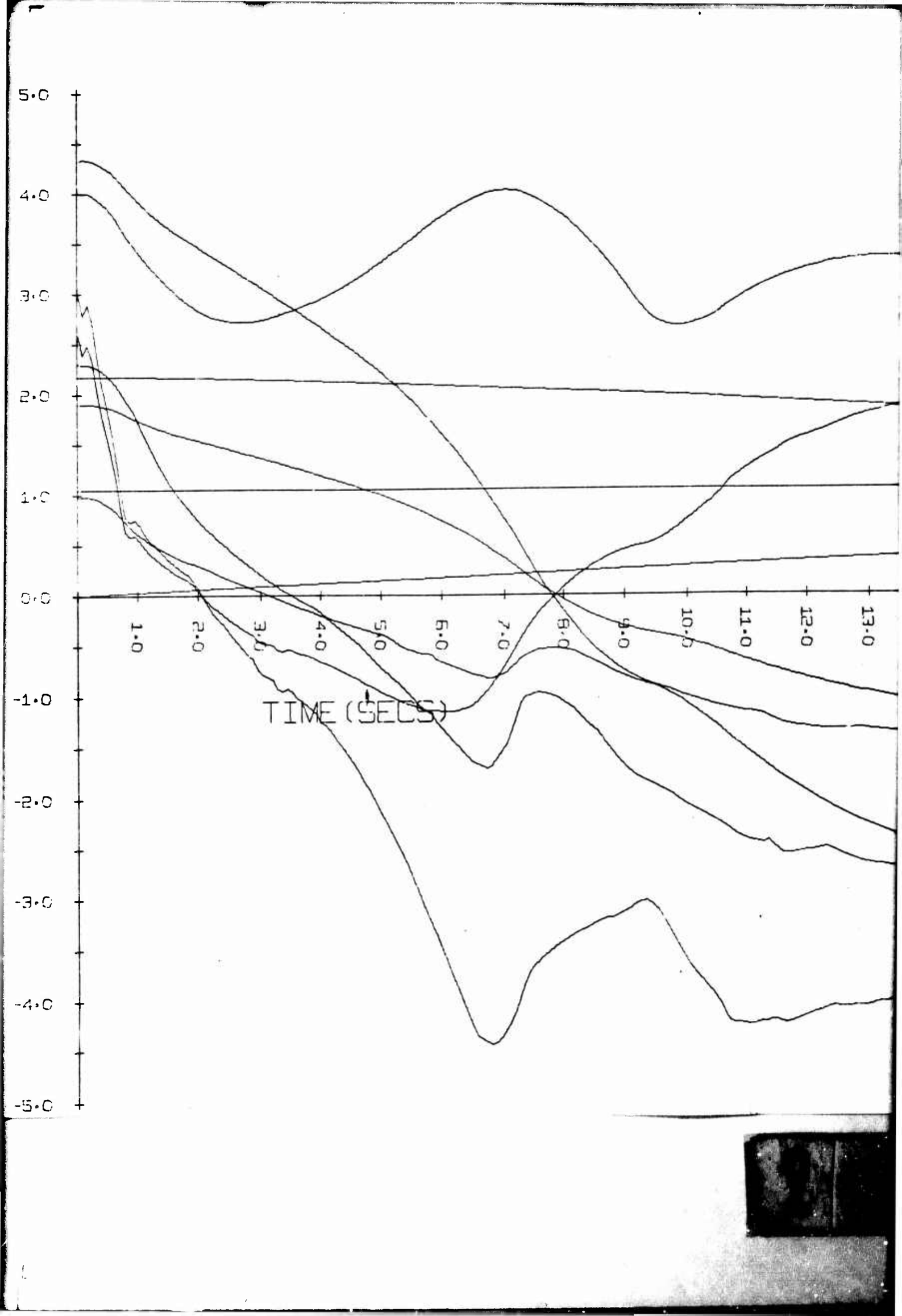
T(SEC)	RPMPM	CFGI	RPMP	VK	QDE(FT-LB)	QG	EB(V)	FMI	EC	QM(FT-LB)	AI(A)	QP	THRUST(T)	RESIST	HP/SH
0.000	799.9	34.3	152.3	17.3	46666.5	46678.4	864.8	84.0	859.0	238121.7	5997.1	229760.0	97.7	97.7	6906.0
0.399	780.9	31.0	150.6	17.3	19641.2	32765.8	854.9	84.0	849.7	167137.3	4209.4	222967.4	92.8	97.7	4794.8
0.799	717.9	26.8	143.6	17.3	-8733.8	12495.6	811.7	84.0	810.3	61945.6	1560.1	195226.2	72.7	97.4	1694.5
1.199	654.7	27.6	135.0	17.3	-8593.5	8026.0	762.8	84.0	761.6	43006.7	1083.1	146571.0	52.1	96.6	1105.7
1.599	605.1	23.4	128.3	17.3	-8483.3	5230.1	724.3	84.0	723.7	24573.5	618.8	104755.8	38.5	95.4	600.4
1.999	566.9	28.8	122.9	17.2	-8398.6	1119.7	693.0	84.0	693.3	5163.4	130.0	74314.1	28.5	94.0	120.8
2.399	547.1	26.4	117.5	17.2	-8354.6	-4742.2	662.2	84.0	663.1	-22159.7	-558.1	50993.6	18.1	92.5	-496.1
2.799	543.6	22.0	112.0	17.1	-8346.9	-9514.8	630.3	84.0	632.0	-46569.2	-1172.8	29961.6	8.0	90.7	-993.6
3.199	553.6	17.2	106.3	17.1	-8369.0	-12627.4	597.5	84.0	599.8	-66594.6	-1677.2	10677.9	-1.8	88.7	-1348.4
3.599	569.6	13.7	100.7	17.0	-8404.6	-14328.5	565.3	84.0	568.2	-80668.0	-2031.6	-5963.7	-11.8	86.5	-1547.4
3.999	588.3	10.3	95.1	17.0	-8448.4	-15763.4	533.1	84.0	536.6	-99826.0	-2514.1	-17723.4	-20.6	84.1	-1809.3
4.399	614.2	7.4	89.3	16.9	-8503.7	-17212.0	499.2	84.0	503.8	-121886.8	-3069.7	-38178.0	-28.8	82.7	-2073.0
4.799	644.4	4.0	83.0	16.8	-8570.6	-18807.4	462.8	84.0	468.6	-151782.0	-3822.6	-60381.0	-36.4	81.5	-2401.1
5.199	679.1	0.4	76.1	16.8	-8647.8	-20116.4	422.2	84.0	429.5	-189496.4	-4772.5	-83769.4	-46.8	80.1	-2747.5
5.599	716.2	-3.4	68.1	16.7	-8730.1	-20466.5	375.1	84.0	384.1	-232219.5	-5848.5	-108759.1	-58.4	78.6	-3011.3
5.999	752.2	-7.4	59.6	16.6	-8809.9	-19487.2	320.0	84.0	330.8	-278201.1	-7006.5	-133124.5	-67.7	77.0	-3107.6
6.399	782.3	-11.6	47.7	16.5	-8877.9	-17113.0	256.8	84.0	269.5	-328805.3	-8281.0	-161448.4	-77.9	75.3	-2992.1
6.799	802.0	-14.4	35.5	16.4	-8920.4	-12343.4	187.6	84.0	200.6	-351877.2	-8862.1	-169369.4	-83.8	73.5	-2383.8
7.199	809.2	-14.6	22.2	16.3	-8923.2	-6255.6	113.9	84.0	125.3	-324762.3	-8179.2	-121124.4	-68.7	71.7	-1274.2
7.599	785.8	-14.3	8.4	16.3	-8884.5	-787.5	37.3	84.0	47.6	-287237.7	-7234.1	-97147.9	-55.4	70.2	-461.9
7.999	753.4	-15.0	-3.8	16.2	-8812.5	3460.4	-31.7	84.0	-21.9	-271641.8	-6841.3	-106818.4	-55.5	68.7	201.2
8.399	708.3	-16.0	-14.2	16.1	-8712.4	7273.1	-89.8	84.0	-80.4	-260920.6	-6571.3	-127588.0	-64.6	67.2	708.3
8.799	653.2	-17.0	-22.2	16.0	-8587.9	10712.4	-134.6	84.0	-125.3	-253439.0	-6382.9	-156060.6	-77.7	65.5	1072.6
9.199	596.4	-17.7	-27.7	15.9	-7662.0	13252.8	-164.9	84.0	-156.2	-242410.1	-6105.1	-179584.3	-88.1	63.8	1276.6
9.599	539.4	-19.5	-31.6	15.8	10509.5	17437.1	-188.3	84.0	-178.5	-253537.1	-6385.4	-192579.3	-95.4	62.8	1526.4
9.999	535.8	-22.2	-36.5	15.8	25985.3	22900.0	-217.1	84.0	-206.0	-287419.5	-7238.7	-207509.3	-103.9	61.7	1999.2
10.399	562.9	-24.2	-42.4	15.7	34738.1	27936.0	-251.0	84.0	-239.1	-310420.5	-7818.0	-220668.0	-111.7	60.6	2506.1
10.799	584.2	-26.2	-49.0	15.5	42000.0	32892.5	-289.1	84.0	-276.4	-335922.0	-8460.3	-235529.1	-117.1	59.3	3134.9
11.199	611.2	-26.8	-55.5	15.4	42000.0	36953.6	-325.8	84.0	-313.2	-336952.1	-8486.2	-244883.3	-121.8	58.1	3562.7
11.599	632.3	-27.3	-61.4	15.3	42000.0	39451.3	-359.5	84.0	-346.8	-335747.1	-8455.9	-255562.4	-131.8	56.8	3930.9
11.999	647.3	-27.6	-66.8	15.2	42000.0	37916.7	-389.6	84.0	-377.2	-331186.5	-8341.0	-253406.7	-136.0	55.4	4217.6
12.399	659.1	-27.8	-72.0	15.1	42000.0	38750.2	-415.3	84.0	-406.1	-323775.6	-8154.4	-251980.3	-138.2	54.0	4439.0
12.799	666.8	-28.2	-76.4	15.0	42000.0	40422.7	-443.2	84.0	-431.0	-323551.2	-8148.7	-263253.5	-137.5	52.6	4707.8
13.199	669.7	-28.6	-80.1	14.9	42000.0	41992.0	-464.1	84.0	-452.0	-320316.3	-8067.2	-268368.3	-139.8	51.5	4888.1
13.599	669.4	-28.9	-83.3	14.8	42000.0	42815.7	-481.9	84.0	-470.0	-316053.7	-7959.9	-271090.6	-141.9	50.5	5014.9
13.999	666.6	-29.1	-86.0	14.7	42000.0	43125.0	-496.9	84.0	-485.2	-311010.0	-7832.9	-272452.6	-144.2	49.5	5095.0
14.399	661.8	-29.3	-88.3	14.5	42000.0	43790.0	-509.6	84.0	-498.1	-305654.0	-7698.0	-272735.1	-145.8	48.4	5139.8
14.799	655.4	-29.5	-90.2	14.4	42000.0	44236.6	-520.1	84.0	-508.8	-300120.0	-7558.6	-272170.1	-146.9	47.4	5155.5
15.199	647.6	-29.7	-91.7	14.3	42000.0	44696.0	-528.6	84.0	-517.4	-295252.8	-7436.0	-273077.0	-146.2	46.3	5158.0
15.599	639.3	-29.9	-92.8	14.2	42000.0	45116.2	-534.8	84.0	-523.8	-290689.6	-7321.1	-273836.3	-144.8	45.3	5141.0
15.999	631.8	-30.1	-93.6	14.1	42000.0	45433.7	-539.1	84.0	-528.3	-285961.8	-7202.0	-273664.8	-143.2	44.2	5100.3
16.399	625.5	-30.2	-94.1	14.0	42000.0	45663.5	-541.7	84.0	-531.1	-281130.3	-7080.3	-272725.5	-141.5	43.1	5040.5
16.799	620.6	-30.3	-94.4	13.9	42000.0	45761.2	-542.9	84.0	-532.4	-275898.0	-6946.5	-271208.2	-139.7	42.3	4959.4
17.199	592.5	-30.8	-94.4	13.8	42000.0	45787.2	-542.7	84.0	-532.4	-270842.1	-6821.2	-269117.5	-137.8	41.5	4868.6
17.599	580.5	-31.0	-94.2	13.7	42000.0	45738.0	-541.4	84.0	-531.4	-265861.4	-6695.8	-266577.0	-136.0	40.7	4769.5
17.999	563.8	-31.2	-93.8	13.6	42000.0	45650.2	-539.3	84.0	-529.5	-261116.8	-6576.3	-263706.2	-134.1	40.0	4667.6

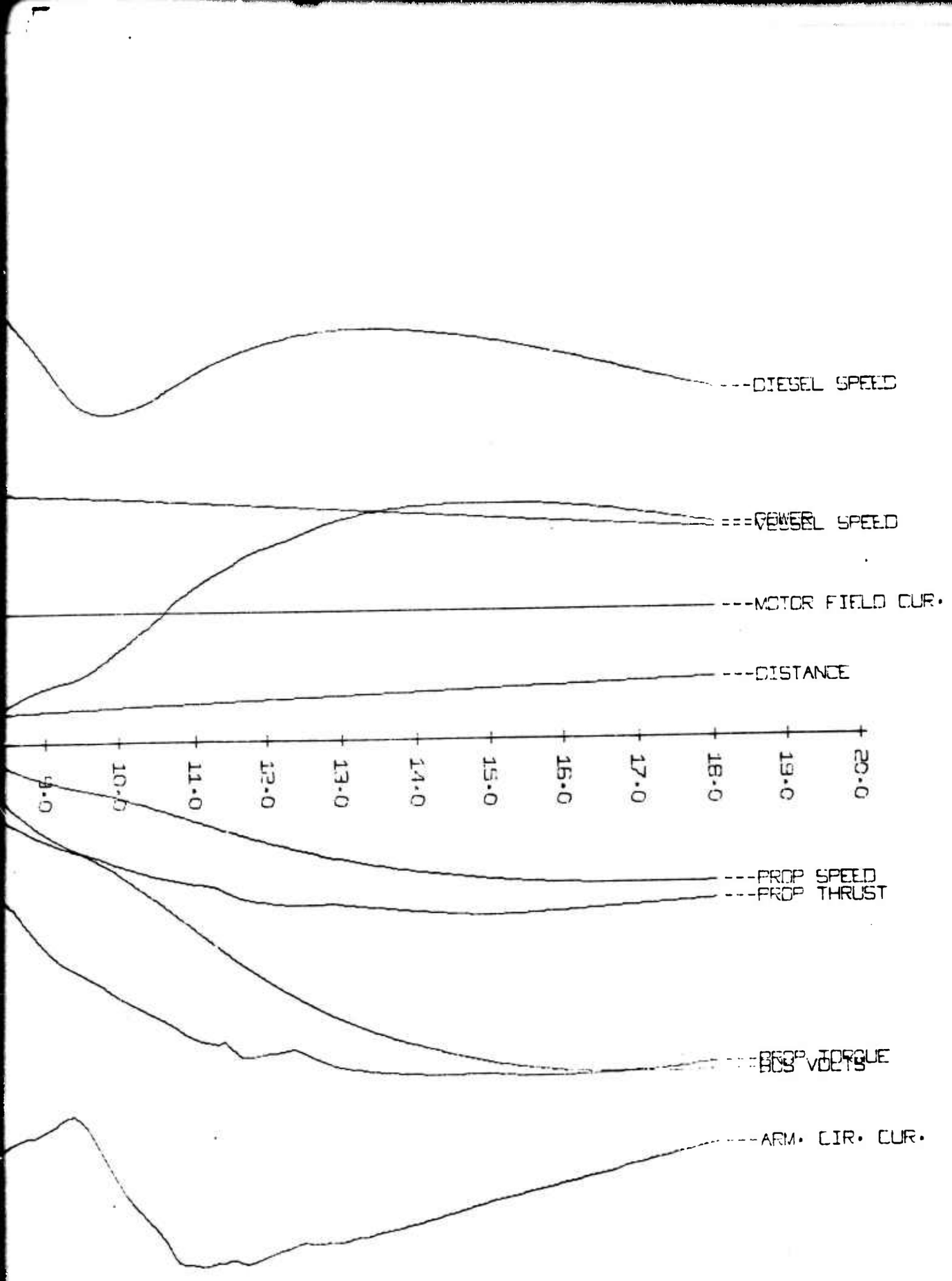
XS(1 IN.=1000.0 FEET)
TP(1 IN.=100000.0 LBS)
QP(1 IN.=100000.0 FT-LBS)
VK(1 IN.= 8.0 KTS.)
KW(1 IN.=2000.0 KW)
NP(1 IN.= 80.0 RPM)
FMI(1 IN.= 80.0 AMPS)
NPM(1 IN.=200.0 RPM)
EB(1 IN.=200.0 VOLTS)
AI(1 IN.= 2000.0 AMPS)

SERIAL NUMBER-- 1.

SIMULATION OF TEST SHIP FOR DOCUMENTATION

CRASH REVERSAL OF A 12000 TON SHIP


5.0
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-4.0
-5.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
TIME (SECS)


---DIESEL SPEED
---POWER
---VESSEL SPEED
---MOTOR FIELD CUR.
---DISTANCE
9.0
10.0
11.0
12.0
13.0
14.0
15.0
16.0
17.0
18.0
19.0
20.0
---PROP SPEED
---PROP THRUST
---PROP TORQUE
---BUS VOLTS
---ARM. CIR. CUR.

3

APPENDIX

SYMBOL TABLE

FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
RO	ρ		Mass density of salt water equal to 1.99
PI	π		Constant equal to 3.14159
IZB		√	Number of propeller blades
DAR	DAR	√	Developed area ratio of propeller
PDR	P/D	√	Pitch-diameter ratio of propeller
D	D	√	Propeller diameter
XJPA	J_{pair}	√	Propeller inertia in air
XJM	J_m	√	Motor inertia
XJSH	J_{sh}	√	Shaft inertia
XJD		√	Diesel inertia
XJG		√	Generator inertia
EFG	e_{fg}		Ordered generator field voltage
XJMP	J_{mp}		Inertia of propeller-motor system
XJDG	J_{pmg}		Combined diesel-generator inertia
FIMAX	i_{fmax}	√	Maximum value of motor field current
RA	R_a		Armature circuit resistance
RG	R_g	√	Generator armature resistance
C1	C_1	√	Constants used in calculating ordered armature current in motor regulator
XK1	K_1	√	
QBRG		√	Y-intercept of developed diesel torque vs. angular velocity of prime mover curve
CSHFR	C_2	√	Shaft friction constant
CWM		√	Slope of developed diesel torque vs. angular velocity of prime mover curve
PCMS		√	Entrained water inertia

**a check(√) implies the variable is initially read in by subroutine REDAT.

FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
XLA	L_a	√	Armature circuit inductance
TAU1	τ_1	√	Time for bridge controller to reach position θ_1 from initial position θ_{ic}
TAU2	τ_2	√	Time for bridge controller to go from position θ_1 to the final (ordered) position θ_c
XK6	K_6	√	Developed diesel torque constant
ALP1	α_1	√	Buffer spring scale
ALP_2	α_2	√	Governor to engine constant
ALP_3	α_3	√	Needle valve adjustment
ALPD	α_d	√	Speed droop coefficient
XMASS			Used to contain the value $2240.(1.+C_{ad})\Delta/32.18$ used in the computation of the ship acceleration.
NØVAR			(Graph routine) indicates number of variables to be plotted
NVAR			(Graph routine) 10-word array containing the code numbers of up to 10 variables to be plotted
JJ			(Graph routine) index to NVAR array
ICNTR		√	Indicates particular command given; equal to one (1) for ahead flank, 2 for back flank, 3 for ahead full, and 4 for ahead full in ice.
NPLOT			(Graph routine) indicates if a grid is to be plotted; equal to zero (0) if a grid not desired, equal to one (1) if a grid is requested.

** a check (√) implies the variable is initially read in by subroutine REDAT.

FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
NDATA			(Not used)
NON		√	Indicates the number of time intervals between listings of the output on the printer.
NIP			(Graph routine) index to NVAR array
SX		√	(Graph routine) plotter scale factor for time axis
SY			(Graph routine) plotter scale factor for variable currently being plotted.
SY1		√	(Graph routine) plotter scale factor for armature circuit current
SY2		√	(Graph routine) plotter scale factor for bus volts
SY3		√	(Graph routine) plotter scale factor for diesel speed
SY4		√	(Graph routine) plotter scale factor for motor field current
SY5		√	(Graph routine) plotter scale factor for propeller speed
SY6		√	(Graph routine) plotter scale factor for power
SY7		√	(Graph routine) plotter scale factor for vessel speed
SY8		√	(Graph routine) plotter scale factor for propeller torque
SY9		√	(Graph routine) plotter scale factor for propeller thrust.

** a check (√) implies the variable is initially read in by subroutine REDAT.

FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
SY10		√	(Graph routine) plotter scale factor for ship's distance
SY11		√	(Graph routine) plotter scale factor for ice torque
SY12		√	(Graph routine - not used)
DT	dt	√	Time interval
CT			Save area for current time
T	t	√	Current time
TEND		√	Time ending value
TIMSC		√	Time increment for writing values on the disk
Z	Z	√	Servo stroke position
Y	dz/dt	√	Servo stroke velocity
XNPM	n_{pm}	√	Angular velocity of prime mover
FGI	i_{fg}	√	Generator field current
FIP	i'_f	√	Motor field adjusting current
XNP	n_p	√	Propeller angular velocity
VS	V_s	√	Ship speed
XS		√	Ship distance
SIE			Angular position of propeller
AI	i_a	√	Armature circuit current
RV		√	(Not used)
TEMP		√	(Not used)
CZ			These 12 variables are related to the 12 variables immediately above (relationship indicated by matching labels after dropping the initial C)
CY			
CXNPM			
CFGI			

** a check (√) implies the variable is initially read in by subroutine REDAT.

FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
CFIP			(continued from previous page)
CXNP			and are used to save the values
CVS			of the above variables at each
CXS			step when the next set of values
CSIE			are being calculated.
CAI			(CRV and CTEMP are not used)
CRV			
CTEMP			
ZDO	dz/dt		Servo stroke velocity
YDO	d^2z/dt^2		Servo stroke acceleration
XNPMD	dn_{pm}/dt	√	Time rate of prime mover speed
FGIDO	di_{fg}/dt		Time rate of generator field current
FIPDO	di'_f/dt		Time rate of motor field adjusting current
XNPDO	dn_p/dt		Propeller acceleration
VSDO	dV_s/dt		Ship's acceleration
XSDO	V_s		Ship's velocity
SIEDO	n_p		Propeller angular velocity
AIDO	di_a/dt	√	Time rate of armature circuit current
RVDO			(Not used)
TEMDO			(Not used)
XNE	n_e		Diesel error speed
XNEDO	dn_e/dt		time rate of diesel error speed
VA	V_a		Advance velocity of propeller
THETO	θ^*		Ordered (final) bridge lever position
THET1	θ_1	√	Bridge lever position reached in τ_1 seconds

** a check (√) implies the variable was initially read in by subroutine REDAT.

FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
THETA	θ		Actual bridge level position
THETD	$d\theta/dt$		Time rate of bridge lever position
THEIC	θ_{ic}	√	Initial value of bridge lever position
QDE	Q_{de}		Developed diesel torque
QG	Q_g		Generator torque
EG	e_g		Generator voltage
EB	e_b		Bus voltage
EC	e_c		Back emf
QM	Q_m		Motor torque
QFRM	Q_{frm}		Shaft friction torque
AIHAT	$\hat{i}_a$		Ordered armature circuit current
QP	Q_p		Propeller torque
QIP			Ideal propeller torque
QI			Ice torque
TP	T_p		Total propeller thrust
XPROI	T		Thrust per propeller
R	R		Ship's resistance
VAR			(Graph routine) contains variable retrieved from the disk
XKW			Ship's power in kilowatts
CVK			Ship's speed in knots
CFG	C_{fg}		Generator flux
SPGEN		√	Spacing of generator, independent axis for CFGA array
CFGA		√	13-word array containing generator flux for spacing indicated by SPGEN
TRNFG	N_{fg}	√	Number of turns in generator
ZGP	Z'_g	√	Generator flux conversion factor

** a check (√) implies the variable was initially read in by subroutine REDAT.

FORTRAN LABEL	MATH SYMBOL (if any)	**	DEFINITION
RFG	R_{fg}	√	Generator field resistance
XLFG	L_{fg}		Generator field inductance
TAUFG	τ_{fg}		Generator time constant
TRNSF	N_{sf}	√	Number of turns in generator field
RX		√	Variable resistance in generator field
XM			Mutual inductance between generator separate and in series fields
XLASF	L_{asf}		Inductance of generator series field
AMPTR			Ampere turns of generator
XGEN		√	Number of generators
CFM	C_{fm}		Motor field flux
SPMOT		√	Spacing of motor, independent axis for CFMA array
CFMA		√	13-word array containing motor flux for spacing indicated by SPMOT
TRNFM	N_{fm}	√	Number of turns in motor
ZMP	Z'_m	√	Motor flux conversion constant
RFM	R_{fm}	√	Motor field resistance
XLFM	L_{fm}		Motor field inductance
TAUFM	τ_{fm}		Motor field time constant
FMIRV		√	Field current increment for reverse direction
TAUA	τ_A	√	Time constants controlling how fast
TAUB	τ_B	√	FMIRV gets added to FMI
B	K_T	√	64-word array containing propeller thrust coefficients using J
BB	K_Q	√	64-word array containing propeller torque coefficients using J

** a check (√) implies the variable was initially read in by subroutine REDAT.

FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
E	C_T	√	32-word array containing propeller thrust coefficients using 1/J
EE	C_Q	√	32-word array containing propeller torque coefficients using 1/J
EHP		√	20-word array containing effective horsepower for hull
WDED	(1-w)		Wake deduction for hull
WADED		√	20-word array of wake deduction factors for hull
TDED	(1-t)		Thrust deduction for hull
THDED		√	20-word array of thrust deduction factors for hull
XKK	K_n	√	Number of propeller shafts
JELO			Disk file record index
JELO			Controls initial computation of ice torque
KELO			Indicates position of ship in ice flow
LELO			(Not used)
MELO			(Not used)
SPARE			5-word array, the first 3 words of which are used as follows: SPARE(1) equivalenced to FMI (i_{fm}) -motor field current SPARE(2) equivalenced to XNGR(n_{gr}) -ordered diesel speed SPARE(3) equivalenced to XNGRD (dn_{gr}/dt) -ordered diesel speed rate

** a check (√) implies the variable was initially read in by subroutine REDAT.

FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
ISPAR			5-word array, the first 2 words of which are used as follows: ISPAR(1)-controls calling of GENIS mainline from EXEC mainline ISPAR(2) -controls motor field plugging in CALVR
CAD	C_{ad}	√	"Added mass" coeffieient of hull
DELTA	Δ	√	Vessel displacement
RM		√	Motor armature resistance
RBA		√	25-word array containing random depths of cut propeller blade makes in the ice as a percentage of the diameter
RBMX	R_{bx}		Maximum depth of cut the propeller blade makes in the ice
BLOC1		√	Length of first block of ice
DBB12		√	Distance between ice blocks 1 and 2
BLOC2		√	Length of ice block 2
DBB21		√	Distance between ice blocks 2 and 1 in repeating sequence
EO	e_o	√	Tip thickness of propeller blade
EK	e_k	√	Root thickness of propeller blade
DCT	D_{ct}	√	Diameter of propeller hub
DCR	D_{cr}	√	Crushing pressure of ice
DSH	D_{sh}	√	Shear pressure of ice
CICE		√	Constant used to increase hull resistance for ice conditions.

**a check (√) implies the variable was initially read in by subroutine REDAT.

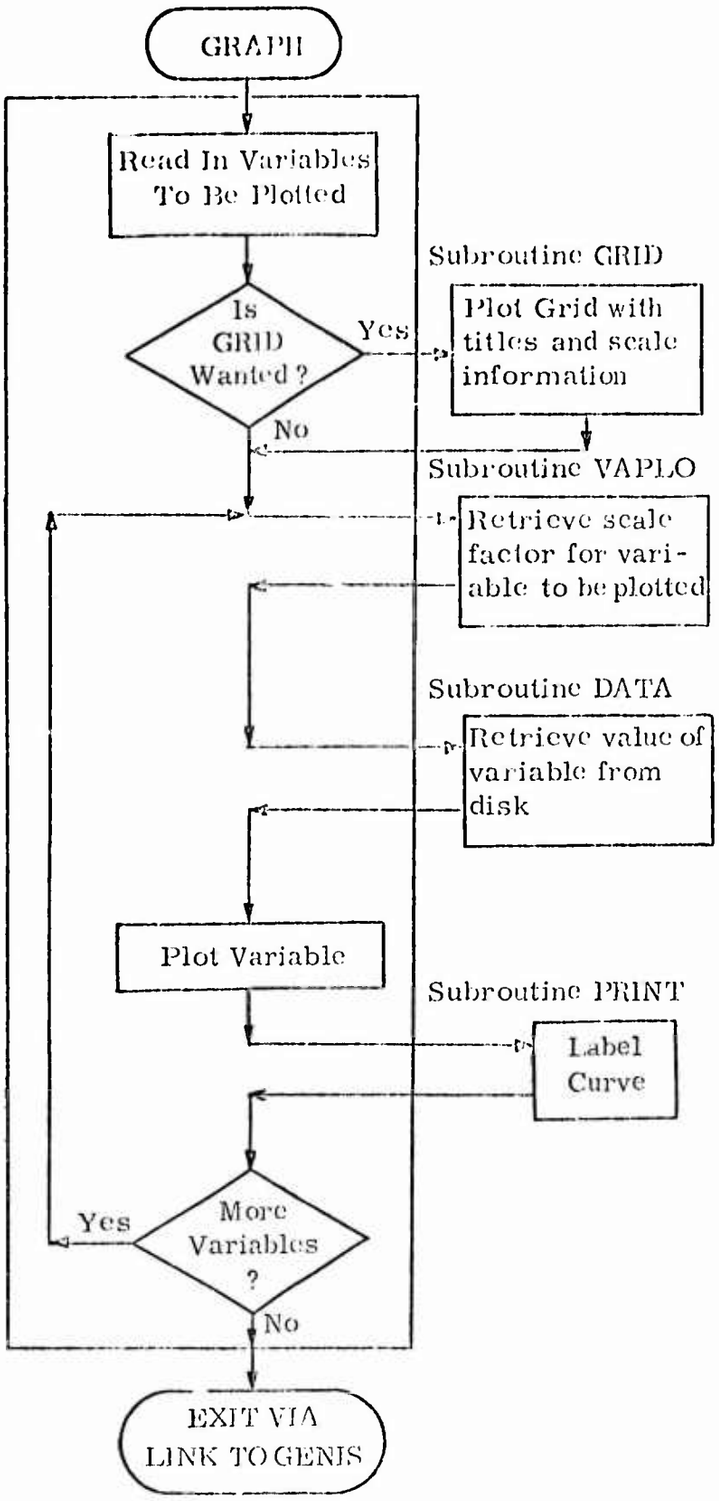
FORTRAN LABEL	MATH SYMBOL (if any)	**	DESCRIPTION
TICE		√	Relative time from beginning of run until ice is encountered
CK1		√	(Not used)
CK2		√	(Not used)
CK3		√	(Not used)

** a check (√) implies the variable initially read in by subroutine REDAT.

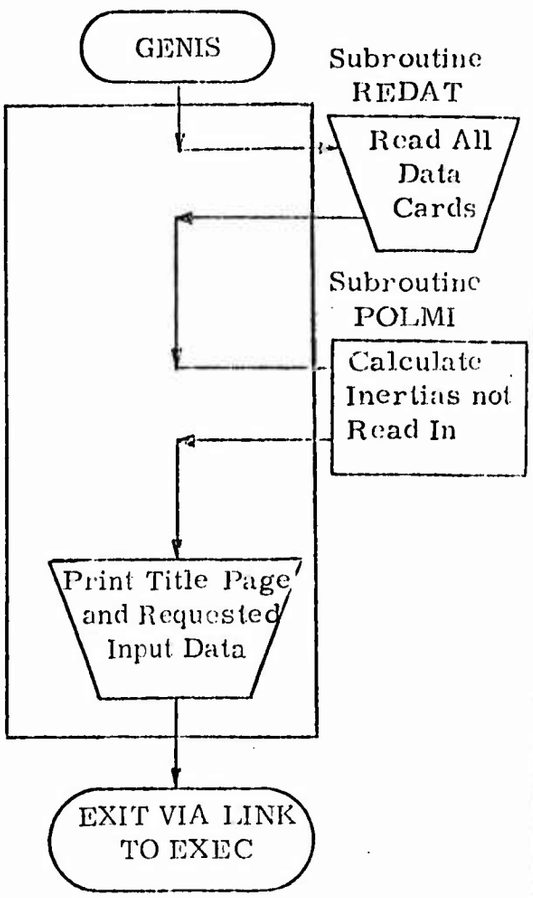
SYSTEM FLOW CHARTS

The following flow charts outline the flow of control in the three mainline programs (GENIS, EXEC, and GRAPH) during a simulation run.

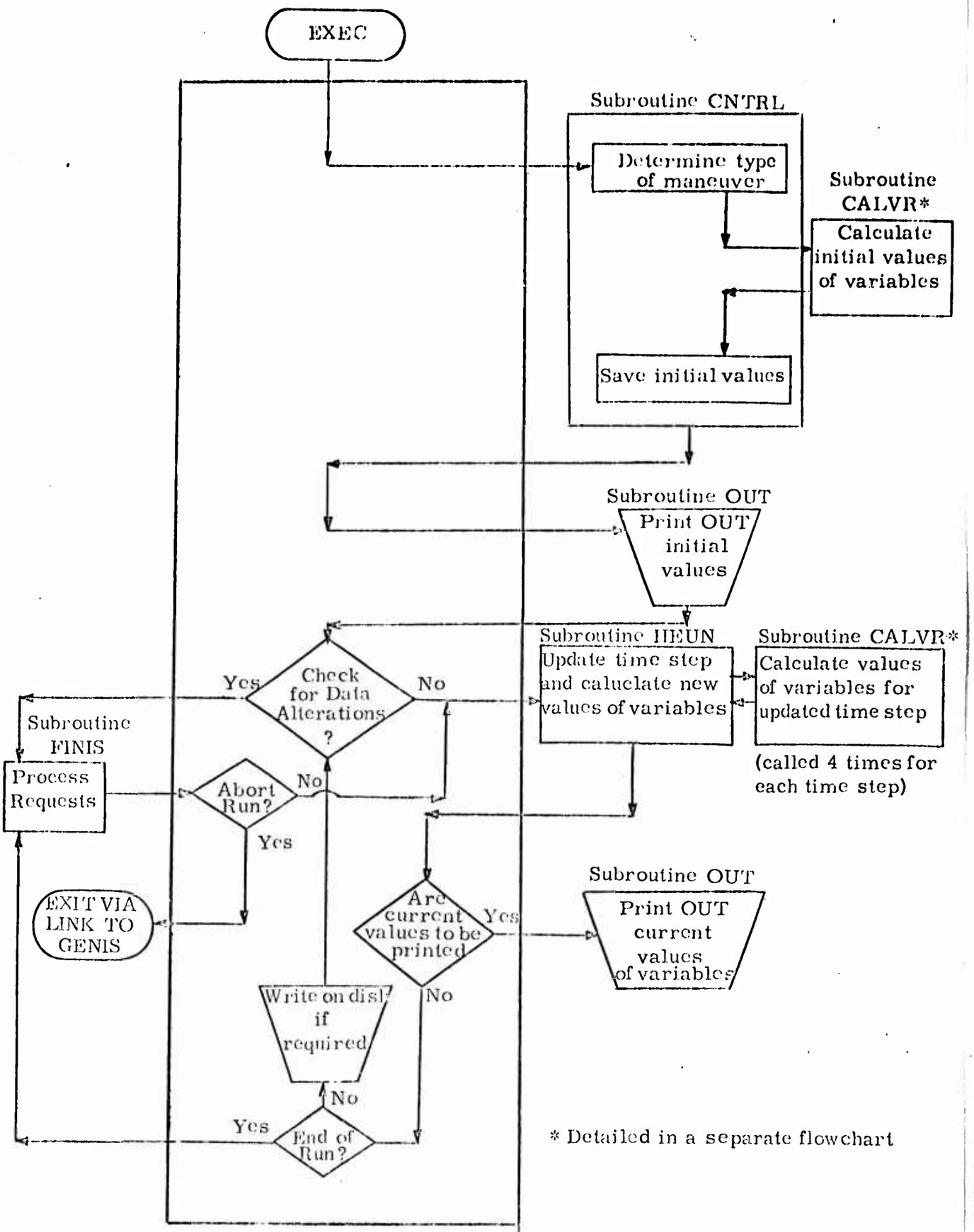
GRAPH MAINLINE



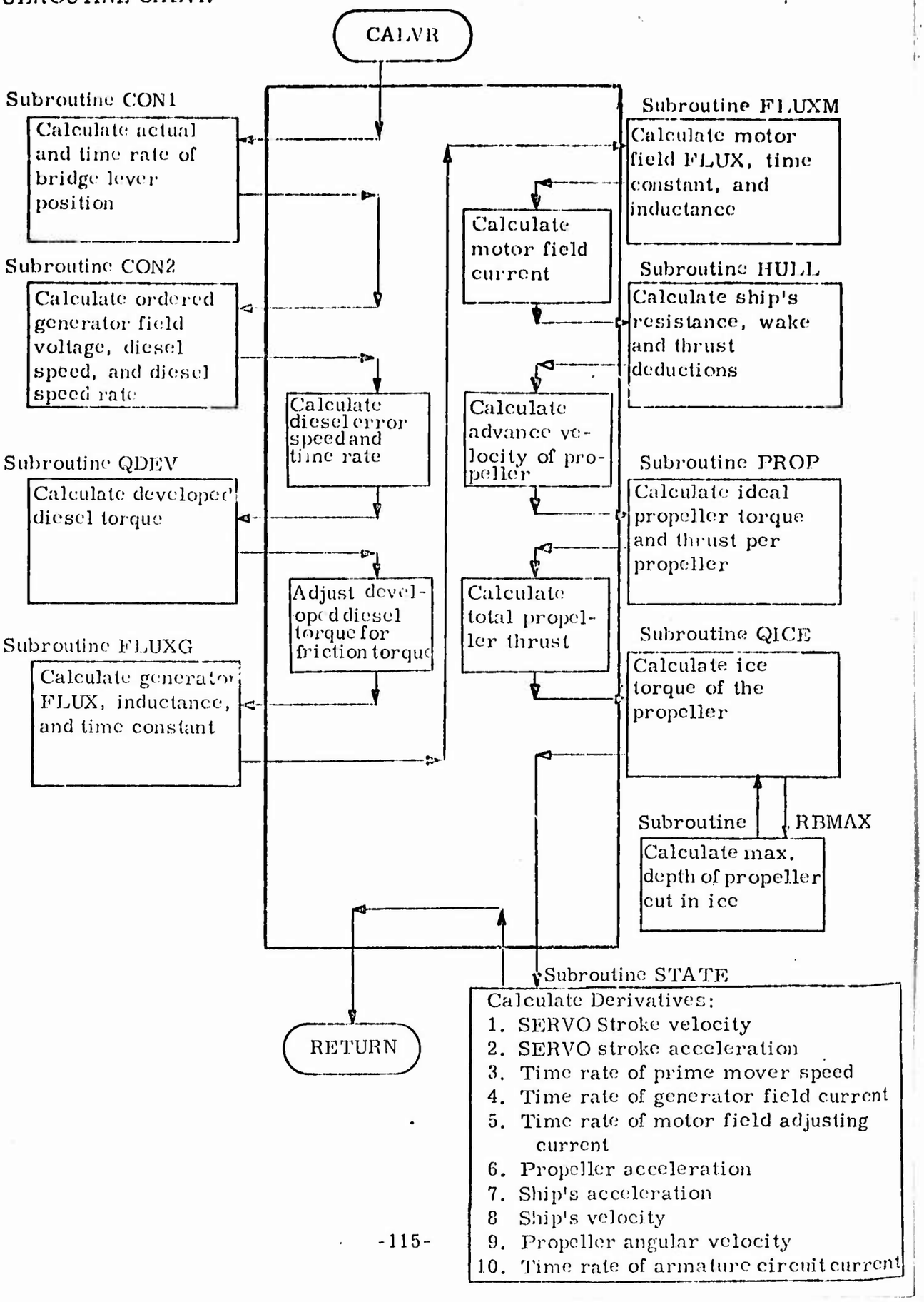
GENIS MAINLINE



EXEC MAINLINE



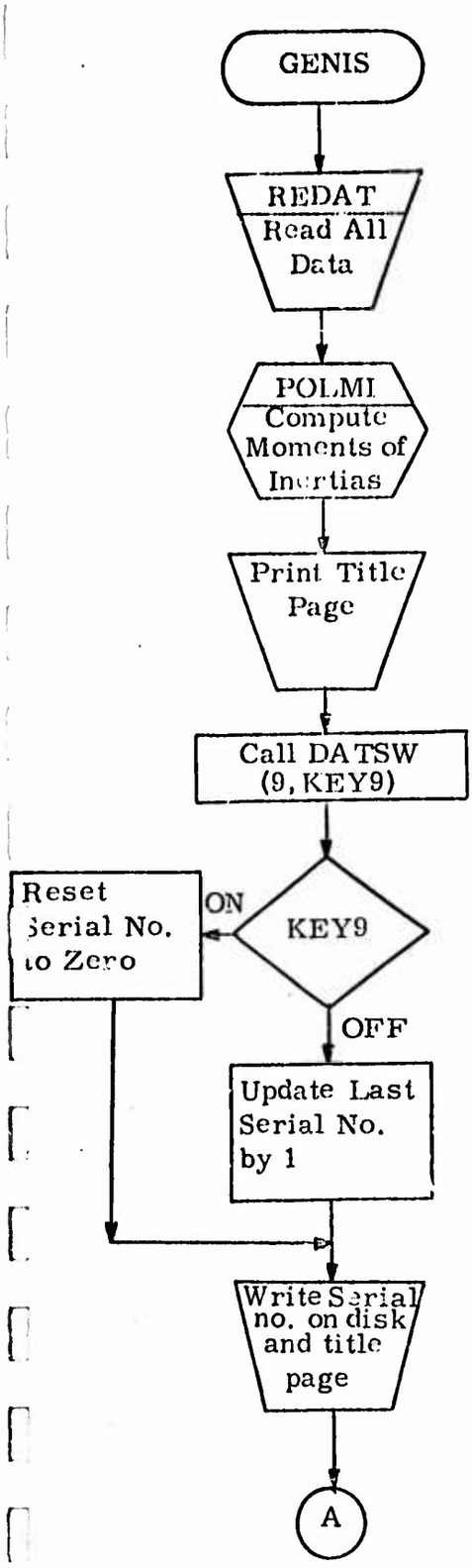
SUBROUTINE CALVR

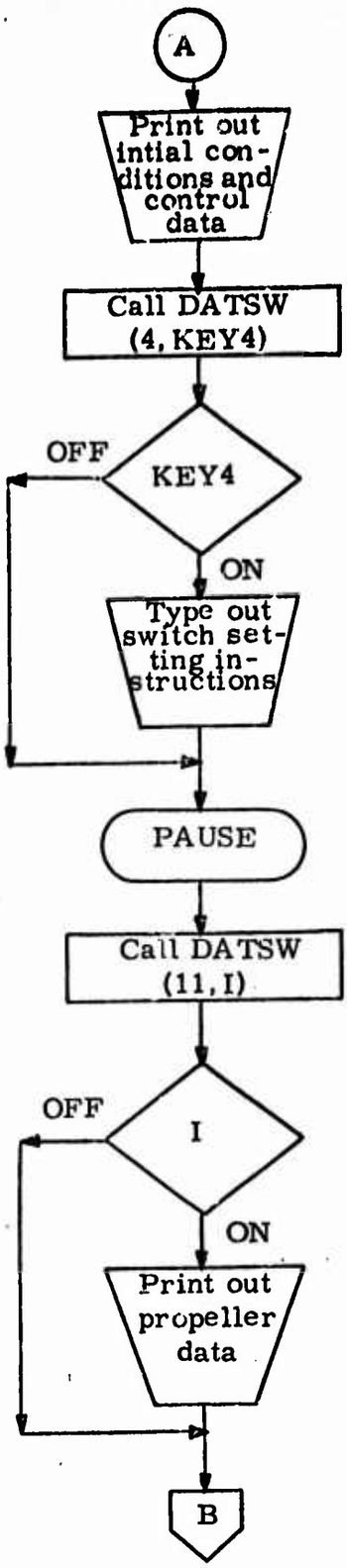


MAINLINE PROGRAMS AND SUBROUTINE FLOW CHARTS

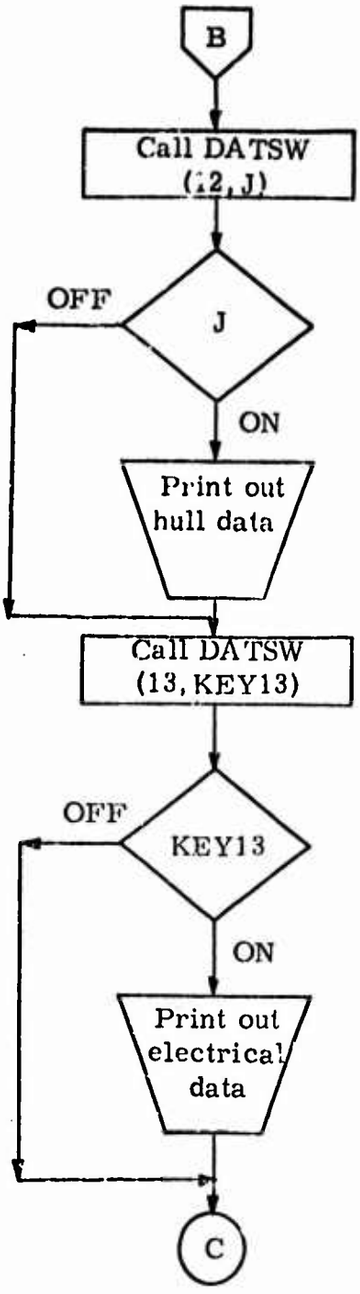
The following flowcharts outline in detail the processing associated with each mainline program and each subroutine.

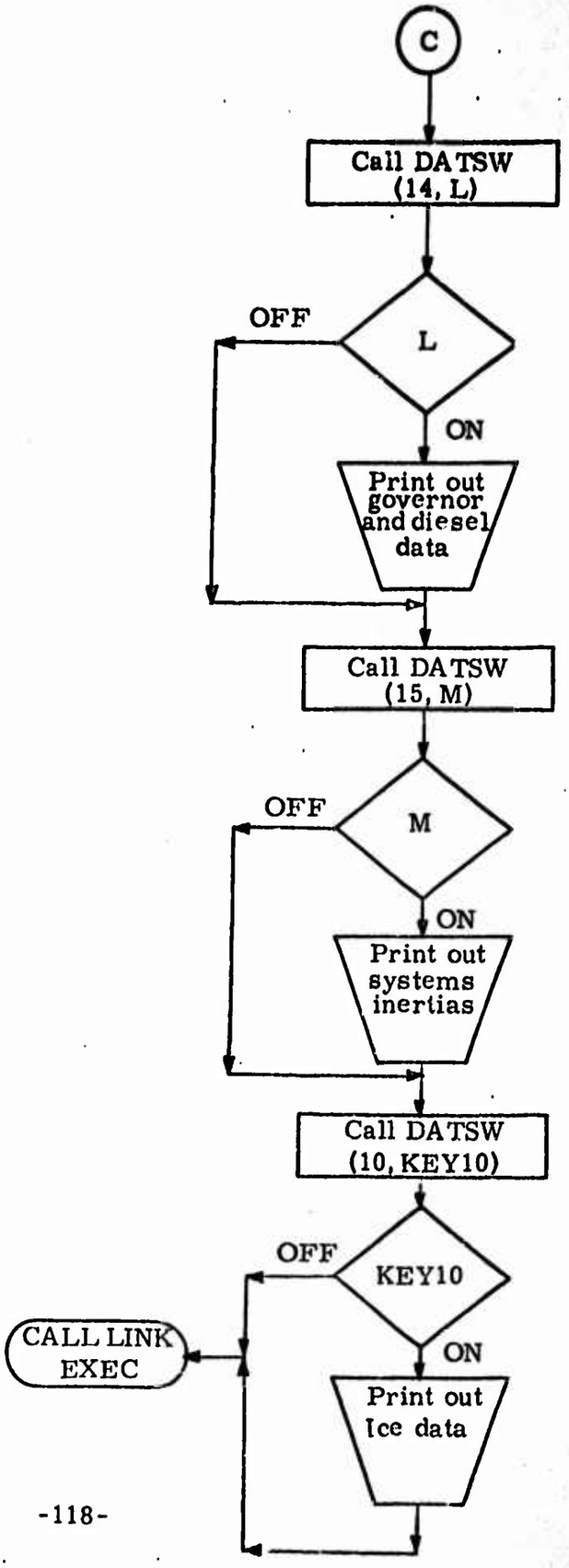
GENIS MAINLINE



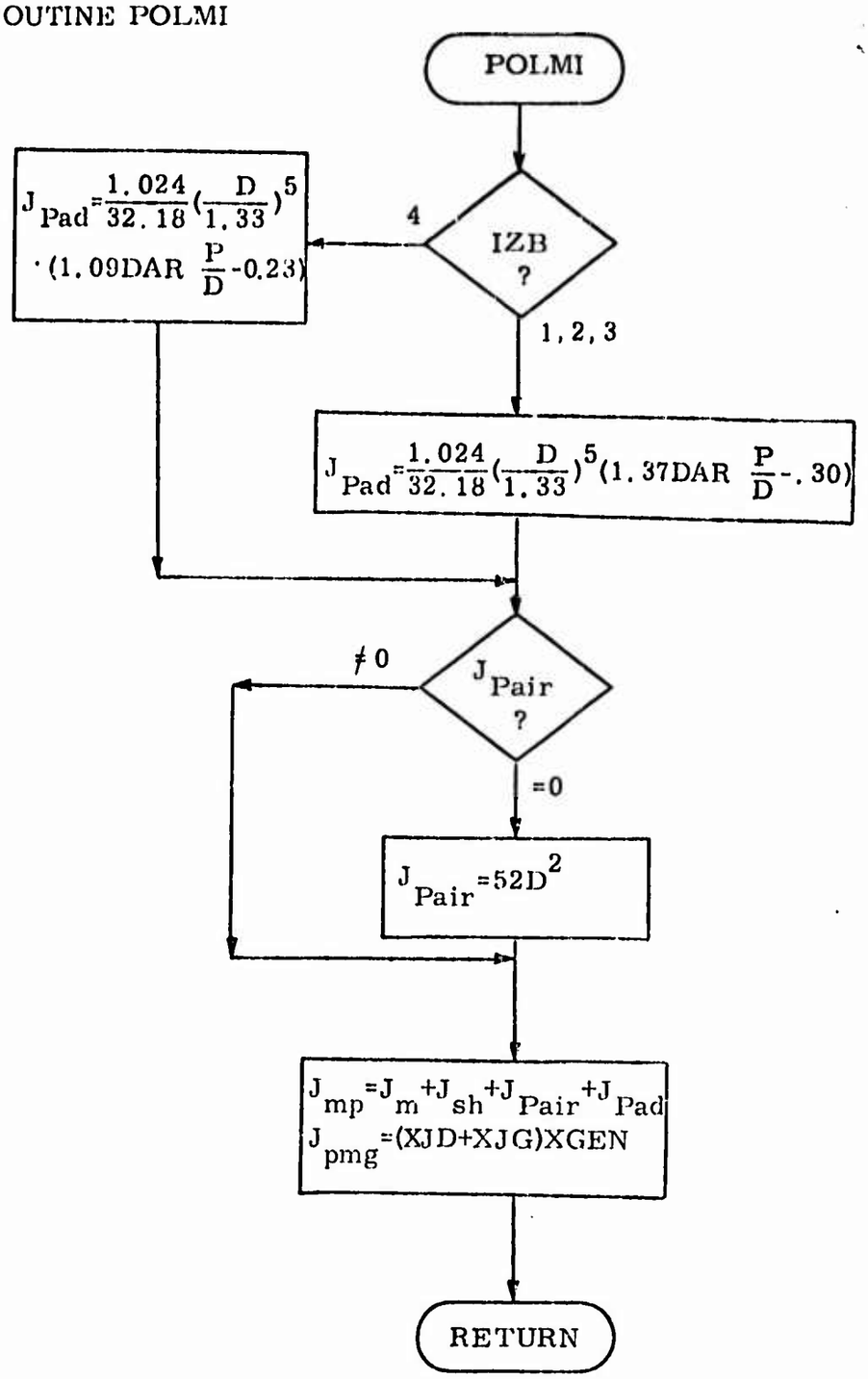


GENIS MAINLINE (continued)

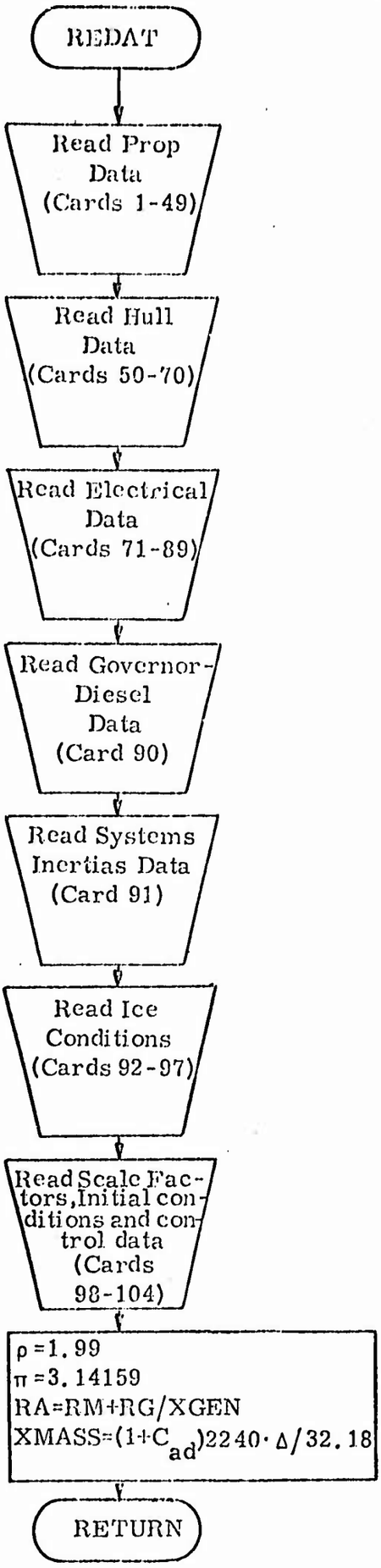




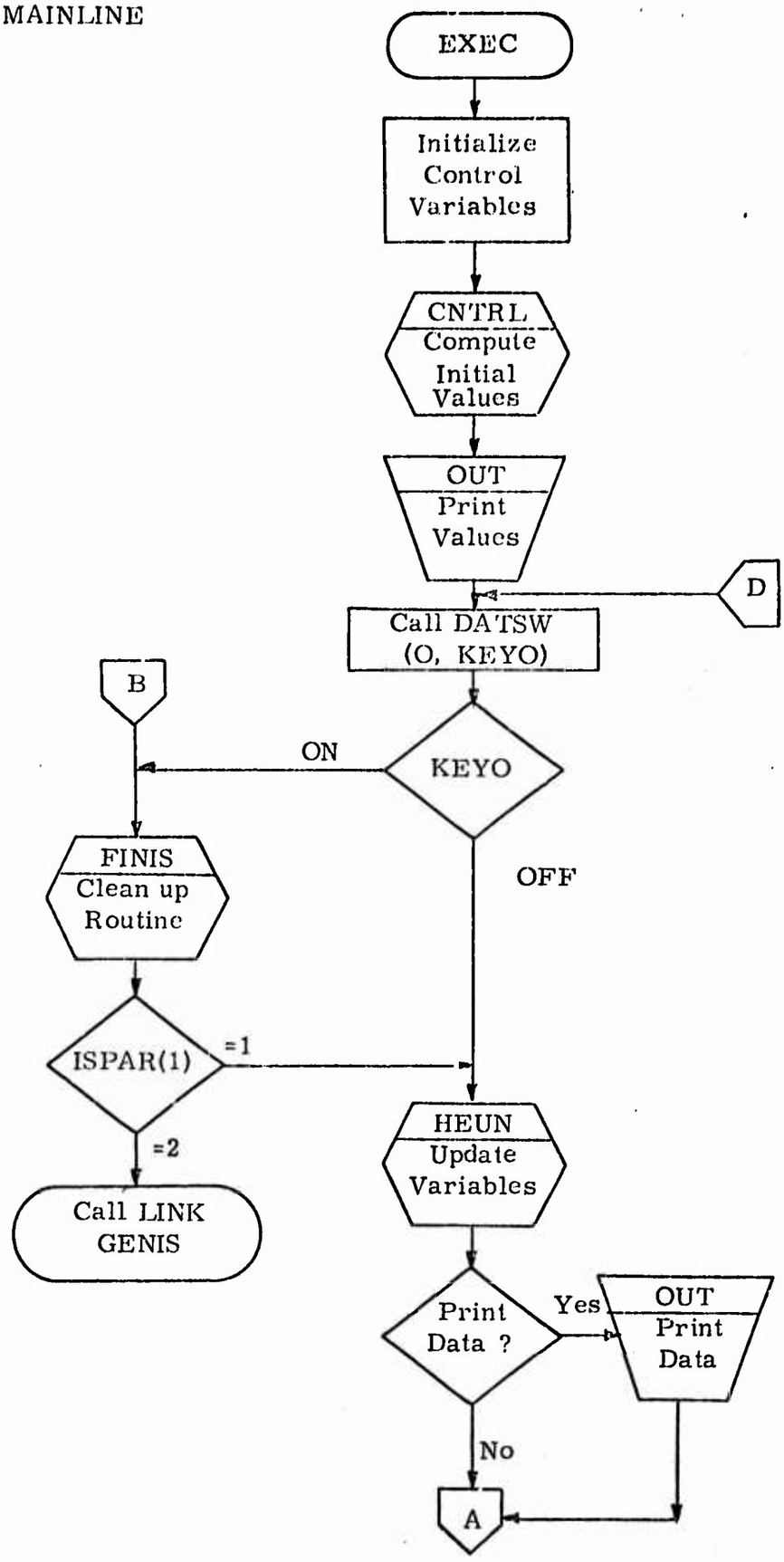
SUBROUTINE POLMI



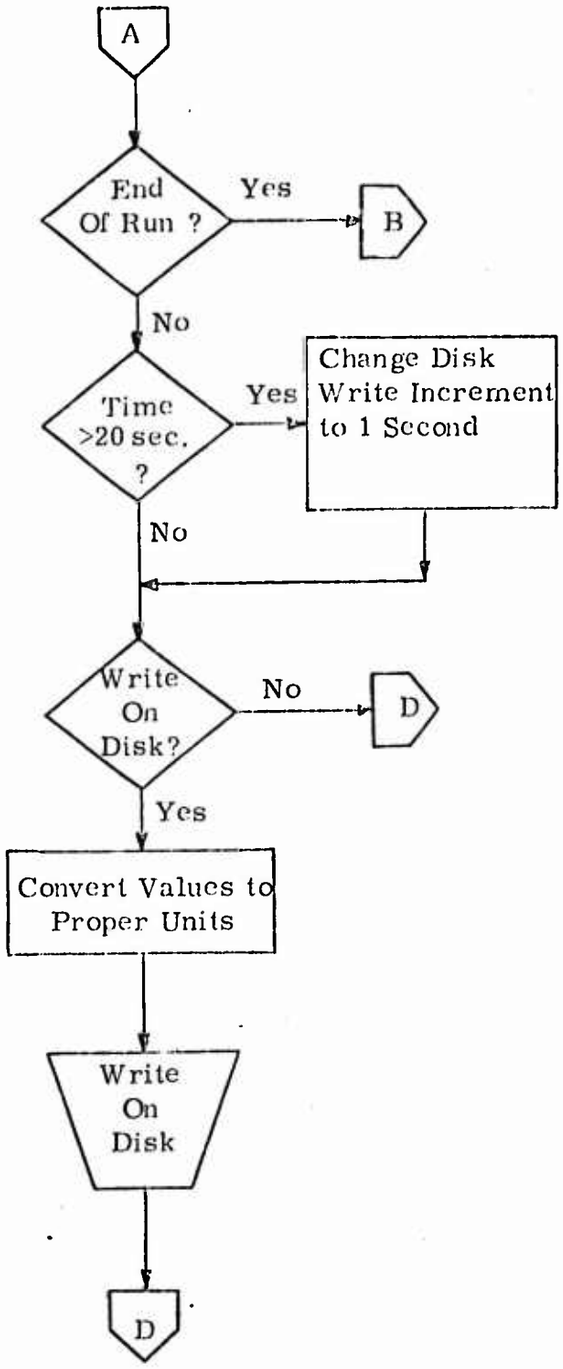
SUBROUTINE REDAT



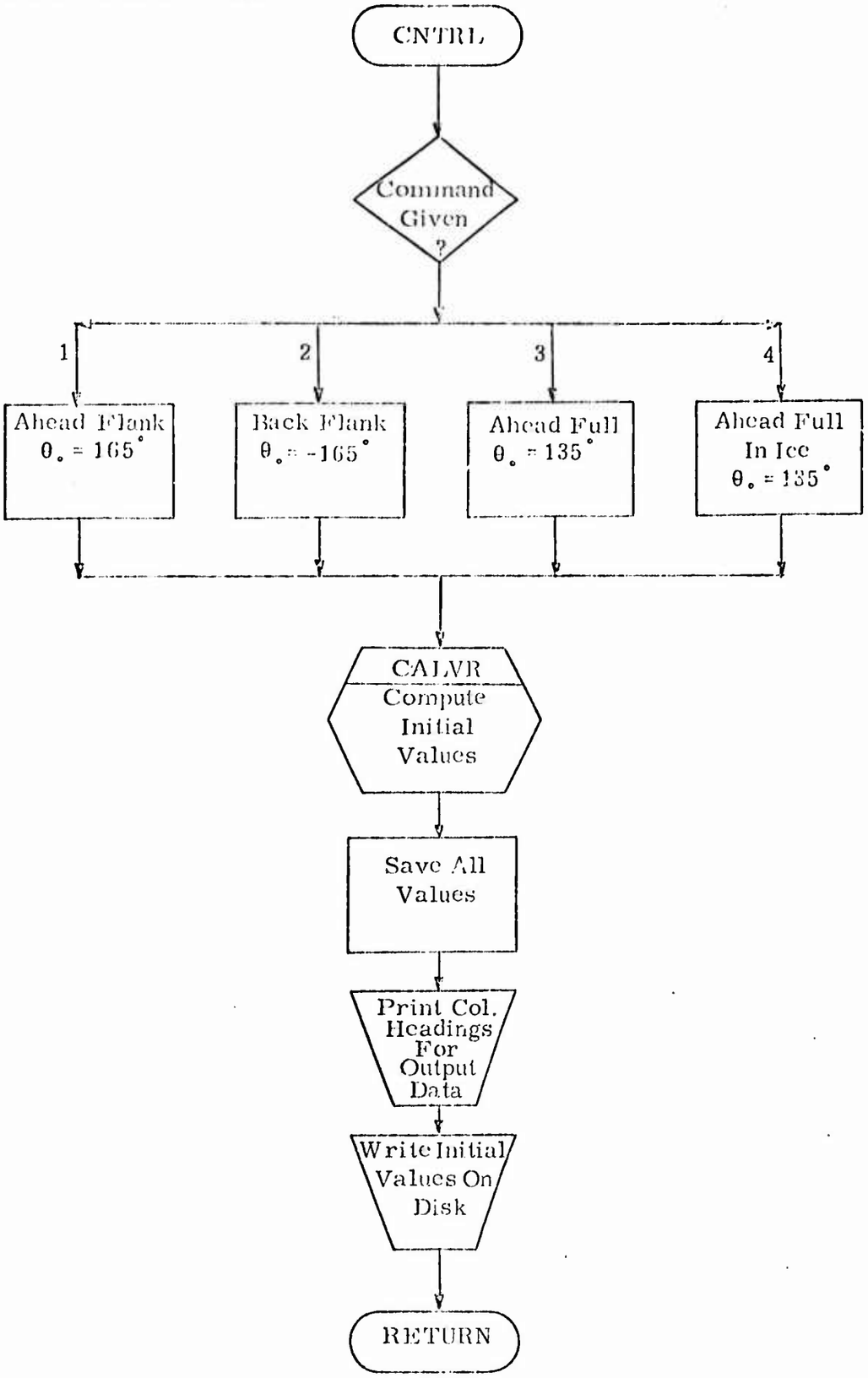
EXEC MAINLINE



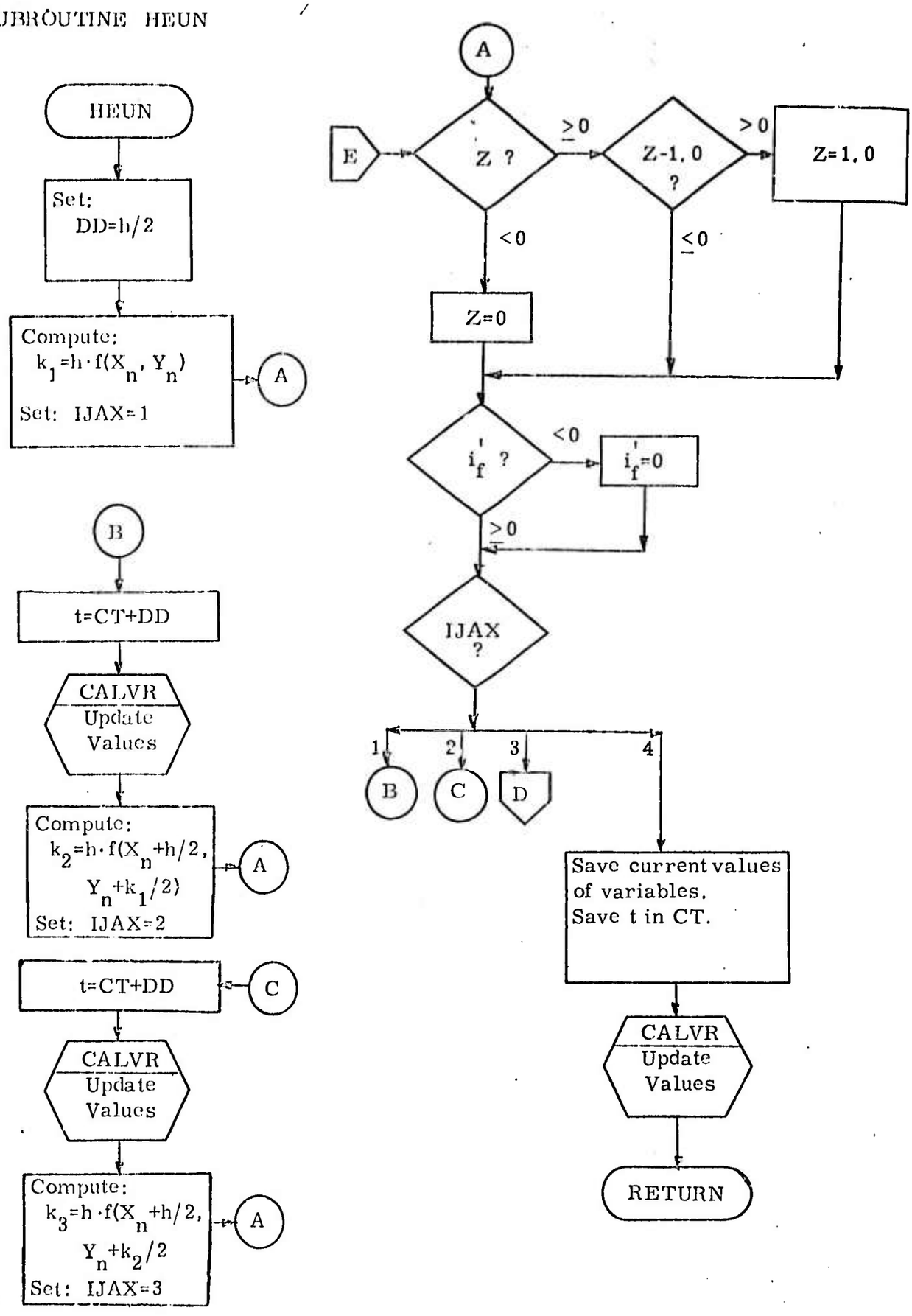
EXEC MAINLINE (continued)

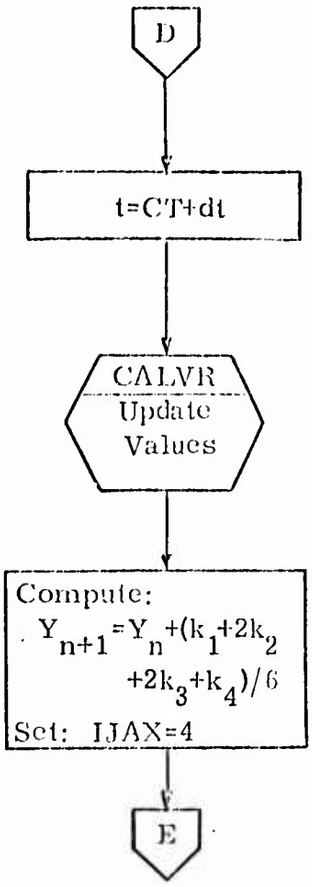


SUBROUTINE CNTRL

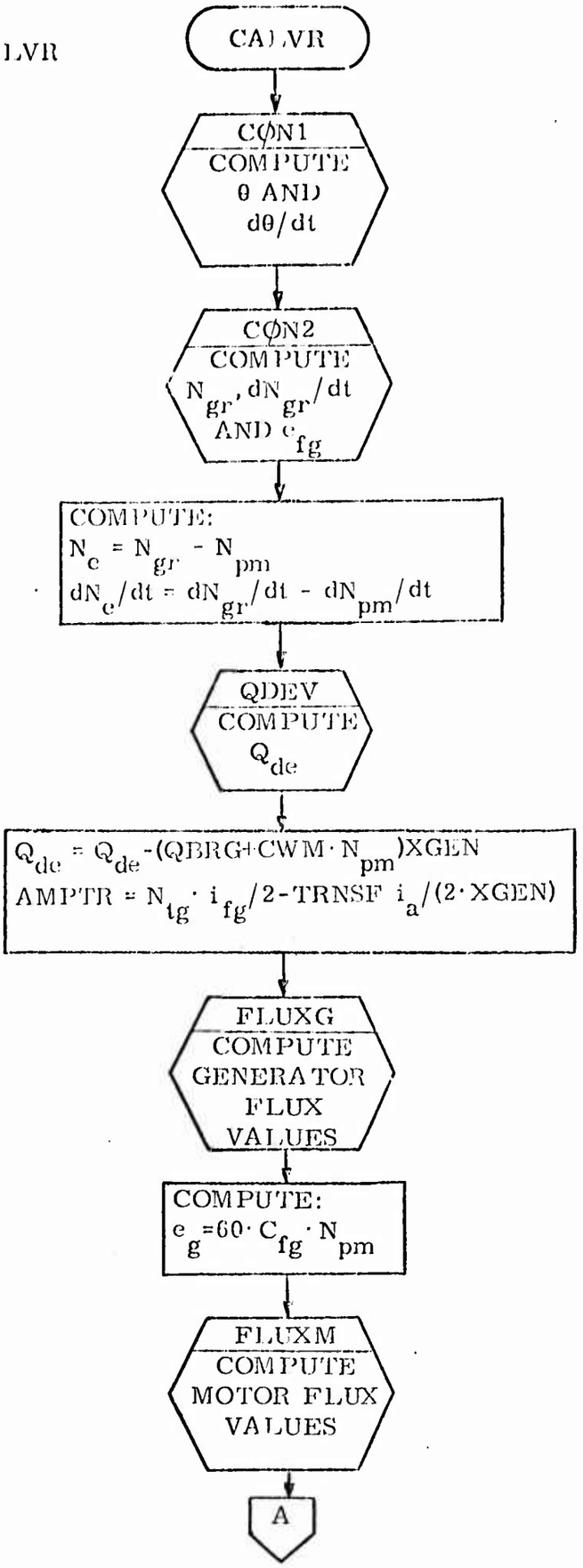


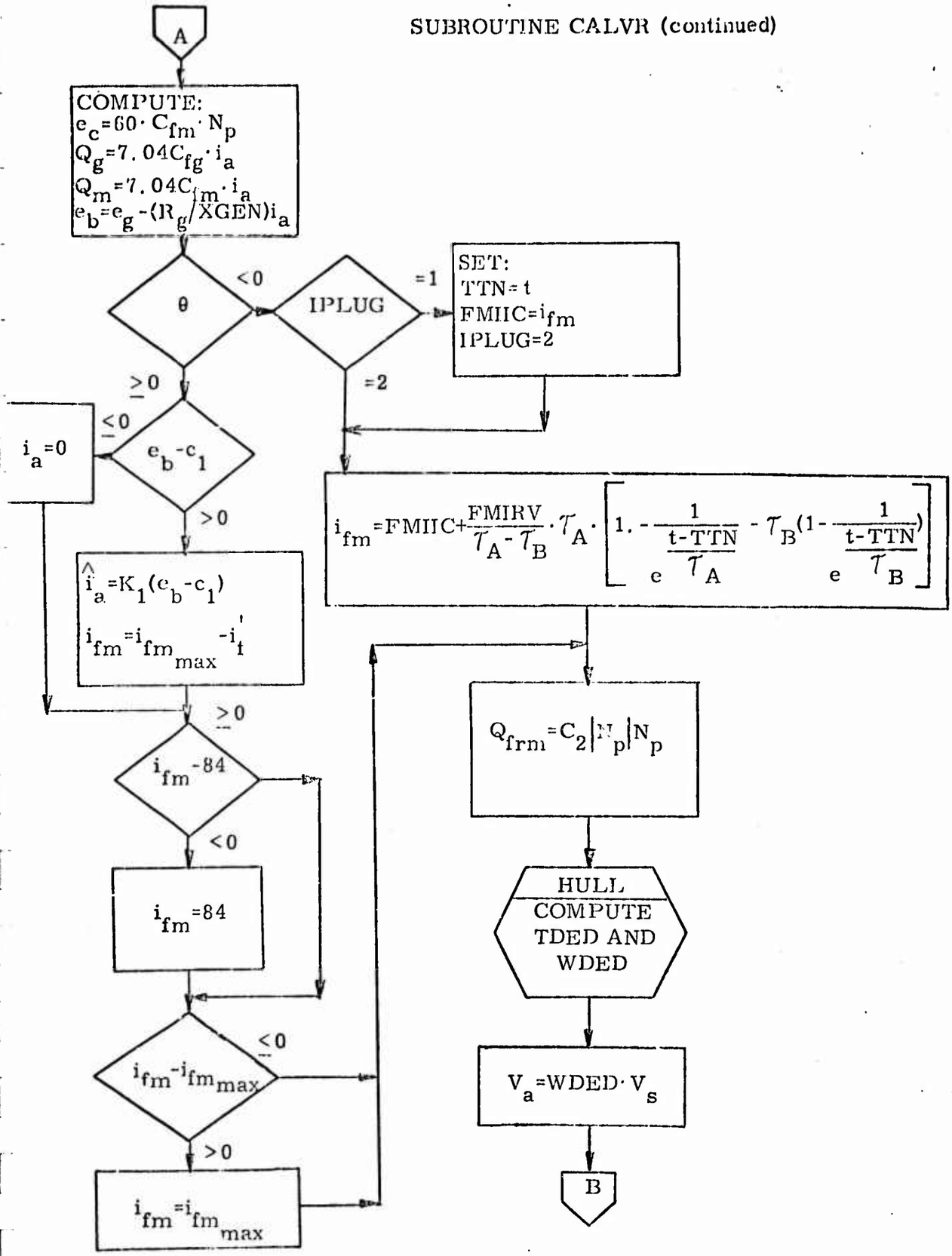
SUBROUTINE HEUN



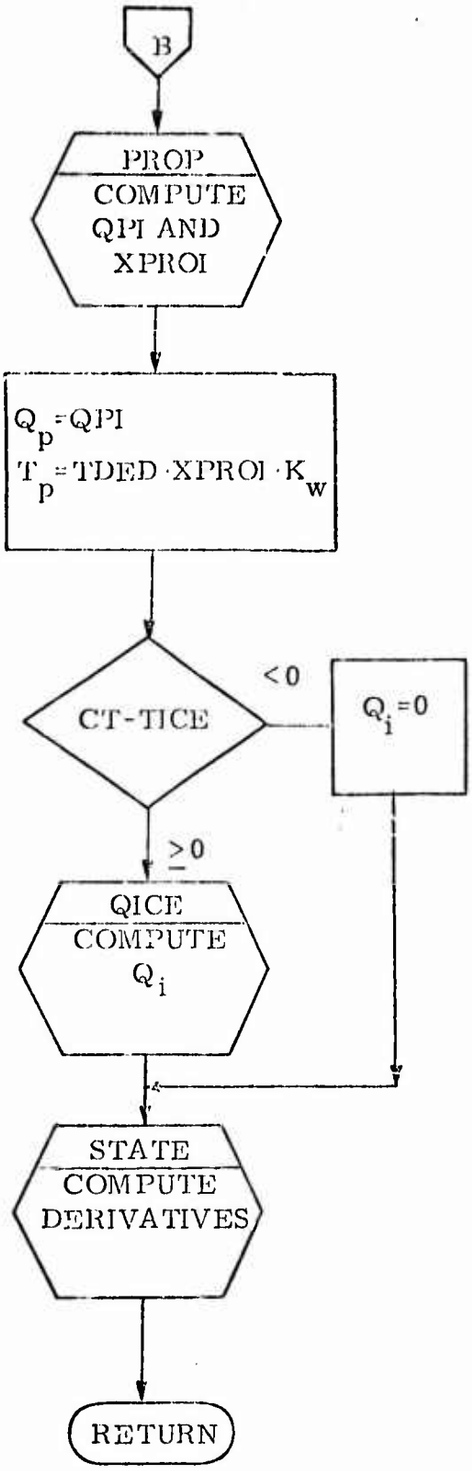

D
t=CT+dt
CALVR
Update
Values
Compute:
$Y_{n+1}=Y_n+(k_1+2k_2+2k_3+k_4)/6$
Set: IJAX=4
E

SUBROUTINE CALVR

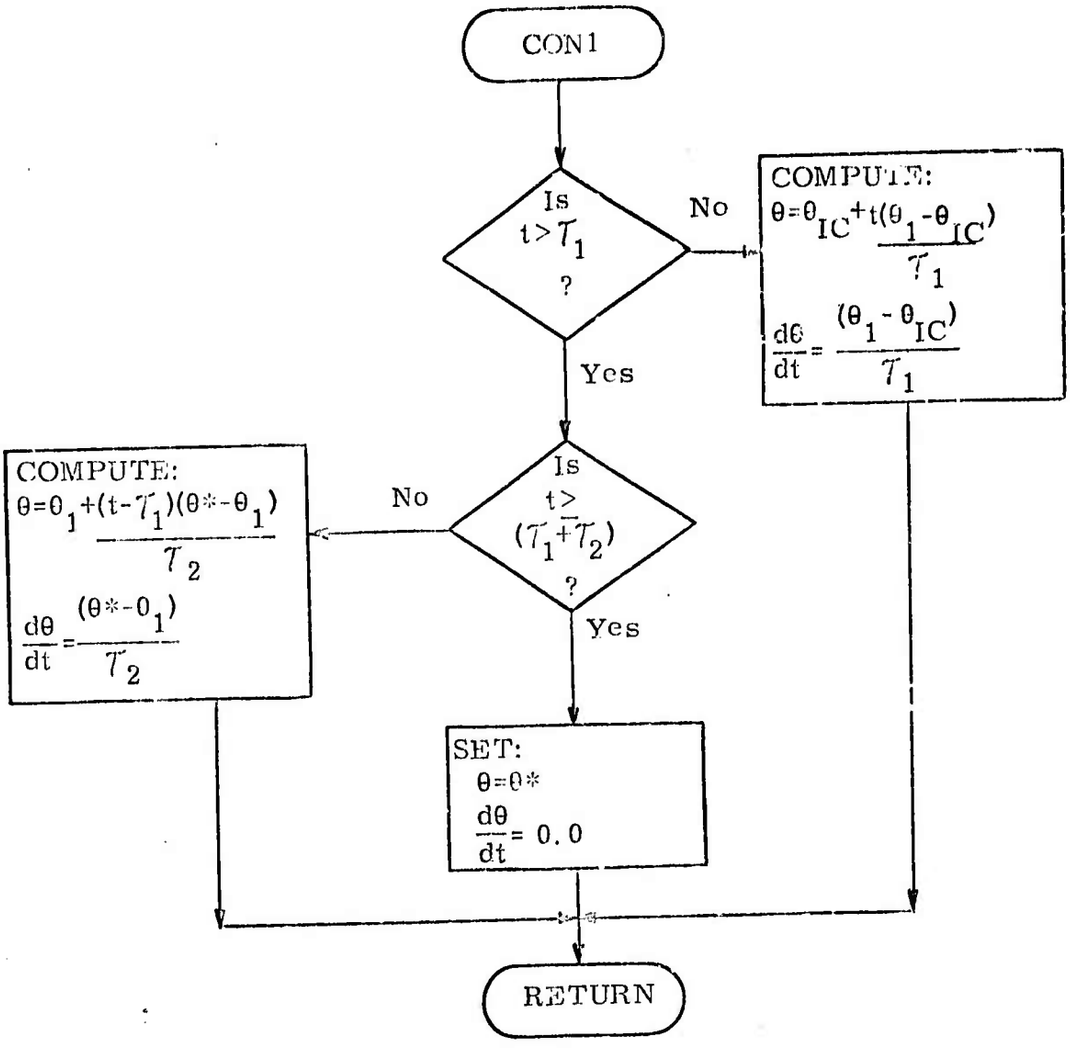



A
COMPUTE:
$e_c=60 \cdot C_{fm} \cdot N_p$
$Q_g=7.04C_{fg} \cdot i_a$
$Q_m=7.04C_{fm} \cdot i_a$
$e_b=e_g-(R_g/XGEN)i_a$
θ
<0
IPLUG
=1
SET:
TTN=t
FMIIC=i_{fm}
IPLUG=2
≥0
=2
≤0
$i_a=0$
e_b-c_1
>0
$\hat{i}_a=K_1(e_b-c_1)$
$i_{fm}=i_{fm_{max}}-i'_t$
$i_{fm}=FMIIC+\frac{FMIRV}{\tau_A-\tau_B} \cdot \tau_A \cdot \left[1.-\frac{1}{e^{\frac{t-TTN}{\tau_A}}}-\tau_B\left(1-\frac{1}{e^{\frac{t-TTN}{\tau_B}}}\right)\right.$
≥0
$i_{fm}-84$
<0
$i_{fm}=84$
$i_{fm}-i_{fm_{max}}$
≤0
>0
$i_{fm}=i_{fm_{max}}$
$Q_{frm}=C_2|N_p|N_p$
HULL
COMPUTE TDED AND WDED
$V_a=WDED \cdot V_s$
B

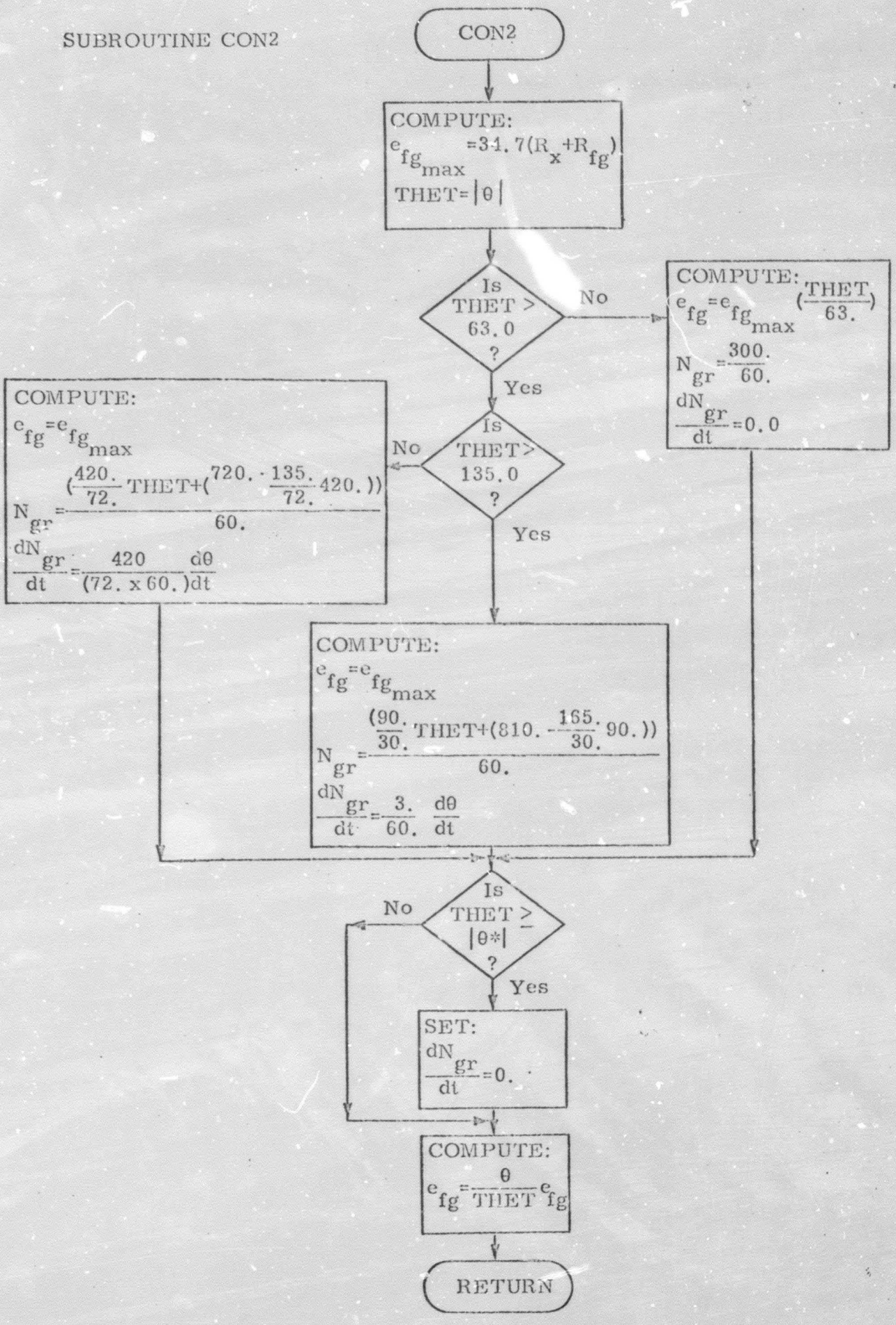
SUBROUTINE CALVR (continued)



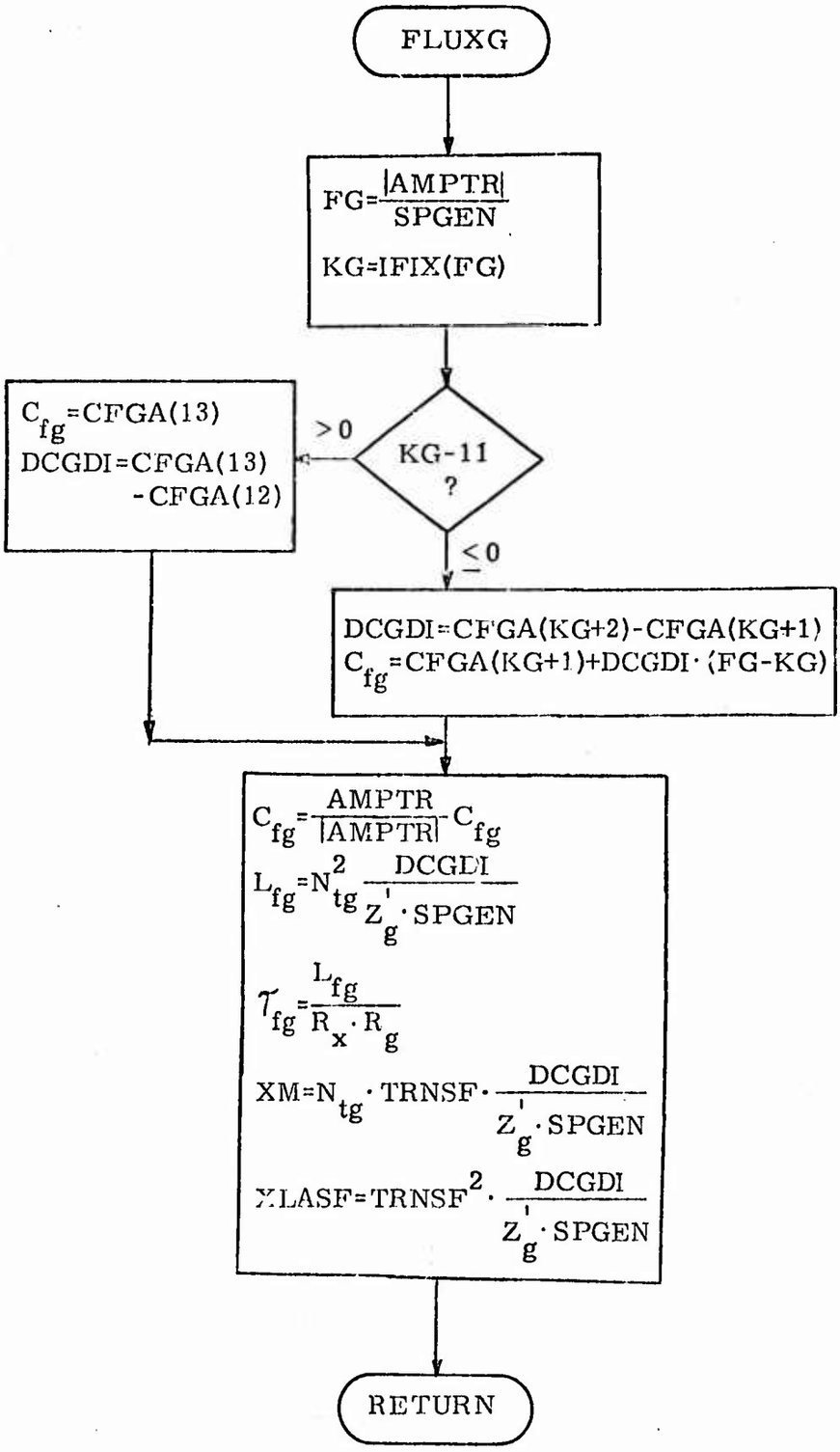
SUBROUTINE CON1



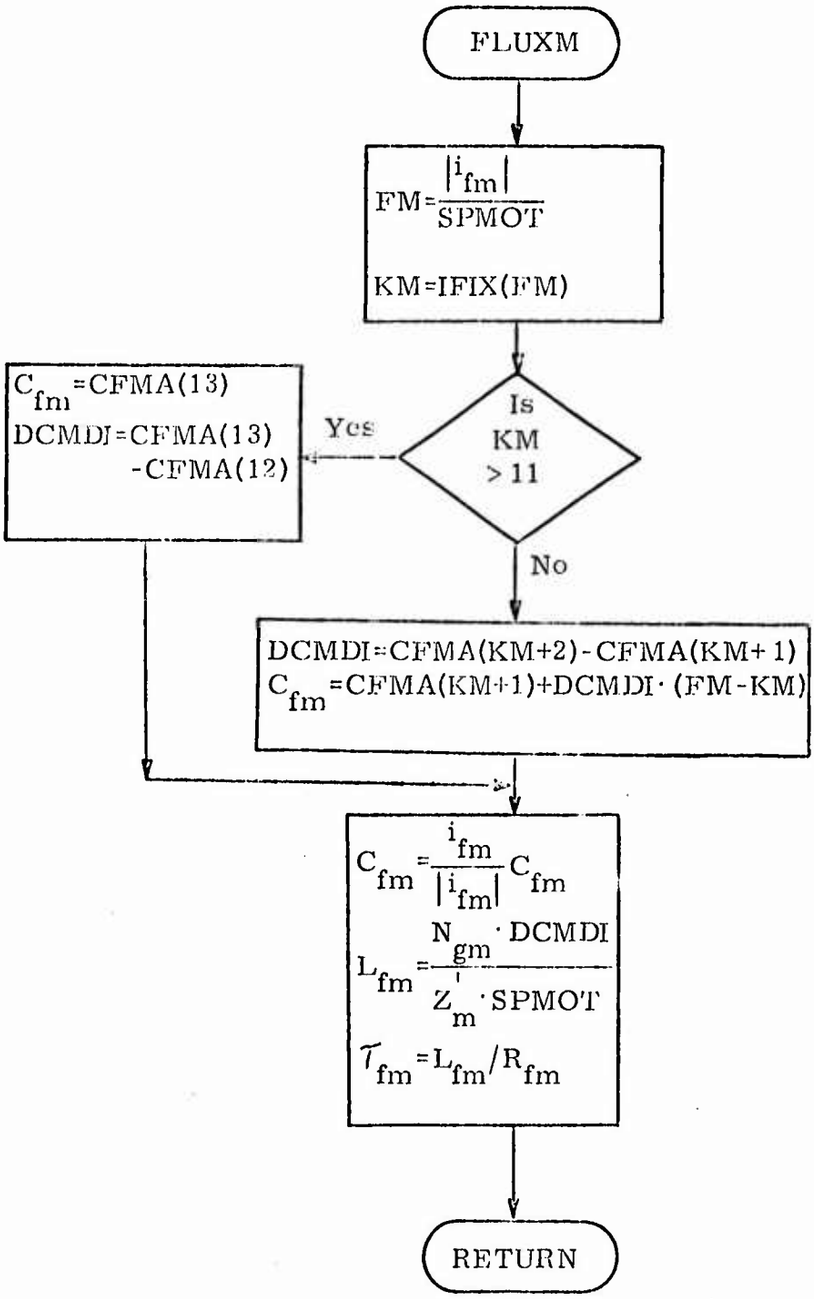
SUBROUTINE CON2



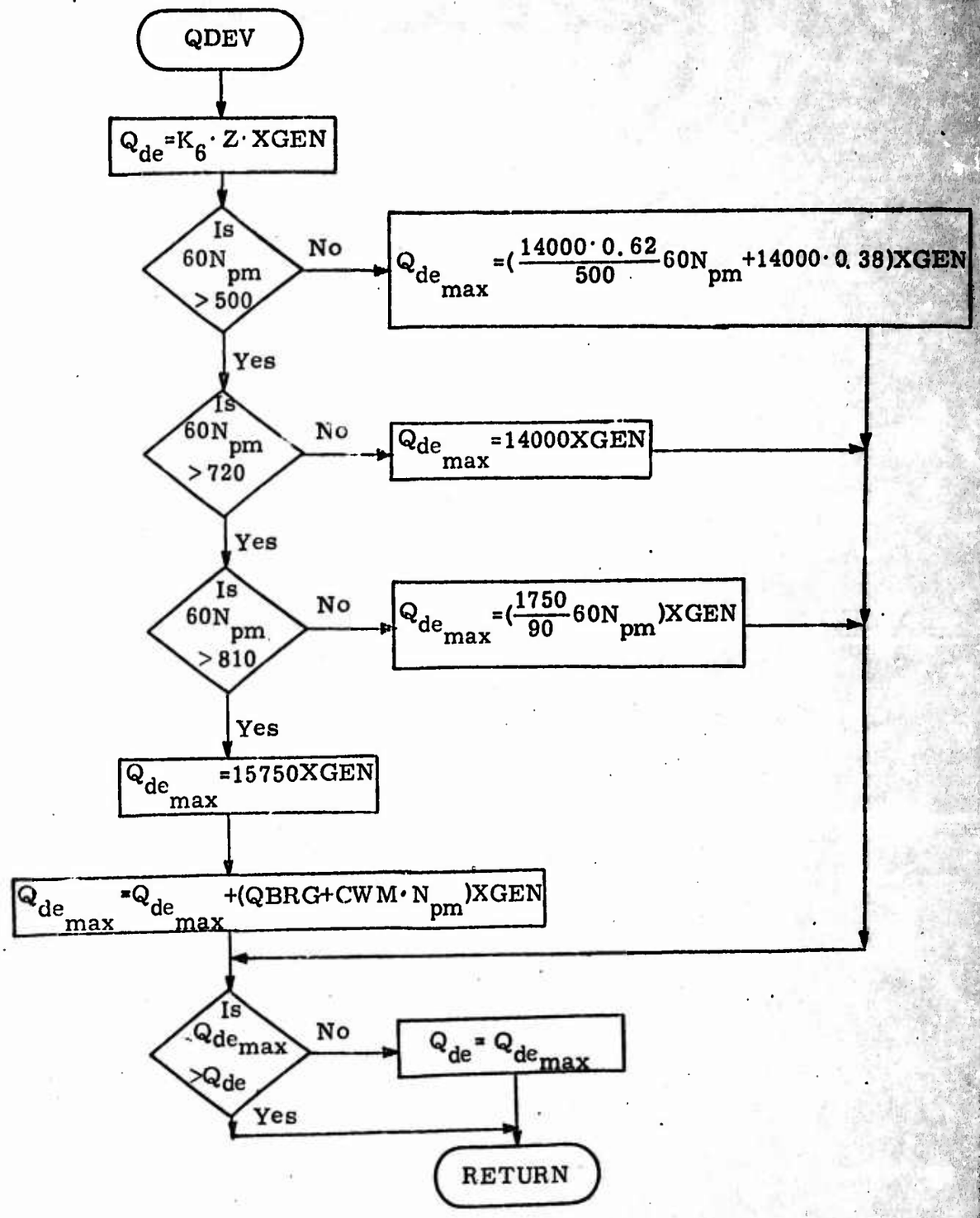
SUBROUTINE FLUXG



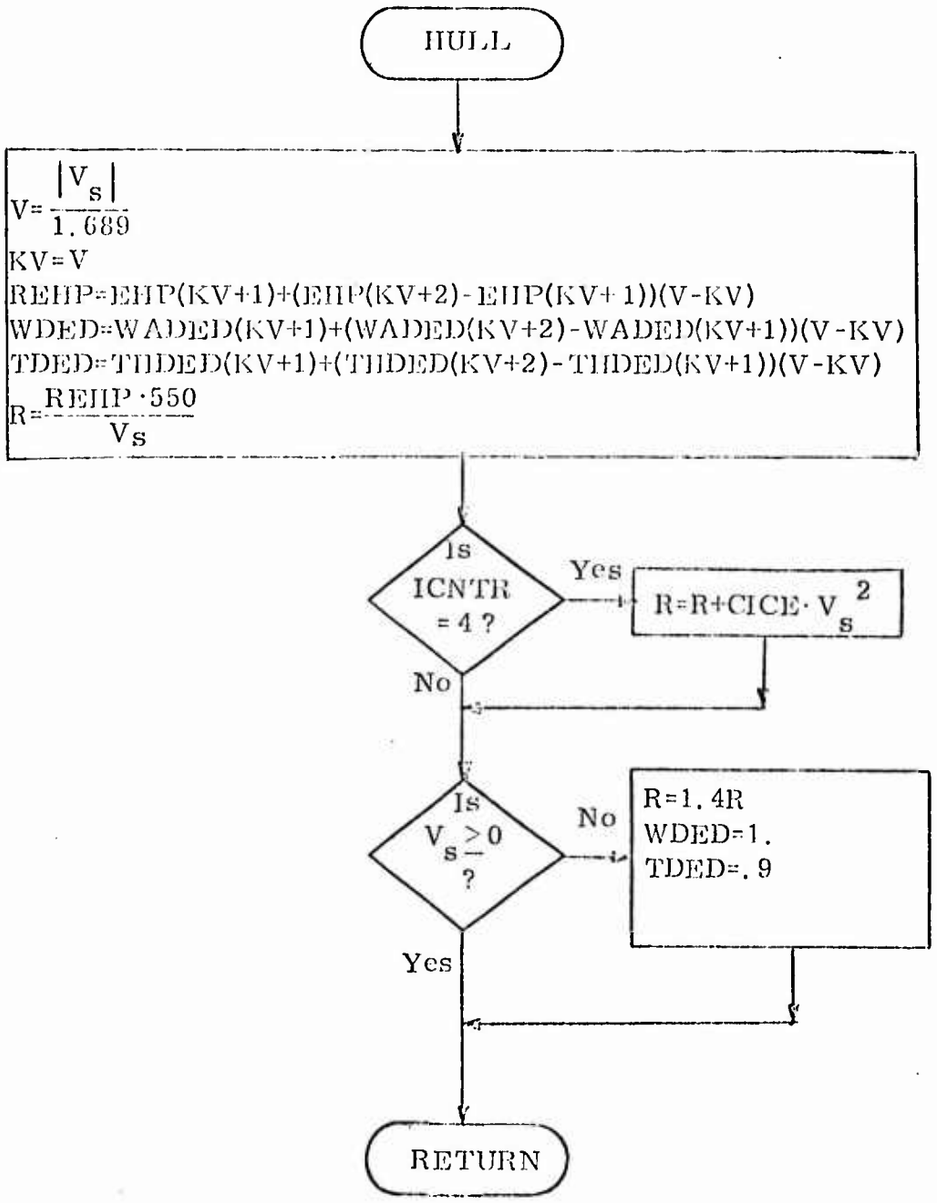
SUBROUTINE FLUXM



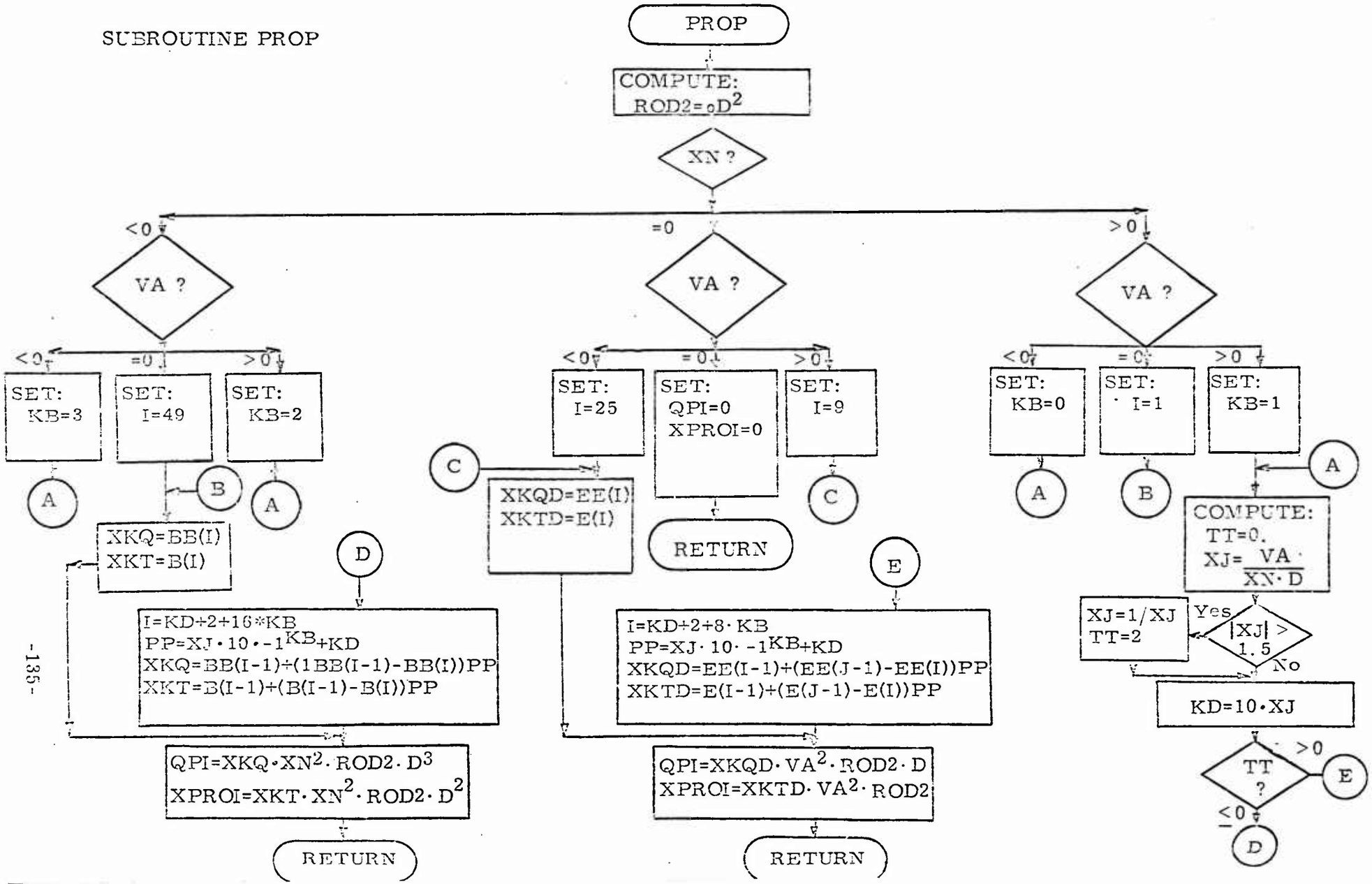
SUBROUTINE QDEV



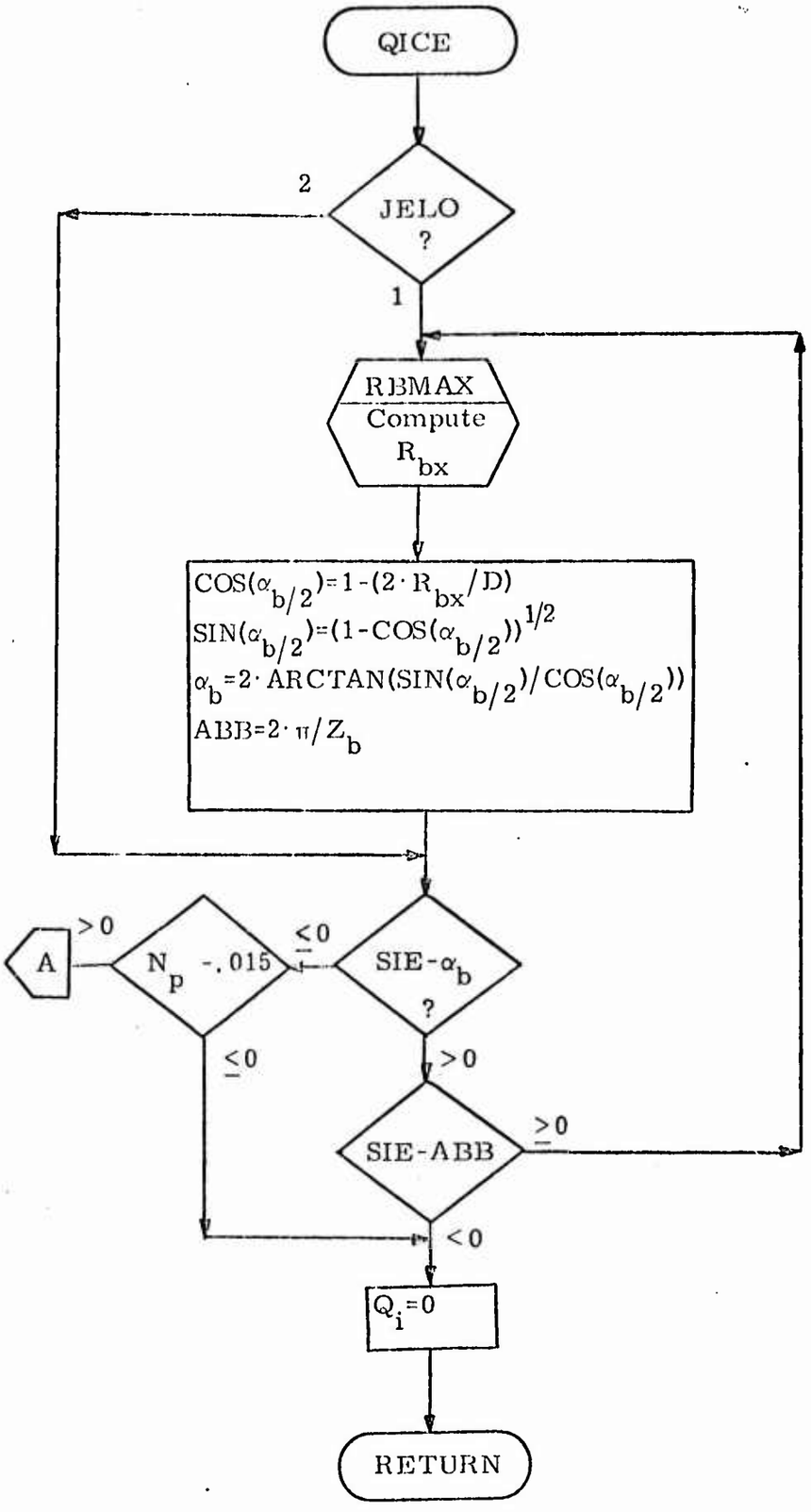
SUBROUTINE HULL



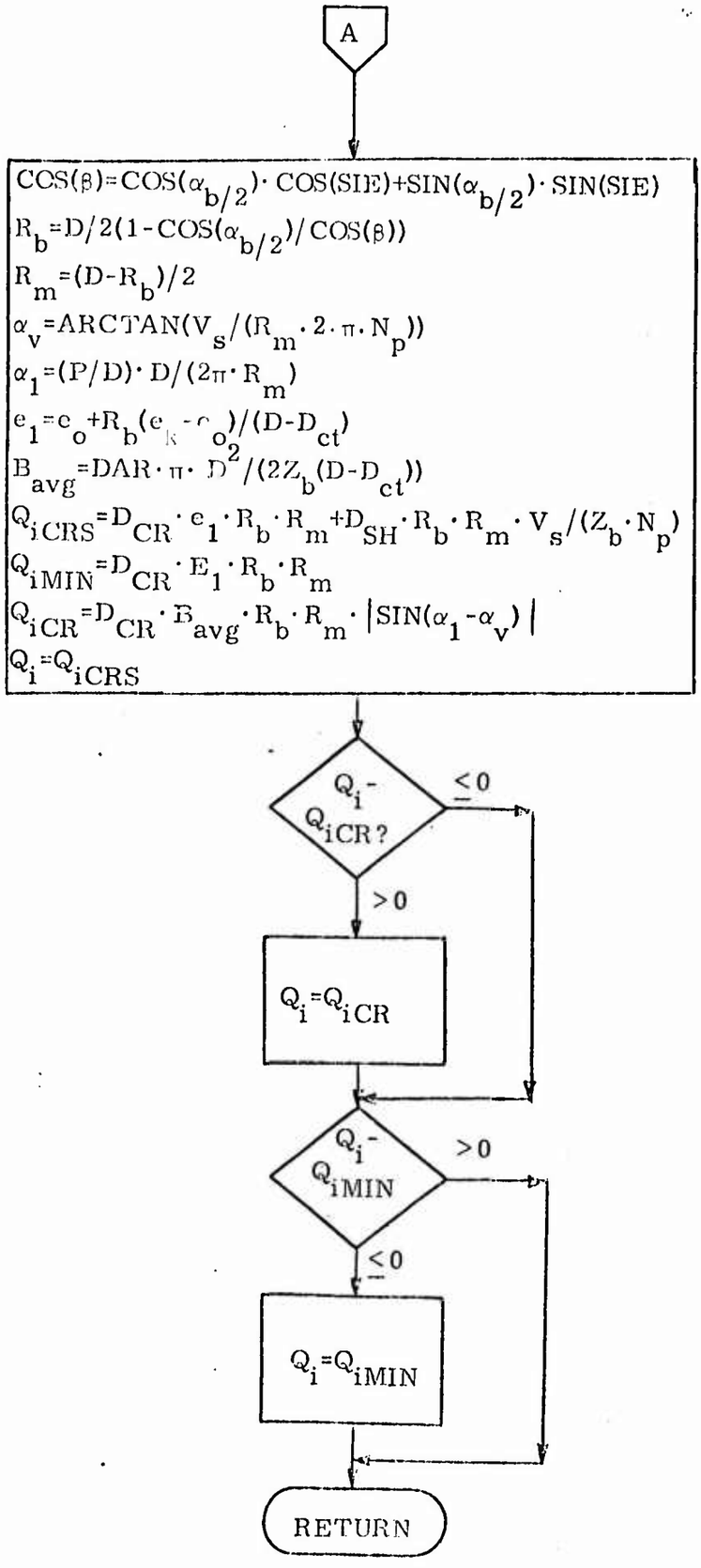
SUBROUTINE PROP



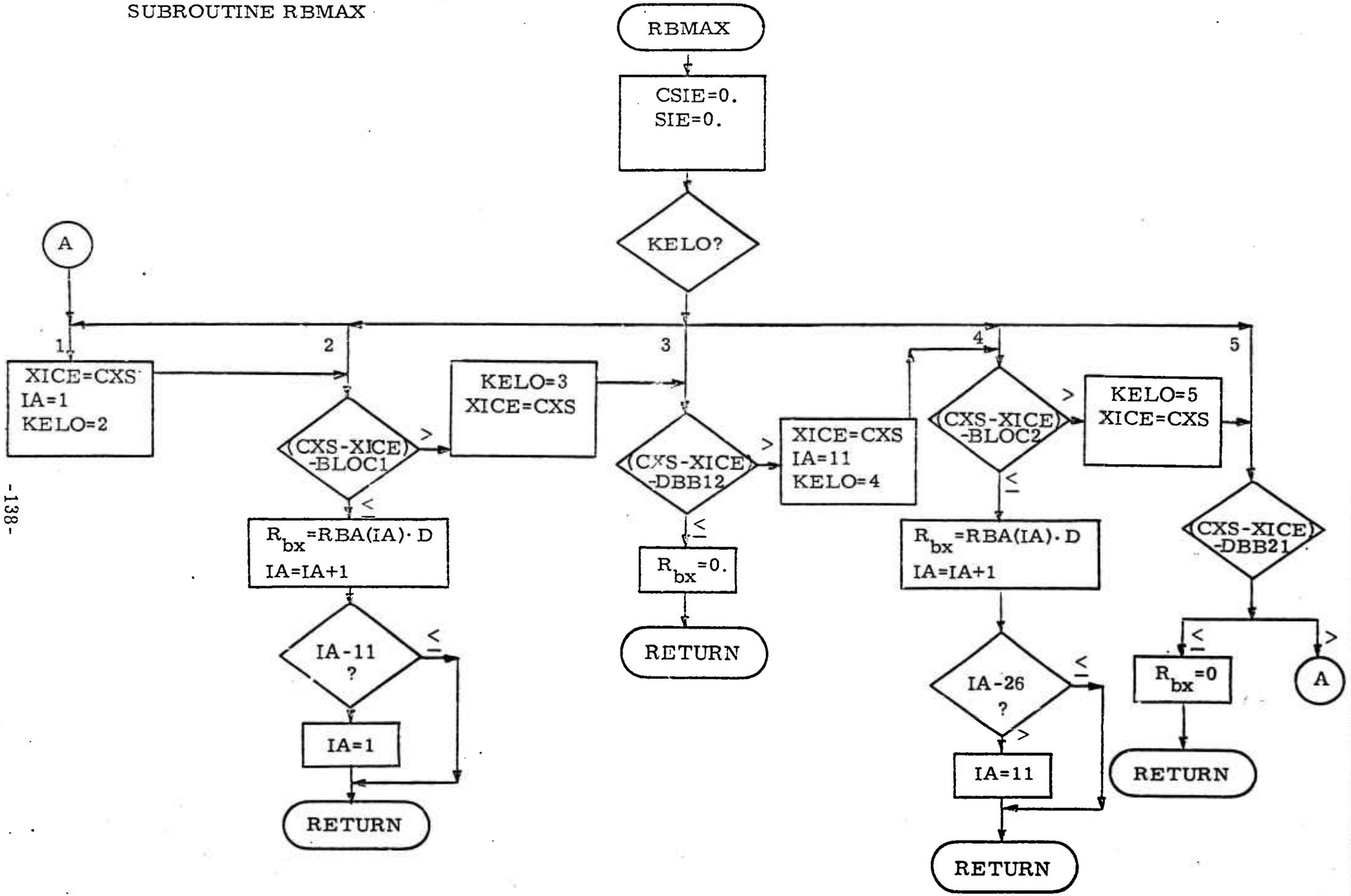
SUBROUTINE QICE



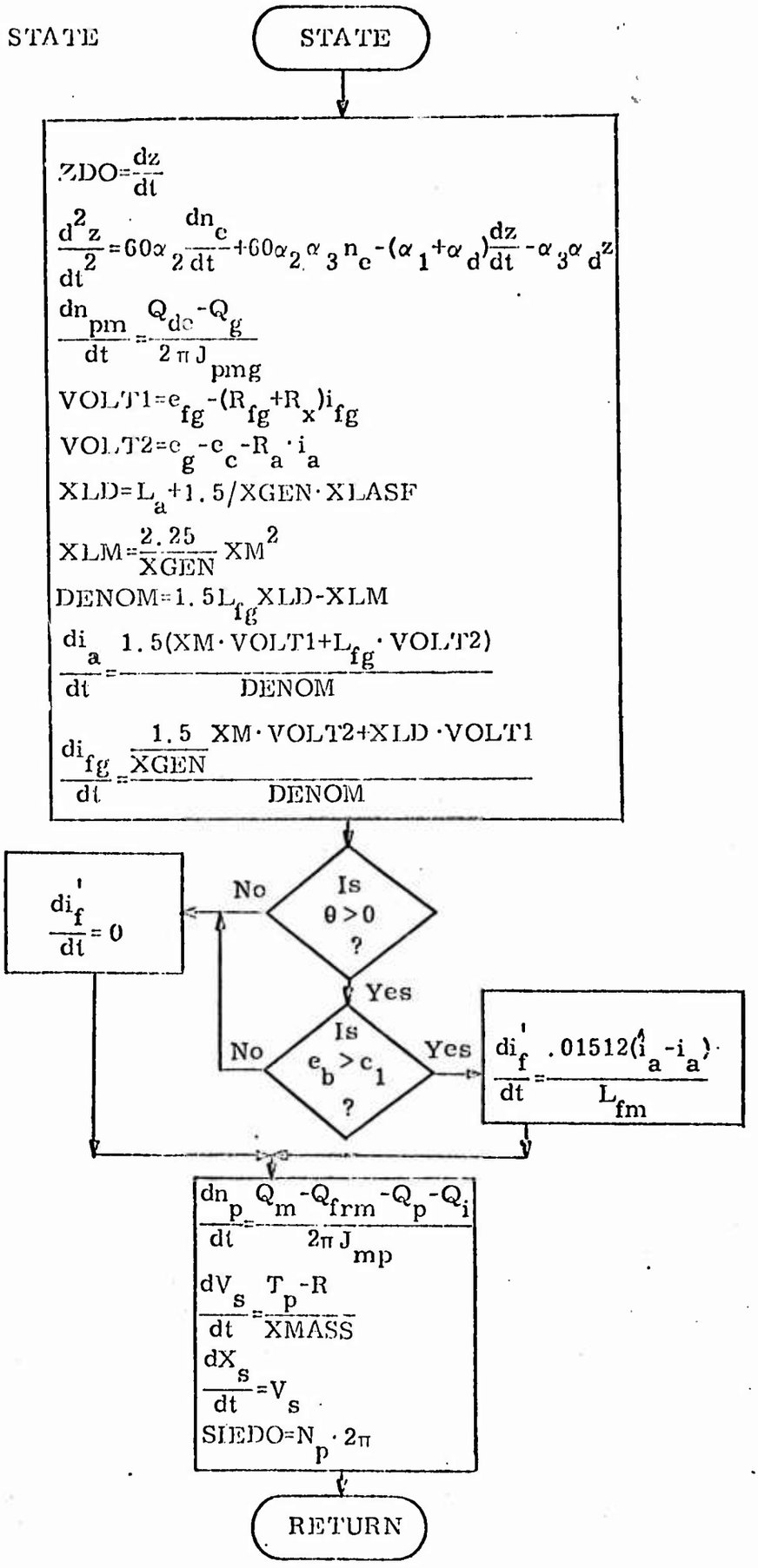
SUBROUTINE QICE (continued)



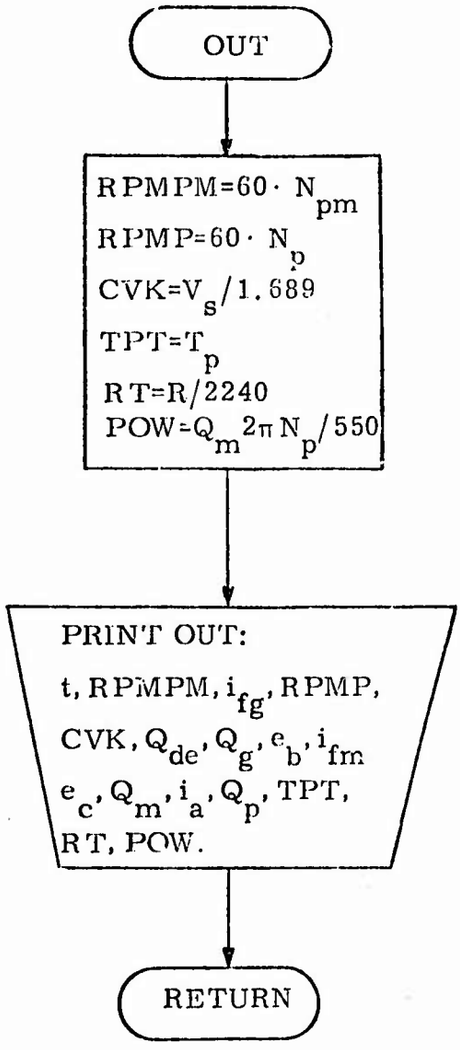
SUBROUTINE RBMAX



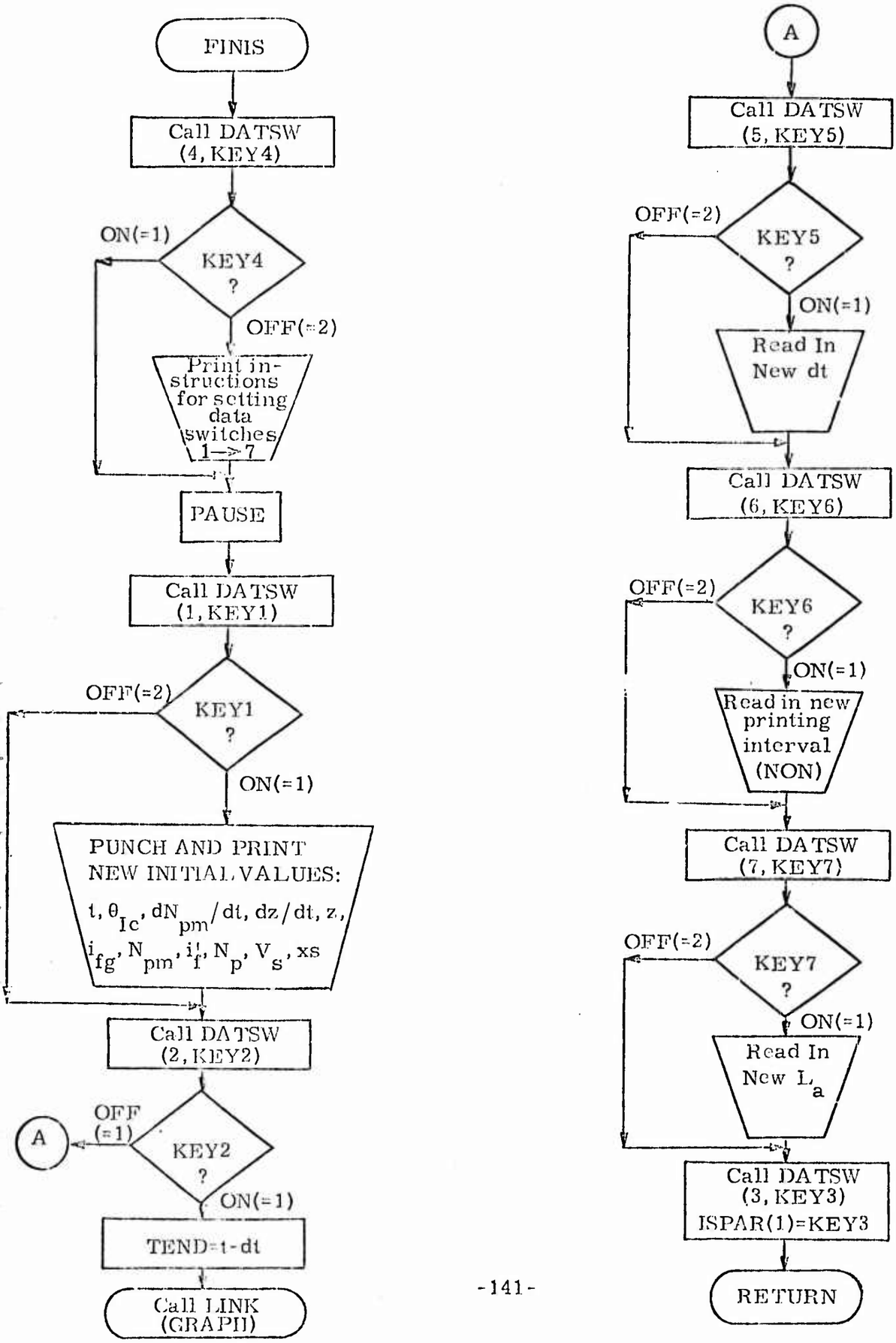
SUBROUTINE STATE



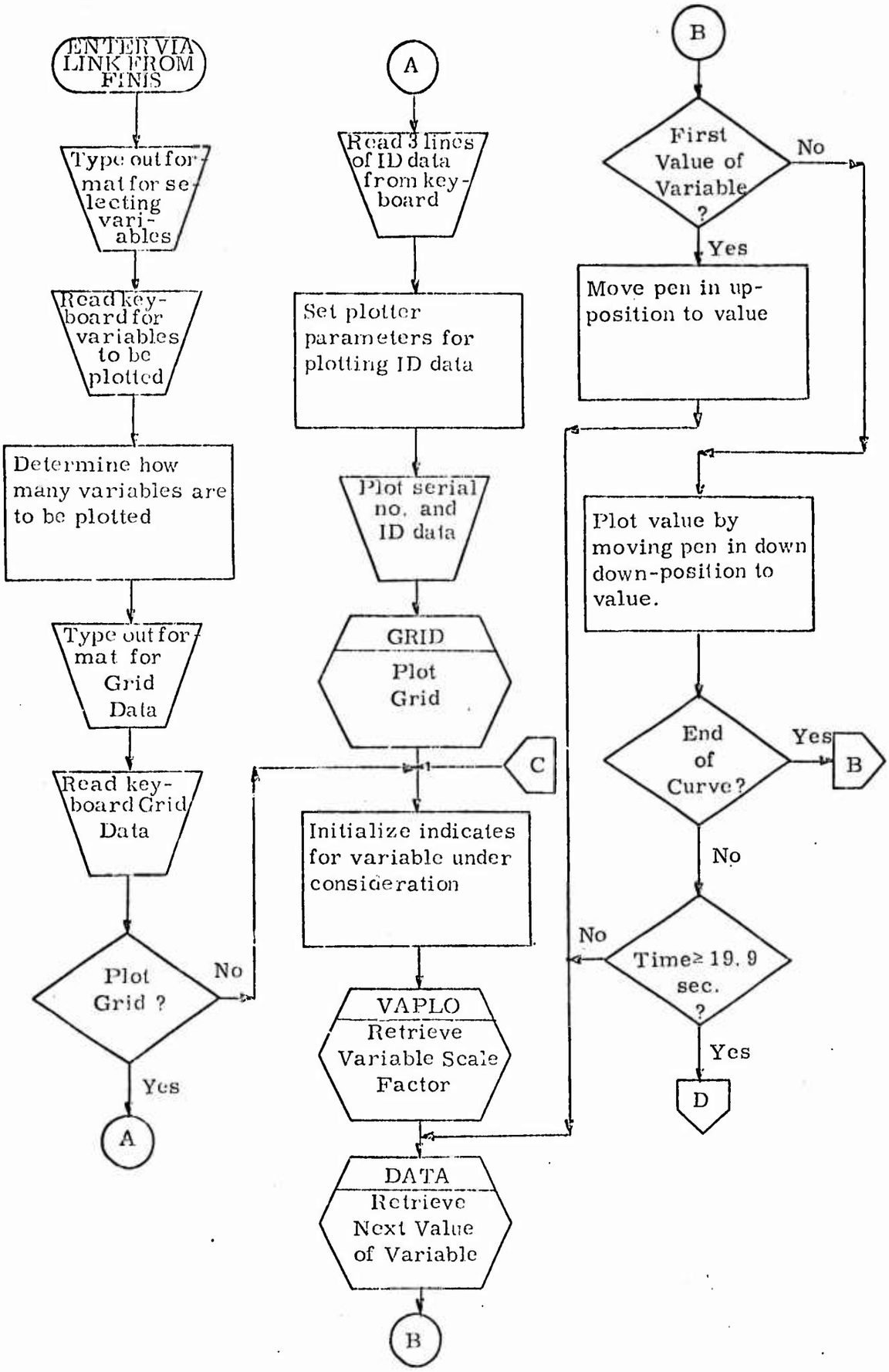
SUBROUTINE OUT



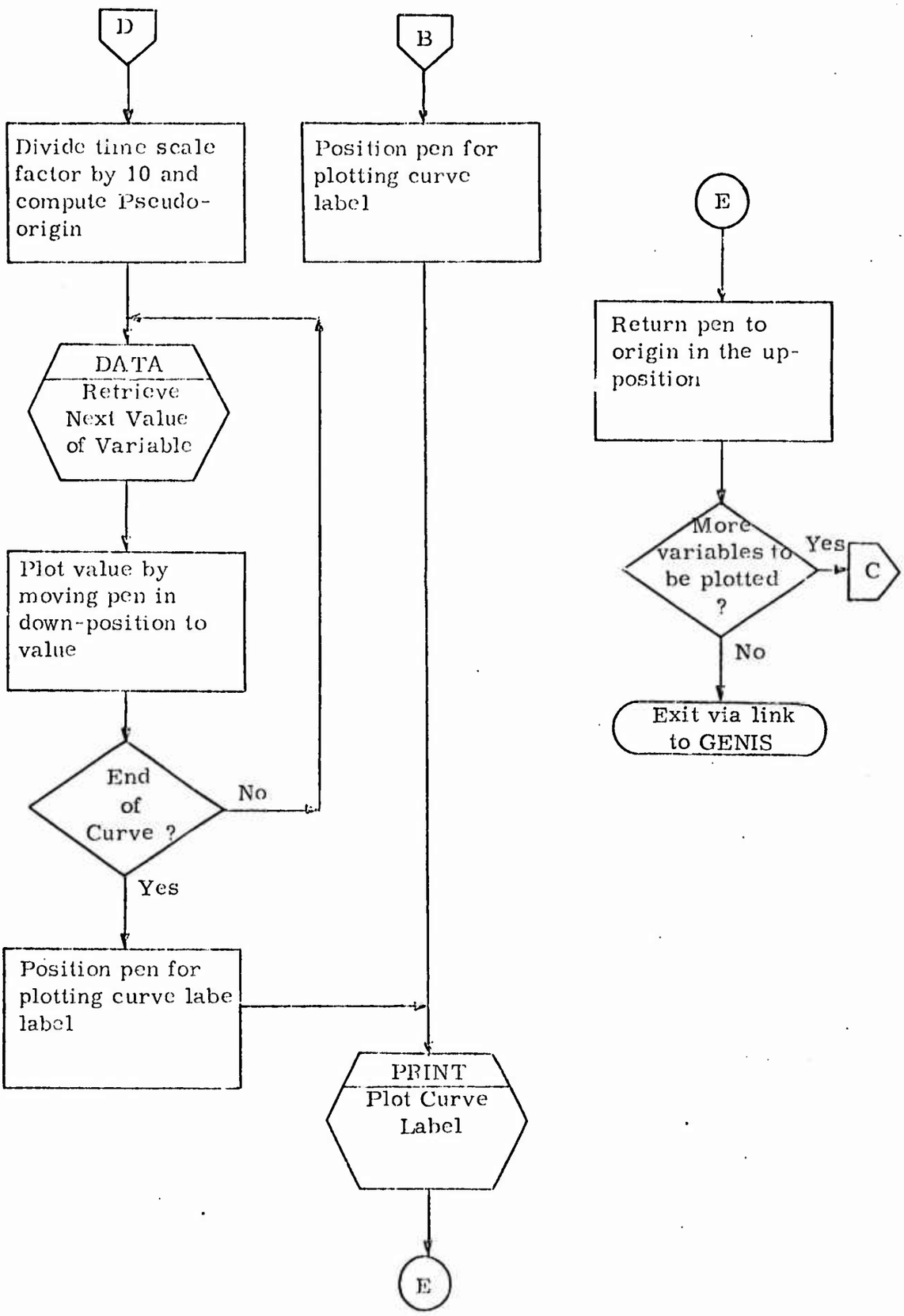
SUBROUTINE FINIS



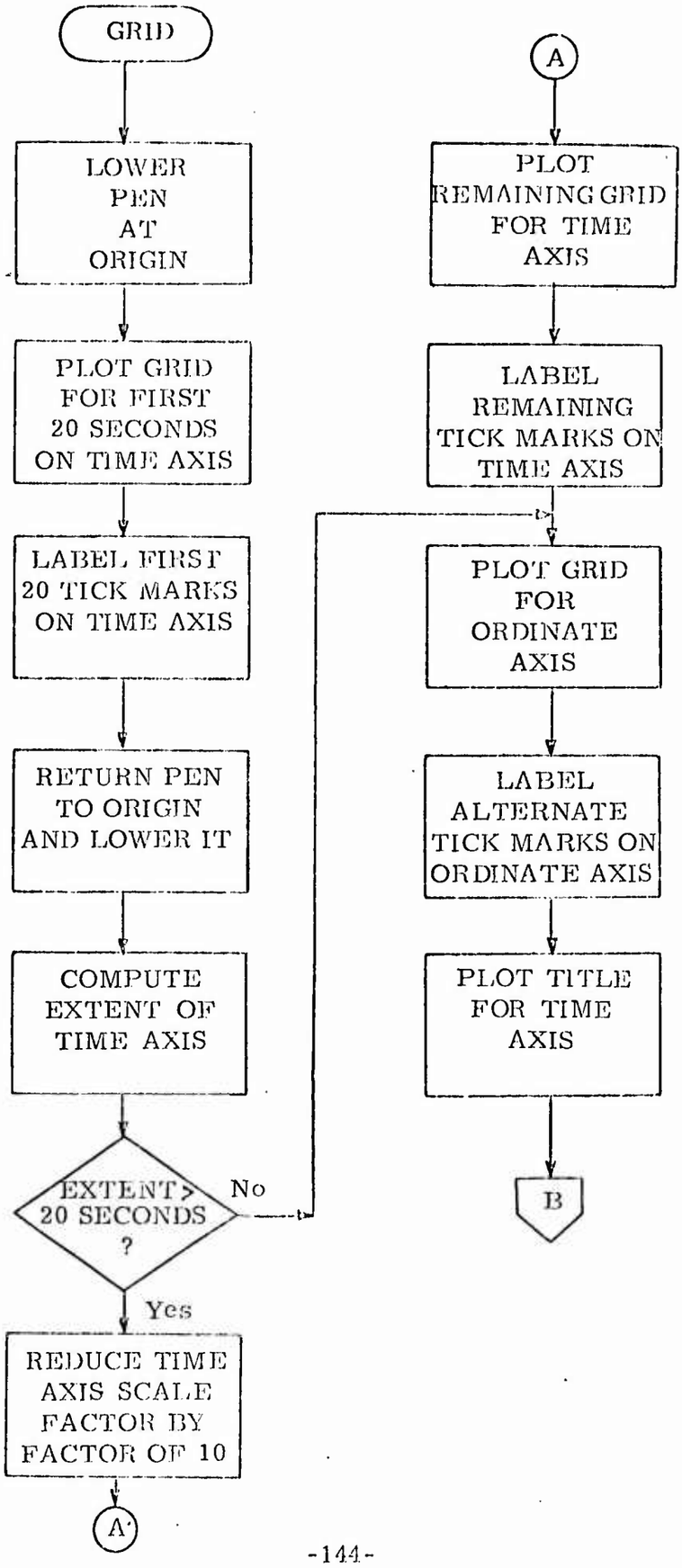
GRAPH MAINLINE



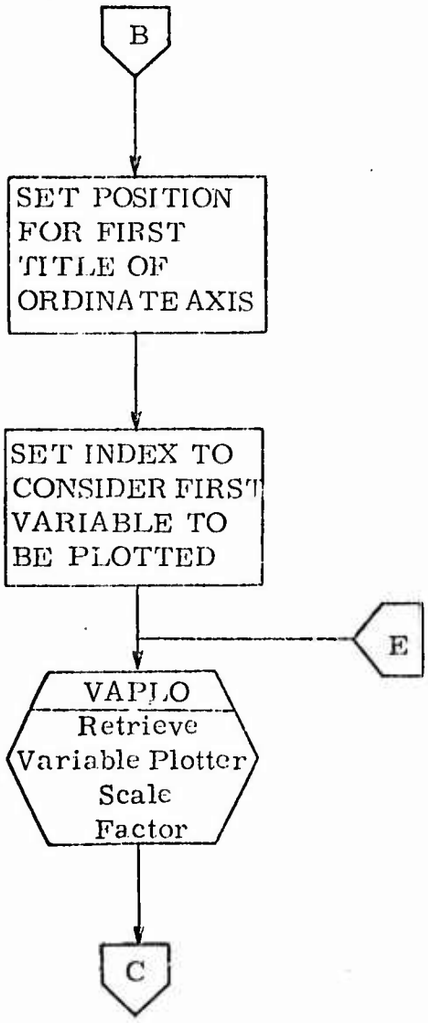
GRAPH MAINLINE (continued)



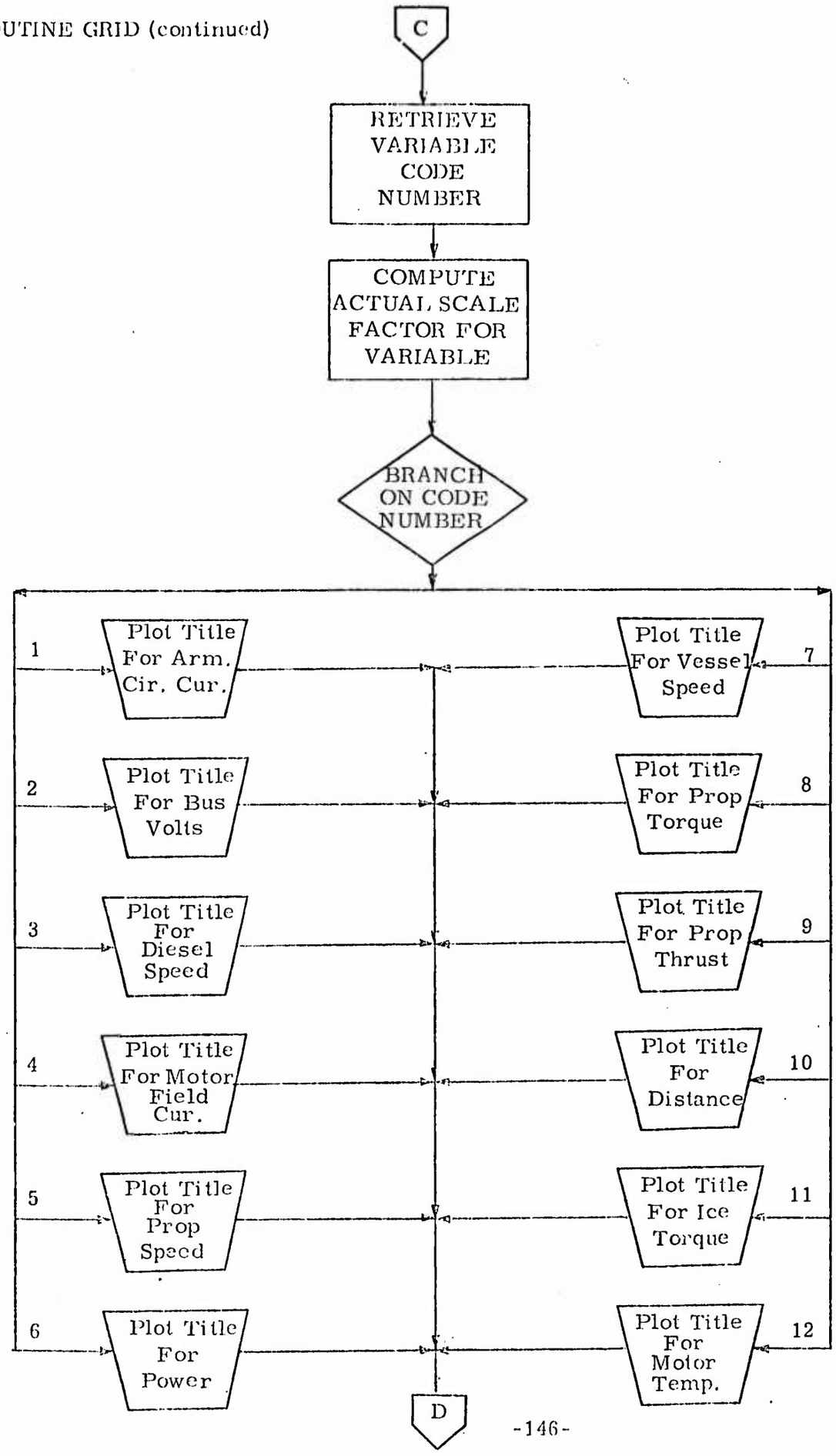
SUBROUTINE GRID

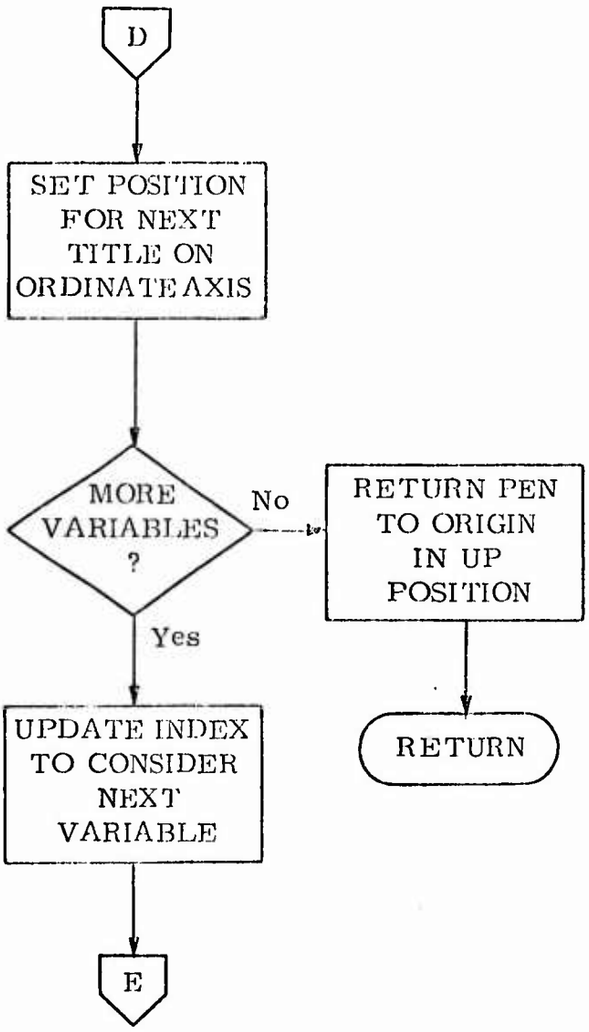


SUBROUTINE GRID (continued)

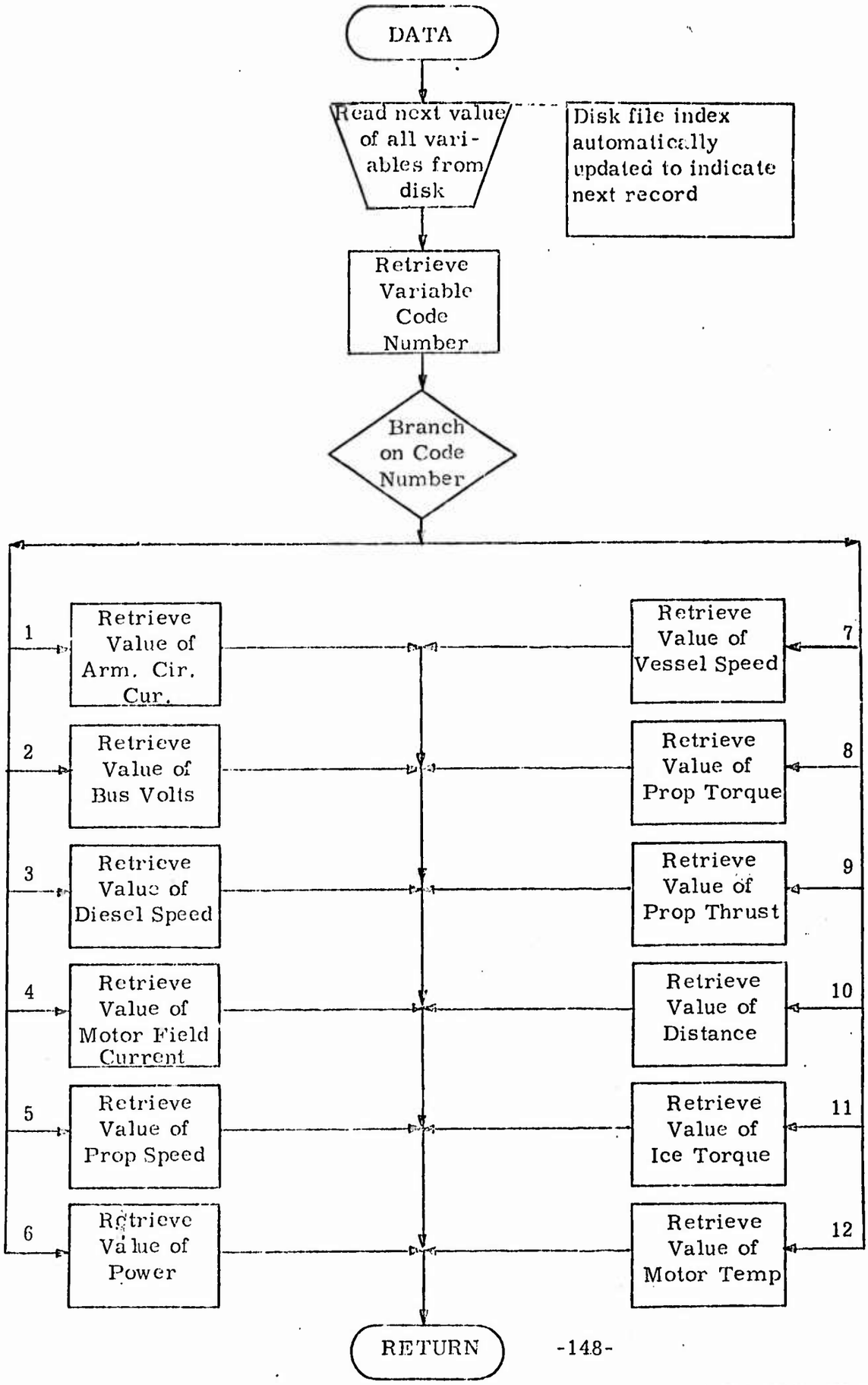


SUBROUTINE GRID (continued)

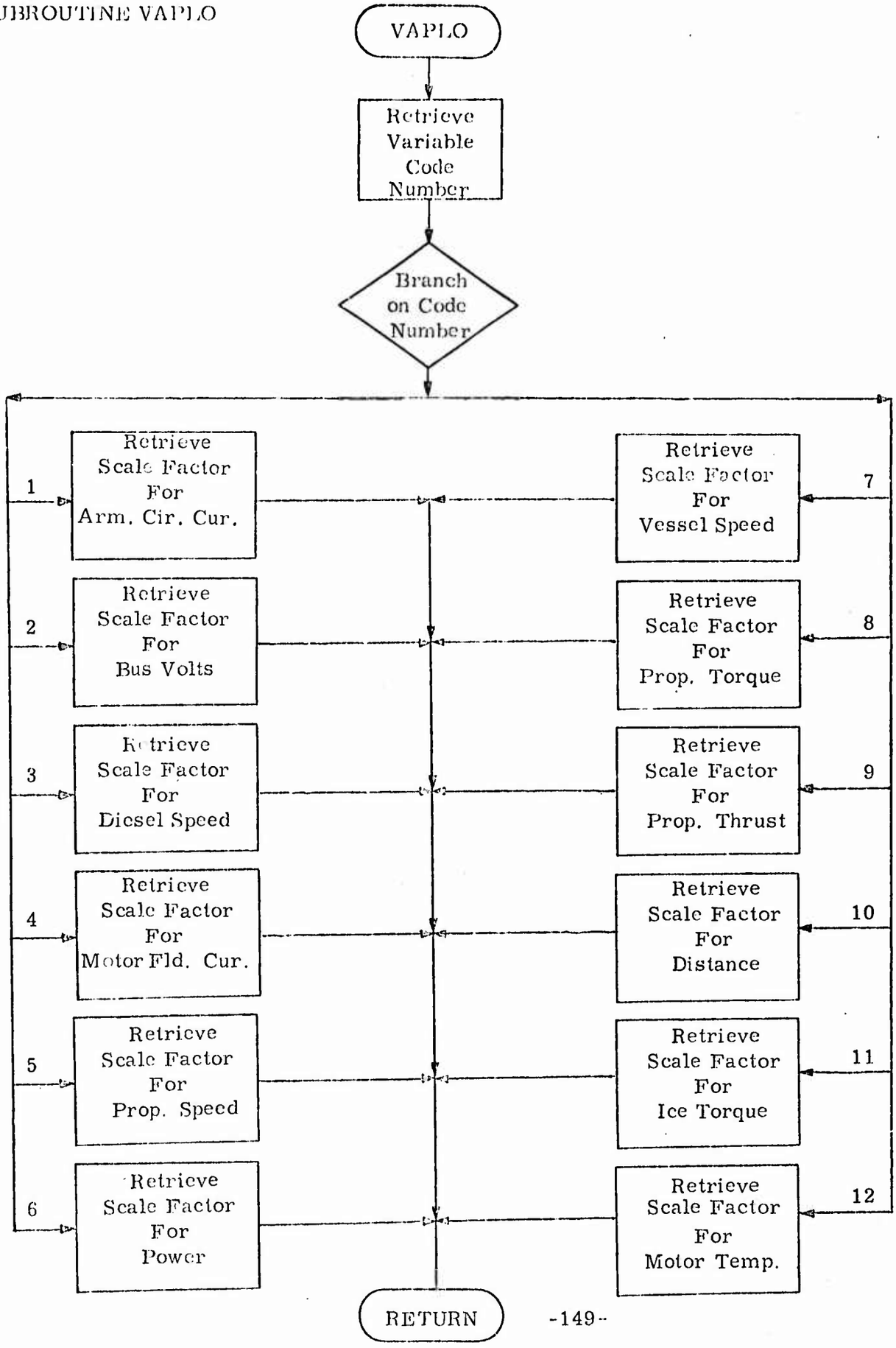



D
SET POSITION FOR NEXT TITLE ON ORDINATE AXIS
MORE VARIABLES ?
No
RETURN PEN TO ORIGIN IN UP POSITION
Yes
RETURN
UPDATE INDEX TO CONSIDER NEXT VARIABLE
E

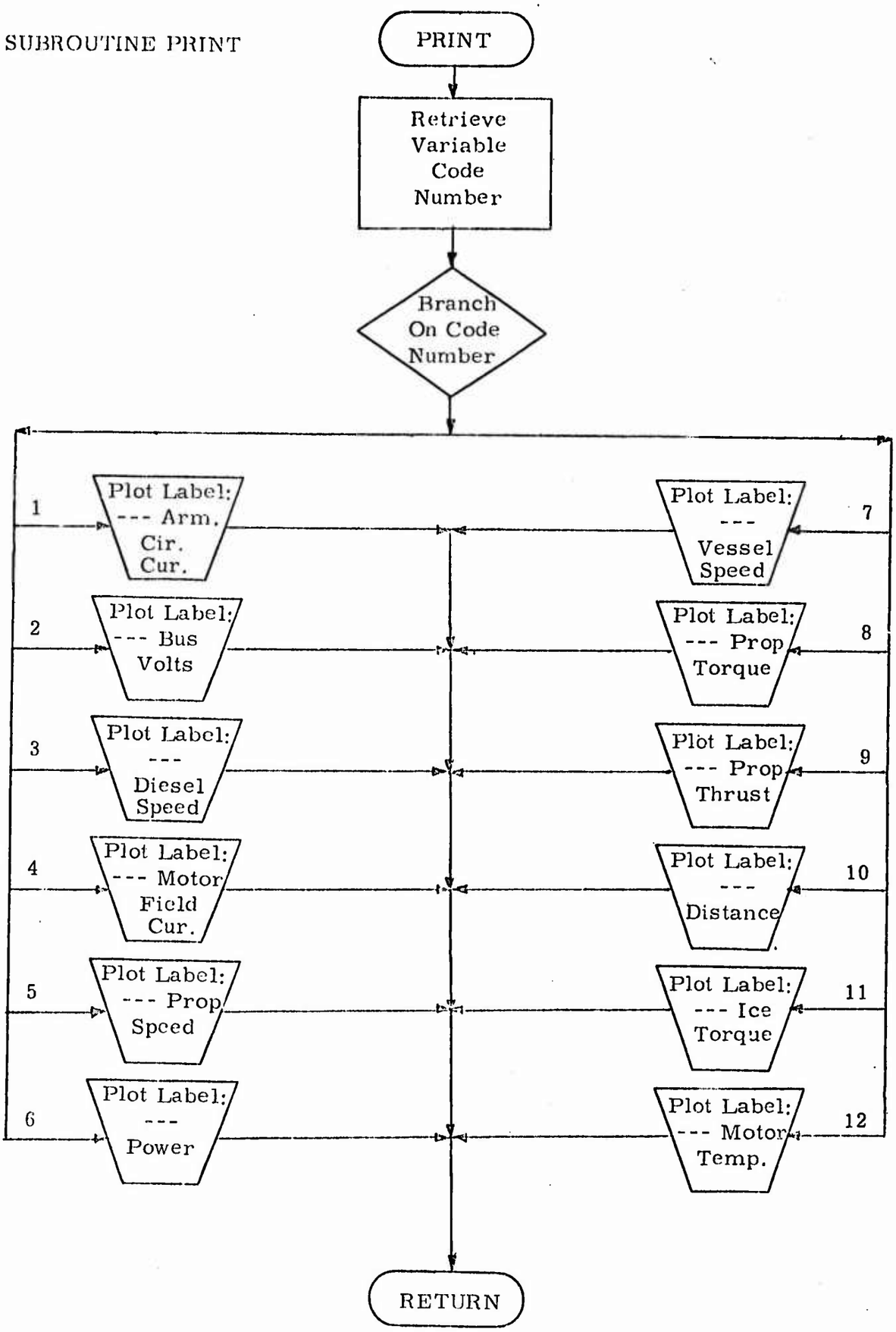
SUBROUTINE DATA



SUBROUTINE VAPLO



SUBROUTINE PRINT



MAINLINE PROGRAMS AND SUBROUTINE

DOCUMENTATION

1. Program Name: Mainline Routine GENIS

2. Program Description: The GENIS mainline routine, after calling subroutine REDAT to read in all the input data and subroutine POLMI to compute the remaining inertias, prints out the control data and initial conditions and depending on the setting of data switches 10 through 15 prints out the ice data, propeller data, hull data, electrical data, governor and bridge controller data, and inertias respectively. The EXEC mainline routine is then loaded and executed (by use of a CALL LINK (EXEC) statement) to begin the simulation.

3. Program Logic: Disk data file 2 contains the serial number of the last simulation. If data switch 9 is on, the serial number will be reset to zero (switch 9 must be on for the first run), otherwise the serial number will be updated by one (1).

4. Listing of Parameters:

Values taken from COMMON - All values read in by subroutine REDAT plus XJPE, the entrained water inertia which was computed in POLMI.
Values taken from the disk - SER, the previous serial number if data switch 9 is off.
Results stored in COMMON - SER, the updated serial number corresponding to this run.

5. Program Length:

a. Source program - 245 statements (including comments).
b. Object program - 1828 machine words.

6. Programs Called By This Program: Subroutines REDAT and POLM1; the EXEC mainline is loaded and executed through use of a CALL LINK (EXEC) statement.

7. Programs Calling This Program: This mainline is loaded and executed either by 1) the initial execution as specified on the IBM 1130 Monitor Control Record // XEQ GENIS; 2) the EXEC mainline upon the completion of a run; or 3) the GRAPH mainline upon the completion of a graph.

8. Local Variables:

LIST - 40 word integer array used to store the title (80 characters or less)

KEY4, KEY9, KEY10, KEY11, KEY12, KEY13, KEY14, and KEY15, integer variables used to store the setting of the appropriate data switches; 1 equals on, 2 equals off.

9. Printed Output: The following lines are printed on the 1132 printer as a title page -

UNITED STATES COAST GUARD ICEBREAKER PROPULSION SYSTEM SIMULATOR

SIMULATION OF (80 character title)

SERIAL NUMBER xxxxxxxxxx

At the top of the next page is printed - SELECTED INPUT DATA followed by the following headings with their appropriate data-

INITIAL CONDITIONS

CONTROL DATA

PROP DATA
HULL DATA
ELECTRICAL DATA
GOVERNOR AND DIESEL DATA
SYSTEM INERTIAS
ICE DATA

Finally, at the top of a new page is printed OUTPUT DATA.

The following instructions pertaining to the selection of the output data is typed out -

TYPE IN CONCLUSION OF PHRASE 'SIMULATION OF'
USING 80 CHARACTERS OR LESS.

TURN ON APPROPRIATE SWITCHES
ICE DATA-10
PROP DATA-11
HULL DATA-12
ELECTRICAL DATA-13
GOVERNOR AND BRIDGE CONTROLLER DATA -14
INERTIAS-15

TURN ON SELECTOR SWITCHES AND PRESS START.

1. Program Name: SUBROUTINE POLMI

2. Program Description: Subroutine POLMI computes the inertia of the propeller-motor system (J_{mp}) and the combined inertia of the diesel engine and generator armature (J_{pmg}).

3. Program Logic: The number of propeller blades (IZB) determines which of two formulas will be used in computing the "added inertia" of the propeller in water (J_{Pad}). If a value for the propeller inertia in air (J_{Pai}) was not entered in, its value is computed. The values of J_{mp} and J_{pmg} are then calculated.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON - IZB, XJPA, XJM, XJSH, XJD, XJG, XGEN, and D.
Results stored in COMMON - XJPA, XJMP, XJDG, and XJPE.

5. Program Length:

a. Source program - 45 statements (including comments).
b. Object program - 112 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: GENIS mainline.

8. Local Variables: None

9. Printed Output: None

1. Program Name: SUBROUTINE REDAT

2. Program Description: Subroutine REDAT reads in all 104 data cards that represent the propeller data, hull data, electrical data, governor-diesel data, systems inertias, ice conditions, scale factors, initial conditions and control data.

3. Program Logic: After reading in all the data cards, the program sets the values of ρ and π to 1.99 and 3.14159 respectively. The armature circuit resistance is calculated from the resistance of the motor armature (R_M), the resistance of the generator armature (R_G), and the number of generators (XGEN). The "mass" term is also calculated.

4. Listing of Parameters: All parameters are located in COMMON.
Values taken from COMMON- none.
Results stored in COMMON - the variables initially read in by this program are so indicated under the COMMON data description section.

5. Program Length:

   a. Source program - 127 statements (including comments).
   b. Object program - 498 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: GENIS mainline.

8. Local Variables: None

9. Printed Output: None

1. Program Name: Mainline Routine EXEC

2. Program Description: The EXEC mainline routine coordinates the actual simulation. Subroutine CNTRL is called to compute the initial values afterwhich subroutine OUT is called to print them out. Thereafter subroutine HEUN is called each time a new time step is taken. Subroutine OUT is called each time the values are to be printed out. Upon conclusion of the simulation subroutine FINIS is called to select control for the remainder of the run. If FINIS returns(implies a graph was not desired) then the GENIS mainline may be loaded and executed if it is desired to start a new problem, otherwise HEUN is called to continue the present problem. The EXEC mainline writes the values on the disk for use by the GRAPH mainline.

3. Program Logic: If data switch zero (0) is on after the initial conditions are computed subroutine FINIS will be called, otherwise it will not be. The variable NON controls the ship interval for printing with the variable MIKE being a counter of the steps taken.

4. Listing of Parameters: All parameters are in COMMON .
Values taken from COMMON - TIMSC, NON, CT, TEND, XNPM, XNP, CVS, QM, PI, AI, EB, FMI, CXS, QPI, XPROI, IELO, QI, and ISPAR(1).
Results stored in COMMON - IELO, JELO, KELO, and ISPAR(2).

5. Program Length:

a. Source program - 108 statements (including comments).
b. Object program - 264 machine words.

6. Programs Called By This Program: Subroutines CNTRL, OUT, HEUN, and FINIS. The GENIS mainline may be loaded and executed through use of a CALL LINK (GENIS) statement.

7. Programs Calling This Program: Loaded and executed by the GENIS mainline through use of a CALL LINK(EXEC) statement.

8. Local Variables:

III - initially set to one (1) but will take on a value of two(2) the first time the value of the current time (CT) exceeds 20 seconds; is used to reset the disk write time step (DTT) to 1.0 after CT exceeds 20 seconds.

MIKE - initially set to one (1) but is updated by one each time a new time step is completed; is used to count time steps for comparison with print out time step (NON).

DTT - disk write time step; is mutially set to the value of TIMSC cut will take on the value of one (1) after CT exceeds 20.0 seconds.

TTT - time value for next writing on disk; is updated by DDT each time the values are written on the disk.

KEYO - contains setting of data switch zero; equals one (1) if on, two (2) if off.

RPMPM- angular velocity of prime mover in revolutions per minute.

RPMP - propeller angular velocity in revolutions per minute.

CVK - ship's speed in knots

XKW - ship's horsepower

JENE - set to value of ISPAR(1) for use in computed GO TO Statement.

9. Printed Output: None

1. Program Name: SUBROUTINE CNTRL

2. Program Description: Subroutine CNTRL sets the value of the ordered bridge lever position (θ_o) depending on the value of ICNTR which indicates the command that was given. Subroutine CALVR is then called to calculate the initial values, afterwhich the current values are saved.

3. Program Logic: The value of θ_o is set to 165.0° if ahead flank was given (ICNTR=1), 135.0° if either ahead full (ICNTR=3) or ahead full in ice (ICNTR=4) was given, and -165.0° if back flank (ICNTR=2) was given. The current values of the eleven state variables ($t, i_a, dz/dt, z, n_{pm}, i'_f, i_{fg}, n_p, V_s$, XS, and SIE) are placed into alternate locations. The initial values are then written on the disk and a heading is printed out to label the columns for succeeding print outs.

4. Listing of Parameters: All parameters are in COMMON.

Values taken from COMMON - ICNTR, FIMAX, FIP, T, AI, Y, Z, XNPM, FIP, FGI, XNP, VS, XS, QM, PI, EB, QPI, XPROI, QI and IELO.

Results stored in COMMON -SIE, THETO, FMI, CT, CAI, CY, CZ, CXNPM, CFIP, CGI, CXNP, CUS, CXS, CSIE, and IELO.

5. Program Length:

a. Source program - 84 statements (including comments).

b. Object program - 242 machine words.

6. Programs Called By This Program: Subroutine CALVR

7. Programs Calling This Program: EXEC mainline

8. Local Variables:

RPMPM -	angular velocity of prime mover in revolutions per minute.
RPMP -	propeller angular velocity in revolutions per minute.
CVK -	ship's speed in knots
XKW -	ship's horsepower

9. Printed Output: Prints out the column headings for the output data that appears in the printer as follows:

T(SEC) RPMPM CFGI RPMP VK QDE(FT-LB)
QG EB(V) FMI EC QM(FT-LB) AI(A)
QP THRUST(T) RESIST HP/SH

1. Program Name: SUBROUTINE HEUN

2. Program Description: Subroutine HEUN is the numerical integration routine. The Runge-Kutta 4^{th} order method is used. Very briefly, the method is as follows: we are seeking to solve the equation $dy/dx = f(x, y)$. Given an (x_n, y_n), we calculate: $y_{n+1} = y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6$

where: $k_1 = hf(x_n, y_n)$

$k_2 = hf(x_n + h/2, y_n + k_1/2)$

$k_3 = hf(x_n + h/2, y_n + k_2/2)$

$k_4 = hf(x_n + h, y_n + k_3)$

and h = independent axis increment (dx).

3. Program Logic: Before subroutine CALVR is called each time, a check is made on the values of Z and FIP. If Z is greater than 1.000 Z is set to 1.000 if FIP is negative, FIP is set to zero. The local variable IJAX is used to return control to the point in the routine where the branch was taken to check on the values of Z and FIP.

4. Listing of Parameters: All parameters are in COMMON. Values taken from and stored in COMMON - the values and derivatives are represented by the three arrays STVR, CSTVR, and STVRD. The array STVRD corresponds to Z through TEMP in COMMON, CSTVR corresponds to CZ through CTEMP, and STVRD corresponds to ZDO through TEMDO. T and DT are also used.

1. Program Name: SUBROUTINE HEUN

2. Program Description: Subroutine HEUN is the numerical integration routine. The Runge-Kutta 4th order method is used. Very briefly, the method is as follows: we are seeking to solve the equation $dy/dx = f(x, y)$. Given an (x_n, y_n), we calculate: $y_{n+1} = y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6$

where: $k_1 = hf(x_n, y_n)$

$k_2 = hf(x_n + h/2, y_n + k_1/2)$

$k_3 = hf(x_n + h/2, y_n + k_2/2)$

$k_4 = hf(x_n + h, y_n + k_3)$

and h = independent axis increment (dx).

3 Program Logic: Before subroutine CALVR is called each time, a check is made on the values of Z and FIP. If Z is greater than 1.000 Z is set to 1.000 if FIP is negative, FIP is set to zero. The local variable IJAX is used to return control to the point in the routine where the branch was taken to check on the values of Z and FIP.

4. Listing of Parameters: All parameters are in COMMON. Values taken from and stored in COMMON - the values and derivatives are represented by the three arrays STVR, CSTVR, and STVRD. The array STVRD corresponds to Z through TEMP in COMMON, CSTVR corresponds to CZ through CTEMP, and STVRD corresponds to ZDO through TEMDO. T and DT are also used.

5. Program Length:

   a. Source program - 84 statements (including comments).
   b. Object program - 354 machine words.

6. Programs Called By This Program: Subroutine CALVR.

7. Programs Calling This Program: EXEC mainline.

8. Local Variables:

   IJAX - takes on the values one (1) through 4 to control the 4 times subroutine CALVR is called.

   DD - set to DT/2 for use in solving for k_2 and k_3.

9. Printed Output: None

1. Program Name: SUBROUTINE CALVR

2. Program Description: Subroutine CALVR updates the values of all the working variables in the program. This is accomplished by calling the following subroutines: CON1, CON2, QDEV, FLUXG, FLUXM, HULL, PROP, QICE (if the ship has encountered ice), and STATE. Intermixed between calls to the above subroutines, subroutine CALVR does some calculations and adjusting of its own. This subroutine is called by subroutine HEUN (the numerical integration routine) each time an updated set of values is desired.

3. Program Logic: Subroutine QICE is called only after the ship has encountered ice which is controlled by the variable TICE - the relative time since the beginning of the run before the ship encounters ice. The variable IPLUG (equivalenced to ISPAR(2)) is initially set to one (1) so that the initial values of T and FMI will be saved for use in the calculation of the updated values of FMI. Thereafter IPLUG is set to two(2) to immediately calculate a new value of FMI using the saved initial values.

4. Listing of Parameters: All parameters are in COMMON. Values taken from COMMON-XNGR, NXPM, NXGRD, XNPMD, QDE, QBRG, CWM, XNPM, XGEN, TRNFG, FGI, TRNSF, AI, CFG, CFM, XNP, RG, IPLUG, T, FMI, TAUA, TAUB, FMIRV, THETA, C1, XK1, FIMAX, FIP, CSHFR, EB, WDED, VS, QPI, TDED, XPROI, XKK, CT, and TICE.
Results stored in COMMON - XNE, XNEDO, QDE, AMPTR, EG, EC, QG, EB, QM, FMI, AIHAT, QFRM, VA, QP, TP, and QI.

5. Program Length:

a. Source program - 110 statements (including comments).

b. Object program - 310 machine words.

6. Programs Called By This Program: Subroutines CON1, CON2, QDEV, FLUXG, FLUXM, HULL, PROP, QICE, and STATE.

7. Programs Calling This Program: Subroutines HEUN and CNTRL.

8. Local Variables:

TTN - contains initial value of T.

FMIIC - contains initial value of FMI

ABLE, BAKER - set to exponential expressions used in the calculation of FMI.

9. Printed Output: None

1. Program Name: SUBROUTINE CON1

2. Program Description: Subroutine CON1 computes the actual bridge lever position (θ) and bridge lever time rate ($d\theta/dt$) which depends on time (t) and control values τ_1 and τ_2.

3. Program Logic: The control values τ_1 and τ_2 determine the speed of the lever in moving from the initial position (θ_{IC}) to the final or ordered position (θ^*). The lever moves from θ_{IC} to θ_1 in τ_1 seconds and from θ_1 to θ^* in τ_2 seconds. A check is made to determine where the present value of t lies, after-which θ and $d\theta/dt$ are computed accordingly.

4. Listing of Parameters: All parameters are located in COMMON.
Values taken from COMMON - T, TAU1, TAU2, THETØ, THET1, and THEIC
Results stored in COMMON - THETA and THETD.

5. Program Length:

a.	Source program -	45	statements (including comments).
b.	Object program -	82	machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine CALVR.

8. Local Variables: None

9. Printed Output: None

1. Program Name: SUBROUTINE CON2

2. Program Description: Subroutine CON2 computes the ordered generator field voltage (e_{fg}), ordered diesel speed (n_{gr}) and ordered diesel speed rate (dn_{gr}/dt). The values calculated are dependent on the values of the actual bridge lever position (θ) and bridge lever time rate ($d\theta/dt$) computed in subroutine CON1.

5. Program Logic: A maximum value of e_{fg} is calculated from the generator field resistance (R_{fg}) and variable resistor in the generator field (R_x). If the value of θ is below 63.0 degrees, the diesel is still idling and the value of dn_{gr}/dt will be zero. If θ is greater than 63.0 degrees, a check is made to determine if a full (θ=135.0 degrees) or maximum (θ=165.0 degrees) command was given. The sign of e_{fg} is determined to be negative if θ is negative and positive if θ is positive.

4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON RX, RFG, THETA, THETØ and THETD. Results stored in COMMON - EFG, XNGR, and XNGRD.

5. Program Length:

   a. Source program - 56 statements (including comments).
   b. Object program - 178 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine CALVR.

8. Local Variables:

EFGM - real variable containing the computed maximum value of e_{fg}.

THET - real value containing the absolute value of θ.

9. Printed Output: None.

1. Program Name: SUBROUTINE FLUXG

2. Program Description: Subroutine FLUXG calculates the generator flux (C_{fg}), generator field inductance (L_{fg}), generator time constant (τ_{fg}), mutual inductance between generator separate and in series fields (XM), and the inductance of the generator in series field (XLASF) from the value of ampere-turns (AMPTR) , a table of field flux (CFMA), values of control field turns (TRNFG), series field tunrs (TRNSF), field resistances (RX, RG) and flux conversion constant (ZGP).

3. Program Logic: The generator field flux is taken from the input array (CFGA) with the value of the generator spacing (SPGEN) being the independent axis. Since the ind ependent axis runs from zero to 12 it is necessary to truncate the independent axis index (KG) and then compute the enclosing indices by adding two and one to the value.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON-AMPTR, SPGEN, CFGA, TRNFG
ZGP, RX, RG, and TRNSF.
Results stored in COMMON-CFG, XLFG, TAUFG, XM, and XLASF.

5. Program Length:

a. Source program - 52 statements (including comments).
b. Object program - 148 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine:CALVR

8. Local Variables:

FG- real variable containing the actual value of the independent axis index.

KG - integer variable containing the truncated value of FG.

DCGDI - real variable containing the interval that the value of FG falls in.

9. Printed Output: None

1. Program Name: SUBROUTINE FLUXM

2. Program Description: Subroutine FLUXM calculates the motor field flux (C_{fm}), the motor field time constant (τ_{fm}), and the motor field inductance (L_{fm}) from the value of field current (FMI), a table of field flux (CFMA), values of field turns (TRNFM), field resistance (RFM) and flux conversion constant (ZMP).

3. Program Logic: The motor field flux is taken from the input array (CFMA) with the value of the motor spacing (SPMOT) being the independent axis. Since the independent axis runs from zero to 12, it is necessary to truncate the independent axis index (KM) and then compute the enclosing indices by adding two and one to the value.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON-FMI, SPMOT, DFMA, TRNFM, ZMP, and RFM.
Results stored in COMMON-DFM, XLFM, and TAUFM.

5. Program Length:

   a. Source program - 52 statements (including comments).
   b. Object program - 112 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine CALVR.

8. Local Variables:

   FM - real variable containing actual value of independent axis index.

KM- integer value containing truncated value of FM

DCMDI - real value containing the interval that the value of FM falls in.

9. Printed Output: None

1. Program Name: SUBROUTINE QDEV

2. Program Description: Subroutine QDEV calculates the developed diesel torque (Q_{de}) using the developed torque constant (K_6), the servo stroke position (Z), the number of generators (XGEN), and the angular velocity of the prime mover (N_{pm}).

3. Program Logic: The maximum developed diesel torque ($Q_{de_{max}}$) is computed from the angular velocity of the prime mover (N_{pm}). A diesel engine friction torque is computed using a friction torque - N_{pm} curve with a Y-intercept indicated by the value of QBRG and a slope indicated by CWM. This value is added to the old value of $Q_{de\,max}$ to arrive at an augmented $Q_{de\,max}$. If the value of Q_{de} is greater than $Q_{de\,max}$, Q_{de} is set equal to $Q_{de_{max}}$.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON - XK6, Z, XNPM, XGEN, QBRG, and CWM.
Results stored in COMMON - QDE

5. Program Length:

a. Source program - 53 statements (including comments).
b. Object program - 138 machine words.

6. Programs Called By This Program : None

7. Programs Calling This Program : Subroutine CALVR

8. Local Variables:

QDEMX - real variable containing the maximum developed diesel torque.

9. Printed Output: None

1. Program Name: SUBROUTINE HULL

2. Program Description: Subroutine HULL computes the ship's resistance (R), wake deduction (WDED), and thrust deduction (TDED) from the ship's speed (V_S), ice resistance factor (CICE) and input tables of effective horsepower (EHP), wake deduction (WADED) and thrust deduction (THDED).

3. Program Logic: If the control value ICNTR implies ahead full in ice (equal to 4), the ice resistance factor (CICE) is taken into account in computing the resistance of the hull. If the ship's speed is negative, the hull resistance is taken as 1.4 times the computed value and the wake and thrust deductions are taken as 1.0 and 0.9 respectively.

Since the independent axis of the three tables (EHP, WADED, and THDED) runs from zero to 19.0, it is necessary to add two and one to the truncated index (KV) in order to arrive at the interval the actual index (V) falls between.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON-VS, EHP, WADED, THDED, ICNTR, and CICE.
Results stored in COMMON - WDED, TDED, and R.

5. Program Length:

a. Source program - 52 statements (including comments).
b. Object program - 152 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine CALVR

8. Local Variables:

V -	real variable containing the absolute value of the ship's speed in knots.
KV -	integer variable containing truncated value of V.
REHP -	real variable containing the effective horsepower used in calculating R.

9. Printed Output: None

1. Program Name: SUBROUTINE PROP

2. Program Description: Subroutine PROP makes use of the propeller data tables to compute the ideal propeller torque (QPI) and thrust per propeller (XPROI) given the propeller angular velocity (n_p) advance velocity of the propeller (Va).

3. Program Logic: The values of the propeller thrust and torque coefficients are computed from the tables by a complex but efficient means. Depending on the signs of VA and XN, a value of the variable KB is set (see Local Variables). This value is used in computing the table subscript and also to set the sign of the interpolation fraction (PP) to the negative. A value of XJ or 1/XJ is computed. The integer variable KD is set to the truncated value of 10 times $|XJ|$. This will determine the relative position of I in its quadrant. The actual value of I in the table is computed using KD and KB. The interpolation fraction (PP) is computed and its sign adjusted by using KB to be negative. The coefficients can now be computed from the tables.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON -RO, D, XN, VA, B, BB, E, and EE.
Results stored in COMMON - QPI, and XPROI.

5. Program Length:

a. Source program - 106 statements (including comments).
b. Object program - 430 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine CALVR

8. Local Variables:

KB - takes on the values zero (0) through 3 depending on the signs of XN and VA as follows.

KB	0	1	2	3
XN	+	+	-	-
VA	-	-	+	-

I - index to data arrays (B, BB, E, and EE).

TT - equal equal to zero (0) if using J, 1 if using 1/J

XJ - equal to VA/(XN· D) if result within range /XJ/= 1.5, otherwise equal to XN· D/VA

KD - equal to 10· /XJ/ and used in computing I

PP - negative fraction used in interp lation

XKQ - propeller torque coefficient

XKT - propeller thrust coefficient

XKQD - modified propeller torque coefficient

XKTD - modified propeller thrust coefficient

9. Printed Output: None

1. Program Name: SUBROUTINE QICE

2. Program Description: Subroutine QICE computes the ice torque (Q_i) of the propeller. If necessary, subroutine RBMAX is called to determine a new value of the maximum depth of cut in feet that the propeller will make in the ice. The value of Q_i is taken as the lesser of crushing torque (Q_{iCR}) and combined crushing and shearing torque (Q_{iCRS}) except that Q_i will never be less than the minimum ice torque (Q_{iMIN}).

3. Program Logic: The variable JELO is initially set to one (1) in the EXEC mainline so that subroutine RBMAX will be called immediately the first time this subroutine is executed. Thereafter, subroutine RBMAX is called only when a new value of the depth of cut is needed (when the angular position of the propeller is greater than the cut angle, α_b, and greater than or equal to the angle between propeller baldes.

4. Listing of Parameters: All parameters are located in COMMON.
Values taken from COMMON-JELO, RBMX, D, IZB, PI, SIE, XNP, PDR, DAR, DCR, EO, EK, DCT, DSH, and VS.
Results stored in COMMON-JELO and QI.

5. Program Length:

   a. Source program - 75 statements (including comments).
   b. Object program - 352 machine words.

6. Programs Called By This Program: Subroutine RBMAX

7. Programs Calling This Program: Subroutine CALVR

8. Local Variables:

CSAB2 -	real variable containing the cosine of $\alpha_b/2$
SNAB2 -	real variable containing the sine of $\alpha_b/2$
AB2 -	real variable containing the value of α_b
XIZB -	real variable containing the number of propeller blades.
ABB -	real variable containing the number of radians between prop blades.
CSBET -	real variable containing the cosine of β
RB -	real variable containing value of R_b
RM -	although appearing in COMMON, this local real variable contains the value of R_m
ALFAV-	real variable containing value of α_v
ALFA1 -	real variable containing value of α_1
E1 -	real variable containing value of e_1
BAV -	real variable containing value of B_{avg}
QICRS -	real variable containing value of Q_{iCRS}
QIMIN -	real variable containing value of Q_{iMIN}
QICR -	real variable containing value of Q_{iCR}

9. Printed Output: None

1. Program Name: SUBROUTINE RBMAX

2. Program Description: Subroutine RBMAX calculates the maximum depth of cut in feet that the propeller blade maker in a block of ice. This value is chosen at random from an array of values that represent the maximum depth of cut as a percentage of the propeller diameter.

3. Program Logic: The ice pattern is made up of two blocks of ice in a repeating sequence. Four variables control the spacing and size of the blocks (BLOC1, BLOC2, DBB12, and DBB21.) The value of the variable KELO determines where in the ice pattern the ship is at a given instance. The value of KELO is initially set to one (1) in the EXEC mainline. When this subroutine is initially executed, the ship had just entered block one and KELO is set to two (2). The variable XICE is used to save the ship's distance upon entering the ice to determine how long the ship will remain in a particular block of ice. The variable KELO assumes other values depending on whether the ship is between blocks one and two, in block two , or between blocks two and one. The first 10 values of the RBA array are used when in block one, the remaining 15 values when in block two.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON-KELO , CXS, BLOC1, RBA, D, DBB21, BLOC2, and DBB12.
Results stored in COMMON - CSIE, SIE, RBMX, and KELO.

5. Program Length:

a. Source program - 87 statements (including comments).

b. Object program - 178 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine QICE

8. Local Variables:

XICE - real variable used to save the present value of the ship's distance each time a block is entered or exited from.

IA - integer variable used to reference the RBA array in determining the maximum depth of cut; updated by one (1) each time a new value is needed.

1. Program Name: SUBROUTINE STATE

2. Program Description: Subroutine STATE calculates the values of all the derivatives using the values calculated in the previous subroutines. The derivatives calculated are: 1) dz/dt -servo stroke velocity; 2) d^2z/dt^2- servo stroke acceleration; 3) dn_{pm}/dt - time rate of prime mover speed; 4) di_{fg}/dt - time rate of generator field current; 4) di'_f/dt - time rate of motor field adjusting current; 6) dn_p/dt -propeller acceleration; 7) dv_s/dt - ship's acceleration; 8) dx_s/dt - ship's velocity; 9)SIEDO - peopeller angular velocity; and 10) di_a/dt - time rate of armature circuit current.

3. Program Logic: If the value of the actual bridge lever position (θ) is negative or if the bus voltage (e_b) is less than or equal to the constant C_1, then the time rate of the motor field adjusting current is set to zero.

4. Listing of Parameters: All parameters are located in COMMON.
Values taken from COMMON, Y, ALP1, ALPZ, ALP3, ALPD, XNEDO, XNE, Z, QDE, QG, PI, XJDG, EFG, RFG, RX, FGI, EG, EC, RA, AI, XLA, XGEN, XLASF, XM, XLFG, THETA, EB, C1, AIHAT, XLFM, QM, QFRM, QP, QI, XJMP, TP, R, XMASS, VS and XNP.
Results stored in COMMON -ZDO, YDO, XNPMD, AIDO, FGIDO, FIPDO, XNPDO, VSDO, XSDO, and SIEDO.

5. Program Length:

a. Source program - 66 statements (including comments).
b. Object program - 252 machine words.

6. Programs Called By This Program: None

7. Program Calling This Program: Subroutine CALVR.

8. Local Variables:

VOLT1- real variable used to contain the value of $e_{fg} - (R_{fg} + R_x) i_{fg}$

VOLT2- real variable used to contain the value of $e_g - e_c - R_a i_a$

XLD - real variable used to contain the value of $L_a + 1.5 \cdot XLASF/XGEN$

XLM - real variable used to contain the value of $2.25\ XM^2/XGEN$

DENOM - real variable used to contain the value of $1.5\ L_{fg}\ (XLD-XLM)$

9. Printed Output: None

1. Program Name: SUBROUTINE OUT

2. Program Description: Subroutine OUT prints out the program variables in question on the IBM 1132 printer.

3. Program Logic: Before printing the variables in question, the following values are changed to different units: the value of the angular velocity of the prime mover (n_{pm}) is changed from revolutions per second to revolutions per minute as is the propeller angular velocity (n_p); the ship's speed (V_s) is converted from feet per second, to knots; the total propeller thrust (T_p) and ship resistance (R) are changed from pounds to long tons. The shaft horsepower is also computed from the motor torque (Q_M) and n_p.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON - CXNPM, CXNP, CVS, TP, R, QM, PI, XNP, CT, CFGI, QDE, QG, EB, FM1, EC, AI, and QP.
Results stored in COMMON - None

5. Program Length:

a. Source program - 49 Statements (including comments).
b. Object program - 132 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: EXEC mainline.

8. Local Variables:

RPMPM - real variable containing angular velocity of prime mover in revolutions per minute.

RPMP - real variable containing propeller angular velocity in revolutions per minute

CVK - real variable containing ship's speed in knots.

TPT - real variable containing total propeller thrust in long tons.

RT - real variable containing ship's resistance in long tons.

POW - Real variable containing ship's horsepower.

9. Printed Output: This program prints out the 16 variables in question on the IBM 1130 printer. The values occupy one line in the following format:

```
     /   RPMPM  ifg     RPMP   CVK  /   Qde
bxxx. xxxbxxx. x bxxx. xbxxxx. xbxxx. x xxxxxx. x

   Qg      eb      ifm      ec       Qm
xxxxxx. xbxxxx. xbxxx. xbxxxx. xbxxxxxxx. x

   ia          Qp     /   TPT    RT   /   POW
bxxxxxx. xbxxxxxxx. x  xxxxx. xbxxxx. x  xxxxxx. x
```

1. Program Name: SUBROUTINE FINIS

2. Program Description: Subroutine FINIS interrogates data switches 1 through 7 to achieve the following results:

   Switch 1 on implies punch new initial condition cards, 2 on implies plot data, 3 on implies continue present solution, 4 on prevents future typing of data switch setting instcuctions. 5 on implies a new value of the time step (dt), 6 on implies a new value of the print interval, and 7 on implies a new value of the armature circuit inductance (L_a).

3. Program Logic: If a graph is indicated, the GRAPH mainline is loaded and executed by use of a CALL LINK (GRAPH) statement in this routine. The terminating time value (TEND) is set to the last time value considered before the graph routine is executed.

4. Listing of Parameters: All parameters are in COMMON.
   Values taken from COMMON - THETA, XNPMD, Y, Z, FGI, XNPM, FIP, XNP, VS, AI, AIDO, CT and DT.
   Results stored in COMMON - TEND, DT, NON, XLA, and ISPAR(1)

5. Program Length:

   a. Source program - 92 statements (including comments).
   b. Object program - 498 machine words.

6. Programs Called By This Program: The GRAPH mainline may be loaded and executed by use of a CALL LINK (GRAPH) statement.

7. Programs Calling This Program: EXEC mainline

8. Local Variables: KEY1, KEY2, KEY3, KEY4, KEY5, KEY6, KEY7- interger variables containing the switch settings for data switches 1 through 7 respectively. A value of one (1) indicates the switch was on, a value of 2 indicates off.

9. Printed Output: If switch 4 was off, the following switch setting instructions are typed out:

TURN ON SWITCH --
1 - TO PUNCH NEW INITIAL CONDITION CARDS
2 - TO PLOT DATA
3 - TO CONTINUE SOLUTION OF CURRENT PROBLEM- SWITCH OFF STARTS NEW PROBLEM
4 - TO SUPPRESS FUTURE TYPING OF THESE I INSTRUCTIONS
5 - TO CHANGE TIME STEP
6 - TO CHANGE PRINT INTERVAL
7 - TO CHANGE XLA
TURN ON SELECTOR SWITCHES AND PRESS START

If new initial condition cards are punched, they are also printed on the IBM 1132 printer.

If new values of either dt, NON, or L_a are to be entered, any of the following messages, whichever be appropriate, will be typed out:

( ) - TIME STEP
( ) - NON
( ) - XLA

1. Program Name: Mainline Routine GRAPH

2. Program Description: Mainline program GRAPH controls the plotting of the data stored on the disk by the EXEC mainline program. The program first requires that the code numbers of the variables to be plotted be entered from the console keyboard. A request is then made to determine if a grid is to be plotted by requiring a code number to be entered from the console keyboard. If a grid is to be plotted, the program requires three lines of graph identification to be entered from the keyboard after which the information as well as the graph serial number (computed or read in by the GENIS mainline) is printed out along with the grid. The program then loops through the array of code numbers to plot each variable in turn. If the curve extends beyond 20 seconds, the time axis plotter scale factor is reduced by a factor of 10 to plot the remainder of the curve. Upon the conclusion of the plotting for each curve, the pen is positioned for printing out the curve label. When all the requested variables have been plotted, the program loads and executes the GENIS mainline to begin a new problem.

3. Program Logic: When the curves extend beyond 20 seconds, it is necessary to compute a pseudo-origin for the remainder of the curve since the time axis plotter scale factor is reduced by a factor of 10 from the 20 second mark until the end of the graph. The new origin 'XDARN', is computed by multiplying the last time value encountered less than or equal to 19.9 seconds by 10 and subtracting the original value from the result. Following time values are added to this pseudo-origin to arrive at an augmented time value, 'CTT', for plotting the corresponding value of the variable in question on the curve.

4. Listing of Parameters: All parameters are located in COMMON. Values are taken from COMMON-SPARE(4), TEND, SX, SY, CT and VAR. Results stored in COMMON-NOVAR, NVAR, IELO, NIP, and JJ.

5. Program Length:

a. Source program - 246 statements (including comments).
b. Object program - 738 machine words.

6. Programs Called By This Program: Subroutines GRID, VAPLO, DATA, and PRINT. The GENIS mainline is loaded and executed by use of a CALL LINK (GENIS) statement following the completion of the entire graph.

7. Programs Calling This Program: This mainline is loaded and executed by use of the CALL LINK (GRAPH) statement in subroutine FINIS (of the EXEC mainline).

8. Local Variables:

NPLOT - this integer variable, although in COMMON, is local to this program and is used to contain the input from the keyboard that determines if a grid is to be plotted; a value greater than or equal to one (1) implies plot a grid, a value less than one (1) implies no grid is to be plotted.

TH - character rotation value: real variable containing the number of radians from horizontal (+X direction) that a line of characters is to be plotted.

L - integer variable used to control the up or down position of the pen during the plotting of a variable; initially set to one (1) so that the pen can be moved to the first value

to be plotted in the up-position for a particular variable, thereafter assumes a value of 2 since the pen will remain down in plotting the remainder of the curve.

XD, YD- real variables used to position the pen for plotting of the curve label after the curve has been plotted; set to start the first character 0.05 inches to the right and below the last value plotted.

The following local variables come into use only if the curves extend beyond 20 seconds:

SXX - time axis plotter scale factor: real variable set to one-tenth of the initial time axis plotter scale factor, 'SX', for plotting of the curves should they extend beyond 20 seconds since after 20 seconds the increments become 10 seconds.

XD - real variable set to 10 times the actual time value for the first time value exceeding or equal to 19.90 seconds.

XDARN- real variable set to the value of 'XD' minus the first time value exceeding or equal to 19.90 seconds. This value will then be used as a pseudo-origin for plotting the remainder of the curve since the plotter scale factor has been reduced by a factor of 10.

CTT - real variable equal to the actual time value, 'CT', plus the value of 'XDARN' for each time step in plotting a curve after the 20 second mark.

9. Printed Output: The following messages and requests for data are typed out on the console typewriter:

AI--1, EB--2, XNPM--3, FMI--4, XNP--5, XKW--6,

CVK--7, QPI--8, XPROI--9, CXS--10, QI--11, TEMP--12

WHICH VARIABLES TO BE PLOTTED (10 ONLY) -

( )( )( )( )( )( )( )( )( )

DO YOU WANT A GRID (0-NO, 1-YES)

( )

If a grid is requested, three lines of not more than 50 characters each are required to be input after which they will be plotted out preceeded by the following information:

SERIAL NUMBER - - xxxxxxxxxx

1. Program Name: SUBROUTINE GRID

2. Program Description: Subroutine GRID is called to plot and label the time axis which is represented by the +X direction and the ordinate axis which is represented by the +Y and -Y directions. In plotting the time axis, the first 20 seconds are plotted in increments of one second. A test is then made to determine if the axis is to extend beyond 20 seconds and if so, the additional length is determined. The time axis is then plotted from the 20 second mark to the end in increments of 10 seconds. The ordinate axis is plotted from +5.0 to -5.0 with each unit representing one inch on the graph. The ordinate axis is labeled with the scale factors (quantity of each variable corresponding to one inch in the Y-direction) for the variables to be plotted.

3. Program Logic: In labeling the time axis, the local real variables 'XD' and 'WYE' are set to control the positioning of each label so that they will be 0.05 inches below and centered on each tick mark. If the time axis extends beyond 20 seconds, the additional length is computed as the number of tick marks beyond the one at 20 seconds (one less than the number of intervals between tick marks).

4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON-SX, PI, TEND, NOVAR, NVAR, and SY. Results stored in COMMON-NIP.

5. Program Length:

   a. Source program - 223 statements (including comments).
   b. Object program - 696 machine words.

6. Programs Called By This Program: Subroutine VAPLO.

7. Programs Calling This Program: GRAPH mainline.

8. Local Variables:

SXX - time axis plotter scale factor: real variable initially set to 'SX' for the plotting of the first 20 seconds of the time axis and then to one-tenth of 'SX' for the plotting of the remainder of the time axis since after 20 seconds the time interval becomes 10 seconds over the same actual distance initially used for the one (1) second intervals.

TH - character rotation value: real variable containing the number of radians from horizontal (+X direction) that a line of characters is to be plotted.

XD - label centering value: real variable used to center the axes labels at each tick mark; equal to 0.05 divided by the plotter scale factor for the axis in question.

WYE - real variable used initially to place the X-axis labels 0.05 inches below each tick mark and then to indicate the relative position of each label for the +Y and -Y directions.

X - real variable initially used to label the time axis and then used to start the Y-axis labels 0.77 inches to the left of the Y-axis. Finally used to indicate the starting position (to the left of the Y-axis) that each curve actual scale factor will be listed on the graph.

TEST - real variable equal to the extent of the time axis beyond 20 seconds.

TEST1 - real variable equal to one tenth of the extent of the time axis beyond 20 seconds plus a constant of one (1) to plot out the remaining tick marks.

ITEST - integer variable containing the number of tick marks to be plotted beyond the one at the 20 second mark.

I - integer variable used as a DO-loop index in labeling the axes.

MIKE - integer variable used to store in turn the code number for each variable to be plotted for use in a computed GO TO statement to plot the actual scale factor for that variable.

SYI - real value containing the actual scale factor (quantity of variable per inch along the Y-axis) for each variable in turn as the scale factor is plotted.

9. Printed Output: Any of the following 12 statements concerning the Y-axis scale factors may be listed on the graph:

AI (1 IN. = xxxxx.x AMPS)
EB (1 IN. - xxx.x VOLTS)
NPM (1 IN. = xxx.x RPM)
FMI (1 IN. - xxx.x AMPS)
NP (1 IN. = xxx.x RPM)
KW (1 IN. = xxxx.x KW)
VK (1 IN. = xx.x KTS.)
QP (1 IN. = xxxxxx.x FT-LBS)
TP (1 IN. = xxxxxx.x LBS)
XS (1 IN. = xxxx.x FEET)
QI (1 IN. = xxxxxx.x FT-LBS)
TEMP (1 IN. = xxx.x DEG. CENT.)

The following time axis label is plotted out:

TIME (SECS)

Each time axis tick mark is labeled starting at 1.0 and going through 20.0 by one's and then through the final value by 10's. Every whole ordinate (Y) axis tick mark is labeled from 5.0 through 0.0 to -5.0.

1. Program Name: SUBROUTINE DATA

2. Program Description: Subroutine DATA retrieves from the disk file the next time value and next value of the variable currently being plotted. The disk file index, 'IELO', is automatically updated to the next record after the values are retrieved. The time value is placed in location 'CT' and the new value of the variable being plotted in 'VAR' , both in COMMON.

3. Program Logic: The current value of the disk file index ('IELO') is used to reference the record of the disk file that is next to be retrieved. Since there is no way to read just one value from the disk, the entire record must be retrieved, after which the value of 'IELO' is automatically updated by one (1) to point to the next record. The code number of the variable currently being plotted is found in the word of the variable list array, 'NVAR', currently being considered which is indicated by value of the variable list array index 'JJ'. A computed GO TO statement is then used to return the new value for the variable in question.

4. Listing of Parameters: All parameters are located in COMMON.
Values taken from COMMON-NVAR, JJ, and IELO.
Results stored in COMMON -IELO, CT, and VAR.

5. Program Length:

a. Source program - 90 statements (including comments).
b. Object program - 136 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: GRAPH mainline

8. Local Variables:

NOOK - integer variable used to store the code number of the variable in question for use in a computed GO TO statement.

The following real variables, although appearing in COMMON, are actually local to this program and are used to store the value of each variable when the file record is read from the disk:

AI -	armature circuit current
EB-	bus volts
RPMPM-	diesel speed
FMI -	motor field current
RPMP-	propeller speed
XKW -	power
CVK -	vessel speed
QPI -	propeller torque
XPROI -	propeller thrust
CXS -	distance
QI -	ice torque
TEMP-	motor terperature

9. Printed Output: None

1. Program Name: SUBROUTINE VAPLO

2. Program Description: Subroutine VAPLO retrieves the plotter scale factor for the variable that is about to be plotted. The plotter scale factor is the actual number of inches along the Y-axis that corresponds to one unit of the variable in question and is placed in 'SY'.

3. Program Logic: The code number of the variable about to be plotted is found in the word of the variable list array, 'NVAR', that is currently being examined which is indicated by the value of the variable list array index, 'NIP'. A computed GO TO statement is then used to return the appropriate plotter scale factor.

4. Listing of Parameters: All parameters are located in COMMON.
Values taken from COMMON-NVAR, NIP, SY1, SY2, SY3, SY4, SY5, SY6, SY7, SY8, SY9, SY10, SY11, and SY12.
Results stored in COMMON - SY.

5. Program Length:

a. Source program - 85 statements (including comments).
b. Object program - 100 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: GRAPH mainline, subroutine GRID.

8. Local Variables:

NAN - integer variable used to store the code number of the variable in question for use in a computed GO TO statement.

9. Printed Output: None

1. Program Name: SUBROUTINE PRINT

2. Program Description: Subroutine PRINT is executed upon the completion of the curve for a particular variable in order to plot the a appropriate curve label for that variable. The curve label is plotted starting at the current location of the pen which was set by the GRAPH mainline before calling this routine to be immediately following the last value plotted for the variable.

3. Program Logic: The code number of the variable in question is found in the word of the variable list array, 'NVAR', that is currently being examined which is indicated by the value of the variable list array index 'JJ'. A computed GO TO statement is then used to plot out the appropriate curve label.

4. Listing of Parameters: All parameters are located in COMMON.
Values taken from COMMON -NVAR and JJ.
Results stored in COMMON - none.

5. Program Length:

   a. Source program - 97 statements (including comments).
   b. Object program - 212 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: GRAPH mainline

8. Local Variables:

   MIKE - integer variable used to store the code number of the variable in question for use in a computed GO TO statement.

9. Printed Output: This program may plot out any one of the following 12 curve labels depending on the variable in question:

--- ARM. CIR. CUR.
--- BUS VOLTS
--- DIESEL SPEED
--- MOTOR FIELD CUR.
--- PROP SPEED
--- POWER
--- VESSEL SPEED
--- PROP TORQUE
--- PROP THRUST
--- DISTANCE
--- ICE TORQUE
--- MOTOR TEMP.

PROGRAM LISTINGS

```
// FOR                                                                  GEN.
*NAME GENIS                                                           GENI0010
*ONE WORD INTEGERS                                                    GENI0020
*LIST ALL                                                             GENI0030
*IOCS(CARD,TYPEWRITER,KEYBOARD,1132PRINTER,DISK)                      GENI0040
C      GENIS MAINLINE                                                 GENI0050
C      VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                 GENI0060
C                                                                     GENI0070
       DIMENSION LIST(40)                                             GENI0080
       COMMON RO,PI,IZR,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,          GENI0090
      1 XJMR,XJDG,FIMAX,RA,RG,C1,XK1,OBRG,CSHFR,CWM,PCMS,XLA,         GENI0100
      2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                       GENI0110
       COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP        GENI0120
       COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12 GENI0130
       COMMON DT,CT,T,TEND,TIMSC                                      GENI0140
       COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP               GENI0150
       COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP   GENI0160
       COMMON ZDO,YDO,XNPMO,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TGENI0170
      1 EMDO                                                          GENI0180
       COMMON XNF,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,ER,EC,GENI0190
      1CM,DERM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                 GENI0200
       COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAGENI0210
      1SF,AMPTR,XGEN                                                  GENI0220
       COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBGENI0230
       COMMON B(64),BB(64),E(32),EE(32)                               GENI0240
       COMMON FHP(20),WDED,WADED(20),THDED,THDED(20),XKK              GENI0250
       COMMON IELO,JELO,KELO,LELO,MELO                                GENI0260
       COMMON SPARE(5),ISPAR(5)                                       GENI0270
       COMMON CAD,DELTA,RM                                            GENI0280
       COMMON RBA(26),RHMX,BLOC1,DBB12,BLOC2,DBB21                    GENI0290
       COMMON F0,EK,DCT,DCR,DSM,CICE,TICE                             GENI0300
       COMMON CK1,CK2,CK3                                             GENI0310
C                                                                     GENI0320
       EQUIVALENCE (SER,SPARE(4))                                     GENI0330
       EQUIVALENCE (XJPE,PCMS)                                        GENI0340
C                                                                     GENI0350
C                          DEFINE FILE TO STORE SERIAL NUMBER         GENI0360
       DEFINE FILE 2(1,2,U,N)                                         GENI0370
C                                                                     GENI0380
C                          READ ALL DATA CARDS                        GENI0390
       CALL REDAT                                                     GENI0400
C                          COMPUTE INERTIAS NOT READ IN               GENI0410
       CALL POLMI                                                     GENI0420
C                                                                     GENI0430
       WRITE(3,100)                                                   GENI0440
  100  FORMAT(1H1)                                                    GENI0450
       WRITE(3,1)                                                     GENI0460
     1 FORMAT(////////////////////////////,47X,'UNITED STATES COAST GUARD',//GENI0470
      1///,42X,''ICEBREAKER PROPULSION SYSTEM SIMULATOR')             GENI0480
C                                                                     GENI0490
C    DESCRIPTION OF OUTPUT NOW ENTERED ON KEYBOARD WILL BE ON TITLE PAGEGENI0500
C                                                                     GENI0510
       WRITE(1,2)                                                     GENI0520
     2 FORMAT('TYPE IN CONCLUSION OF PHRASE''SIMULATION OF''USING 80 CHARGENI0530
```

```
     1ACTERS OR LESS')                                                  GEN:0540
      READ(6,3) (LIST(I),I=1,40)                                        GEN:0550
    3 FORMAT(40A2)                                                      GEN:0560
      WRITE(3,4) (LIST(I),I=1,40)                                       GEN:0570
    4 FORMAT('1',20('/'),' SIMULATION OF ',40A2)                        GEN:0580
C                       CHECK IF SERIAL NUMBER IS TO BE UPDATED.        GEN:0590
C                       RESET TO ZERO IF DATA SWITCH 9 ON               GEN:0600
      CALL DATSW(9,KEY9)                                                GEN:0610
      GO TO(80,81),KEY9                                                 GEN:0620
   80 SER=0.0                                                           GEN:0630
      WRITE(2'1)SER                                                     GEN:0640
   81 READ(2'1) SER                                                     GEN:0650
      SER=SER+1.                                                        GEN:0660
      WRITE(3,82) SER                                                   GEN:0670
   82 FORMAT(' SERIAL NUMBER=',F10.0)                                   GEN:0680
      WRITE'2'1) SER                                                    GEN:0690
      WRITE(3,5)                                                        GEN:0700
    5 FORMAT(1H1,50X,'SELECTED INPUT DATA')                             GEN:0710
C                                                                       GEN:0720
C*******************     PRINT OUT INITIAL CONDITIONS                  GEN:0730
C                                                                       GEN:0740
      WRITE(3,6)                                                        GEN:0750
    6 FORMAT(////,1X,'INITIAL CONDITIONS',/)                            GEN:0760
      WRITE (3,1100)                                                    GEN:0770
 1100 FORMAT ('       T          THEIC      XNPMD      Y          Z     RGEN:0780
     1V         TEMP')                                                 GEN:0790
      WRITE(3,215) T,THEIC,XNPMD,Y,Z,RV,TEMP                            GEN:0800
  215 FORMAT(8F10.4)                                                    GEN:0810
      WRITE (3,1200)                                                    GEN:0820
 1200 FORMAT ('       FGI        XNPM       FIP        XNP        VS    XGEN:0830
     1S         AI         AIDO')                                      GEN:0840
      WRITE (3,216) FGI,XNPM,FIP,XNP,VS,XS,AI,AIDO                      GEN:0850
  216 FORMAT(8F10.4)                                                    GEN:0860
C                                                                       GEN:0870
C*******************     PRINT OUT CONTROL DATA                        GEN:0880
C                                                                       GEN:0890
      WRITE(3,10)                                                       GEN:0900
   10 FORMAT(///,1X,'CONTROL DATA')                                     GEN:0910
      WRITE(3,11) ICNTR,NON,DT,TICE,TEND,TIMSC                          GEN:0920
   11 FORMAT(/,1X,'ICNTR=',I5,'        NON  =',I5,6X,'DT   =',F10.4,' TICGEN:0930
     1E=',F10.4,' TEND =',F10.4,' TIMSC=',F10.4)                        GEN:0940
      WRITE(3,55) THET1,TAU1,TAU2                                       GEN:0950
   55 FORMAT(1X,'THET1=',F10.4,'   TAU1 =',F10.4,' TAU2=',F10.4)        GEN:0960
      WRITE(3,13) SY1,SY2,SY3,SY4,SY5                                   GEN:0970
   13 FORMAT(1X,'SY1  =',F10.7,'   SY2  =',F10.7,' SY3 =',F10.7,' SY4 ='GEN:0980
     1,F10.7,' SY5 =',F10.7)                                            GEN:0990
      WRITE(3,14) SY6,SY7,SY8,SY9,SY10                                  GEN:1000
   14 FORMAT(1X,'SY6  =',F10.7,'   SY7  =',F10.7,' SY8 =',F10.7,' SY9 ='GEN:1010
     1,F10.7,' SY10=',F10.7)                                            GEN:1020
      WRITE(3,115) SY11,SY12,SX                                         GEN:1030
  115 FORMAT(1X,'SY11 =',F10.7,'   SY12 =',F10.7,' SX  =',F10.7)        GEN:1040
C*******************                                                   GEN:1050
C*******************     DETERMINE WHAT DATA IS TO BE PRINTED OUT      GEN:1060
C*******************                                                   GEN:1070
```

```
      CALL DATSW (4,KEY4)                                                GENI1080
      GO TO (1191,1151), KEY4                                            GENI1090
 1151 WRITE(1,15)                                                        GENI1100
   15 FORMAT('TURN ON APPROPRIATE SWITCHES')                             GENI1110
      WRITE(1,116)                                                       GENI1120
  116 FORMAT('    ICE DATA- 10 ')                                        GENI1130
      WRITE(1,16)                                                        GENI1140
   16 FORMAT('    PROP DATA- 11 ')                                       GENI1150
      WRITE(1,17)                                                        GENI1160
   17 FORMAT('    HULL DATA- 12 ')                                       GENI1170
      WRITE(1,18)                                                        GENI1180
   18 FORMAT('    ELECTRICAL DATA- 13 ')                                 GENI1190
      WRITE(1,19)                                                        GENI1200
   19 FORMAT('    GOVERNOR AND BRIDGE CONTROLLER DATA- 14 ')             GENI1210
      WRITE(1,119)                                                       GENI1220
  119 FORMAT('    INERTIAS- 15 ')                                        GENI1230
 1191 WRITE(1,1192)                                                      GENI1240
 1192 FORMAT(/////,' TURN ON SELECTOR SWITCHES AND PRESS START')         GENI1250
      PAUSE                                                              GENI1260
      CALL DATSW(11,I)                                                   GENI1270
      IF(I-1) 30,20,30                                                   GENI1280
C*******************                                                    GENI1290
C*******************     PRINT OUT PROPELLER DATA                       GENI1300
C*******************                                                    GENI1310
   20 WRITE(3,21)                                                        GENI1320
   21 FORMAT(///,1X,'PROP DATA')                                         GENI1330
      WRITE(3,22) IZB,DAR,EO,DCT                                         GENI1340
   22 FORMAT(/,1X,'IZB=',I5,'          DAR=',F10.4,'    EO=',F10.4,'    GENI1350
     1DCT=',F10.4)                                                       GENI1360
      WRITE(3,23) PDR,D,EK                                               GENI1370
   23 FORMAT(1X,'PDR=',F10.4,'     D  =',F10.4,'    EK=',F10.4)         GENI1380
      WRITE(3,24)                                                        GENI1390
   24 FORMAT(9X,'B',20X,'BB',18X,'E',19X,'EE')                           GENI1400
      DO 26 I=1,32                                                       GENI1410
      WRITE(3,25) B(I),BB(I),E(I),EE(I)                                  GENI1420
   25 FORMAT(1X,F10.4,11X,F10.4,11X,F10.4,11X,F10.4)                     GENI1430
   26 CONTINUE                                                           GENI1440
      DO 28 I=33,64                                                      GENI1450
      WRITE(3,27) B(I),BB(I)                                             GENI1460
   27 FORMAT(1X,F10.4,11X,F10.4)                                         GENI1470
   28 CONTINUE                                                           GENI1480
   30 CALL   ,TSW(12,J)                                                  GENI1490
      IF(J-1) 40,31,40                                                   GENI1500
C*******************                                                    GENI1510
C*******************     PRINT OUT HULL DATA                            GENI1520
C*******************                                                    GENI1530
   31 WRITE(3,32)                                                        GENI1540
   32 FORMAT(///,1X,'HULL DATA')                                         GENI1550
      WRITE(3,33) CAD,DELTA,XKK                                          GENI1560
   33 FORMAT(/,1X,'CAD=',F10.4,' DELTA=',F10.1,' XKK=',F10.0)            GENI1570
      WRITE(3,34)                                                        GENI1580
   34 FORMAT(3X,'VK',15X,'EHP',20X,'WADED',20X,'THDED')                  GENI1590
      DO 37 I=1,20                                                       GENI1600
      K=I-1                                                              GENI1610
```

```
      WRITE(3,36) K,EMP(I),WADED(I),THDED(I)                            GENI1620
   36 FORMAT(3X,I2,10X,F10.4,13X,F10.4,13X,F10.4)                       GENI1630
   37 CONTINUE                                                          GENI1640
   40 CALL DATSW(12,KEY12)                                              GENI1650
      GO TO(41,50),KEY12                                                GENI1660
C*******************                                                   GENI1670
C*******************     PRINT OUT ELECTRICAL DATA                       GENI1680
C*******************                                                   GENI1690
   41 WRITE(3,42)                                                       GENI1700
   42 FORMAT(///,1X,'ELECTRICAL DATA')                                  GENI1710
      WRITE(3,43) RM,RG,TRNFG,ZGP                                       GENI1720
   43 FORMAT(/,1X,'RM   =',F10.6,'  RG   =',F10.6,' TRNFG=',F10.1,' ZGP GENI1730
     1=',F10.2)                                                         GENI1740
      WRITE(3,44) RFG,TRNFM,ZMP,RFM                                     GENI1750
   44 FORMAT(1X,'RFG  =',F10.6,'  TRNFM=',F10.2,' ZMP  =',F10.2,' RFM ='GENI1760
     1,F10.6)                                                           GENI1770
      WRITE(3,45) FIMAX,C1,XK1                                          GENI1780
   45 FORMAT(1X,'FIMAX=',F10.4,'  C1   =',F10.4,' XK1  =',F10.4)        GENI1790
      WRITE(3,46) CSHFR,TAUB,FMIRV,TAUA                                 GENI1800
   46 FORMAT(1X,'CSHFR=',F10.4,'  TAUB =',F10.4,' FMIRV=',F10.4,' TAUA='GENI1810
     1,F10.4)                                                           GENI1820
      WRITE(3,47) TRNSF,RX,XGEN,XLA,SPGEN,SPMOT                         GENI1830
   47 FORMAT(1X,'TRNSF=',F10.4,'  RX   =',F10.4,' XGEN =',F10.4,' XLA ='GENI1840
     1,F10.4,' SPGEN=',F10.4,' SPMOT=',F10.4)                           GENI1850
      WRITE(3,48)                                                       GENI1860
   48 FORMAT(6X,'CFGA',16X,'CFVA')                                      GENI1870
      DO 49 I=1,13                                                      GENI1880
      WRITE(3,419) CFGA(I),CFVA(I)                                      GENI1890
  419 FORMAT(1X,F10.4,10X,F10.4)                                        GENI1900
   49 CONTINUE                                                          GENI1910
   50 CALL DATSW(14,L)                                                  GENI1920
      IF(L-1) 60,51,60                                                  GENI1930
C*******************                                                   GENI1940
C*******************     PRINT OUT GOVERNOR AND DIESEL DATA              GENI1950
C*******************                                                   GENI1960
   51 WRITE(3,52)                                                       GENI1970
   52 FORMAT(///,1X,'GOVERNOR AND DIESEL DATA')                         GENI1980
      WRITE(3,53) ALP1,ALP2,ALP3,ALPD                                   GENI1990
   53 FORMAT(/,1X,'ALP1 =',F10.4,'  ALP2=',F10.4,' ALP3=',F10.4,'  ALPD=GENI2000
     1',F10.4)                                                          GENI2010
      WRITE(3,54) XK6,QBRG,CWM                                          GENI2020
   54 FORMAT(1X,'XK6  =',F10.2,'  QBRG=',F10.4,' CWM =',F10.4)          GENI2030
   60 CALL DATSW(15,M)                                                  GENI2040
      IF(M-1) 70,61,70                                                  GENI2050
C*******************                                                   GENI2060
C*******************     PRINT OUT SYSTEMS INERTIAS                      GENI2070
C*******************                                                   GENI2080
   61 WRITE(3,62)                                                       GENI2090
   62 FORMAT(///,1X,'SYSTEM INERTIAS')                                  GENI2100
      WRITE(3,63) XJPA,XJM,XJSH                                         GENI2110
   63 FORMAT(/,1X,'XJPA=',F10.2,'  XJM=',F10.2,'  XJSH=',F10.2)         GENI2120
      WRITE(3,64) XJD,XJG,XJPE                                          GENI2130
   64 FORMAT(1X,'XJD =',F10.2,'  XJG=',F10.2,'  XJPE=',F10.2)           GENI2140
   70 CALL DATSW(10,KEY10)                                              GENI2150
```

```
      GO TO (91,99), KEY10                                                  GENI2160
C*********************                                                     GENI2170
C*********************     PRINT ICE DATA                                  GENI2180
C*********************                                                     GENI2190
   91 WRITE(3,92)                                                         GENI2200
   92 FORMAT(///,1X,'ICE DATA')                                           GENI2210
      WRITE(3,93) BLOC1,DBB12,BLOC2,DBB21                                 GENI2220
   93 FORMAT(/,1X,'BLOC1=',F10.3,' DBB12=',F10.3,' BLOC2=',F10.3,' DBB21GENI2230
     1=',F10.3)                                                           GENI2240
      WRITE(3,94)                                                         GENI2250
   94 FORMAT(6X,'RBA')                                                    GENI2260
      DO 95 I=1,25                                                        GENI2270
      WRITE(3,96) RBA(I)                                                  GENI2280
   96 FORMAT(1X,F10.3)                                                    GENI2290
   95 CONTINUE                                                            GENI2300
      WRITE(3,97) DCR,DSH,CICE                                            GENI2310
   97 FORMAT(1X,'DCR=',F10.3,' DSH=',F10.3,' CICE=',F10.3)                GENI2320
      WRITE(3,98) CK1,CK2,CK3                                             GENI2330
   98 FORMAT(1X,'CK1=',F10.3,' CK2=',F10.3,' CK3 =',F10.3)                GENI2340
   99 CONTINUE                                                            GENI2350
      WRITE(3,71)                                                         GENI2360
   71 FORMAT(1H1,55X,'OUTPUT DATA',///)                                   GENI2370
C*********************                                                     GENI2380
C*********************     BEGIN SIMULTAION     ***************************GENI2390
C*********************                                                     GENI2400
      CALL LINK(EXEC)                                                     GENI2410
      END                                                                 GENI2420
// DUP                                                                    GENI2430
*STORE      WS  UA  GENIS                                                GENI2440
```

```
// FOR                                                                  REDA0 00
*LIST ALL                                                               REDA0010
*ONE WORD INTEGERS                                                      REDA0020
      SUBROUTINE REDAT                                                  REDA0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    REDA0040
C                                                                       REDA0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJW,XJSH,XJD,XJG,EFG,             REDA0060
     1 XJWB,XJDG,EIWAX,RA,RG,C1,XK1,GBRG,CSHFR,CWX,PCMS,XLA,            REDA0070
     2 TAU1,TAU2,XK4,ALP1,ALP2,ALP3,ALPD,XMASS                          REDA0080
      COMMON NOVAR,MVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           REDA0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   REDA0100
      COMMON DT,CT,T,TEND,TIVSC                                         REDA0110
      COMMON Z,Y,XNPW,EGI,EIP,XNP,VS,XS,SIE,AI,RV,TEMP                  REDA0120
      COMMON CZ,CY,CXNPW,CEGI,CEIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      REDA0130
      COMMON ZDO,YDO,XNPWD,EGIDO,EIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TSEDA0140
     1 EMDO                                                             REDA0150
      COMMON XNE,XNEDO,VA,THETC,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,REDA0160
     1QW,QERV,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    REDA0170
      COMMON CEG,SPGEN,CEGA(13),TRNEG,ZGP,REG,XLEG,TAUEG,TRNSE,RX,XW,XLAREDA0180
     1SE,AMBTR,XGEN                                                     REDA0190
      COMMON CEW,SPWOT,CEWA(13),TRNEW,ZWP,REW,XLEW,TAUEW,EMIRV,TAUA,TAUBREDA0200
      COMMON B(64),BB(64),E(32),EE(32)                                  REDA0210
      COMMON FHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  REDA0220
      COMMON IELO,JELO,KELO,LELO,MELO                                   REDA0230
      COMMON SPARE(5),ISPAR(5)                                          REDA0240
      COMMON CAD,DELTA,RM                                               REDA0250
      COMMON BBA(25),BBMX,BLOC1,DBB12,BLOC2,DBB21                       REDA0260
      COMMON EO,EK,DCT,DCR,DSH,CICE,TICE                                REDA0270
      COMMON CK1,CK2,CM3                                                REDA0280
C                                                                       REDA0290
C                          FORMAT STATEMENTS FOR READING DATA           REDA0300
  200 FORMAT (I5,5X,6F10.4)                                             REDA0310
  201 FORMAT(2(10X,2F10.4))                                             REDA0320
  202 FORMAT(7F10.4)                                                    REDA0330
  204 FORMAT(10X,3F10.4)                                                REDA0340
  207 FORMAT(2(10X,F10.4))                                              REDA0350
  214 FORMAT(7F10.7)                                                    REDA0360
  221 FORMAT(2I5,4F10.4)                                                REDA0370
  230 FORMAT(7F10.3)                                                    REDA0380
 2066 FORMAT(F10.4,F10.7)                                               REDA0390
 4000 FORMAT(4F10.6)                                                    REDA0400
C                                                                       REDA0410
C*********************                                                 REDA0420
C*********************       PROPELLER DATA                            REDA0430
C*********************                                                 REDA0440
C                          READ DATA CARD 1                             REDA0450
    1 READ (2,200) IZB,DAR,PDR,D,EC,EK,DCT                              REDA0460
C                          READ DATA CARDS 2 - 33                       REDA0470
      DO 100 I=1,64,2                                                   REDA0480
  100 READ(2,201) B(I),BB(I),B(I+1),BB(I+1)                             REDA0490
C                          READ DATA CARDS 34 - 49                      REDA0500
      DO 101 I=1,32,2                                                   REDA0510
  101 READ(2,201) E(I),EE(I),E(I+1),EE(I+1)                             REDA0520
C*********************                                                 REDA0530
```

```
C*******************     HULL DATA                                          REDA0540
C*******************                                                        REDA0550
C                         READ DATA CARD 50                                REDA0560
      READ (2,203) CAD,DELTA,XKK                                           REDA0570
C                         READ DATA CARDS 51 - 70                          REDA0580
      DO 102 I=1,20                                                        REDA0590
  102 READ(2,204) EHP(I),WADED(I),THDED(I)                                 REDA0600
C*******************                                                        REDA0610
C*******************     ELECTRICAL DATA                                    REDA0620
C*******************                                                        REDA0630
C                         READ DATA CARDS 71 AND 72                        REDA0640
      READ(2,4000) RM,TRNFM,ZMP,REM,RG,TRNFG,ZGP,RFG                       REDA0650
C                         READ DATA CARD 73                                REDA0660
      READ(2,203) FIMAX,C1,XK1,CSHFR,TRNSF,RX                              REDA0670
C                         READ DATA CARD 74                                REDA0680
      READ(2,203) FMIRV,TAUA,TAUB                                          REDA0690
C                         READ DATA CARD 75                                REDA0700
      READ(2,2066) XGEN,XLA                                                REDA0710
C                         READ DATA CARD 76                                REDA0720
      READ(2,203) SPGEN,SPMOT                                              REDA0730
C                         READ DATA CARDS 77 - 89                          REDA0740
      DO 103 I=1,13                                                        REDA0750
  103 READ(2,207) CFGA(I),CFMA(I)                                          REDA0760
C*******************                                                        REDA0770
C*******************     GOVERNOR - DIESEL DATA                             REDA0780
C*******************                                                        REDA0790
C                         READ DATA CARD 90                                REDA0800
      READ(2,203) ALP1,ALP2,ALP3,ALPD,XK6,QBRG,CWM                         REDA0810
C*******************                                                        REDA0820
C*******************     SYSTEMS INERTIAS                                   REDA0830
C*******************                                                        REDA0840
C                         READ DATA CARD 91                                REDA0850
      READ(2,203) XJPA,XJM,XJSH,XJD,XJG                                    REDA0860
C*******************                                                        REDA0870
C*******************     ICE CONDITIONS                                     REDA0880
C*******************                                                        REDA0890
C                         READ DATA CARDS 92 - 96                          REDA0900
      DO 104 I=1,25,5                                                      REDA0910
  104 READ(2,230) RBA(I),RBA(I+1),RBA(I+2),RBA(I+3),RBA(I+4)               REDA0920
C                         READ DATA CARD 97                                REDA0930
      READ(2,230) BLOC1,DBB12,BLOC2,DBB21,CICE,DCR,DSH                     REDA0940
C*******************                                                        REDA0950
C*******************     SCALE FACTORS, INITIAL CONDITIONS, CONTROL DATA    REDA0960
C*******************                                                        REDA0970
C                         READ DATA CARD 98                                REDA0980
      READ(2,214) SY1,SY2,SY3,SY4,SY5,SY6,SY7                              REDA0990
C                         READ DATA CARD 99                                REDA1000
      READ(2,214) SY8,SY9,SY10,SY11,SY12,SX                                REDA1010
C                         READ DATA CARD 100                               REDA1020
      READ(2,203) T,THEIC,XNPMD,Y,Z,RV,TEMP                                REDA1030
C                         READ DATA CARD 101                               REDA1040
      READ(2,203) FGI,XNPM,FIP,XNP,VS,XS                                   REDA1050
C                         READ DATA CARD 102                               REDA1060
      READ(2,203) AI,AIDO                                                  REDA1070
```

```
C                          READ DATA CARD 103                                   REDA1080
      READ(2,203) THET1,TAU1,TAU2,CK1,CK2,CK3                                   REDA1090
C                          READ DATA CARD 104 (THE LAST DATA CARD.)             REDA1100
  220 READ(2,221) ICNTR,NON,DT,TIMSC,TICE,TEND                                  REDA1110
C                                                                               REDA1120
C                                                                               REDA1130
C                          SET CONSTANTS 'RHO' AND 'PI'                         REDA1140
      RO=1.99                                                                   REDA1150
      PI=3.14159                                                                REDA1160
C                          COMPUTE 'RA' THUS LEAVING VARIABLE 'RM' OPEN         REDA1170
C                          FOR USE AS LOCAL VARIABLE IN SUBROUTINE QICE         REDA1180
      RA=RM+RG/XGEN                                                             REDA1190
C                          COMPUTE 'XMASS' TERM FOR THRUST COMPUTATION          REDA1200
      XMASS=(1.+CAD)*2240.*DELTA/32.18                                          REDA1210
C                                                                               REDA1220
      RETURN                                                                    REDA1230
      END                                                                       REDA1240
// DUP                                                                          REDA1250
*STORE      WS  UA  REDAT                                                       REDA1260
```

```
// FOR                                                                  POLM0 00
*LIST ALL                                                               POLM0010
*ONE WORD INTEGERS                                                      POLM0020
      SUBROUTINE POLMI                                                  POLM0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    POLM0040
C                                                                       POLM0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             POLM0060
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,OBRG,CSHFR,CWM,PCMS,XLA,            POLM0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          POLM0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           POLM0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   POLM0100
      COMMON DT,CT,T,TEND,TIMSC                                         POLM0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  POLM0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      POLM0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TPOLM0140
     1 EMDO                                                             POLM0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,POLM0160
     1QV,QERM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    POLM0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAPOLM0180
     1SF,AMPTR,XGEN                                                     POLM0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBPOLM0200
      COMMON B(64),BB(64),E(32),EE(32)                                  POLM0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  POLM0220
      COMMON IELO,JELO,KELO,LELO,MELO                                   POLM0230
      COMMON SPARE(5),ISPAR(5)                                          POLM0240
C                                                                       POLM0250
      EQUIVALENCE (XJPE,PCMS)                                           POLM0260
C                                                                       POLM0270
C                          BRANCH ON NUMBER OF PROPELLER BLADES         POLM0280
      GO TO (3,3,3,4),IZB                                               POLM0290
C                          3 PROPELLER BLADES                           POLM0300
    3 XJPE=(1.024*((1.37*DAR*PDR)-0.30)*(D/1.33)**5)/32.18              POLM0310
      GO TO 20                                                          POLM0320
C                          4 PROPELLER BLADES                           POLM0330
    4 XJPE=(1.024*((1.09*DAR*PDR)-0.23)*(D/1.33)**5)/32.18              POLM0340
C                          COMPUTE 'XJPA' IF A VALUE OF ZERO WAS READ IN POLM0350
   20 IF (XJPA) 10,5,10                                                 POLM0360
    5 XJPA=52.*D**2                                                     POLM0370
C                          COMPUTE REMAINING INERTIAS                   POLM0380
   10 XJMP=XJM+XJSH+XJPA+XJPE                                           POLM0390
      XJDG=(XJD+XJG)*XGEN                                               POLM0400
      RETURN                                                            POLM0410
      END                                                               POLM0420
// DUP                                                                  POLM0430
*STORE      WS  UA  POLMI                                               POLM0440
```

```
// FOR                                                                    EXEC0 00
*LIST ALL                                                                 EXEC0010
*NAME EXEC                                                                EXEC0020
*ONE WORD INTEGERS                                                        EXEC0030
*IOCS(CARD,TYPEWRITER,KEYBOARD,1132PRINTER,DISK)                          EXEC0040
C      ICEBREAKER PROPULSION PLANT SIMULATION                             EXEC0050
C      EXECUTIVE ROUTINE                                                  EXEC0060
C      VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                     EXEC0070
C                                                                         EXEC0080
       COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,              EXEC0090
      1 XJVP,XJDG,FIMAX,PA,RG,C1,XK1,QBRG,CSHFR,CWM,PCMS,XLA,             EXEC0100
      2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                           EXEC0110
       COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP            EXEC0120
       COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   EXEC0130
       COMMON DT,CT,T,TEND,TIMSC                                         EXEC0140
       COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  EXEC0150
       COMMON CZ,CY,CXNPM,CTGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      EXEC0160
       COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TEXEC0170
      1 EMDO                                                              EXEC0180
       COMMON XNE,XNEDO,VA,THET0,THET1,THETA,THETD,THEIC,GDE,GG,EG,EB,EC,EXEC0190
      1CM,QERM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                     EXEC0200
       COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAEXEC0210
      1SF,AMPTR,XGEN                                                      EXEC0220
       COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBEXEC0230
       COMMON B(64),BB(64),E(32),EE(32)                                   EXEC0240
       COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                   EXEC0250
       COMMON IELO,JELO,KELO,LELO,NELO                                    EXEC0260
       COMMON SPARE(5),ISPAR(5)                                           EXEC0270
       COMMON CAD,DELTA,RM                                                EXEC0280
       COMMON BBA(25),RBMX,BLOC1,DBB12,BLOC2,DBB21                        EXEC0290
       COMMON F0,FK,DCT,DCR,DSH,CICE,TICE                                 EXEC0300
       COMMON CK1,CK2,CK3                                                 EXEC0310
C                                                                         EXEC0320
       EQUIVALENCE (FMI,SPARE(1))                                         EXEC0330
C                                                                         EXEC0340
C      SPARE(1)--FMI                                                      EXEC0350
C      SPARE(2)--XNGR                                                     EXEC0360
C      SPARE(3)--XNGRD                                                    EXEC0370
C      SPARE(4)--VACANT                                                   EXEC0380
C      SPARE(5)--VACANT                                                   EXEC0390
C      ISPAR(1)--CONTROLS CALLING OF GENIS FROM EXEC                      EXEC0400
C      ISPAR(2)--CONTROLS MOTOR FIELD PLUGGING IN CALVR                   EXEC0410
C      ISPAR(3)--VACANT                                                   EXEC0420
C      ISPAR(4)--VACANT                                                   EXEC0430
C      ISPAR(5)--VACANT                                                   EXEC0440
C                                                                         EXEC0450
       DEFINE FILE 1(600,26,U,IELO)                                       EXEC0460
C                                                                         EXEC0470
C      INITIALIZE CONTROL VARIABLES                                       EXEC0480
C                        SET DISK WRITE TIME STEP CONTROL                 EXEC0490
1      III=1                                                              EXEC0500
C                        SET PRINTER STEP COUNTER                         EXEC0510
       MIKE=1                                                             EXEC0520
C                        SET DISK FILE INDEX                              EXEC0530
```

```
      IELO=1                                                                EXEC0540
C                          SET ICE CONTROL                                  EXEC0550
      JELO=1                                                                EXEC0560
      KELO=1                                                                EXEC0570
C                          SET MOTOR FIELD PLUG                             EXEC0580
      ISPAR(2)=1                                                            EXEC0590
C                          SAVE DISK WRITE TIME STEP                        EXEC0600
      DTT=TIMSC                                                             EXEC0610
C                          COMPUTE NEXT TIME FOR WRITING ON DISK            EXEC0620
      TTT=T+TIMSC                                                           EXEC0630
C                                                                           EXEC0640
C     COMPUTE INITIAL VALUES                                                EXEC0650
      CALL CNTRL                                                            EXEC0660
C     PRINT OUT INITIAL VALUES                                              EXEC0670
      CALL OUT                                                              EXEC0680
C     CHECK IF DATA SWITCHES ARE TO BE INTERROGATED                         EXEC0690
 50   CALL DATSW (0,KEY0)                                                   EXEC0700
      GO TO (40,1300), KEY0                                                 EXEC0710
C     CALL NUMERICAL INTEGRATION ROUTINE TO UPDATE VALUES                   EXEC0720
 1300 CALL HEUN                                                             EXEC0730
C     CHECK FOR PRINTING OUT OF DATA                                        EXEC0740
    7 IF (MIKE-NON) 31,32,32                                                EXEC0750
   32 CALL OUT                                                              EXEC0760
      MIKE=0                                                                EXEC0770
   31 MIKE=MIKE+1                                                           EXEC0780
C     CHECK FOR END OF SIMULATION                                           EXEC0790
      IF (CT-(TEND+DTT)) 51,51,40                                           EXEC0800
C     CHANGE DISK WRITE INTERVAL IF TIME GREATER THAN 20 SECONDS            EXEC0810
 51   IF (CT-20.) 30,30,10                                                  EXEC0820
 10    GO TO (11,30),III                                                    EXEC0830
   11 DTT=1.0                                                               EXEC0840
      TTT=TTT+DTT-TIMSC                                                     EXEC0850
      III=2                                                                 EXEC0860
   30 IF (CT-TTT) 50,55,55                                                  EXEC0870
C     CONVERT VALUES TO UNITS FOR WRITING ON DISK                           EXEC0880
 55   RPMPM=60.*XNPM                                                        EXEC0890
      RPMP=60.*XNP                                                          EXEC0900
      CVK=CVS/1.689                                                         EXEC0910
      XKW=.746*CVK*2.*PI*XNP/550.                                           EXEC0920
C     WRITE VALUES ON DISK                                                  EXEC0930
      WRITE (1'IELO) CT,AI,EB,RPMPM,FMI,RPMP,XKW,CVK,QPI,XPROI,CXS          EXEC0940
     1,QI,TEMP                                                              EXEC0950
C     UPDATE TIME FOR NEXT DISK WRITING                                     EXEC0960
      TTT=TTT+DTT                                                           EXEC0970
      GO TO 50                                                              EXEC0980
C     INTERROGATE DATA SWITCHES                                             EXEC0990
   40 CALL FINIS                                                            EXEC1000
C     BEGIN NEW SIMULATION IF SO DIRECTED                                   EXEC1010
      JENE=ISPAR(1)                                                         EXEC1020
      GO TO (1300,45), JENE                                                 EXEC1030
 45   CALL LINK(GENIS)                                                      EXEC1040
      END                                                                   EXEC1050
// DUP                                                                      EXEC1060
*STORE      WS  UA  EXEC                                                    EXEC1070
```

```
// FOR                                                                    CNTR0000
*LIST ALL                                                               CNTR0010
*ONE WORD INTEGERS                                                      CNTR0020
      SUBROUTINE CNTRL                                                  CNTR0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    CNTR0040
C                                                                       CNTR0050
      COMMON RO,PI,IZR,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             CNTR0060
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHFR,CWM,PCMS,XLA,            CNTR0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          CNTR0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           CNTR0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   CNTR0100
      COMMON DT,CT,T,TEND,TIMSC                                         CNTR0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  CNTR0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      CNTR0130
      COMMON ZDO,YDO,XNPMO,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TCNTR0140
     1 EMDO                                                             CNTR0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,OG,EG,EB,EC,CNTR0160
     1QV,QFRM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    CNTR0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLACNTR0180
     1SF,AMPTR,XGEN                                                     CNTR0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBCNTR0200
      COMMON B(64),BB(64),E(32),EE(32)                                  CNTR0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  CNTR0220
      COMMON IELO,JELO,KELO,LELO,MELO                                   CNTR0230
      COMMON SPARE(5),ISPAR(5)                                          CNTR0240
      COMMON CAD,DELTA,RM                                               CNTR0250
      COMMON BBA(25),RBMX,BLOC1,DBB12,BLOC2,DBB21                       CNTR0260
      COMMON E0,EK,DCT,DCR,DSH,CICE,TICE                                CNTR0270
      COMMON CK1,CK2,CK3                                                CNTR0280
C                                                                       CNTR0290
      EQUIVALENCE (FMI,SPARE(1))                                        CNTR0300
C                                                                       CNTR0310
C     RESET ANGULAR POSITION OF PROPELLER BLADE                         CNTR0320
      SIE=0.                                                            CNTR0330
C     BRANCH ACCORDING TO MANEUVER TO SET FINAL BRIDGE LEVER POSITION   CNTR0340
      GO TO(10,20,21,22), ICNTR                                         CNTR0350
C                                                                       CNTR0360
C     AHEAD FLANK                                                       CNTR0370
   10 THETO=165.                                                        CNTR0380
      GO TO 30                                                          CNTR0390
C                                                                       CNTR0400
C     BACK FLANK                                                        CNTR0410
   20 THETO=-165.                                                       CNTR0420
      GO TO 30                                                          CNTR0430
C                                                                       CNTR0440
C     AHEAD FULL                                                        CNTR0450
   21 THETO=135.                                                        CNTR0460
      GO TO 30                                                          CNTR0470
C                                                                       CNTR0480
C     AHEAD FULL IN ICE                                                 CNTR0490
   22 THETO=135.                                                        CNTR0500
      GO TO 30                                                          CNTR0510
C                                                                       CNTR0520
   30 FMI=FIMAX-FIP                                                     CNTR0530
```

```
C     COMPUTE INITIAL VALUES OF VARIABLES                                    CNTR0540
      CALL CALVR                                                             CNTR0550
C     SAVE INITIAL VALUES AS CURRENT VALUES                                  CNTR0560
      CT=T                                                                   CNTR0570
      CAI=AI                                                                 CNTR0580
      CY=Y                                                                   CNTR0590
      CZ=Z                                                                   CNTR0600
      CXNPM=XNPM                                                             CNTR0610
      CFIP=FIP                                                               CNTR0620
      CFGI=FGI                                                               CNTR0630
      CXNP=XNP                                                               CNTR0640
      CVS=VS                                                                 CNTR0650
      CXS=XS                                                                 CNTR0660
      CSIE=SIE                                                               CNTR0670
C     PRINT COLUMN HEADINGS FOR OUTPUT DATA                                  CNTR0680
    1 WRITE (3,201)                                                          CNTR0690
  201 FORMAT (1X,'T(SEC) RPMPM CFGI   RPMP   VK  QDE(FT-LB)  QG    EB(V)CNTR0700
     1  FMI   EC    QM(FT-LB)   AI(A)   QP   THRUST(T)  RESIST   HP/SH')CNTR0710
C     CONVERT TO APPROPRIATE UNITS FOR DISK WRITING                          CNTR0720
      RPMPM=60.*XNPM                                                         CNTR0730
      RPMP=60.*XNP                                                           CNTR0740
      CVK=CVS/1.689                                                          CNTR0750
      XKW=.746*QM*2.*PI*XNP/550.                                             CNTR0760
C     WRITE INITIAL VALUES ON DISK                                           CNTR0770
      WRITE (1'IELO) CT,AI,EB,RPMPM,FMI,RPMP,XKW,CVK,QPI,XPROI,CXS           CNTR0780
     1,QI,TEMP                                                               CNTR0790
      RETURN                                                                 CNTR0800
      END                                                                    CNTR0810
// DUP                                                                       CNTR0820
*STORE      WS  UA  CNTRL                                                    CNTR0830
```

```
// FOR                                                                    HEUN0 00
*LIST ALL                                                               HEUN0010
*ONE WORD INTEGERS                                                      HEUN0020
      SUBROUTINE HEUN                                                   HEUN0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    HEUN0040
      DIMENSION STVR(12), CSTVR(12),STVRD(12),STVR0(12),STVR1(12),STVR2(HEUN0050
     112)                                                               HEUN0060
      COMMON RO,PI,IZR,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             HEUN0070
     1 XJVP,XJDG,FIMAX,RA,RG,C1,XK1,GBRG,CSHER,CWM,PCMS,XLA,            HEUN0080
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          HEUN0090
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           HEUN0100
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   HEUN0110
      COMMON DT,CT,T,TEND,TIMSC                                         HEUN0120
      COMMON Z,Y,XNPV,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  HEUN0130
      COMMON CZ,CY,CXNPV,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      HEUN0140
      COMMON ZDO,YDO,XNPVD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,THEUN0150
     1 EMDO                                                             HEUN0160
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,OG,EG,EB,EC,HEUN0170
     1OM,OFRM,AIHAT,OP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    HEUN0180
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAHEUN0190
     1SF,AMPTR,XGEN                                                     HEUN0200
      COMMON CFY,SPMOT,CFYA(13),TRNFY,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBHEUN0210
      COMMON B(64),BB(64),E(32),EE(32)                                  HEUN0220
      COMMON CHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  HEUN0230
      COMMON IFLO,JFLO,KFLO,LFLO,MFLO                                   HEUN0240
      COMMON SPARE(5),ISPAR(5)                                          HEUN0250
      EQUIVALENCE (Z,STVR(1))                                           HEUN0260
      EQUIVALENCE (CZ,CSTVR(1))                                         HEUN0270
      EQUIVALENCE (ZDO,STVRD(1))                                        HEUN0280
C     CSTVR ARRAY REPRESENTS OLD VALUES                                 HEUN0290
C     STVRD ARRAY REPRESENTS DERIVATIVES                                HEUN0300
C     STVR ARRAY REPRESENTS CURRENT SOLUTION IN INTEGRATION, UPDATED    HEUN0310
C     STVR0 ARRAY WILL CONTAIN VALUES OF FIRST ORDER                    HEUN0320
C     STVR1 ARRAY WILL CONTAIN VALUES OF SECOND ORDER                   HEUN0330
C     STVR2 ARRAY WILL CONTAIN VALUES OF THIRD ORDER                    HEUN0340
C         AS EACH NEW STEP IS CALCULATED                                HEUN0350
      N=12                                                              HEUN0360
      DD=DT/2.                                                          HEUN0370
      DO 10 I=1,N                                                       HEUN0380
      STVR0(I)=STVRD(I)*DT                                              HEUN0390
   10 STVR(I)=CSTVR(I)+STVR0(I)/2.0                                     HEUN0400
      IJAX=1                                                            HEUN0410
      GO TO 100                                                         HEUN0420
  200 T=CT+DD                                                           HEUN0430
      CALL CALVR                                                        HEUN0440
      DO 20 I=1,N                                                       HEUN0450
      STVR1(I)=STVRD(I)*DT                                              HEUN0460
   20 STVR(I)=CSTVR(I)+STVR1(I)/2.0                                     HEUN0470
      IJAX=2                                                            HEUN0480
      GO TO 100                                                         HEUN0490
  300 T=CT+DD                                                           HEUN0500
      CALL CALVR                                                        HEUN0510
      DO 30 I=1,N                                                       HEUN0520
```

```
      STVR2(I)=STVRD(I)*DT                                                  HEUN0530
   30 STVR(I)=CSTVR(I)+STVR2(I)                                             HEUN0540
      IJAX=3                                                                HEUN0550
      GO TO 100                                                             HEUN0560
  400 T=CT+DT                                                               HEUN0570
      CALL CALVR                                                            HEUN0580
      DO 40 I=1,N                                                           HEUN0590
   40 STVR(I)=CSTVR(I)+(STVR0(I)+2.0*(STVR1(I)+STVR2(I))+STVRD(I)*DT)/6.    HEUN0600
      IJAX=4                                                                HEUN0610
C     RESET SERVO STROKE POSITION TO ZERO IF LESS THAN ZERO OR TO 0.6 IF    HEUN0620
C         GREATER THAN 0.6                                                  HEUN0630
  100 IF(Z) 162,163,163                                                     HEUN0640
  162 Z=0.                                                                  HEUN0650
      GO TO 16                                                              HEUN0660
  163 IF  Z-1.0  16,16,160                                                  HEUN0670
  160 Z 1.0                                                                 HEUN0680
C     SET MOTOR FIELD ADJUSTING CURRENT TO ZERO IF NEGATIVE                 HEUN0690
   16 IF(FIP) 172,172,173                                                   HEUN0700
  172 FIP=0.0                                                               HEUN0710
C     CONTINUE WITH SOLUTION                                                HEUN0720
  173 GO TO (200,300,400,500), IJAX                                         HEUN0730
C     SAVE NEW VALUES AS CURRENT VALUES                                     HEUN0740
  500 DO 50 I=1,N                                                           HEUN0750
   50 CSTVR(I)=STVR(I)                                                      HEUN0760
      CT=T                                                                  HEUN0770
C     UPDATE VALUES AND COMPUTE DERIVATIVES                                 HEUN0780
      CALL CALVR                                                            HEUN0790
      RETURN                                                                HEUN0800
      END                                                                   HEUN0810
// DUP                                                                      HEUN0820
*STORE      WS  UA  HEUN                                                    HEUN0830
```

```
// FOR                                                                  CALV0 00
*LIST ALL                                                               CALV0010
*ONE WORD INTEGERS                                                      CALV0020
      SUBROUTINE CALVR                                                  CALV0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    CALV0040
C                                                                       CALV0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             CALV0060
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHFR,CWM,PCMS,XLA,            CALV0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          CALV0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           CALV0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   CALV0100
      COMMON DT,CT,T,TEND,TIMSC                                         CALV0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  CALV0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      CALV0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TCALV0140
     1 EMDO                                                             CALV0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,CALV0160
     1VA,QFRM,AIMAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    CALV0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,REG,XLFG,TAUFG,TRNSF,RX,XM,XLACALV0180
     1SF,AMPTR,XGEN                                                     CALV0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,REM,XLFM,TAUFM,FMIRV,TAUA,TAUBCALV0200
      COMMON B(64),BB(64),E(32),EE(32)                                  CALV0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  CALV0220
      COMMON IELO,JELO,KELO,LELO,MELO                                   CALV0230
      COMMON SPARE(5),ISPAR(5)                                          CALV0240
      COMMON CAD,DELTA,RM                                               CALV0250
      COMMON RBA(25),RBMX,BLOC1,DBB12,BLOC2,DBB21                       CALV0260
      COMMON E0,EK,DCT,DCR,DSH,CICE,TICE                                CALV0270
      COMMON CK1,CK2,CK3                                                CALV0280
C                                                                       CALV0290
      EQUIVALENCE (XNGR,SPARE(2)),(XNGRD,SPARE(3)),(FMI,SPARE(1))       CALV0300
      EQUIVALENCE(IPLUG,ISPAR(2))                                       CALV0310
      EQUIVALENCE (FLOW,SPARE(5))                                       CALV0320
C                                                                       CALV0330
C                          COMPUTE ACTUAL AND TIME RATE OF BRIDGE       CALV0340
C                          LEVER POSITION                               CALV0350
      CALL CON1                                                         CALV0360
C                          COMPUTE ORDERED GENERATOR FIELD VOLTAGE,     CALV0370
C                          DIESEL SPEED, AND DIESEL SPEED RATE          CALV0380
      CALL CON2                                                         CALV0390
C                                                                       CALV0400
C                                                                       CALV0410
C                          CALCULATE DIESEL ERROR SPEED AND TIME RATE   CALV0420
      XNE=XNGR-XNPM                                                     CALV0430
      XNEDO=XNGRD-XNPMD                                                 CALV0440
C                          CALCULATE DEVELOPED DIESEL TORQUE            CALV0450
      CALL QDEV                                                         CALV0460
C                                                                       CALV0470
C                          ADJUST DEVELOPED DIESEL TORQUE FOR           CALV0480
C                          FRICTION TORQUE                              CALV0490
      QDE=QDE-(QBRG+CWM*XNPM)*XGEN                                      CALV0500
      AMPTR=TRNFG*FGI/2.-TRNSF*AI/(2.*XGEN)                             CALV0510
C                                                                       CALV0520
C                          CALCULATE GENERATOR FLUX, INDUCTANCE, AND    CALV0530
```

```
C                               TIVE CONSTANT                                   CALV0540
      CALL FLUXG                                                                CALV0550
      EG=60.*CFG*XNPM                                                           CALV0560
C                               CALCULATE MOTOR FIELD FLUX, INDUCTANCE, AND     CALV0570
C                               TIVE CONSTANT                                   CALV0580
      CALL FLUXM                                                                CALV0590
      EC=60.*CFM*XNP                                                            CALV0600
      DG=7.04*CFG*AI                                                            CALV0610
      DM=7.04*CFM*AI                                                            CALV0620
C                               CALCULATE MOTOR FIELD CURRENT                   CALV0630
      FB=FG-(RG/XGEN)*AI                                                        CALV0640
C                               CHECK FOR REVERSE DIRECTION                     CALV0650
      IF (THETA) 5,6,6                                                          CALV0660
    5 GO TO(7,8),IPLUG                                                          CALV0670
C                               SAVE INITIAL TIME AND MOTOR FIELD CURRENT       CALV0680
    7 TTN=T                                                                     CALV0690
      FMIIC=FMI                                                                 CALV0700
      IPLUG=2                                                                   CALV0710
    8 ABLE=1./EXP((T-TTN)/TAUA)                                                 CALV0720
      BAKER=1./EXP((T-TTN)/TAUB)                                                CALV0730
      FMI=FMIIC+FMIRV/(TAUA-TAUB)*(TAUA*(1.-ABLE)-TAUB*(1.-BAKER))              CALV0740
      GO TO 34                                                                  CALV0750
    6 IF(FB-C1) 10,10,20                                                        CALV0760
   20 AIMAT=XK1*(FB-C1)                                                         CALV0770
      FMI=FMAX-FIP                                                              CALV0780
      GO TO 30                                                                  CALV0790
   10 AIMAT=0.                                                                  CALV0800
   30 IF (FMI-84.0) 31,32,32                                                    CALV0810
   31 FMI=84.0                                                                  CALV0820
   32 IF (FMI-FMAX) 34,34,33                                                    CALV0830
   33 FMI=FMAX                                                                  CALV0840
   34 QFRM=CSHFR*ABS(XNP)*XNP                                                   CALV0850
C                               CALCULATE SHIP'S RESISTANCE, WAKE AND THRUST    CALV0860
C                               DEDUCTIONS                                      CALV0870
      CALL HULL                                                                 CALV0880
C                               CALCULATE ADVANCE VELOCITY OF PROPELLER         CALV0890
      VA=WDED*VS                                                                CALV0900
C                               CALCULATE IDEAL PROPELLER TORQUE AND            CALV0910
C                               THRUST PER PROPELLER                            CALV0920
      CALL PROP                                                                 CALV0930
C                               CALCULATE TOTAL PROPELLER THRUST                CALV0940
      QP=QPI                                                                    CALV0950
      TP=TDED*XPROI*XKK                                                         CALV0960
C                               CHECK IF SHIP HAS ENCOUNTERED ICE AND SET       CALV0970
C                               ICE TORQUE TO ZERO IF NOT                       CALV0980
      IF(CT-TICE)50,51,51                                                       CALV0990
   50 QI=0.0                                                                    CALV1000
      GO TO 52                                                                  CALV1010
C                               CALCULATE ICE TORQUE OF PROPELLER               CALV1020
   51 CALL QICE                                                                 CALV1030
C                               CALCULATE DERIVATIVES                           CALV1040
   52 CALL STATE                                                                CALV1050
      RETURN                                                                    CALV1060
      END                                                                       CALV1070
```

```
// DUP
*STORE  WS  UA  CALVR
```

CALV1080
CALV1090

```
// FOR                                                                  C1ON0 00
*LIST ALL                                                              C1ON0010
*ONE WORD INTEGERS                                                     C1ON0020
      SUBROUTINE CON1                                                  C1ON0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                   C1ON0040
C                                                                      C1ON0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,            C1ON0060
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,GBRG,CSHFR,CWM,PCMS,XLA,           C1ON0070
     2 TAU1,TAU2,XKG,ALP1,ALP2,ALP3,ALPD,XMASS                         C1ON0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP          C1ON0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12  C1ON0100
      COMMON DT,CT,T,TEND,TIMSC                                        C1ON0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                 C1ON0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP     C1ON0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TC1ON0140
     1 EMDO                                                            C1ON0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,GDE,GG,EG,EB,EC,C1ON0160
     1GY,GFRM,AIHAT,GP,GPI,GI,TP,XPROI,R,VAR,XKW,CVK                   C1ON0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAC1ON0180
     1SF,AMPTR,XGEN                                                    C1ON0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUSC1ON0200
      COMMON B(64),BB(64),E(32),EE(32)                                 C1ON0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                 C1ON0220
      COMMON IELO,JELO,KELO,LELO,VELO                                  C1ON0230
      COMMON SPARE(5),ISPAR(5)                                         C1ON0240
C                                                                      C1ON0250
C                          CHECK FOR RANGE ZERO TO TAU1 SECONDS        C1ON0260
      IF (T-TAU1) 5,5,10                                               C1ON0270
C                          TIME LESS THAN TAU1 SECONDS                 C1ON0280
    5 THETA=THEIC+(THET1-THEIC)*T/TAU1                                 C1ON0290
      THETD=(THET1-THEIC)/TAU1                                         C1ON0300
      GO TO 25                                                         C1ON0310
C                          CHECK FOR RANGE TAU1 TO TAU2 SECONDS        C1ON0320
   10 IF (T-(TAU1+TAU2)) 15,20,20                                      C1ON0330
C                          TIME BETWEEN TAU1 AND TAU2 SECONDS          C1ON0340
   15 THETA=THET1+(THETO-THET1)*(T-TAU1)/TAU2                          C1ON0350
      THETD=(THETO-THET1)/TAU2                                         C1ON0360
      GO TO 25                                                         C1ON0370
C                          TIME GREATER THAN OR EQUAL TO TAU2 SECONDS  C1ON0380
   20 THETA=THETO                                                      C1ON0390
      THETD=0.0                                                        C1ON0400
   25 RETURN                                                           C1ON0410
      END                                                              C1ON0420
// DUP                                                                 C1ON0430
*STORE      WS  UA  CON1                                               C1ON0440
```

```
// FOR                                                                  C2ON0 00
*LIST ALL                                                               C2ON0010
*ONE WORD INTEGERS                                                      C2ON0020
      SUBROUTINE CON2                                                   C2ON0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    C2ON0040
C                                                                       C2ON0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             C2ON0060
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHFR,CWX,PCMS,XLA,            C2ON0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          C2ON0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           C2ON0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   C2ON0100
      COMMON DT,CT,T,TEND,TIMSC                                         C2ON0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  C2ON0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      C2ON0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TC2ON0140
     1 EVDO                                                             C2ON0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,C2ON0160
     1QM,QERM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    C2ON0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAC2ON0180
     1OF,AMPTR,XGEN                                                     C2ON0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZVP,RFM,XLFM,TAUFM,EMIRV,TAUA,TAUBC2ON0200
      COMMON B(64),BB(64),E(32),EE(32)                                  C2ON0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  C2ON0220
      COMMON IELO,JELO,KELO,LELO,MELO                                   C2ON0230
      COMMON SPARE(5),ISPAR(5)                                          C2ON0240
      EQUIVALENCE (XNGR,SPARE(2)),(XNGRD,SPARE(3))                      C2ON0250
C                                                                       C2ON0260
C                          COMPUTE MAXIMUM EFG                          C2ON0270
      EFGM=34.7*(RX*RFG)                                                C2ON0280
C                          COMPUTE VALUES BASED ON ABSOLUTE VALUE       C2ON0290
C                          OF THETA                                     C2ON0300
      THET=ABS(THETA)                                                   C2ON0310
      IF(THET-63.)10,10,20                                              C2ON0320
   10 EFG=EFGM/63.*THET                                                 C2ON0330
      XNGR=300./60.                                                     C2ON0340
      XNGRD=0.0                                                         C2ON0350
      GO TO 49                                                          C2ON0360
   20 IF (THET-135.) 30,30,40                                           C2ON0370
   30 XNGR=(420./72.*THET+(720.-135./72.*420.))/60.                     C2ON0380
      XNGRD=420./(72.*60.)*THETD                                        C2ON0390
      EFG=EFGM                                                          C2ON0400
      GO TO 49                                                          C2ON0410
   40 XNGR=(90./30.*THET+(810.-165./30.*90.))/60.                       C2ON0420
      XNGRD=3./60.*THETD                                                C2ON0430
      EFG=EFGM                                                          C2ON0440
      GO TO 49                                                          C2ON0450
C                          CHECK FOR ORDERED POSITION OF BRIDGE CONTROL C2ON0460
C                          LEVER                                        C2ON0470
   49 IF(THET-ABS(THETO)) 496,495,495                                   C2ON0480
  495 XNGRD=0.0                                                         C2ON0490
C                          SET SIGN OF EFG TO THAT OF THETA             C2ON0500
  496 EFG=THETA/THET*EFG                                                C2ON0510
      RETURN                                                            C2ON0520
      END                                                               C2ON0530
```

```
C2DN0540
C2DN0550
// DUP
*STORE    WS  UA  CONZ
```

```
// FOR                                                                    FLXG0 00
*LIST ALL                                                               FLXG0010
*ONE WORD INTEGERS                                                      FLXG0020
      SUBROUTINE FLUXG                                                  FLXG0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    FLXG0040
C                                                                       FLXG0050
      COMMON RO,PI,IZE,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             FLXG0060
     1 XJVP,XJDG,FIMAX,RA,RG,C1,XK1,OBRG,CSHER,CWM,PCMS,XLA,            FLXG0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          FLXG0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           FLXG0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   FLXG0100
      COMMON DT,CT,T,TEND,TINSC                                         FLXG0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  FLXG0120
      COMMON CZ,CY,CXNPY,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      FLXG0130
      COMMON ZDO,YDO,XNPYD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TFLXG0140
     1 EMDO                                                             FLXG0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,QG,EG,EB,EC,FLXG0160
     1QV,QFRM,AIHAT,QP,CPI,QI,TP,XPROI,R,VAR,XKW,CVK                    FLXG0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAFLXG0180
     1SF,AMPTR,XGEN                                                     FLXG0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBFLXG0200
      COMMON B(64),BB(64),F(32),EE(32)                                  FLXG0210
      COMMON EMP(20),WDED,WADED(20),TDED,THDED(20),XKK                  FLXG0220
      COMMON IELO,JELO,KELO,LELO,VELO                                   FLXG0230
      COMMON SPARE(5),ISPAR(5)                                          FLXG0240
C                                                                       FLXG0250
C                          COMPUTE ABSCISSA                             FLXG0260
      FG=ABS(AMPTR)/SPGEN                                               FLXG0270
C                          TRUNCATE TO INTEGER                          FLXG0280
      KG=FG                                                             FLXG0290
C                          CHECK IF ABSCISSA EQUAL TO LAST VALUE, AND IF FLXG0300
C                          SO SET ORDINATE TO LAST VALUE AND INTERVAL TO FLXG0310
C                          LAST RANGE                                   FLXG0320
      IF(KG-11) 1,1,2                                                   FLXG0330
    2 CFG=CFGA(13)                                                      FLXG0340
      DCGDI=CFGA(13)-CFGA(12)                                           FLXG0350
      GO TO 3                                                           FLXG0360
C                          COMPUTE RANGE ORDINATE FALLS IN              FLXG0370
    1 DCGDI=CFGA(KG+2)-CFGA(KG+1)                                       FLXG0380
C                          COMPUTE ORDINATE                             FLXG0390
      CFG=CFGA(KG+1)+DCGDI*(FG-KG)                                      FLXG0400
C                          SET SIGN OF ORDINATE                         FLXG0410
    3 CFG=AMPTR/ABS(AMPTR)*CFG                                          FLXG0420
C                          COMPUTE REMAINING GENERATOR VALUES           FLXG0430
      XLFG=TRNFG*TRNFG*DCGDI/(ZGP*SPGEN)                                FLXG0440
      TAUFG=XLFG/(RX+RG)                                                FLXG0450
      XM=TRNFG*TRNSF*DCGDI/(ZGP*SPGEN)                                  FLXG0460
      XLASF=TRNSF*TRNSF*DCGDI/(ZGP*SPGEN)                               FLXG0470
      RETURN                                                            FLXG0480
      END                                                               FLXG0490
// DUP                                                                  FLXG0500
*STORE      WS  UA  FLUXG                                               FLXG0510
```

```
// FOR                                                                  FLXM0 00
*LIST ALL                                                               FLXM0010
*ONE WORD INTEGERS                                                      FLXM0020
      SUBROUTINE FLUXM                                                  FLXM0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    FLXM0040
C                                                                       FLXM0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             FLXM0060
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,GBRG,CSHFR,CWM,PCMS,XLA,            FLXM0070
     2 TAU1,T12,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                           FLXM0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           FLXM0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   FLXM0100
      COMMON DT,CT,T,TEND,TIMSC                                         FLXM0110
      COMMON Z,Y,XNPY,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  FLXM0120
      COMMON CZ,CY,CXNPY,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      FLXM0130
      COMMON ZDO,YDO,XNPYO,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TFLXM0140
     1 EMPO                                                             FLXM0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,FLXM0160
     1QM,QFRM,AIHAT,CP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    FLXM0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAFLXM0180
     1SF,AMPTD,XGEN                                                     FLXM0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBFLXM0200
      COMMON B(64),BB(64),E(32),EF(32)                                  FLXM0210
      COMMON EMP(20),WDED,WADED(20),TDED,TWDED(20),XKK                  FLXM0220
      COMMON IFLO,JFLO,KFLO,LFLO,MFLO                                   FLXM0230
      COMMON SPARE(5),ISPAR(5)                                          FLXM0240
C                                                                       FLXM0250
      EQUIVALENCE (FMI,SPARE(1))                                        FLXM0260
C                                                                       FLXM0270
C                          COMPUTE ABSCISSA                             FLXM0280
      FM=ABS(FMI)/SPMOT                                                 FLXM0290
C                          TRUNCATE TO INTEGER                          FLXM0300
      KM=FM                                                             FLXM0310
C                          CHECK IF ABSCISSA EQUAL TO LAST VALUE, AND IF FLXM0320
C                          SO SET ORDINATE TO LAST VALUE AND INTERVAL TO FLXM0330
C                          LAST RANGE                                   FLXM0340
      IF (KM-11) 1,1,2                                                  FLXM0350
    2 CFM=CFMA(13)                                                      FLXM0360
      DCMDI=CFMA(13)-CFMA(12)                                           FLXM0370
      GO TO 3                                                           FLXM0380
C                          COMPUTE RANGE ORDINATE FALLS IN              FLXM0390
    1 DCMDI=CFMA(KM+2)-CFMA(KM+1)                                       FLXM0400
C                          COMPUTE ORDINATE                             FLXM0410
      CFM=CFMA(KM+1)+DCMDI*(FM-KM)                                      FLXM0420
C                          SET SIGN OF ORDINATE                         FLXM0430
    3 CFM=FMI/ABS(FMI)*CFM                                              FLXM0440
C                          COMPUTE REMAINING MOTOR FIELD VALUES         FLXM0450
      XLFM=TRNFM*DCMDI/(ZMP*SPMOT)                                      FLXM0460
      TAUFM=XLFM/RFM                                                    FLXM0470
      RETURN                                                            FLXM0480
      END                                                               FLXM0490
// DUP                                                                  FLXM0500
*STORE      WS  UA  FLUXM                                               FLXM0510
```

```
// FOR                                                                    QDEV0 00
*LIST ALL                                                                 QDEV0010
*ONE WORD INTEGERS                                                        QDEV0020
      SUBROUTINE QDEV                                                     QDEV0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                      QDEV0040
C                                                                         QDEV0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,               QDEV0060
     1 XJVP,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHFR,CWM,PCYS,XLA,              QDEV0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                            QDEV0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP             QDEV0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12     QDEV0100
      COMMON DT,CT,T,TEND,TIMSC                                           QDEV0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                    QDEV0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP        QDEV0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TQDEV0140
     1 END0                                                               QDEV0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,CG,EG,EB,EC,QDEV0160
     1QV,QERM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                      QDEV0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAQDEV0180
     1SF,AMPTR,XGEN                                                       QDEV0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBQDEV0200
      COMMON B(64),BB(64),E(32),EE(32)                                    QDEV0210
      COMMON EMP(20),WDED,WADED(20),TDED,THDED(20),XKK                    QDEV0220
      COMMON IELO,JELO,KELO,LELO,MELO                                     QDEV0230
      COMMON SPARE(5),ISPAR(5)                                            QDEV0240
C                                                                         QDEV0250
C                          COMPUTE DIESEL DEVELOPED TORQUE                QDEV0260
      QDE=XK6*Z*XGEN                                                      QDEV0270
C                          COMPUTE MAXIMUM DIESEL DEVELOPED TORQUE USING  QDEV0280
C                          ANGULAR VELOCITY OF PRIME MOVER AND THE        QDEV0290
C                          NUMBER OF GENERATORS                           QDEV0300
      IF (60.*XNPM-500.) 10,10,20                                         QDEV0310
   10 QDEMX=(14000.*0.62/500.*60.*XNPM+14000.*0.38)*XGEN                  QDEV0320
      GO TO 70                                                            QDEV0330
   20 IF (60.*XNPM-720.) 30,30,40                                         QDEV0340
   30 QDEMX=(14000.)*XGEN                                                 QDEV0350
      GO TO 70                                                            QDEV0360
   40 IF(60.*XNPM-810.) 50,50,60                                          QDEV0370
   50 QDEMX=(1750./90.*60.*XNPM)*XGEN                                     QDEV0380
      GO TO 70                                                            QDEV0390
   60 QDEMX=(15750.)*XGEN                                                 QDEV0400
C                                                                         QDEV0410
C    THE FOLLOWING TERM IN BRACKETS IS THE DIESEL ENGINE FRICTION TORQUE  QDEV0420
C                                                                         QDEV0430
   70 QDEMX=QDEMX+(QBRG+CWM*XNPM)*XGEN                                    QDEV0440
C                          SET DEVELOPED TORQUE TO MAXIMUM VALUE IF       QDEV0450
C                          COMPUTED VALUE IS GREATER THAN MAXIMUM VALUE   QDEV0460
      IF (QDEMX-QDE) 65,65,80                                             QDEV0470
   65 QDE=QDEMX                                                           QDEV0480
   80 RETURN                                                              QDEV0490
      END                                                                 QDEV0500
// DUP                                                                    QDEV0510
*STORE    WS  UA  QDEV                                                    QDEV0520
```

```
// FOR                                                                    HULL0 00
*LIST ALL                                                                 HULL0010
*ONE WORD INTEGERS                                                        HULL0020
      SUBROUTINE HULL                                                     HULL0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                      HULL0040
C                                                                         HULL0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,               HULL0060
     1 XJVP,XJDG,FIMAX,RA,RG,C1,XK1,OBRG,CSHFR,CWM,PCMS,XLA,              HULL0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                            HULL0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP             HULL0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12     HULL0100
      COMMON DT,CT,T,TEND,TIMSC                                           HULL0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                    HULL0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP        HULL0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,THULL0140
     1 EMDO                                                               HULL0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,QG,EG,EB,EC,HULL0160
     1QM,QFRM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                      HULL0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAHULL0180
     1SF,AMPTR,XGEN                                                       HULL0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBHULL0200
      COMMON B(64),BB(64),E(32),EE(32)                                    HULL0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                    HULL0220
      COMMON IELO,JELO,KELO,LELO,MELO                                     HULL0230
      COMMON SPARE(5),ISPAR(5)                                            HULL0240
      COMMON CAD,DELTA,RV                                                 HULL0250
      COMMON RRA(25),RRMX,BLOC1,DBB12,BLOC2,DBB21                         HULL0260
      COMMON EO,EK,DCT,DCR,DSH,CICE,TICE                                  HULL0270
      COMMON CK1,CK2,CK3                                                  HULL0280
C                                                                         HULL0290
C                          CONVERT SHIP'S SPEED TO KNOTS                  HULL0300
      V=ABS(VS)/1.689                                                     HULL0310
C                          CONVERT TO INTEGER (TRUNCATING VALUE)          HULL0320
      KV=V                                                                HULL0330
C                          COMPUTE VALUES USING TABLES                    HULL0340
      REHP=EHP(KV+1)+(EHP(KV+2)-EHP(KV+1))*(V-KV)                         HULL0350
      WDED=WADED(KV+1)+(WADED(KV+2)-WADED(KV+1))*(V-KV)                   HULL0360
      TDED=THDED(KV+1)+(THDED(KV+2)-THDED(KV+1))*(V-KV)                   HULL0370
      R=REHP*550./VS                                                      HULL0380
C                          ADD ICE RESISTANCE IF APPLICABLE               HULL0390
      IF(ICNTR-4) 67,66,67                                                HULL0400
   66 R=R+CICE*VS*VS                                                      HULL0410
C                          UPDATE VALUES IF IN REVERSE DIRECTION          HULL0420
   67 IF(VS)60,65,65                                                      HULL0430
 65   RETURN                                                              HULL0440
   60 R=1.4*R                                                            HULL0450
      WDED=1.                                                             HULL0460
      TDED=.9                                                             HULL0470
      RETURN                                                              HULL0480
      END                                                                 HULL0490
// DUP                                                                    HULL0500
*STORE      WS  UA  HULL                                                  HULL0510
```

```
// FOR                                                                    PROP0 00
*LIST ALL                                                                 PROP0010
*ONE WORD INTEGERS                                                        PROP0020
      SUBROUTINE PROP                                                     PROP0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                      PROP0040
C                                                                         PROP0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,               PROP0060
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHFR,CWM,PCMS,XLA,              PROP0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                            PROP0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP             PROP0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12     PROP0100
      COMMON DT,CT,T,TEND,TIMSC                                           PROP0110
      COMMON Z,Y,XNPP,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEVP                    PROP0120
      COMMON CZ,CY,CXNPY,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP        PROP0130
      COMMON ZDO,YDO,XNPYD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TPROP0140
     1 EVDO                                                               PROP0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,CDE,QG,EG,EB,EC,PROP0160
     1PM,QFRY,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                      PROP0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAPROP0180
     1SF,AYPTR,XGEN                                                       PROP0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBPROP0200
      COMMON B(64),BB(64),E(32),EE(32)                                    PROP0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                    PROP0220
      COMMON IELO,JELO,KELO,LELO,MELO                                     PROP0230
      COMMON SPARE(5),ISPAR(5)                                            PROP0240
C                                                                         PROP0250
      EQUIVALENCE (XN,XNP)                                                PROP0260
C                                                                         PROP0270
      ROD2=RO*D*D                                                         PROP0280
      IF(XN) 31,32,41                                                     PROP0290
C                          NEGATIVE PORP ANGULAR VELOCITY                 PROP0300
   31 IF(VA) 100,200,300                                                  PROP0310
C                          NEGATIVE ADVANCE VELOCITY OF PROP              PROP0320
  100 KH=3                                                                PROP0330
      GO TO 42                                                            PROP0340
C                          ZERO ADVANCE VELOCITY OF PROP                  PROP0350
  200 I=49                                                                PROP0360
      GO TO 8                                                             PROP0370
C                          POSITIVE ADVANCE VELOCITY OF PROP              PROP0380
  300 KH=2                                                                PROP0390
      GO TO 42                                                            PROP0400
C                          PROP ANGULAR VELOCITY OF ZERO                  PROP0410
   32 IF(VA)101,33,301                                                    PROP0420
C                          NEGATIVE ADVANCE VELOCITY OF PROP              PROP0430
  101 I=25                                                                PROP0440
      GO TO 18                                                            PROP0450
C                          ZERO ADVANCE VELOCITY OF PROP                  PROP0460
   33 QPI=0.                                                              PROP0470
      XPROI=0.                                                            PROP0480
      RETURN                                                              PROP0490
C                          POSITIVE ADVANCE VELOCITY OF PROP              PROP0500
  301 I=9                                                                 PROP0510
      GO TO 18                                                            PROP0520
C                          POSITIVE PROP ANGULAR VELOCITY                 PROP0530
```

```
   41 IF(VA) 102,202,302                                                PROP0540
C                          NEGATIVE ADVANCE VELOCITY OF PROP            PROP0550
  102 KB=0                                                              PROP0560
      GO TO 42                                                          PROP0570
C                          ZERO ADVANCE VELOCITY OF PROP                PROP0580
  202 I=1                                                               PROP0590
      GO TO 8                                                           PROP0600
C                          POSITIVE ADVANCE VELOCITY OF PROP            PROP0610
  302 KB=1                                                              PROP0620
C                          SET 'TT' TO ZERO IF USING J, TO 2 IF USING 1/J PROP0630
 42   TT=0.                                                             PROP0640
      XJ=VA/(XN*D)                                                      PROP0650
      IF (ABS(XJ)-1.5) 1,1,2                                            PROP0660
    2 XJ=1./XJ                                                          PROP0670
      TT=1.                                                             PROP0680
    1 KD=10.*ABS(XJ)                                                    PROP0690
      IF (TT) 43,43,44                                                  PROP0700
C                          CALCULATE IDEAL PROP TORQUE AND THRUST PER   PROP0710
C                          PROP USING REGULAR COEFFICIENTS (BASED ON J) PROP0720
C                          CALCULATE ARRAY INDEX                        PROP0730
   43 I=KD+2+16*KB                                                      PROP0740
C                          CALCULATE FRACTIONAL PART AND ADJUST TO BE   PROP0750
C                          NEGATIVE                                     PROP0760
    0 PP=XJ*10.*(-1)**KB+KD                                             PROP0770
      XKQ=BB(I-1)+(BB(I-1)-BB(I))*PP                                    PROP0780
      XKT=B(I-1)+(B(I-1)-B(I))*PP                                       PROP0790
C                          COMPUTE TORQUE AND THRUST                    PROP0800
 50   QPI=XKQ*XN*XN*ROD2*D*D*D                                          PROP0810
      XPROI=XKT*XN*XN*ROD2*D*D                                          PROP0820
      RETURN                                                            PROP0830
  8   XKQ=BB(I)                                                         PROP0840
      XKT=B(I)                                                          PROP0850
      GO TO 50                                                          PROP0860
C                          CALCULATE IDEAL PROP TORQUE AND THRUST PER   PROP0870
C                          PROP USING MODIFIED COEFFICIENTS (BASED ON 1/J)PROP0880
C                          CALCULATE ARRAY INDEX                        PROP0890
   44 I=KD+2+9*KB                                                       PROP0900
C                          CALCULATE FRACTIONAL PART AND ADJUST TO BE   PROP0910
C                          NEGATIVE                                     PROP0920
   19 PP=XJ*10.*(-1)**KB+KD                                             PROP0930
      XKQD=EE(I-1)+(EE(I-1)-EE(I))*PP                                   PROP0940
      XKTD=E(I-1)+(E(I-1)-E(I))*PP                                      PROP0950
C                          COMPUTE TORQUE AND THRUST                    PROP0960
 60   QPI=XKQD*VA*VA*ROD2*D                                             PROP0970
      XPROI=XKTD*VA*VA*ROD2                                             PROP0980
      RETURN                                                            PROP0990
 18   XKQD=EE(I)                                                        PROP1000
      XKTD=E(I)                                                         PROP1010
      GO TO 60                                                          PROP1020
      END                                                               PROP1030
// DUP                                                                  PROP1040
*STORE      WS  UA  PROP                                               PROP1050
```

```
// FOR                                                                  QICE0000
*LIST ALL                                                               QICE0010
*ONE WORD INTEGERS                                                      QICE0020
      SUBROUTINE QICE                                                   QICE0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    QICE0040
C                                                                       QICE0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             QICE0060
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHFR,CWM,PCMS,XLA,            QICE0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          QICE0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           QICE0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   QICE0100
      COMMON DT,CT,T,TEND,TIMSC                                         QICE0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  QICE0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      QICE0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TQICE0140
     1 EMDO                                                             QICE0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,QG,EG,EB,EC,QICE0160
     1QV,QFRM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    QICE0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAQICE0180
     1SF,AMPTR,XGEN                                                     QICE0190
      COMMON CFX,SPVOT,CFMA(13),TRNFM,ZVP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBQICE0200
      COMMON P(64),BB(64),E(32),EE(32)                                  QICE0210
      COMMON EMP(20),WDED,WADED(20),TDED,THDED(20),XKK                  QICE0220
      COMMON IELO,JELO,KELO,LELO,MELO                                   QICE0230
      COMMON SPARE(5),ISPAR(5)                                          QICE0240
      COMMON CAD,DELTA,RM                                               QICE0250
      COMMON RBA(25),RBMX,BLOC1,DBB12,BLOC2,DBB21                       QICE0260
      COMMON EO,EK,DCT,DCR,DSH,CICE,TICE                                QICE0270
      COMMON CK1,CK2,CK3                                                QICE0280
C                                                                       QICE0290
C                          CHECK FOR INITIAL CALCULATION OF MAX CUT     QICE0300
      GO TO (1,2), JELO                                                 QICE0310
 1    CALL RBMAX                                                        QICE0320
C                          CALCULATE ANGLE OF CUT                       QICE0330
      CSAB2=1.-(2.*RBMX/D)                                              QICE0340
      SNAB2=SQRT(1.-CSAB2**2)                                           QICE0350
      AB2=2.*ATAN(SNAB2/CSAB2)                                          QICE0360
C                          CALCULATE ANGLE BETWEEN PROP BLADES          QICE0370
      XIZB=IZB                                                          QICE0380
      ABB=2.*PI/XIZB                                                    QICE0390
      JELO=2                                                            QICE0400
C                          CHECK IF ANGULAR POSITION OF PROP LESS THAN  QICE0410
C                          OR EQUAL TO ANGLE OF CUT                     QICE0420
 2    IF(SIE-AB2) 30,30,20                                              QICE0430
C                          CHECK IF ANGULAR POSITION OF PROP GREATER    QICE0440
C                          THAN OR EQUAL TO ANGLE BETWEEN PROP BLADES   QICE0450
 20   IF(SIE-ABB) 25,1,1                                                QICE0460
C                          SET ICE TORQUE TO ZERO                       QICE0470
 25   QI=0.                                                             QICE0480
      RETURN                                                            QICE0490
C                          CHECK IF PROP ANGULAR VELOCITY BELOW 0.015   QICE0500
 30   IF(ABS(XNP)-.015) 25,25,31                                        QICE0510
 31   CSBET=CSAB2*COS(SIE)+SNAB2*SIN(SIE)                               QICE0520
```

```
      RB=D/2*(1.-CSAB2/CSBET)                                          QICE0530
      RM=(D-RB)/2.                                                     QICE0540
      ALFAV=ATAN(VS/(RM*2.*PI*XNP))                                    QICE0550
      ALFA1=PDR*D/(2.*PI*RM)                                           QICE0560
      E1=E0+RB*((CK-E0)/(D-DCT))                                       QICE0570
      BAV=DAR*PI*D**2/(2.*XIZB*(D-DCT))                                QICE0580
C                          CALCULATE CRUSHING-SHEARING ICE TORQUE     QICE0590
      QICRS=DCR*E1*RB*RM+DSH*RB*RM*VS/(XIZB*XNP)                       QICE0600
C                          CALCULATE MINIMUM ICE TORQUE               QICE0610
      QIMIN=DCR*E1*RB*RM                                               QICE0620
C                          CALCULATE CRUSHING ICE TORQUE              QICE0630
      QICR=DCR*BAV*RB*RM*ABS(SIN(ALFA1-ALFAV))                         QICE0640
C                          SET ICE TORQUE                             QICE0650
      QI=QICRS                                                         QICE0660
      IF(QI-QICR) 45,45,40                                             QICE0670
 40   QI=QICR                                                          QICE0680
 45   IF(QI-QIMIN) 50,50,55                                            QICE0690
 50   QI=QIMIN                                                         QICE0700
 55   RETURN                                                           QICE0710
      END                                                              QICE0720
// DUP                                                                 QICE0730
*STORE      WS  UA  QICE                                              QICE0740
```

```
// FOR                                                                    RBMA0000
*LIST ALL                                                                 RBMA0010
*ONE WORD INTEGERS                                                        RBMA0020
      SUBROUTINE RBMAX                                                    RBMA0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                      RBMA0040
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSM,XJD,XJG,EFG,               RBMA0050
     1 XJM2,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHER,CWM,PCMS,XLA,              RBMA0060
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                            RBMA0070
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP             RBMA0080
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12    RBMA0090
      COMMON DT,CT,T,TEND,TIMSC                                           RBMA0100
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                    RBMA0110
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP        RBMA0120
      COMMON ZDO,YDO,XNPMO,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TRBMA0130
     1 EMDO                                                               RBMA0140
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,RBMA0150
     1QM,QERM,AIMAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                      RBMA0160
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLARBMA0170
     1SF,AMPTR,XGEN                                                       RBMA0180
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBRBMA0190
      COMMON B(64),BB(64),E(32),EE(32)                                    RBMA0200
      COMMON EMP(20),XDED,WADED(20),TDED,THDED(20),XKK                    RBMA0210
      COMMON IFLO,JELO,KELO,LELO,MELO                                     RBMA0220
      COMMON SPARE(5),ISPAR(5)                                            RBMA0230
      COMMON CAD,DELTA,RM                                                 RBMA0240
      COMMON RBA(25),RBMX,BLOC1,DB512,BLOC2,DBB21                         RBMA0250
      COMMON E0,EK,DCT,DCR,DSH,CICE,TICE                                  RBMA0260
      COMMON CK1,CK2,CK3                                                  RBMA0270
C                          SET PROP ANGULAR POSITION TO ZERO              RBMA0280
      CSIE=0.                                                             RBMA0290
      SIE=0.                                                              RBMA0300
C                          BRANCH ON 'KELO' AS FOLLOWS...                 RBMA0310
C                               KELO=1 -- FIRST ENCOUNTER WITH BLOCK 1    RBMA0320
C                                   =2 -- IN BLOCK 1                      RBMA0330
C                                   =3 -- BETWEEN BLOCKS 1 AND 2          RBMA0340
C                                   =4 -- IN BLOCK 2                      RBMA0350
C                                   =5 -- BETWEEN BLOCKS 2 AND 1          RBMA0360
      GO TO (1,2,3,4,5), KELO                                             RBMA0370
C                          SAVE PRESENT DISTANCE                          RBMA0380
 1    XICE=CXS                                                            RBMA0390
C                          INITIALIZE ARRAY INDEX                         RBMA0400
      IA=1                                                                RBMA0410
C                          UPDATE 'KELO'                                  RBMA0420
      KELO=2                                                              RBMA0430
C                          CHECK IF IN BLOCK 1                            RBMA0440
 2    IF((CXS-XICE)-BLOC1) 55,55,10                                       RBMA0450
C                          COMPUTE MAX CUT AND UPDATE ARRAY INDEX         RBMA0460
 55   RBMX=RBA(IA)*D                                                      RBMA0470
      IA=IA+1                                                             RBMA0480
C                          REINITIALIZE ARRAY INDEX IF BEYOND MAX VALUE   RBMA0490
      IF (IA-11) 7,7,6                                                    RBMA0500
 6    IA=1                                                                RBMA0510
 7    RETURN                                                              RBMA0520
C                          UPDATE 'KELO' AND SAVE PRESENT DISTANCE        RBMA0530
```

```
 10   KELO=3                                                          RBMA0540
      XICE=CXS                                                        RBMA0550
C                          CHECK IF BETWEEN BLOCKS 1 AND 2 AND RETURN RBMA0560
C                          MAX CUT OF ZERO IF SO                     RBMA0570
 3    IF((CXS-XICE)-DB812) 15,15,20                                   RBMA0580
 15   RBMX=0.                                                         RBMA0590
      RETURN                                                          RBMA0600
C                          ENTERING BLOCK 2, INITIALIZE ARRAY INDEX AND RBMA0610
C                          UPDATE 'KELO'                             RBMA0620
 20   XICE=CXS                                                        RBMA0630
      IA=11                                                           RBMA0640
      KELO=4                                                          RBMA0650
C                          CHECK IF IN BLOCK 2                       RBMA0660
 4    IF((CXS-XICE)-BLOC2) 25,25,30                                   RBMA0670
C                          COMPUTE MAX CUT AND UPDATE ARRAY INDEX    RBMA0680
 25   RBMX=RBA(IA)*D                                                  RBMA0690
      IA=IA+1                                                         RBMA0700
C                          REINITIALIZE ARRAY INDEX IF BEYOND MAX VALUE RBMA0710
      IF(IA-26) 27,27,26                                              RBMA0720
 26   IA=11                                                           RBMA0730
 27   RETURN                                                          RBMA0740
C                          BETWEEN BLOCKS 2 AND 1, UPDATA 'KELO' AND RBMA0750
C                          SAVE PRESENT DISTANCE                     RBMA0760
 30   KELO=5                                                          RBMA0770
      XICE=CXS                                                        RBMA0780
C                          CHECK IF BETWEEN BLOCKS 2 AND 1 AND RETURN A RBMA0790
C                          MAX CUT OF ZERO IF SO                     RBMA0800
 5    IF((CXS-XICE)-DB821) 35,35,1                                    RBMA0810
 35   RBMX=0.                                                         RBMA0820
      RETURN                                                          RBMA0830
      END                                                             RBMA0840
// DUP                                                                RBMA0850
*STORE      WS  UA  RBMAX                                            RBMA0860
```

```
// FOR                                                                    STAT0000
*LIST ALL                                                                 STAT0010
*ONE WORD INTEGERS                                                        STAT0020
      SUBROUTINE STATE                                                    STAT0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                      STAT0040
C                                                                         STAT0050
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,               STAT0060
     1 XJVP,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHFR,CWM,PCMS,XLA,              STAT0070
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                            STAT0080
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP             STAT0090
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12     STAT0100
      COMMON DT,CT,T,TEND,TIMSC                                           STAT0110
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                    STAT0120
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP        STAT0130
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TSTAT0140
     1 EMDO                                                               STAT0150
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,STAT0160
     1QK,QERM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                      STAT0170
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLASTAT0180
     1SF,AMPTR,XGEN                                                       STAT0190
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBSTAT0200
      COMMON B(64),BB(64),E(32),EE(32)                                    STAT0210
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                    STAT0220
      COMMON IELO,JELO,KELO,LELO,MELO                                     STAT0230
      COMMON SPARE(5),ISPAR(5)                                            STAT0240
      COMMON CAD,DELTA,RM                                                 STAT0250
      COMMON RBA(25),RBMX,BLOC1,DBB12,BLOC2,DBB21                         STAT0260
      COMMON EO,EK,DCT,DCR,DSH,CICE,TICE                                  STAT0270
      COMMON CK1,CK2,CK3                                                  STAT0280
C                                                                         STAT0290
      EQUIVALENCE (FLOW,SPARE(5))                                         STAT0300
C                                                                         STAT0310
C                          COMPUTE SERVO STROKE VELOCITY                  STAT0320
      ZDO=Y                                                               STAT0330
C                          COMPUTE SERVO STROKE ACCELERATION              STAT0340
      YDO=60.*ALP2*XNEDO+60.*ALP2*ALP3*XNE-(ALP1+ALPD)*Y-ALP3*ALPD*Z      STAT0350
C                          COMPUTE TIME RATE OF PRIME MOVER SPEED         STAT0360
      XNPMD=(QDE-QG)/(2.*PI*XJDG)                                         STAT0370
C                          COMPUTE TIME RATE OF ARMATURE CIRCUIT CURRENT  STAT0380
      VOLT1=EFG-(RFG+RX)*FGI                                              STAT0390
      VOLT2=EG-EC-RA*AI                                                   STAT0400
      XLD=XLA+1.5/XGEN*XLASF                                              STAT0410
      XLM=2.25/XGEN*XM*XM                                                 STAT0420
      DENOM=1.5*XLFG*XLD-XLM                                              STAT0430
      AIDO=1.5*(XM*VOLT1+XLFG*VOLT2)/DENOM                                STAT0440
C                          COMPUTE TIME RATE OF GENERATOR FIELD CURRENT   STAT0450
      FGIDO=(1.5/XGEN*XM*VOLT2+XLD*VOLT1)/DENOM                           STAT0460
C                          COMPUTE TIME RATE OF MOTOR FIELD ADJUSTING CUR.STAT0470
      IF (THETA) 10,6,6                                                   STAT0480
    6 IF(EB-C1) 10,10,20                                                  STAT0490
   10 FIPDO=0.0                                                           STAT0500
      GO TO 30                                                            STAT0510
   20 FIPDO=.01512*(AIHAT-AI)/XLFM                                        STAT0520
```

```
C                                COMPUTE PROPELLER ACCELERATION                 STAT0530
   30 XNPDO=(QM-QFRM-QP-QI)/(2.*PI*XJMP)                                        STAT0540
C                                COMPUTE SHIP'S ACCELERATION                    STAT0550
      VSDO=(TP-R)/XMASS                                                         STAT0560
C                                COMPUTE SHIP'S VELOCITY                        STAT0570
      XSDO=VS                                                                   STAT0580
C                                COMPUTE PROPELLER ANGULAR POSITION             STAT0590
      SIEDO=XNP*2.*PI                                                           STAT0600
C                                                                               STAT0610
      RETURN                                                                    STAT0620
      END                                                                       STAT0630
// DUP                                                                          STAT0640
*STORE      WS  UA  STATE                                                       STAT0650
```

```
// FOR                                                                    OUT0000
*LIST ALL                                                                 OUT0010
*ONE WORD INTEGERS                                                        OUT0020
      SUBROUTINE OUT                                                      OUT0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                      OUT0040
      COMMON RO,PI,IZR,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,               OUT0050
     1 XJMP,XJDG,FIXAX,RA,RG,C1,XK1,QBRG,CSHFR,CWM,PCMS,XLA,              OUT0060
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                            OUT0070
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP             OUT0080
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12     OUT0090
      COMMON DT,CT,T,TEND,TIMSC                                           OUT0100
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                    OUT0110
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP        OUT0120
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,T  OUT0130
     1,EMDO                                                               OUT0140
      COMMON XNF,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,   OUT0150
     1QM,QFM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                       OUT0160
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLA  OUT0170
     1SF,AMPTR,XGEN                                                       OUT0180
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUB  OUT0190
      COMMON B(64),BB(64),E(32),EE(32)                                    OUT0200
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                    OUT0210
      COMMON IELO,JELO,KELO,LELO,MELO                                     OUT0220
      COMMON SPARE(5),ISPAR(5)                                            OUT0230
C                                                                         OUT0240
      EQUIVALENCE (FMI,SPARE(1))                                          OUT0250
C                                                                         OUT0260
C                          CONVERT ANGULAR VELOCITIES TO REV. PER MIN.     OUT0270
   32 RPMPM=60.*CXNPM                                                     OUT0280
      RPMP=60.*CXNP                                                       OUT0290
C                          CONVERT SHIP'S SPEED TO KNOTS                  OUT0300
      CVK=CVS/1.689                                                       OUT0310
C                          CONVERT TOTAL PROPELLER THRUST TO TONS         OUT0320
      TPT=TP/2240.                                                        OUT0330
C                          CONVERT TOTAL RESISTANCE TO TONS               OUT0340
      RT=R/2240.                                                          OUT0350
C                          COMPUTE SHIP'S HORSEPOWER FROM MOTOR TORQUE    OUT0360
C                          AND PROPELLER ANGULAR VELOCITY                 OUT0370
      POW=QM*2.*PI*XNP/550.                                               OUT0380
C                          PRINT VALUES                                   OUT0390
      WRITE (3,301) CT,RPMPM,CFGI,RPMP,CVK,QDE,QG,EB,FMI,EC,QM,AI,QP,     OUT0400
     1 TPT,RT,POW                                                         OUT0410
  301 FORMAT (1X,F7.3,1X,F5.1,1X,F5.1,1X,F6.1,1X,F5.1,F8.1,               OUT0420
     1 F8.1,1X,F6.1,1X,F5.1,1X,F6.1,1X,F9.1,1X,F8.1,1X,F9.1,F7.1,1X,      OUT0430
     2 F6.1,F8.1)                                                         OUT0440
      RETURN                                                              OUT0450
      END                                                                 OUT0460
// DUP                                                                    OUT0470
*STORE      WS  UA  OUT                                                   OUT0480
```

```
// FOR                                                                  FINI0000
*LIST ALL                                                              FINI0010
*ONE WORD INTEGERS                                                     FINI0020
      SUBROUTINE FINIS                                                 FINI0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                   FINI0040
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,            FINI0050
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,OBRG,CSHFR,CWM,PCMS,XLA,           FINI0060
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALP0,XMASS                         FINI0070
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,MON,NIP           FINI0080
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12  FINI0090
      COMMON DT,CT,T,TEND,TIMSC                                        FINI0100
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                 FINI0110
      COMMON CZ,CY,CXNPY,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP     FINI0120
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TFINI0130
     1 EMDO                                                            FINI0140
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,FINI0150
     1QM,QERM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                   FINI0160
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAFINI0170
     1EP,ANPTR,XGEN                                                    FINI0180
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBFINI0190
      COMMON B(64),BB(64),E(32),EE(32)                                 FINI0200
      COMMON EMP(20),WDED,WADED(20),TDED,THDED(20),XKK                 FINI0210
      COMMON ISLO,JSLO,KSLO,LSLO,MSLO                                  FINI0220
      COMMON SPARE(5),ISPAR(5)                                         FINI0230
C*******************     CHECK IF DATA SWITCH SETTING INSTRUCTIONS      FINI0240
C*******************     ARE TO BE TYPED OUT                           FINI0250
      CALL DATSW(4,KEY4)                                               FINI0260
      GO TO(110,50),KEY4                                               FINI0270
   50 WRITE(1,100)                                                     FINI0280
  100 FORMAT(' TURN ON SWITCH --')                                     FINI0290
      WRITE(1,101)                                                     FINI0300
  101 FORMAT('         1 - TO PUNCH NEW INITIAL CONDITION CARDS')      FINI0310
      WRITE(1,102)                                                     FINI0320
  102 FORMAT('         2 - TO PLOT DATA')                              FINI0330
      WRITE(1,103)                                                     FINI0340
  103 FORMAT('         3 - TO CONTINUE SOLUTION OF CURRENT PROBLEM - SWITFINI0350
     1CH OFF STARTS NEW PROBLEM')                                      FINI0360
      WRITE(1,104)                                                     FINI0370
  104 FORMAT('         4 - TO SUPPRESS FUTURE TYPING OF THESE INSTRUCTIONFINI0380
     1S')                                                              FINI0390
      WRITE(1,1045)                                                    FINI0400
 1045 FORMAT('         5 - TO CHANGE TIME STEP')                       FINI0410
      WRITE (1,1050)                                                   FINI0420
 1050 FORMAT ('         6 - TO CHANGE PRINT INTERVAL')                  FINI0430
      WRITE (1,1055)                                                   FINI0440
 1055 FORMAT ('         7 - TO CHANGE XLA')                             FINI0450
      WRITE(1,105)                                                     FINI0460
  105 FORMAT(///' TURN ON SELECTOR SWITCHES AND PRESS START')          FINI0470
  110 PAUSE                                                            FINI0480
      CALL DATSW(1,KEY1)                                               FINI0490
      GO TO(200,250),KEY1                                              FINI0500
C*******************     PUNCH NEW INITIAL CONDITION CARDS              FINI0510
  200 WRITE(2,201) THETA,XNPMD,Y,Z                                     FINI0520
      WRITE(3,201) THETA,XNPMD,Y,Z                                     FINI0530
```

```
  201 FORMAT (10X,4F10.4,22X,'DATA0100')                                FINI0540
      WRITE(2,202) FGI,XNPM,FIP,XNP,VS,AI,AIDO                          FINI0550
      WRITE(3,202) FGI,XNPM,FIP,XNP,VS,AI,AIDO                          FINI0560
  202 FORMAT (5F10.4,22X,'DATA0101',/,2F10.4,52X,'DATA0102')            FINI0570
  250 CALL DATSW(2,KEY2)                                                FINI0580
      GO TO(300,350),KEY2                                               FINI0590
C******************     LOAD AND EXECUTE GRAPH ROUTINE TO PLOT DATA      FINI0600
  300 TEND=CT-DT                                                        FINI0610
      CALL LINK (GRAPH)                                                 FINI0620
  350 CALL DATSW(5,KEY5)                                                FINI0630
      GO TO (400,450), KEY5                                             FINI0640
C******************     NEW TIME STEP TO BE ENTERED                      FINI0650
  400 WRITE(1,120)                                                      FINI0660
  120 FORMAT('(       ) - TIME STEP')                                   FINI0670
      READ (6,121) DT                                                   FINI0680
  121 FORMAT(1X,F6.4)                                                   FINI0690
  450 CALL DATSW (6,KEY6)                                               FINI0700
      GO TO (500,550), KEY6                                             FINI0710
C******************     NEW PRINTER TIME STEP TO BE ENTERED              FINI0720
  500 WRITE (1,501)                                                     FINI0730
  501 FORMAT ('(    ) - NON')                                           FINI0740
      READ (6,502) NON                                                  FINI0750
  502 FORMAT (1X,I3)                                                    FINI0760
  550 CALL DATSW (7,KEY7)                                               FINI0770
      GO TO (600,650), KEY7                                             FINI0780
C******************     NEW ARMATURE CIRCUIT INDUCTANCE TO BE ENTERED    FINI0790
  600 WRITE (1,601)                                                     FINI0800
  601 FORMAT('(        ) - XLA')                                        FINI0810
      READ (6,602) XLA                                                  FINI0820
  602 FORMAT (1X,F7.4)                                                  FINI0830
C******************     STORE INFO FOR ABORTING RUN                      FINI0840
C******************          SWITCH 3 ON TO CONTINUE, OFF TO ABORT       FINI0850
  650 CALL DATSW (3,KEY3)                                               FINI0860
      ISPAR(1)=KEY3                                                     FINI0870
      RETURN                                                            FINI0880
      END                                                               FINI0890
// DUP                                                                  FINI0900
*STORE      WS  UA  FINIS                                              FINI0910
```

```
// FOR                                                                    GRAP0000
*LIST ALL                                                                 GRAP0010
*NAME GRAPH                                                               GRAP0020
*ONE WORD INTEGERS                                                        GRAP0030
*IOCS(CARD,TYPEWRITER,KEYBOARD,1132PRINTER,DISK,PLOTTER)                  GRAP0040
C  GRAPH MAINLINE PROGRAM                                                 GRAP0050
C      VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                     GRAP0060
C                                                                         GRAP0070
C      THE VALUES OF THE VARIABLES TO BE PLOTTED ARE TAKEN FROM THE       GRAP0080
C      DISK FILE AS CREATED BY 'EXEC'.                                    GRAP0090
C      THE FOLLOWING VARIABLES ARE ASSUMED TO BE IN 'COMMON' (HAVING      GRAP0100
C      ALREADY BEEN READ IN OR COMPUTED IN 'GENIS')...                    GRAP0110
C          SPARE(4), TEND, SX, SY1, SY2, SY3, SY4, SY5, SY6,              GRAP0120
C               SY7, SY8, SY9, SY10, SY11, SY12.                          GRAP0130
C          SPARE(4)          -  SERIAL NUMBER TO BE PRINTED ON THE GRAPH.  GRAP0140
C          TEND              -  TIME END VALUE                           GRAP0150
C          SX                -  SCALE FACTOR FOR TIME AXIS (X-AXIS).     GRAP0160
C          SY1, ..., SY12    -  SCALE FACTORS FOR THE 12 VARIABLES THAT  GRAP0170
C                                 MAY BE PLOTTED.                         GRAP0180
C          'SPARE(4)' IS ONLY REQUIRED IF A GRID IS REQUESTED.            GRAP0190
C                                                                         GRAP0200
C                         PEN MUST BE INITIALLY SET IN THE CENTER OF THE  GRAP0210
C                         ROLL AT LEAST 6 INCHES FROM THE LEFT END OR     GRAP0220
C                         LAST PLOT ON THE CHART.                         GRAP0230
C                                                                         GRAP0240
C                                                                         GRAP0250
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,               GRAP0260
     1 XJVP,XJDG,FIMAX,RA,RG,C1,XK1,OBRG,CSHER,CWN,CMS,XLA,               GRAP0270
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                           GRAP0280
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP             GRAP0290
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12    GRAP0300
      COMMON DT,CT,T,TEND,TIMSC                                          GRAP0310
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                   GRAP0320
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP       GRAP0330
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TGRAP0340
     1 EMDO                                                              GRAP0350
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETO,THEIC,QDE,QG,EG,EB,EC,GRAP0360
     1QV,QERV,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                     GRAP0370
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAGRAP0380
     1SF,AMPTR,XGEN                                                      GRAP0390
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBGRAP0400
      COMMON B(64),BB(64),E(32),EE(32)                                   GRAP0410
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                   GRAP0420
      COMMON IELO,JELO,KELO,LELO,MELO                                    GRAP0430
      COMMON SPARE(5),ISPAR(5)                                           GRAP0440
      COMMON CAD,DELTA,RM                                                GRAP0450
      COMMON RBA(25),RBMX,BLOC1,DBB12,BLOC2,DBB21                        GRAP0460
      COMMON E0,EK,DCT,DCR,DSH,CICE,TICE                                 GRAP0470
      COMMON CK1,CK2,CK3                                                 GRAP0480
C                                                                        GRAP0490
      EQUIVALENCE (SER,SPARE(4))                                         GRAP0500
C                                                                        GRAP0510
      DEFINE FILE 1(600,26,U,IELO)                                       GRAP0520
C                                                                        GRAP0530
```

```
C                                                                       GRAP0540
C*********  DETERMINE WHICH VARIABLES ARE TO BE PLOTTED      ********   GRAP0550
C                                                                       GRAP0560
C                       ZERO COUNTER FOR NUMBER OF VARIABLES.           GRAP0570
      NOVAR=0                                                           GRAP0580
C                       TYPE OUT VARIABLE CODE NUMBERS.                 GRAP0590
C                                                                       GRAP0600
C                       CODE NO.  VARIABLE   DESCRIPTION                GRAP0610
C                           1       'AI'      ARM. CIR. CUR.            GRAP0620
C                           2       'EB'      BUS VOLTS                 GRAP0630
C                           3       'RPMPM'   DIESEL SPEED              GRAP0640
C                           4       'FMI'     MOTOR FIELD CUR.          GRAP0650
C                           5       'RPMP'    PROP. SPEED               GRAP0660
C                           6       'XKW'     POWER                     GRAP0670
C                           7       'CVK'     VESSEL SPEED              GRAP0680
C                           8       'QPI'     PROP. TORQUE              GRAP0690
C                           9       'XPROI'   PROP. THRUST              GRAP0700
C                          10       'CXS'     DISTANCE                  GRAP0710
C                          11       'QI'      ICE TORQUE                GRAP0720
C                          12       'TEMP'    MOTOR TEMP.               GRAP0730
C                                                                       GRAP0740
      WRITE (1,1000)                                                    GRAP0750
 1000 FORMAT (' AI--1,EB--2,XNPM--3,FMI--4,XNP--5,XKW--6,CVK--7,QPI--8, GRAP0760
     1XPROI--9,CXS--10,QI--11,TEMP--12'//)                              GRAP0770
C                       TYPE OUT REQUEST AND FORMAT FOR CODE NOS.       GRAP0780
      WRITE (1,1004)                                                    GRAP0790
 1004 FORMAT (' WHICH VARIABLES TO BE PLOTTED (10 ONLY) -')             GRAP0800
      WRITE (1,1005)                                                    GRAP0810
 1005 FORMAT (' (  )(  )(  )(  )(  )(  )(  )(  )(  )(  )')             GRAP0820
C                       READ CODE NOS. FROM KEYBOARD.                   GRAP0830
      READ (6,1006) NVAR(1),NVAR(2),NVAR(3),NVAR(4),NVAR(5)             GRAP0840
     1,NVAR(6),NVAR(7),NVAR(8),NVAR(9),NVAR(10)                         GRAP0850
 1006 FORMAT (2X,I2,2X,I2,2X,I2,2X,I2,2X,I2,2X,I2,2X,I2,2X,I2,2X,I2,2X,I2, GRAP0860
     12X,I2)                                                            GRAP0870
C                                                                       GRAP0880
C                                                                       GRAP0890
C*********  DETERMINE THE NUMBER OF VARIABLES TO BE PLOTTED  ********   GRAP0900
C                                                                       GRAP0910
      DO 699 I=1,10                                                     GRAP0920
      IF (NVAR(I)) 699,699,700                                          GRAP0930
  700 NOVAR=NOVAR+1                                                     GRAP0940
  699 CONTINUE                                                          GRAP0950
C                                                                       GRAP0960
C                                                                       GRAP0970
C*********  DETERMINE IF A GRID IS TO BE PLOTTED             ********   GRAP0980
C                                                                       GRAP0990
C                       TYPE OUT REQUEST AND FORMAT FOR GRID PLOT.      GRAP1000
      WRITE (1,1007)                                                    GRAP1010
 1007 FORMAT(' DO YOU WANT A GRID (0 - NO,1 - YES)')                    GRAP1020
      WRITE (1,1008)                                                    GRAP1030
 1008 FORMAT(' ( )')                                                    GRAP1040
C                       READ GRID PLOT INFO.                            GRAP1050
      READ (6,1003) NPLOT                                               GRAP1060
 1003 FORMAT (2X,I1)                                                    GRAP1070
```

```
C                       DETERMINE IF GRID WAS DESIRED.                  GRAP1080
      IF (NPLOT-1) 2,1,1                                                GRAP1090
C                                                                       GRAP1100
C                                                                       GRAP1110
C*********  READ ID. INFO. AND PLOT GRID                        ********GRAP1120
C                                                                       GRAP1130
C                       GRID REQUESTED.                                 GRAP1140
C                       READ IN THREE LINES OF INFO. FROM KEYBOARD      GRAP1150
C                       FOR GRAPH LABELS.                               GRAP1160
    1 READ (6,2000)                                                     GRAP1170
 2000 FORMAT ('                                                  ')     GRAP1180
      READ (6,2001)                                                     GRAP1190
 2001 FORMAT ('                                                  ')     GRAP1200
      READ (6,2002)                                                     GRAP1210
 2002 FORMAT ('                                                  ')     GRAP1220
C                       SET GRAPH SCALE TO UNITY AND INDICATE PEN IS    GRAP1230
C                       PRESENTLY AT ORIGIN.                            GRAP1240
      CALL SCALE (1.,1.,0.,0.)                                          GRAP1250
C                       SET CHARACTER ROTATION VARIABLE FOR             GRAP1260
C                       HORIZONTAL PRINTING (ALONG X-AXIS).             GRAP1270
      TH=0.                                                             GRAP1280
C                       SERIAL INFO. AND GRAPH LABELS WILL BE PRINTED   GRAP1290
C                       OUT STARTING 6 INCHES TO THE LEFT AND 2 INCHES  GRAP1300
C                       BELOW THE ORIGIN.  EACH SUCCEEDING LINE WILL    GRAP1310
C                       BE 0.5 INCHES BELOW THE PREVIOUS ONE.  THE      GRAP1320
C                       CHARACTERS WILL BE THE MINIMUM SIZE ALLOWED     GRAP1330
C                       (.10 INCHES SQUARE).                            GRAP1340
      CALL FCHAR (-6.0,-2.0,.10,.10,TH)                                 GRAP1350
      WRITE (7,2003) SER                                                GRAP1360
 2003 FORMAT (10X,'SERIAL NUMBER--',F10.0)                              GRAP1370
      CALL FCHAR (-6.0,-2.5,.10,.10,TH)                                 GRAP1380
      WRITE (7,2000)                                                    GRAP1390
      CALL FCHAR (-6.0,-3.0,.10,.10,TH)                                 GRAP1400
      WRITE (7,2001)                                                    GRAP1410
      CALL FCHAR (-6.0,-3.5,.10,.10,TH)                                 GRAP1420
      WRITE (7,2002)                                                    GRAP1430
C                       RETURN PEN TO ORIGIN IN THE UP-POSITION.        GRAP1440
      CALL FPLOT (0,0.,0.)                                              GRAP1450
C                       PLOT OUT GRID AND LABEL AXES.                   GRAP1460
      CALL GRID                                                         GRAP1470
C                                                                       GRAP1480
C                                                                       GRAP1490
C*********  PLOT CURVE FOR EACH VARIABLE INDICATED IN TURN    ********GRAP1500
C                                                                       GRAP1510
C                       LOOP THROUGH NUMBER OF VARIABLES TO BE PLOTTED. GRAP1520
 2    DO 100 JJ=1,NOVAR                                                 GRAP1530
C                       SET SWITCH SO PEN WILL BE MOVED TO THE FIRST    GRAP1540
C                       VALUE TO BE PLOTTED IN THE UP-POSITION AND      GRAP1550
C                       THEN LOWERED TO BEGIN PLOT.                     GRAP1560
      L=1                                                               GRAP1570
C                       SET DISK FILE RECORD INDEX TO FIRST RECORD.     GRAP1580
      IELO=1                                                            GRAP1590
C                       STORE LOOP INDEX FOR USE BY SUBROUTINE 'VAPLO'. GRAP1600
      NIP=JJ                                                            GRAP1610
```

```
C                       RETRIEVE VARIABLE SCALE FACTOR INTO 'SY'.        GRAP1620
      CALL VAPLO                                                         GRAP1630
C                       SET SCALE FACTORS FOR VARIABLE ('SX' WAS         GRAP1640
C.                      READ IN).                                        GRAP1650
      CALL SCALE (SX,SY,0.,0.)                                           GRAP1660
C                       RETRIEVE NEXT VALUE OF VARIABLE INTO 'VAR' AND   GRAP1670
C                       CORRESPONDING TIME VALUE INTO 'CT'.              GRAP1680
    5 CALL DATA                                                          GRAP1690
C                       CHECK FOR FIRST VALUE OF VARIABLE.               GRAP1700
      GO TO (10,20),L                                                    GRAP1710
C                       FIRST VALUE OF VARIABLE INDICATED.  MOVE PEN TO  GRAP1720
C                       LOCATION IN UP-POSITION LOWERING PEN AFTER       GRAP1730
C                       MOVEMENT.                                        GRAP1740
   10 CALL FPLOT (-2,CT,VAR)                                             GRAP1750
C                       SET SWITCH SO FUTURE VALUES WILL BE PLOTTED      GRAP1760
C                       CONTINUOUSLY.                                    GRAP1770
      L=2                                                                GRAP1780
C                       RETRIEVE NEXT VALUE OF VARIABLE.                 GRAP1790
      GO TO 5                                                            GRAP1800
C                       PLOT NEW VALUE CONTINUING GRAPH FROM PREVIOUS    GRAP1810
C                       POINT.                                           GRAP1820
   20 CALL FPLOT (0,CT,VAR)                                              GRAP1830
C                       CHECK IF GRAPH HAS BEEN COMPLETED.               GRAP1840
      IF (CT-TEND) 50,50,60                                              GRAP1850
C                       CHECK IF CHANGE IN SCALE IS NEEDED.              GRAP1860
   50 IF (CT-19.90) 5,55,55                                              GRAP1870
C                       REDUCE X-AXIS SCALE VALUE BY A FACTOR OF TEN.    GRAP1880
   55 SXX=.1*SX                                                          GRAP1890
C                       INCREASE LAST TIME VALUE BY A FACTOR OF TEN.     GRAP1900
      XD=CT/.1                                                           GRAP1910
C                       COMPUTE NEW STARTING POINT FOR REMAINDER         GRAP1920
C                       OF GRAPH.                                        GRAP1930
      XDARN=XD-CT                                                        GRAP1940
C                       SET NEW SCALE FACTORS AND INDICATE PRESENT       GRAP1950
C                       LOCATION OF PEN IS ORIGIN.                       GRAP1960
      CALL SCALE (SXX,SY,XD,VAR)                                         GRAP1970
C                       RETRIEVE NEXT VALUE OF VARIABLE INTO 'VAR' AND   GRAP1980
C                       CORRESPONDING TIME VALUE INTO 'CT'.              GRAP1990
   57 CALL DATA                                                          GRAP2000
C                       COMPUTE MODIFIED TIME VALUE BECAUSE OF NEW       GRAP2010
C                       X-AXIS SCALE FACTOR AND ORIGIN.                  GRAP2020
      CCT=XDARN+CT                                                       GRAP2030
C                       PLOT NEW VALUE CONTINUING GRAPH FROM PREVIOUS    GRAP2040
C                       POINT.                                           GRAP2050
      CALL FPLOT (0,CCT,VAR)                                             GRAP2060
C                       CHECK IF GRAPH HAS BEEN COMPLETED.               GRAP2070
      IF (CT-TEND) 57,57,58                                              GRAP2080
C                                                                        GRAP2090
C                       SET CHARACTER ROTATION VALUE FOR                 GRAP2100
C                       HORIZONTAL PRINTING (ALONG X-AXIS).              GRAP2110
   58 TH=0.0                                                             GRAP2120
C                       POSITION PEN AND SET CHARACTER SIZE FOR          GRAP2130
C                       PRINTING OF LABEL.                               GRAP2140
      XD=.05/SXX                                                         GRAP2150
```

```
      YD=.05/SY                                                          GRAP2160
      CALL FCHAR (CCT+XD,VAR-YD,.10,.10,TH)                              GRAP2170
      GO TO 59                                                           GRAP2180
C                                                                        GRAP2190
C                        SET CHARACTER ROTATION VALUE FOR                GRAP2200
C                        HORIZONTAL PRINTING (ALONG X-AXIS).             GRAP2210
   60 TH=0.0                                                             GRAP2220
C                        POSITION PEN AND SET CHARACTER SIZE FOR         GRAP2230
C                        PRINTING OF LABEL.                              GRAP2240
      XD=.05/SX                                                          GRAP2250
      YD=.05/SY                                                          GRAP2260
      CALL FCHAR (CT+XD,VAR-YD,.10,.10,TH)                               GRAP2270
C                                                                        GRAP2280
C                        PRINT OUT APPROPRIATE LABEL.                    GRAP2290
   59 CALL PRINT                                                         GRAP2300
C                        RAISE PEN AND RETURN IT TO ORIGIN.              GRAP2310
      CALL FPLOT (1,0.,0.)                                               GRAP2320
C                        LOOP IF MORE VARIABLES ARE TO BE PLOTTED.       GRAP2330
  100 CONTINUE                                                           GRAP2340
C                                                                        GRAP2350
C                                                                        GRAP2360
C*********  RELOAD AND EXECUTE 'GENIS' TO START NEW PROBLEM  ********    GRAP2370
C                                                                        GRAP2380
C                        ALL PLOTTING DONE.  LOAD AND EXECUTE 'GENIS'    GRAP2390
C                        MAINLINE TO START NEW PROBLEM.                  GRAP2400
      CALL LINK (GENIS)                                                  GRAP2410
C                                                                        GRAP2420
      END                                                                GRAP2430
// DUP                                                                   GRAP2440
*STORE      WS  UA  GRAPH                                                GRAP2450
```

```
// FOR                                                                  GRID0000
*LIST ALL                                                               GRID0010
*ONE WORD INTEGERS                                                      GRID0020
      SUBROUTINE GRID                                                   GRID0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    GRID0040
CC                                                                      GRID0050
C                                                                       GRID0060
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             GRID0070
     1 XJMP,XJDG,FIMAX,RA,RG,C1,XK1,QBRG,CSHFR,CWM,PCMS,XLA,            GRID0080
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          GRID0090
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           GRID0100
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   GRID0110
      COMMON DT,CT,T,TEND,TIMSC                                         GRID0120
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  GRID0130
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      GRID0140
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TGRID0150
     1 EMDO                                                             GRID0160
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,GRID0170
     1QY,QERM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    GRID0180
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAGRID0190
     1SF,AMPTR,XGEN                                                     GRID0200
      COMMON CFY,SPYOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBGRID0210
      COMMON B(64),BB(64),E(32),EF(32)                                  GRID0220
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  GRID0230
      COMMON IELO,JELO,KELO,LELO,MELO                                   GRID0240
      COMMON SPARE(5),ISPAR(5)                                          GRID0250
CC                                                                      GRID0260
CC                                                                      GRID0270
C*********  PLOT GRID FOR FIRST 20 SECONDS ON TIME AXIS.                 GRID0280
CC                                                                      GRID0290
C                        LOWER PEN AT ORIGIN.                           GRID0300
      CALL FPLOT (2,0.,0.)                                              GRID0310
C                        SET UP SCALE FACTORS...                        GRID0320
CC                            TIME AXIS (X-AXIS) AS READ IN,            GRID0330
CC                            Y-AXIS AS UNIT.                           GRID0340
      SXX=SX                                                            GRID0350
      CALL SCALF (SXX,1.,0.,0.)                                         GRID0360
C                        PLOT FIRST 20 POSITIONS ON +X-AXIS.            GRID0370
      CALL FGRID (0,0.,0.,1.,20)                                        GRID0380
C                        SET CHARACTER ROTATION VALUE FOR LABELING      GRID0390
C                        OF X-AXIS (VERTICAL, TOP TO BOTTOM).           GRID0400
      TH=3.*PI/2.                                                       GRID0410
C                        COMPUTE VALUE FOR CENTERING X-AXIS LABELS      GRID0420
C                        ON TICK MARKS (0.05 INCHES ABSOLUTE SCALE).    GRID0430
      XD=.05/SXX                                                        GRID0440
C                        SET INITIAL VALUE OF LABEL.                    GRID0450
      X=1.                                                              GRID0460
C                        SET VALUE FOR PRINTING LABEL 0.05 INCHES BELOW GRID0470
C                        X-AXIS                                         GRID0480
      WYE=-0.05                                                         GRID0490
C                        LOOP TO PRINT FIRST 20 LABELS.                 GRID0500
      DO 10 I=1,20                                                      GRID0510
C                        MOVE PEN TO APPROPRIATE POSITION.              GRID0520
      CALL FCHAR (X-XD,WYE,.10,.10,TH)                                  GRID0530
```

```
C                          PRINT OUT LABEL.                                     GRID0540
      WRITE (7,5) X                                                            GRID0550
 5    FORMAT (F5.1)                                                            GRID0560
C                          UPDATE LABEL                                         GRID0570
 10   X=X+1.                                                                   GRID0580
C                                                                             GRID0590
C                                                                             GRID0600
C*********  DETERMINE IF TIME AXIS EXTENDS BEYOND 20 SECONDS AND FIND         GRID0610
C*********  END POINT IF SO.                                                  GRID0620
C                                                                             GRID0630
C                          MOVE PEN TO ORIGIN AND LOWER IT.                   GRID0640
      CALL FPLOT (-2.0,,0.)                                                   GRID0650
C                          CHECK IF NECESSARY TO EXTEND X-AXIS BEYOND         GRID0660
C                          20 SECONDS.                                        GRID0670
      TEST=TEND-20.                                                           GRID0680
      IF (TEST) 100,100,200                                                   GRID0690
C                          X-AXIS MUST BE EXTENDED.  COMPUTE FINAL VALUE      GRID0700
C                          TRUNCATE TO INTEGER                                GRID0710
 200  TEST1=TEST/10.0+1.0                                                     GRID0720
      ITEST=TEST1                                                             GRID0730
C                                                                             GRID0740
C                                                                             GRID0750
C*********  PLOT GRID FOR REMAINDER OF TIME AXIS.                             GRID0760
C                                                                             GRID0770
C                          REDUCE X-AXIS SCALE VALUE BY A FACTOR OF 10.       GRID0780
      SXX=.1*SX                                                               GRID0790
C                          SET NEW SCALE FACTORS.                             GRID0800
      CALL SCALF (SXX,1.,0.,0.)                                               GRID0810
C                          PLOT REMAINDER OF X-AXIS GRID.                     GRID0820
      CALL FGRID (0,200.,0.,10.,ITEST)                                        GRID0830
C                          COMPUTE VALUE TO CENTER X-AXIS LABELS ON           GRID0840
C                          TICK MARKS.                                        GRID0850
      XD=.05/SXX                                                              GRID0860
C                          SET INITIAL LABEL.                                 GRID0870
      X=30.                                                                   GRID0880
C                          LOOP TO PRINT REMAINING LABELS.                    GRID0890
      DO 11 I=1,ITEST                                                         GRID0900
C                          MOVE PEN TO APPROPRIATE POSITION.                  GRID0910
      CALL FCHAR (180.+X-XD,WYE,.10,.10,TH)                                   GRID0920
C                          PRINT LABEL.                                       GRID0930
      WRITE (7,5) X                                                           GRID0940
C                          UPDATE LABEL.                                      GRID0950
 11   X=X+10.                                                                 GRID0960
C                                                                             GRID0970
C                                                                             GRID0980
C*********  PLOT GRID FOR +Y AND -Y AXES.                                     GRID0990
C                                                                             GRID1000
C                          MOVE PEN TO ORIGIN AND LOWER IT.                   GRID1010
      CALL FPLOT (-2,0.,0.)                                                   GRID1020
C                          RESET ORIGINAL SCALE FACTORS.                      GRID1030
 100  CALL SCALF (SX,1.,0.,0.)                                                GRID1040
C                          PLOT GRID FOR +Y DIRECTION                         GRID1050
      CALL FGRID(1,0.,0.,.5,10)                                               GRID1060
C                          PLOT GRID FOR -Y DIRECTION                         GRID1070
```

```
      CALL FGRID (3.0,0.,.5,10)                                          GRID1080
C                         SET CHARACTER ROTATION VALUE FOR               GRID1090
C                         HORIZONTAL (ALONG X-AXIS)                      GRID1100
      TH=0.0                                                             GRID1110
C                         SET VALUE TO START PRINTING Y-AXIS LABELS 0.77 GRID1120
C                         INCHES TO THE LEFT OF THE ORIGIN.              GRID1130
      X=-0.77/SX                                                         GRID1140
C                         INITIALIZE Y-AXIS LABEL                        GRID1150
C                         SET VALUE FOR CENTERING OF LABELS ON TICK      GRID1160
C                         MARKS.                                         GRID1170
      WYE=5.                                                             GRID1180
      XD=.05                                                             GRID1190
C                         LOOP TO PRINT OUT LABELS.                      GRID1200
      DO 20 I=1,11                                                       GRID1210
      CALL FCHAR (X,WYE-XD,.10,.10,TH)                                   GRID1220
      WRITE(7,5) WYE                                                     GRID1230
 20   WYE=WYE-1.                                                         GRID1240
C                                                                        GRID1250
C                                                                        GRID1260
C*********  PLOT TITLE ON TIME AXIS.                                     GRID1270
C                                                                        GRID1280
C                         SET CHARACTER ROTATION VALUE FOR               GRID1290
C                         HORIZONTAL.                                    GRID1300
      TH=0.                                                              GRID1310
C                         MOVE PEN TO LOCATION FOR PRINTING X-AXIS       GRID1320
C                         TITLE.                                         GRID1330
      CALL FCHAR (3.,-1.25,.2,.2,TH)                                     GRID1340
C                         PRINT TITLE.                                   GRID1350
      WRITE (7,6)                                                        GRID1360
 6    FORMAT ('TIME(SECS)')                                              GRID1370
C                                                                        GRID1380
C                                                                        GRID1390
C*********  PLOT SCALE FACTORS FOR ALL THE VARIABLES BEING PLOTTED.      GRID1400
C                                                                        GRID1410
C                         SET CHARACTER ROTATION VALUE FOR VERTICAL      GRID1420
C                         BOTTOM TO TOP PRINTING.                        GRID1430
      TH=PI/2.                                                           GRID1440
C                         SET VALUE SO INITIAL LINE WILL BE 1 INCH TO    GRID1450
C                         LEFT OF ORIGIN.                                GRID1460
      X=-1.0/SX                                                          GRID1470
C                         LOOP THROUGH NUMBER OF VARIABLES TO BE PLOTTED.GRID1480
      DO 69 NIP=1,NOVAR                                                  GRID1490
C                         RETRIEVE SCALE FACTOR FOR VARIABLE IN 'SY'.    GRID1500
      CALL VAPLO                                                         GRID1510
C                         STORE VARIABLE CODE NUMBER.                    GRID1520
      MIKE=NVAR(NIP)                                                     GRID1530
C                         MOVE PEN TO BEGINNING OF LINE.  CHARACTERS     GRID1540
C                         WILL BE OF SIZE 0.2 INCHES SQUARE.             GRID1550
      CALL FCHAR (X,-1.5,.2,.2,TH)                                       GRID1560
C                         COMPUTE PLOT SCALE FOR VARIABLE.               GRID1570
      SYI=1./SY                                                          GRID1580
C                         BRANCH ACCORDING TO VARIABLE IN QUESTION.      GRID1590
      GO TO (30,40,50,60,70,30,90,92,94,96,98,101), MIKE                 GRID1600
C                                                                        GRID1610
```

```
C                          VARIABLE 'AI'    - ARM. CIR. CUR.                GRID1620
   30 WRITE (7,31) SYI                                                       GRID1630
   31 FORMAT('AI(1 IN.=',F7.1,'  AMPS)')                                     GRID1640
      GO TO 69                                                               GRID1650
C                          VARIABLE 'EB'    - BUS VOLTS                     GRID1660
   40 WRITE (7,41) SYI                                                       GRID1670
   41 FORMAT ('EB(1 IN.=',F5.1,' VOLTS)')                                    GRID1680
      GO TO 69                                                               GRID1690
C                          VARIABLE 'RPMPM' - DIESEL SPEED                  GRID1700
   50 WRITE (7,51) SYI                                                       GRID1710
   51 FORMAT ('NPM(1 IN.=',F5.1,' RPM)')                                     GRID1720
      GO TO 69                                                               GRID1730
C                          VARIABLE 'FMI'   - MOTOR FIELD CUR.              GRID1740
   60 WRITE (7,61) SYI                                                       GRID1750
   61 FORMAT ('FMI(1 IN.=',F5.1,' AMPS)')                                    GRID1760
      GO TO 69                                                               GRID1770
C                          VARIABLE 'RPMP'  - PROP. SPEED                   GRID1780
   70 WRITE (7,71) SYI                                                       GRID1790
   71 FORMAT ('NP(1 IN.=',F5.1,' RPM)')                                      GRID1800
      GO TO 69                                                               GRID1810
C                          VARIABLE 'XKW'   - POWER                         GRID1820
   80 WRITE (7,81) SYI                                                       GRID1830
   81 FORMAT ('KW(1 IN.=',F6.1,' KW)')                                       GRID1840
      GO TO 69                                                               GRID1850
C                          VARIABLE 'CVK'   - VESSEL SPEED                  GRID1860
   90 WRITE (7,91) SYI                                                       GRID1870
   91 FORMAT ('VK(1 IN.=',F5.1,' KTS.)')                                     GRID1880
      GO TO 69                                                               GRID1890
C                          VARIABLE 'QPI'   - PROP. TORQUE                  GRID1900
   92 WRITE (7,93) SYI                                                       GRID1910
   93 FORMAT ('QP(1 IN.=',F8.1,' FT-LBS)')                                   GRID1920
      GO TO 69                                                               GRID1930
C                          VARIABLE 'XPROI' - PROP. THRUST                  GRID1940
   94 WRITE (7,95) SYI                                                       GRID1950
   95 FORMAT ('TP(1 IN.=',F8.1,' LBS)')                                      GRID1960
      GO TO 69                                                               GRID1970
C                          VARIABLE 'CXS'   - DISTANCE                      GRID1980
   96 WRITE (7,97) SYI                                                       GRID1990
   97 FORMAT ('XS(1 IN.=',F6.1,' FEET)')                                     GRID2000
      GO TO 69                                                               GRID2010
C                          VARIABLE 'QI'    - ICE TORQUE                    GRID2020
   98 WRITE(7,99) SYI                                                        GRID2030
   99 FORMAT('QI(1 IN.=',F8.1,'FT-LBS)')                                     GRID2040
      GO TO 69                                                               GRID2050
C                          VARIABLE 'TEMP'  - MOTOR TEMP.                   GRID2060
  101 WRITE(7,102) SYI                                                       GRID2070
  102 FORMAT('TEMP(1 IN.=',F5.1,'DEG.CENT.)')                                GRID2080
C                                                                            GRID2090
C                          COMPUTE POSITION FOR NEXT LINE TO BE 0.4 INCHES  GRID2100
C                          FURTHER LEFT OF ORIGIN THAN PREVIOUS LINE.       GRID2110
 69   X=X-(.4/SX)                                                           GRID2120
C                                                                            GRID2130
C                                                                            GRID2140
C********  MOVE PEN TO ORIGIN IN THE UP-POSITION AND RETURN.                 GRID2150
```

```
C
      CALL FPLOT (0.,0.,0.)                                          GRID2160
      RETURN                                                         GRID2170
C                                                                    GRID2180
      END                                                            GRID2190
// DUP                                                               GRID2200
*STORE      WS  UA  GRID                                             GRID.210
                                                                     GRID2.20
```

```
// FOR                                                                  SDAT0000
*LIST ALL                                                               SDAT0010
*ONE WORD INTEGERS                                                      SDAT0020
      SUBROUTINE DATA                                                   SDAT0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    SDAT0040
C                                                                       SDAT0050
C                                                                       SDAT0060
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,XJFG,            SDAT0070
     1 XJVP,XJDG,FIMAX,RA,RG,C1,XK1,QBR,ER,CWM,PCMS,XLA,                SDAT0080
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALP4,GS                             SDAT0090
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPL,DATA,NOP,NON,NIP               SDAT0100
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   SDAT0110
      COMMON DT,CT,T,TEND,TIMSC                                         SDAT0120
      COMMON Z,Y,XNPY,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  SDAT0130
      COMMON CZ,CY,CXNPY,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      SDAT0140
      COMMON ZDO,YDO,XNPYD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TSDAT0150
     2 EXDO                                                             SDAT0160
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,SDAT0170
     1QM,QFCM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    SDAT0180
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLASDAT0190
     1SF,AVPTR,XGEN                                                     SDAT0200
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBSDAT0210
      COMMON B(64),BB(64),E(32),EE(32)                                  SDAT0220
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  SDAT0230
      COMMON IELO,JELO,KELO,LELO,MELO                                   SDAT0240
      COMMON SPARE(5),ISPAR(5)                                          SDAT0250
C                                                                       SDAT0260
C                                                                       SDAT0270
C                     READ LATEST VALUES OF VARIABLES FROM DISK         SDAT0280
C                     USING CURRENT VALUE OF 'IELO' AS INDEX. 'IELO'    SDAT0290
C                     WILL BE UPDATED TO NEXT RECORD AUTOMATICALLY.     SDAT0300
      READ(1'IELO)CT,AI,EB,RPMPM,FMI,RPMP,XKW,CVK,QPI,XPROI,CXS         SDAT0310
     1,QI,TEMP                                                          SDAT0320
C                                                                       SDAT0330
C                     STORE NUMBER OF VARIABLE WE ARE DEALING WITH      SDAT0340
C                     AND BRANCH ACCORDINGLY TO RETRIEVE ITS VALUE.     SDAT0350
      NOOK=NVAR(JJ)                                                     SDAT0360
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120), NOOK              SDAT0370
C                                                                       SDAT0380
C                     VARIABLE 'AI'     - ARM. CIR. CUR.                SDAT0390
 10   VAR=AI                                                            SDAT0400
      RETURN                                                            SDAT0410
C                                                                       SDAT0420
C                     VARIABLE 'EB'     - BUS VOLTS                     SDAT0430
 20   VAR=EB                                                            SDAT0440
      RETURN                                                            SDAT0450
C                                                                       SDAT0460
C                     VARIABLE 'RPMPM' - DIESEL SPEED                   SDAT0470
 30   VAR=RPMPM                                                         SDAT0480
      RETURN                                                            SDAT0490
C                                                                       SDAT0500
C                     VARIABLE 'FMI'    - MOTOR FIELD CUR.              SDAT0510
 40   VAR=FMI                                                           SDAT0520
      RETURN                                                            SDAT0530
```

```
C                                                                       SDAT0540
C                          VARIABLE 'RPMP'  - PROP. SPEED               SDAT0550
 50   VAR=RPMP                                                          SDAT0560
      RETURN                                                            SDAT0570
C                                                                       SDAT0580
C                          VARIABLE 'XKW'   - POWER                     SDAT0590
   60 VAR=XKW                                                           SDAT0600
      RETURN                                                            SDAT0610
C                                                                       SDAT0620
C                          VARIABLE 'CVK'   - VESSEL SPEED              SDAT0630
   70 VAR=CVK                                                           SDAT0640
      RETURN                                                            SDAT0650
                                                                        SDAT0660
                           VARIABLE 'QPI'   - PROP. TORQUE              SDAT0670
   80 VAR=QPI                                                           SDAT0680
      RETURN                                                            SDAT0690
C                                                                       SDAT0700
C                          VARIABLE 'XPROI' - PROP. THRUST              SDAT0710
   90 VAR=XPROI                                                         SDAT0720
      RETURN                                                            SDAT0730
C                                                                       SDAT0740
C                          VARIABLE 'CXS'   - DISTANCE                  SDAT0750
  100 VAR=CXS                                                           SDAT0760
      RETURN                                                            SDAT0770
C                                                                       SDAT0780
C                          VARIABLE 'QI'    - ICE TORQUE                SDAT0790
 110  VAR=QI                                                            SDAT0800
      RETURN                                                            SDAT0810
C                                                                       SDAT0820
C                          VARIABLE 'TEMP'  - MOTOR TEMP.               SDAT0830
 120  VAR=TEMP                                                          SDAT0840
      RETURN                                                            SDAT0850
C                                                                       SDAT0860
      END                                                               SDAT0870
// DUP                                                                  SDAT0880
*STORE      WS  UA  DATA                                                SDAT0890
```

```
// FOR                                                                  PRIN0000
*LIST ALL                                                               PRIN0010
*ONE WORD INTEGERS                                                      PRIN0020
      SUBROUTINE PRINT                                                  PRIN0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    PRIN0040
C                                                                       PRIN0050
C                                                                       PRIN0060
      COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             PRIN0070
     1 XJMP,XJDG,EMAX,RA,RG,C1,XK1,GBRG,CSHFR,CWM,PCMS,XLA,             PRIN0080
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          PRIN0090
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           PRIN0100
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   PRIN0110
      COMMON DT,CT,T,TEND,TIMSC                                         PRIN0120
      COMMON Z,Y,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP                  PRIN0130
      COMMON CZ,CY,CXNPM,CFGI,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      PRIN0140
      COMMON ZDO,YDO,XNPMD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TPRIN0150
     1 EMDO                                                             PRIN0160
      COMMON XNS,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,GDE,OG,EG,EB,EC,PRIN0170
     1OM,CFRM,AIMAT,GP,GPI,QI,TP,XPROI,R,VAR,XKW,CVK                    PRIN0180
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAPRIN0190
     1SF,AMPTR,XGEN                                                     PRIN0200
      COMMON CFY,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBPRIN0210
      COMMON B(64),BB(64),E(32),EE(32)                                  PRIN0220
      COMMON EMP(20),WDED,WADED(20),TDED,THDED(20),XKK                  PRIN0230
      COMMON IELO,JELO,KELO,LELO,MELO                                   PRIN0240
      COMMON SPARE(5),ISPAR(5)                                          PRIN0250
C                                                                       PRIN0260
C                                                                       PRIN0270
C                     STORE NUMBER OF VARIABLE WE ARE DEALING WITH      PRIN0280
C                     AND BRANCH ACCORDINGLY TO PRINT OUT THE           PRIN0290
C                     APPROPRIATE LABEL ON THE PLOTTER.                 PRIN0300
      MIKE=NVAR(JJ)                                                     PRIN0310
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120), MIKE              PRIN0320
C                                                                       PRIN0330
C                     VARIABLE 'AI'    - ARM. CIR. CUR.                 PRIN0340
   10 WRITE (7,11)                                                      PRIN0350
   11 FORMAT ('---ARM. CIR. CUR.')                                      PRIN0360
      RETURN                                                            PRIN0370
C                                                                       PRIN0380
C                     VARIABLE 'EB'    - BUS VOLTS                      PRIN0390
   20 WRITE (7,21)                                                      PRIN0400
   21 FORMAT ('---BUS VOLTS')                                           PRIN0410
      RETURN                                                            PRIN0420
C                                                                       PRIN0430
C                     VARIABLE 'RPMPM' - DIESEL SPEED                   PRIN0440
   30 WRITE (7,31)                                                      PRIN0450
   31 FORMAT ('---DIESEL SPEED')                                        PRIN0460
      RETURN                                                            PRIN0470
C                                                                       PRIN0480
C                     VARIABLE 'FMI'   - MOTOR FIELD CUR.               PRIN0490
   40 WRITE (7,41)                                                      PRIN0500
   41 FORMAT ('---MOTOR FIELD CUR.')                                    PRIN0510
      RETURN                                                            PRIN0520
C                                                                       PRIN0530
```

```
C                    /          VARIABLE 'RPMP'  - PROP. SPEED                PRIN0540
   50 WRITE (7,51)                                                            PRIN0550
   51 FORMAT ('---PROP SPEED')                                                PRIN0560
      RETURN                                                                  PRIN0570
C                                                                             PRIN0580
C                             VARIABLE 'XKW'   - POWER                        PRIN0590
   60 WRITE (7,61)                                                            PRIN0600
   61 FORMAT ('---POWER')                                                     PRIN0610
      RETURN                                                                  PRIN0620
C                                                                             PRIN0630
C                             VARIABLE 'CVK'   - VESSEL SPEED                 PRIN0640
   70 WRITE (7,71)                                                            PRIN0650
   71 FORMAT ('---VESSEL SPEED')                                              PRIN0660
      RETURN                                                                  PRIN0670
C                                                                             PRIN0680
C                             VARIABLE 'QPI'   - PROP. TORQUE                 PRIN0690
   80 WRITE (7,81)                                                            PRIN0700
   81 FORMAT ('---PROP TORQUE')                                               PRIN0710
      RETURN                                                                  PRIN0720
C                                                                             PRIN0730
C                             VARIABLE 'XPROI' - PROP. THRUST                 PRIN0740
   90 WRITE (7,91)                                                            PRIN0750
   91 FORMAT ('---PROP THRUST')                                               PRIN0760
      RETURN                                                                  PRIN0770
C                                                                             PRIN0780
C                             VARIABLE 'CXS'   - DISTANCE                     PRIN0790
  100 WRITE (7,101)                                                           PRIN0800
  101 FORMAT ('---DISTANCE')                                                  PRIN0810
      RETURN                                                                  PRIN0820
C                                                                             PRIN0830
C                             VARIABLE 'QI'    - ICE TORQUE                   PRIN0840
  110  WRITE(7,111)                                                          PRIN0850
  111  FORMAT('--- ICE TORQUE')                                              PRIN0860
       RETURN                                                                 PRIN0870
C                                                                             PRIN0880
C                             VARIABLE 'TEMP'  - MOTOR TEMP.                  PRIN0890
  120  WRITE(7,121)                                                          PRIN0900
  121  FORMAT('--- MOTOR TEMP.')                                             PRIN0910
       RETURN                                                                 PRIN0920
C                                                                             PRIN0930
       END                                                                    PRIN0940
// DUP                                                                        PRIN0950
*STORE      WS  UA  PRINT                                                    PRIN0960
```

```
// FOR                                                                  VAPL0000
*LIST ALL                                                               VAPL0010
*ONE WORD INTEGERS                                                      VAPL0020
      SUBROUTINE VAPLO                                                  VAPL0030
C     VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968                    VAPL0040
C                                                                       VAPL0050
C                                                                       VAPL0060
      COMMON RO,PI,IZR,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,             VAPL0070
     1 XJVP,XJDG,FIMAX,RA,RG,C1,XK1,OBRG,CSHFR,CWM,PCMS,XLA,            VAPL0080
     2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS                          VAPL0090
      COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP           VAPL0100
      COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12   VAPL0110
      COMMON DT,CT,T,TEND,TIMSC                                         VAPL0120
      COMMON Z,Y,XNPM,FGI,FID,XNP,VS,XS,SIE,AI,RV,TEMP                  VAPL0130
      COMMON CZ,CY,CXNPM,CFGI,CFID,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP      VAPL0140
      COMMON ZDO,YDO,XNPMD,FGIDO,FIDDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TVAPL0150
     1 TMDO                                                             VAPL0160
      COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,QDE,QG,EG,EB,EC,VAPL0170
     1QF,QFRM,AIHAT,QP,QPI,QI,TP,XPROI,R,VAR,XKW,CVK                    VAPL0180
      COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLAVAPL0190
     1SF,AMPTR,XCFN                                                     VAPL0200
      COMMON CFM,SPMOT,CFMA(13),TRNFM,ZMP,RFM,XLFM,TAUFM,FMIRV,TAUA,TAUBVAPL0210
      COMMON B(64),BB(64),E(32),EE(32)                                  VAPL0220
      COMMON EHP(20),WDED,WADED(20),TDED,THDED(20),XKK                  VAPL0230
      COMMON IELO,JELO,KELO,LELO,MELO                                   VAPL0240
      COMMON SPARE(5),ISPAR(5)                                          VAPL0250
C                                                                       VAPL0260
C                                                                       VAPL0270
C                       STORE NUMBER OF VARIABLE WE ARE DEALING WITH    VAPL0280
C                       AND BRANCH ACCORDINGLY TO RETRIEVE ITS SCALE    VAPL0290
C                       FACTOR FOR PLOTTING.                            VAPL0300
      NAN=NVAR(NIP)                                                     VAPL0310
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120), NAN               VAPL0320
C                                                                       VAPL0330
C                       VARIABLE 'AI'    - ARM. CIR. CUR.               VAPL0340
10    SY=SY1                                                            VAPL0350
      RETURN                                                            VAPL0360
C                                                                       VAPL0370
C                       VARIABLE 'EB'    - BUS VOLTS                    VAPL0380
20    SY=SY2                                                            VAPL0390
      RETURN                                                            VAPL0400
C                                                                       VAPL0410
C                       VARIABLE 'RPMPM' - DIESEL SPEED                 VAPL0420
30    SY=SY3                                                            VAPL0430
      RETURN                                                            VAPL0440
C                                                                       VAPL0450
C                       VARIABLE 'FMI'   - MOTOR FIELD CUR.             VAPL0460
40    SY=SY4                                                            VAPL0470
      RETURN                                                            VAPL0480
C                                                                       VAPL0490
C                       VARIABLE 'RPMP'  - PROP. SPEED                  VAPL0500
50    SY=SY5                                                            VAPL0510
      RETURN                                                            VAPL0520
C                                                                       VAPL0530
```

```
C                              VARIABLE 'XKW'    - POWER                      VAPL0540
   60 SY=SY6                                                                  VAPL0550
      RETURN                                                                  VAPL0560
C                                                                             VAPL0570
C                              VARIABLE 'CVK'    - VESSEL SPEED               VAPL0580
   70 SY=SY7                                                                  VAPL0590
      RETURN                                                                  VAPL0600
C                                                                             VAPL0610
C                              VARIABLE 'QPI'    - PROP. TORQUE               VAPL0620
   80 SY=SY8                                                                  VAPL0630
      RETURN                                                                  VAPL0640
C                                                                             VAPL0650
C                              VARIABLE 'XPROT'  - PROP. THRUST               VAPL0660
   90 SY=SY9                                                                  VAPL0670
      RETURN                                                                  VAPL0680
C                                                                             VAPL0690
C                              VARIABLE 'CXS'    - DISTANCE                   VAPL0700
  100 SY=SY10                                                                 VAPL0710
      RETURN                                                                  VAPL0720
C                                                                             VAPL0730
C                              VARIABLE 'QI'     - ICE TORQUE                 VAPL0740
  110 SY=SY11                                                                 VAPL0750
      RETURN                                                                  VAPL0760
C                                                                             VAPL0770
C                              VARIABLE 'TEMP'   - MOTOR TEMP.                VAPL0780
  120 SY=SY12                                                                 VAPL0790
      RETURN                                                                  VAPL0800
C                                                                             VAPL0810
      END                                                                     VAPL0820
// DUP                                                                        VAPL0830
*STORE      WS  UA  VAPLO                                                     VAPL0840
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
U. S. Coast Guard Headquarters
1300 E Street, N. W.
Washington, D. C. 20591

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
None

3. REPORT TITLE
Ship Design and Maintenance Computer Program
Icebreaker Propulsion System Simulation (Diesel-Electric)

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)
Jack W. Lewis

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1968	257	3

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 1702 Replacement Icebreakers	ENE-11
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT
This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Commandant(E), U.S. Coast Guard Headquarters, 1300 E St., N.W., Washington, D.C. 20591

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U. S. Coast Guard

13. ABSTRACT

This document describes a digital computer program written for the IBM 1130 Computing System that is capable of simulating transient conditions within a diesel-electric propulsion system typical of installations aboard most icebreaker type vessels. The program will simulate various conditions to which an icebreaker may be subjected. These include "crash reversals", acceleration under free route or ice clogged channel conditions, ice ramming and subjection of the propellers to ice encounters. The program accomplishes the simulation by solving numerically the set of simultaneous non-linear differential equations which describe the dynamic behavior of the complete propulsion system including propeller and ship motion dynamics. At the option of the user, output data can be plotted.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Icebreakers						
Ship Design						
IBM 1130						
Computer Program						
Simulation						
Propulsion Control						
Automated Control Systems						
Propulsion Plants						
Biesels						
Electric Drive						
Propellers						

DD FORM 1 NOV 65 1473 (BACK)