COPY NO. 17

FINAL REPORT

DEVELOPMENT OF A WEIGHT/SIZING DESIGN SYNTHESIS COMPUTER PROGRAM

28 FEBRUARY 1973 MDC E0746

VOLUME II
PROGRAM DESCRIPTION

SUBMITTED TO
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058

CONTRACT NAS 9-12989

PREPARED BY: J. M. Garrison
J. M. GARRISON, STUDY MANAGER

APPROVED BY: J. J. Morgan
J. J. MORGAN, MANAGER WEIGHTS

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST
Saint Louis, Missouri 63166 (314) 232-0232

MCDONNELL DOUGLAS CORPORATION

FOREWORD

The Weight/Sizing Design Synthesis Computer Program was developed by McDonnell Douglas Astronautics Company - East under Contract NAS 9-12989 for the National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, Houston, Texas. The contract involved a study to derive basic weight estimation relationships for those elements of the Space Shuttle vehicle which contribute a significant portion of the inert weight. These relationships measure the pacing parameters of load, geometry, material, and environment. The weight estimation relationships are then combined into the Weight/Sizing Design Synthesis Computer Program.

This report is submitted in three volumes;

I	Program Formulation
II	Program Description
III	User Manual

This Volume contains a listing for each module and subroutine of the program. Also included is a generalized flow chart describing the subroutine linkage of the complete program, plus detailed flow charts of each subprogram.

ACKNOWLEDGEMENTS

The following McDonnell Douglas Astronautics Company - East personnel were the major contributors to the technical contents of this study.

L. M. Gnojewski/R. W. Ridenour	Program Coding/Assembly Integration
B. A. Grob	External Tank & Empirical Equations
J. J. Morgan	Wing
J. M. Garrison	Structure Models

The Technical Monitor for the National Aeronautics and Space Administration, Mr. Norman A. Piercy, of the Engineering Technology Branch, provided valuable guidance and direction throughout the study.

TABLE OF CONTENTS

List of Pages

1. INTRODUCTION

The primary objective of this study was the development of a Weight/Sizing Design Synthesis Methodology to be used in support of the mainline Space Shuttle Program. This methodology has a minimum number of data inputs and quick turn around capabilities consistent with the objectives of enabling the NASA to rapidly: (a) make weight comparisons between the current Shuttle configuration and proposed changes, (b) determine the effects of various subsystem trade on total system weight, and (c) determine the effects of weight on performance and performance on weight. The Executive Sizing and Performance (ESPER) program is the culmination of this development effort.

The complete listing of the input and output variables, as well as a complete program description, are found in Volume I, Program Formulation, and again, along with usage instructions, in Volume III, The User Manual. The program listings in this volume are as they actually appear in ESPER. They are comprised of statements belonging to one of two general classes:

1. executable statements that perform computations, input/output operations, and program flow control.
2. nonexecutable statements that provide information to the compiler about storage assignments, data types, and program form, as well as providing information to the program and subroutines during execution about input/output formats and data initialization.

These programs differ considerably from those in Volume I in that they are not self-sustaining, but rather, they are totally dependent upon the iterative computational sequence in ESPER for their execution.

2. ESPER PROGRAM

The ESPER program is a multioption sizing/synthesis program geared to the Solid Rocket Motor (SRM) Booster in parallel with an external hydrogen/oxygen tank Orbiter for either the easterly (28-1/2 deg inclination) polar (90-deg inclination), or resupply (55-deg inclination) missions. Although these are the primary missions of current interest, the program will handle any mission falling within the inclination constraints of 28.5 deg to 90 deg as shown in Figure 2-7, Page 2-17 of Volume I. The program has two primary options:

(a) fixed hardware, and

(b) iterative vehicle sizing.

The fixed hardware option determines the payload capability of a given configuration. This allows the user to determine the effect on performance of configuration and/or criteria changes, either real or proposed.

The iterative vehicle sizing option physically sizes the vehicles for a given payload. It determines the size of the SRM and its propellant load, and the size of the external tank and its corresponding propellant load. The iterative procedure is based on the sizing criteria of a fixed staging velocity or it will size the vehicle to a minimum gross lift off weight (GLOW). The minimum GLOW option is provided as it is generally associated with a minimum cost operation.

In turn, either of the sizing requirements can be run with a fixed thrust option in which both the booster and orbiter thrust are set at given values, and the propellant requirements are determined, or the orbiter thrust can be fixed and the first stage thrust-to-weight ratio input. The fixed thrust-to-weight options determines the booster engine size plus the propellant requirements.

Each of the vehicles has several modes of analysis available. The orbiter, external tank, and booster weight can be determined by the option of detail analysis, while maintaining a user input dry weight, or no analysis but simply utilizing an input weight to represent the vehicle. In addition, the external tank and the booster are represented by simplified equations in which the parameters of interest are curve-fit to determine the vehicle weight.

In addition to printing out the performance parameters, the option is available to print out the detail subsystems weights of each vehicle, providing a line item comparion with the current Shuttle vehicle. The other option would be a simplified printout, containing only the vehicle dry or burn out weight as listed in the performance parameter.

Two performance subroutines are tied into the ESPER programs to allow the user to determine growth characteristics or vehicle sensitivities.

ESPER is fundamentally based on the control logic found in the Analytical Parametric Systems Evaluation (APSE) program delivered to NASA at ATP plus 6 weeks. APSE is primarily a multivehicle program in which many types of vehicles and configurations can be compared, i.e., fully reusable configuration, external hydrogen tank orbiter, pressure feed booster, series burn, as well as the current baseline solid rocket motor (SRM) booster with an external hydrogen/oxygen tank orbiter.

Inherent with the multivehicle concept are extremely simplified weight relations. The weight equations in APSE consisted primarily of mass fractions, with the booster being a function of thrust and propellant load, the orbiter a function of thrust, and the external tank a function of required propellant. These mass fractions were derived from study point designs, and required continual updating to meet the ever changing criteria. With ESPER being based on the current baseline vehicle, and the multivehicle studies dropped, the major emphasis of this study was directed to the expansion of the weight relationships.

Figure 2-1 presents a simplified flow chart of the ESPER program. The program consists of three vehicle modules, two functional modules, and three performance subroutines. The vehicle modules contain the analytical and empirical equations and relationships required to completely define the orbiter and booster,

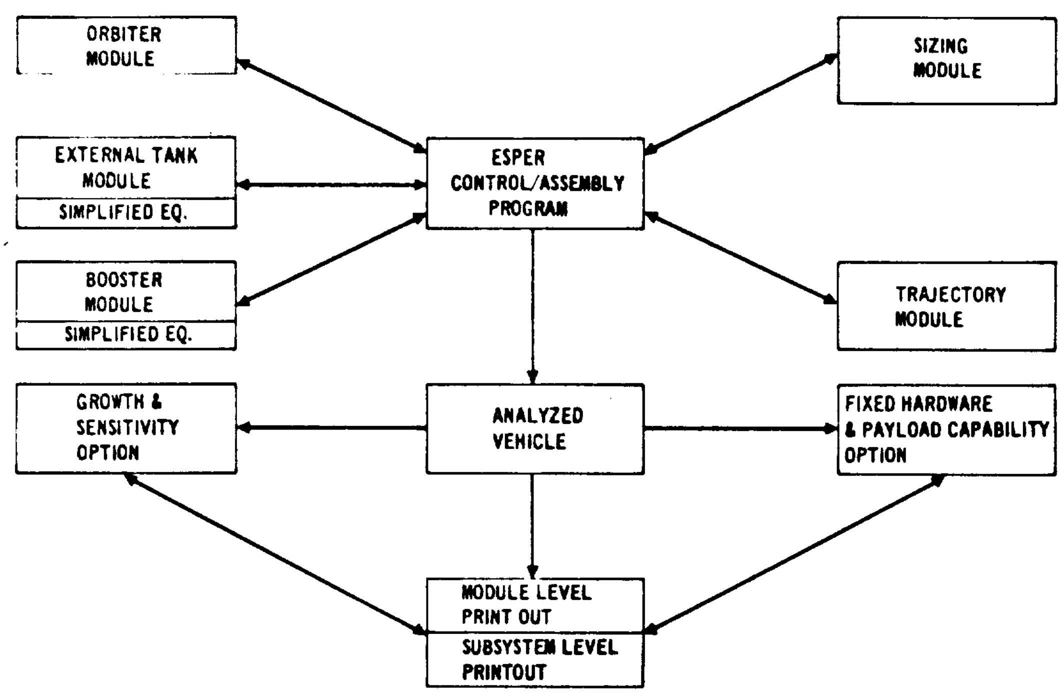

FIGURE 2-1 ESPER FLOW CHART

and the external tank respectively. These equations will measure the pacing parameters of load, material, and geometry. The functional modules describe the vehicle sizing and the trajectory analysis. The output is an analyzed vehicle which, when coupled with the performance subroutines, will allow the user to derive growth accommodations, sensitivities, and payload capabilities. These modules and subroutines operate under the logic and direction of the Control/Assembly program. This program, in essence the heart of ESPER, integrates the vehicle modules and combines them in an iterative sequence for the orbiter, external tank, and booster. This iterative sequence, however, is under the complete control of the user and is altered according to the option specified by the user in the 'PERF' (Performance) data block. The use of separate modules, which contain their own input/output common blocks, provide a systematic means of controlling the logic flow in the programs. The ESPER Control/Assembly logic is shown in the flow diagram, Figure 2-2, which is followed by a listing of this program.

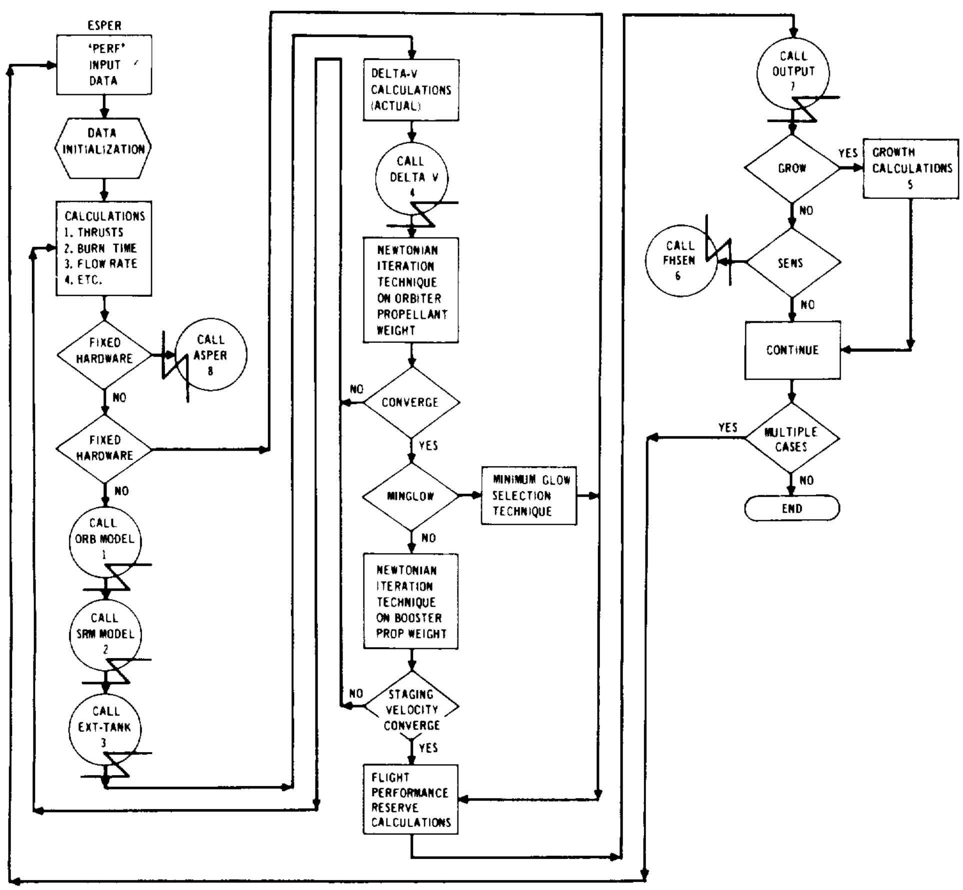

FIGURE 2-2 ESPER FLOW DIAGRAM

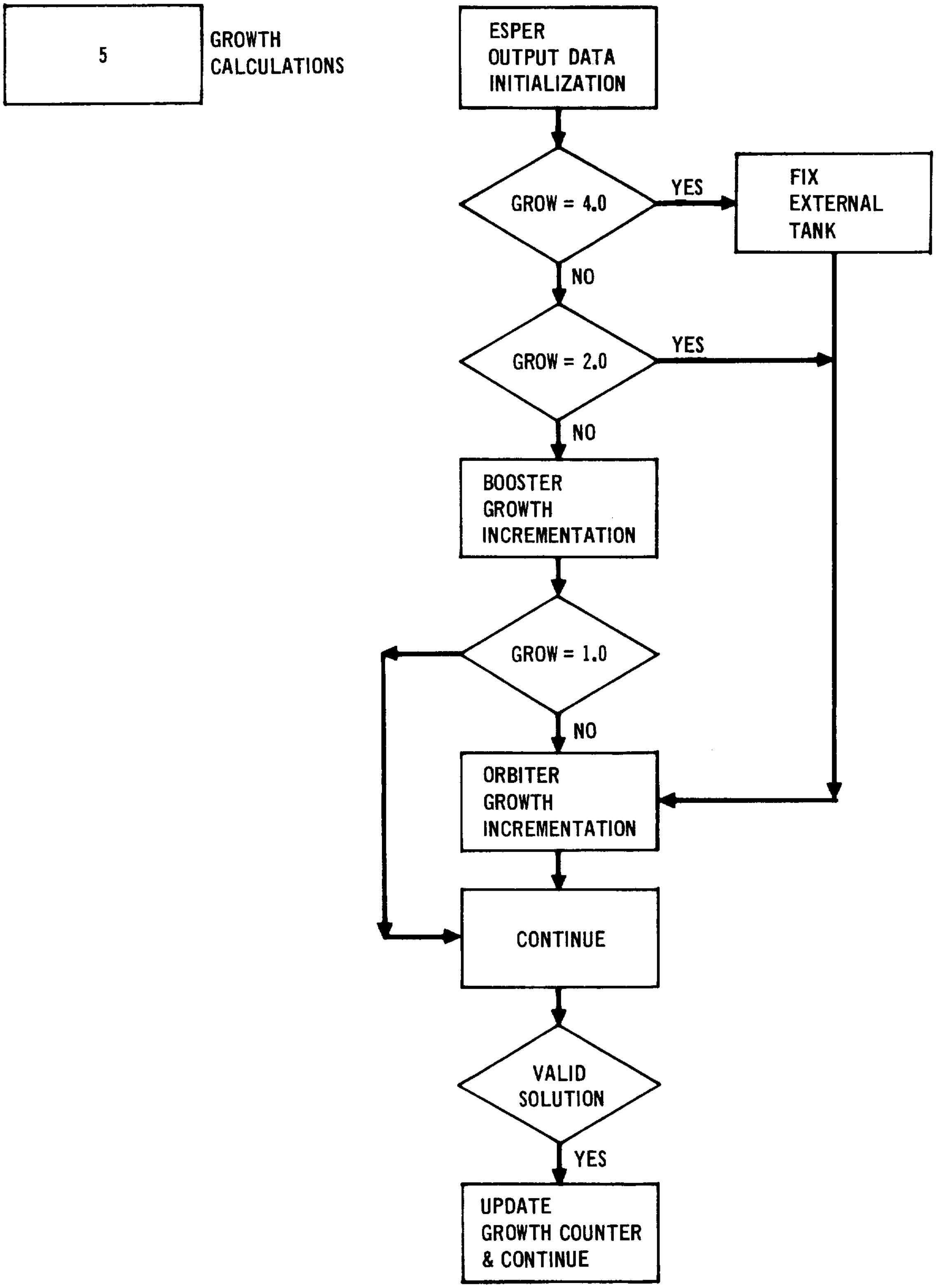

FIGURE 2-2 ESPER FLOW DIAGRAM (Continued)

To facilitate rapid turnaround and ease of updating for major configuration changes, the modularized concept was utilized. The vehicle modules are comprised of data files and subsystem models.

These data files are as follows:

1. ØRB & ØAERØ - ORBITER MODULE DATA
2. EXT - EXTERNAL TANK MODULE DATA
3. SRM - SRM MODULE DATA

Collectively separating the data in this fashion serves two purposes, (1) it greatly facilitates the locating and changing of input parameters, (2) if an entire module is replaced it becomes an easy task to replace the existing data block with the new one.

Linkage of data between the vehicle modules and their subsystem models is accomplished through named common blocks ending in the letter 'D' (for example, 'STRD,' where 'STR' designates the model into which the data is to be transferred, and 'D' designates a Data common). Parameters calculated in each model are in turn linked back to their respective module through argument lists. These calculated module parameters are linked to the rest of the main program through named common blocks ending in the letter 'O' (for example, 'SRMØ,' where 'SRM' designates the module in which they were generated, and 'O' designates an Output common).

The entire Formulation Concept of ESPER, that is, utilizing modularization, utilizing subsystem models, collectively separating data and common block linkage, elevates ESPER from just another sizing program to a dynamically powerful tool that can evolve in complexity with the mainline Shuttle program, thus adverting the pitfall of obsolescence.

This Formulation Concept makes the evoluation process by the user an easy task, for subsystem modifications can be made without affecting the rest of the module, and module replacement can be accomplished without affecting the rest of the program. The user needs only to program the new module or subsystem, correct the common linkage and proper functioning is insured.

COPY LSUB TO LP(K,NC)

```
$FIXED
C
C                                   ESPER
C
C     THIS PROGRAM INTEGRATES THE VEHICLE MODULES AND
C     COMBINES THEM IN AN ITERATIVE COMPUTATIONAL SEQUENCE
C     FOR THE ORBITER,EXTERNAL TANK,AND SRM.THE SPECIFIC
C     WEIGHT RELATIONSHIPS DEVELOPED FOR EACH VEHICLE MODULE
C     AND THE ASCENT TRAJECTORY CURVE FIT OF VELOCITY LOSSES
C     FEED INTO THIS CONTROL/ASSEMBLY PROGRAM.
      IMPLICIT REAL (A-Z)
      DIMENSION BDVC(3),ODVC(3),PRB(3),PRO(3)
      DIMENSION MGOL(2),MGDVB(2)
      INTEGER J,NI
C
C     PERFORMANCE COMMON BLOCK
C
       COMMON/MAIN/ PROPB,PROPO,BOT,BCANT,OCANT,OCANTY
     1,OCANTP,NOENGB,NOENGO,THOSL,THBSL,THOV,TBV,FLOWR
     2,TF,FTW,FIXHRD
     3,ISPOS,ISPOV,ISPBOS,ISPBOV,SCD,BTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THOTC,THOSLT,THBTC,ISPO,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVONC,DVB,DVTOTC,WOSCD,OINWT,OLANWT
     7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
C
C     SRM OUTPUT COMMON BLOCK
C
       COMMON/SRMO/BOOWT,BDRYWT,BGLOW,LAMB,PWOI
     1,MASSRM,WCASE,WJOINT,WNOZZ,WTTER,WINST,WIGN,BSRMC
     2,SRMISS,PWES,PWASLS,PWAS,PWNF,PWTN,PWAV,WNCTPS,SRMIC
     3,WRECOV,PWPAR,PWPI,PWRR,PWRP,PWWR,SRMRC
     4,UNCERT,EXPINS,BOSLUN,SRML,SRMD
     5,PGROSS,PROBWT,PPROPB,PDRYWT,PBOSLJ
     6,FIXBOO,SIMPBO
C
C     ORBITER OUTPUT COMMON BLOCK
C
       COMMON/ORBO/R1,R2,RL,WTAJX,WTACS,ACSENG,ACSSYS,WTACTK
     1,ACSMOD,WTOMS,OMSENG,PROPSY,WTOMTK,MODULE
     1,SURFC,PPWR,ELEC,HYDR,AVIONO,ECLSO,PPROV,OUNCWT
     2,OROMIS,TABPRO,SURFK
     1,PERSON,ORESD,ORESV,PLOADU,PLOADD,ACSPRO,WOPROP,SUDLE
     2,FIXORB,FIXWOR,ODRYWT,ORIFL
     1,WWT,WSS,WBSTR,WTORBE,WTORBC,LEW,WTE,WAIL,WAS,WADR,WAH
     2,WAP,GPROV,PWINGK,TAIL,TSG,TBSTR,TTORQB,TLE,WRUD,WRS
     3,WRDR,WRH,WRP,PTAILK
     1,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36
     1,TOTTPS,TWGWT,WGWT,WGLEWT,TWT,TLEWT,BLBTPS
     2,BASEWT,IBTWT,PTPSCN,TTWT,BTPSWT,MCSWT,LDTWT
     3,PROWT,PPPC,PHYC,SCWT
     1,TAPROP,ENGPAC,ENG,TVC,CONTR,PRPUTL,PROSYS
     2,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC
     1,LNDOK,NG1,NG2,NG3,NGEAR,MG1,MG2,MG3,MGEAR,AX1,AX2
     2,AX3,AXGEAR,LNDOKK
```

```
C
C     EXTERNAL TANK ØUTPUT CØMMØN BLØCK
C
      CØMMØN/EXTØ/BØDGRP,TØTPS,FWDTK,FAIRT,FWDBLF,FCCTPS
     1,CØNSCT,TPSIN,CYLSCT,ACYDM,AFTBLF,WINT,PRØSY,AFTNK
     2,FEDSYS,FWDBLA,PRSVNT,AFTCYL,SUMP,AFTBLA,PNPJ,TWINT
     3,NØSFAR,AVIØNT,JMBPNL,WRETRØ,TUNNEL,MISCT,BAFF,SUBDRY
     4,GU,DRYWT,RESIDT,UNDRAN,FEEDTR,PRSURT,FBIAS,INERT
     5,GRØSSW,TLAMB,ØTRAP,EXTL,EXTD,BLKHD,EXTHØ,EXTHH,SIMPTK
C
C     DELTAV ØUTPUT CØMMØN BLØCK
C
      CØMMØN/DVØ/DVT,DVØN,DVR,DVØR,DVBRP,X2,X3
     1,DVEPR,TØTLSS,DVØNR,DVALT
C
C     PERFØRMANCE (PERF) INPUT DATA BLØCK
C
      NAMELIST PRØPBG,PRØPBG,BCANT,BCANTY,BCANTP
     1,NØENGB,NØENGØ,THBSL,THØSL,THØVI
     2,TF,FTW,FIXHRD,ISPBS
     3,ISPBV,ISPØBS,ISPØBV,SCD,BTW
     4,H,DVCØRR,INC,STAGV,REL,DVCØN,DVCNST
     5,BDRYWT,BBØWT,ØLLPLB,INERT,MATCH,TLSSR
     6,DRYWT,RESIDT,MR,ØMSISP,ØMSDVT,ØMSDVP,ØMR
     7,SENS,GRØW,NI,GRØWØ,GRØWB,LØNGP,MINGLW
C
      CALL EØFSET(9999S)
      F=0.0
      PIN=0.0
    1 INPUT(1)
      IF(F.EQ.0.0) TØV=THØVI
      THØV=TØV
      PIN1=PRØPBG
      PIN2=PRØPØG
      ØCANT=1.0
      ØGLØW=1000000.
      ØLANWT=100000.
      ØLØWT=100000.
      ØLØWLØ=100000.
      THBSLT=2000000.
      P=0.0
      J=1
      MG=0.0
      MGDVB(1)=0.0
      MGDVB(2)=0.0
      MGGL(1)=0.0
      MGGL(2)=40000000.
   10 PRB(2)=0.0
      PRB(3)=0.0
      PRØ(2)=0.0
      PRØ(3)=0.0
      BDVC(2)=0.0
      BDVC(3)=0.0
      ØDVC(2)=0.0
      ØDVC(3)=0.0
      IF(PIN.GT.0.0) PRØPBG=PIN1
      IF(PIN.GT.0.0) PRØPØG=PIN2
      IF(MG.EQ.1.0) STAGV=MGDVB(2)
      IF(GRØW.GT.0.0.AND.J.GT.1) GØ TØ 2
      IF(FIXHRD.EQ.0.0) BBT=100.
      PRØPB=PRØPBG
```

```
2     IF(GRØW.EQ.4..AND.J.GT.1) GØ TØ 3
      PRØPØ=PRØPØG
3     IF(GRØW.EQ.4..AND.J.GT.1) PRØPØ=PRØPØT
      PRØ(1)=PRØPB
      PRØ(1)=PRØPØ
      VALID=50000.
      N=0.0
      S=0.0
      IF(ØCANT.GT.1.0) GØ TØ 20
      IF(NØENGØ.EQ.3.0) CTHØV=(THØV+THØV*2.
     1*CØS(ØCANTY/57.2958))/NØENGØ
      IF(NØENGØ.NE.3.0) CTHØV=THØV*CØS(ØCANTY/57.2958)
      IF(ØCANTP.GT.0.0) CTHØSL=(CTHØV*CØS(ØCANTP/57.2958))*
     1(ISPØBS/ISPØBV)
      THØV=CTHØV
      THØSL=CTHØSL
      ØCANT=ØCANT+1.0
      IF(GRØW.LE.3.0.AND.J.GT.1) GØ TØ 12
      THØSLC=THØSL*CØS(ØCANT/57.2958)
12    THØTC=THØV*NØENGØ
      FLØWR=TØV/ISPØBV
      ØFLØWX=FLØWR/(1.+MR)
      ØFLØWY=(FLØWR*MR)/(1.+MR)
      FSTART=3.2550*ØFLØWX
      ØSTART=2.7912*ØFLØWY
      ØHØLD=FSTART+ØSTART
20    IF(FIXHRD.GT.0.0) CALL ASPER
      IF(FIXHRD.GT.0.0) GØ TØ 100
      PRØPØ1=ØØT*FLØWR*NØENGØ
      PRØPØT=PRØPØ+PRØPØ1
      ISPØ=(PRØPØ*ISPØV+PRØPØ1*ISPØBV)/PRØPØT
      ISPØ=ISPØBV
      IF(GRØW.GE.2.0.AND.J.GT.1) GØ TØ 21
      CALL PRØMØDEL
21    IF(FIXØRØ.GT.0.0) GØ TØ 4
      PRØPØT=PRØPØ+PRØPØ1
      PRØPØ1=PRØPØ1
      PRØPØ2=PRØPØ
      GØ TØ 5
4     PRØPØT=PRØPØ
      PRØPØ1=PRØPØ1
      PRØPØ2=PRØPØT-PRØPØ1
5     IF(S.GT.0.0.AND.N.EQ.3.) GØ TØ 6
      CALL ØRMMØDEL
6     CALL EXTANK
      ØGLØW=ØLØWT+PRØPØT+INERT
      GLØW=ØGLØW+ØGLØW
      TØTAL=GLØW+ØHØLD
      WØSCD=GLØW/SCD
      IF(FTX.GT.0.0) GØ TØ 7
      IF(GRØW.LE.3.0.AND.J.GT.1) GØ TØ 13
      THØTC=(THØSLC*NØENGØ)+(NØENGØ*THØSL)
      THØSLT=THØSL*NØENGØ
      ØØT=(PRØPØ*ISPØS)/(THØSL*NØENGØ)
      GØ TØ 8
7     THØTC=ØTW*GLØW
      THØSLT=(THØTC-(NØENGØ*THØSL))/CØS(ØCANT/57.2958)
      ØØT=(PRØPØ*ISPØS)/THØSLT
8     IF(TF.NE.0.0) ØØT=ØØT/TF
13    TØWØ=THØTC/(ØGLØW-PRØPØ1)
      TØWØ=THØTC/GLØW
```

```
      IF(FTW.GT.0.0) TØWB=BTW
      DVB=32.174*ISPB*ALØG(GLØW/(GLØW-PRØPBT))
      DVØNC=32.174*ISPØ*ALØG((ØGLØW-PRØPØ1)/
     1(ØGLØW-PRØPØT))
      DVTØTC=DVB+DVØNC
      FW(1)=TØWB
      FW(2)=TØWØ
      CALL DELTAV
      S=S+1.0
      P=2.0
      IF(FIXØRB.GT.0.0) GØ TØ 97
C
C     ESPER ITERATIØN TECHNIQUE
C
C     ØRBITER ITERATIØN
C
      ØDVC(1)=DVØNC
      TVALID=DVØNR-DVØNC
      IF((DVØNR-DVØNC).GT.5000.) PRØPØG=PRØPØG+50000.
      IF((DVØNR-DVØNC).GT.5000.) PRØPØG=PRØPØG+10000.
      IF((DVØNR-DVØNC).GT.5000.) GØ TØ 10
      IF(ABS(DVØNR-DVØNC).LT.1.)GØ TØ 50
      IF(VALID.LT.0.0) GØ TØ 52
      IF(VALID.LT.TVALID) GØ TØ 50
   52 ØRACK1=(DVØNR-ØDVC(1))/(ØDVC(1)-ØDVC(2))
      ØRACK2=(PRØ(1)-PRØ(2))/(ØDVC(1)-ØDVC(2))
      ØRACK3=(PRØ(2)-PRØ(3))/(ØDVC(2)-ØDVC(3))
      PRØPØ=PRØ(1)+(PRØ(1)-PRØ(2))*ØRACK1
     &+(((DVØNR-ØDVC(1))*(DVØNR-ØDVC(2)))/(ØDVC(1)-ØDVC(3)))
     &*(ØRACK2-ØRACK3)
      IF(PRØPØ.LE.0.0) PRØPØG=PRØPØG+50000.
      IF(PRØPØ.LE.0.0) GØ TØ 10
      PRØ(3)=PRØ(2)
      PRØ(2)=PRØ(1)
      ØDVC(3)=ØDVC(2)
      ØDVC(2)=ØDVC(1)
      PRØ(1)=PRØPØ
      VALID=TVALID
      N=9.
      IF(FIXØRB.GT.0.0) GØ TØ 57
      IF(REL.EQ.0.0.AND.DVB.LT.(STAGV-50.)) GØ TØ 50
      IF(REL.EQ.1.0.AND.DVBRP.LT.(STAGV-50)) GØ TØ 50
   57 GØ TØ 20
C     THE SELECTØR 'STAGV' ENABLES THE USER TØ CHØØSE THE
C     STAGING VELØCITY HE WISHES THE PRØBLEM TØ BE SØLVED
C     FØR
   50 IF(FIXBØØ.GT.0.0)  GØ TØ 100
C
C     SRM ITERATIØN
C
      IF(MINGLW.EQ.0.0) GØ TØ 15
      IF(MG.EQ.1.0) GØ TØ 15
      IF(VALID.LT.TVALID) GØ TØ 15
      IF(ABS(DVØNR-DVØNC).GT.1.) GØ TØ 15
      MGGL(1)=GLØW
      MGDVB(1)=DVB
      IF(REL.GT.0.0) MGDVB(1)=DVBRP
      IF(MGGL(1).GT.MGGL(2)) MG=1.0
      IF(MGGL(1).GT.MGGL(2)) GØ TØ 10
      MGGL(2)=MGGL(1)
      MGDVB(2)=MGDVB(1)
```

```
 15   IF(REL.GT.0.0) GO TO 55
      IF(ABS(DVB-STAGV).LT.1.) GO TO 100
      BDVC(1)=DVB
      GO TO 56
 55   IF(ABS(DVBRP-STAGV).LT.1.) GO TO 100
      BDVC(1)=DVBRP
 56   BRACK1=(STAGV-BDVC(1))/(BDVC(1)-BDVC(2))
      BRACK2=(PRB(1)-PRB(2))/(BDVC(1)-BDVC(2))
      BRACK3=(PRB(2)-PRB(3))/(BDVC(2)-BDVC(3))
      PROPB=PRB(1)+(PRB(1)-PRB(2))*BRACK1
     1+(((STAGV-BDVC(1))*(STAGV-BDVC(2)))/(BDVC(1)-BDVC(3)))
     2*(BRACK2-BRACK3)
      IF(PROPB.LE.0.0) PROPBG=PROPBG+50000.
      IF(PROPB.LE.0.0) GO TO 10
      PRB(3)=PRB(2)
      PRB(2)=PRB(1)
      BDVC(3)=BDVC(2)
      BDVC(2)=BDVC(1)
      PRB(1)=PROPB
      BDVC(2)=0.0
      BDVC(3)=0.0
      PRB(2)=0.0
      PRB(3)=0.0
      PROPB=PROPBG
      PRB(1)=PROPB
      VALID=50000.
      N=10.
      GO TO 98
 97   IF(ABS(DVT-DVTOTC).LT.1.)  GO TO 100
      BDVC(1)=DVTOTC
      BRACK1=(DVT-BDVC(1))/(BDVC(1)-BDVC(2))
      PROPB=PRB(1)+(PRB(1)-PRB(2))*BRACK1
      IF(PROPB.LE.0.0) PROPBG=PROPBG+50000.
      IF(PROPB.LE.0.0) GO TO 10
      PRB(2)=PRB(1)
      BDVC(2)=BDVC(1)
      PRB(1)=PROPB
 98   GO TO 20
C     FLIGHT PERFORMANCE RESERVE CALCULATIONS
 100  WWOFPR=OLLPLO+PLOADU+WOPROP+INERT
      WWFPR=WWOFPR*EXP(DVFPR/(32.174*ISPO))
      FPROP=WWFPR-WWOFPR
      IF(GROW.GT.0.0.AND.J.GT.1) GO TO 400
      CALL OUTPUT
      IF(GROW.GT.0.0) GO TO 400
      IF(SENS.EQ.0.0) GO TO 110
      IF(SENS.GT.0.0.AND.FIXHRD.NE.1.0) CALL ASPER
      CALL FHSEN
      GO TO 110
C
C     GROWTH OPTION CALCULATIONS
C
 400  IF(J.GT.1) GO TO 447
      WRITE(108,1130)
      WRITE(108,1140)
      IF(GROW.EQ.1.0) WRITE(108,1042)
      IF(GROW.EQ.2.0) WRITE(108,1044)
      IF(GROW.EQ.3.0) WRITE(108,1046)
      IF(GROW.EQ.4.0) WRITE(108,1048)
      IF(GROW.EQ.4.0) WRITE(108,1049)
      IF(GROW.EQ.4.0) GO TO 441
```

```
      WRITE(108,1150)
441   IF(J.EQ.1) GØ TØ 445
447   IF(GRØW.EQ.4.0) WRITE(108,1160) PCENT,BGLØW,BDRYWT
     1,ØLØWT,DRYWT,GØØD,DVBØØ,TØWØ
      IF(GRØW.EQ.4.0) GØ TØ 445
      WRITE(108,1160) PCENT,PRØPØT,BDRYWT,ØLØWT,DRYWT
     1,GØØD,DVBØØ,TØWØ
1130  FØRMAT('1',///////////)
1140  FØRMAT(30X,'GRØWTH STUDY')
1042  FØRMAT(29X,'BØØSTER GRØWTH')
1044  FØRMAT(29X,'ØRBITER GRØWTH')
1046  FØRMAT(24X,'BØØSTER + ØRBITER GRØWTH')
1048  FØRMAT(26X,'FIXED HØ TANK GRØWTH')
1049  FØRMAT(/,3X,'%',3X,'BØØ GLØW',2X,'DRY-WT',3X,
     1'ØRB-WT.',2X,'TNK-WT',2X,'VALID',2X,'STAGE',2X,'T/W(2)')
1150  FØRMAT(/,3X,'%',3X,'ØRB PRØP',2X,'DRY-WT',3X,
     1'ØRB-WT.',2X,'TNK-WT',2X,'VALID',2X,'STAGE',2X,'T/W(2)')
1160  FØRMAT(1X,F4.3,2X,F8.0,2X,F7.0,2X,F7.0,1X,F7.0,
     12X,F3.1,4X,F5.0,1X,F5.3)
1180  FØRMAT(///////////)
445   IF(J.EQ.NI) GØ TØ 171
      IF((J-1).GT.0) GØ TØ 452
      BBG=BØØWT
      BDG=BDRYWT
      ØBG=ØLØWT
      LAMBG=LAMB
      FIXBG=FIXBØØ
      FTWG=FTW
      FIXØBG=FIXØRB
      FIXWØG=FIXWØR
      RELG=REL
452   PCENT=FLØAT(J-1)*.01
      IF(GRØW.EQ.4.0) GØ TØ 449
      FIXBØØ=1.0
      FTW=0.0
      IF(GRØW.EQ.2.0) GØ TØ 449
      BGCØN=BBG-BDG
      BDRYWT=BDG+(PCENT*10.*GRØWB)
      BØØWT=BDRYWT+BGCØN
      IF(GRØW.EQ.1.0) GØ TØ 451
449   IF(GRØW.EQ.4.0) FIXØRB=1.0
      IF(GRØW.EQ.4.0) REL=0.0
      ØLØWT=ØBG+(PCENT*10.*GRØWØ)*
     1EXP(ØMSDVP/(32.174*ØMSISP))
      FIXWØR=1.0
      IF(GRØW.EQ.4.0) GØ TØ 453
451   IF(ABS(DVØNR-DVØNC).GT.1.0) GØØD=2.0
      IF(GØØD.EQ.2.0) GØ TØ 171
      IF(ABS(DVØNR-DVØNC).GT.1.) GØ TØ 448
      GØØD=1.0
      GØ TØ 448
453   IF(FIXØRB.GT.0.0) GØ TØ 454
      IF(ABS(DVB-STAGV).GT.1.) GØØD=2.0
      IF(GØØD.EQ.2.0) GØ TØ 171
      IF(ABS(DVB-STAGV).GT.1.) GØ TØ 448
      GØØD=1.0
      GØ TØ 448
454   IF(ABS(DVT-DVTØTC).GT.1.) GØØD=2.0
      IF(GØØD.EQ.2.0) GØ TØ 171
      IF(ABS(DVT-DVTØTC).GT.1.) GØ TØ 448
      GØØD=1.0
```

```
  448   J=J+1
        GØ TØ 10
  171   WRITE(108,1190)
 1190   FØRMAT(/,3X,'WHEN GØØD=1.0  VALID SØLUTIØN'
       1,/,3X,'ESPER TERMINATES GRØWTH ØPTIØN'
       2,/,3X,'WHEN GØØD=2.0  NØ VALID SØLUTIØN')
        WRITE(108,1180)
        FIXRØØ=FIXBG
        FTW=FIWG
        ØLØWT=ØRG
        ØRØWT=RØG
        LAMR=LAMBG
        FIXØRB=FIXØRG
        FIXWØR=FIXWØG
        REL=RELG
        GØØD=1.0
  110   F=0.0
        PIN=1.0
        GØ TØ 1
 9999   CALL CLØSF1
        STØP
        END
```

3. ORBITER MODULE

The Orbiter Module contains the analytical and empirical weight estimation relationships necessary to completely define the vehicle. These relationships are combined into separate models, each model fully describing a weight group from the NASA functional coding. As an example, the Wing Group Model contains the analytical relationships describing the weight of the torque box plus empirical relationships defining the remaining elements of the wing, such as leading edge, landing gear provision, and elevon.

The Orbiter Module is set up to analyze a point design vehicle with minimum data. The NASA weight report and design data, coupled with a three-view drawing of the Orbiter, supplies all inputs necessary to analyze the configuration. Volume III, the User Manual, lists all required input data, and delineates the interface with the Group Weight Statement and the Design Data Summary. A point design analysis will give a detail line-item comparison with a contractor's weight report. This comparison will provide insight to variations of payload and performance characteristics as well as indicate subsystems that require scrutiny, either updating the model to a more realistic level or possible errors in the contractor data.

To run a point design analysis, it is first necessary to determine the performance characteristics, if unknown, from the ESPER program by running a fixed hardware case. In this case, the vehicle module weights, propellant, thrust, and velocity laws are input, and the payload capability is measured for a given mission. Next, an iterative case is run, using the data from the fixed hardware case. The payload, propellants, laws, and vehicle module dry weights are input. The program then analyzes the various subsystems and determines their weight. The growth/uncertainty of each vehicle module is allowed to "float," i.e., vary either up or down to maintain a constant dry weight, therefore physically sizing each system to the point design loads.

The primary purpose of the Orbiter Module is to provide the capability of analyzing an interated vehicle to determine performance trades and to lend direction to the overall design effort by answering such questions as:

1. What happens if you vary engine characteristics, such as Orbiter thrust, or specific impulse?
2. Is the staging velocity optimized?
3. What is the minimum gross weight vehicle for the users constraints?

4. What is the effect of changes to the primary construction material?
5. How do geometric changes, such as aspect ratio, payload bay length, or width, effect the configuration?

The inputted parameters start the Orbiter Module iteration for which liftoff weight, injected weight, etc., are calculated. These calculated weights, in turn, modify the aerodynamic surfaces; the wing area changes to maintain a constant wing loading and landing speed, and the tail changes to maintain control capability with a constant tail volume coefficient. In turn, these modify the surface controls and the thermal protection system. The auxiliary propulsion system is affected by injected weight and the landing gear by the landing loads. The body is modified by reaction from the above systems which, in turn, changes the interstage loads which ripple changes back through the body. The entire module continues the iteration until a completely balanced system exists.

Figure 3-1 is a flow diagram of the Orbiter Module, followed by the detail listing of the Module.

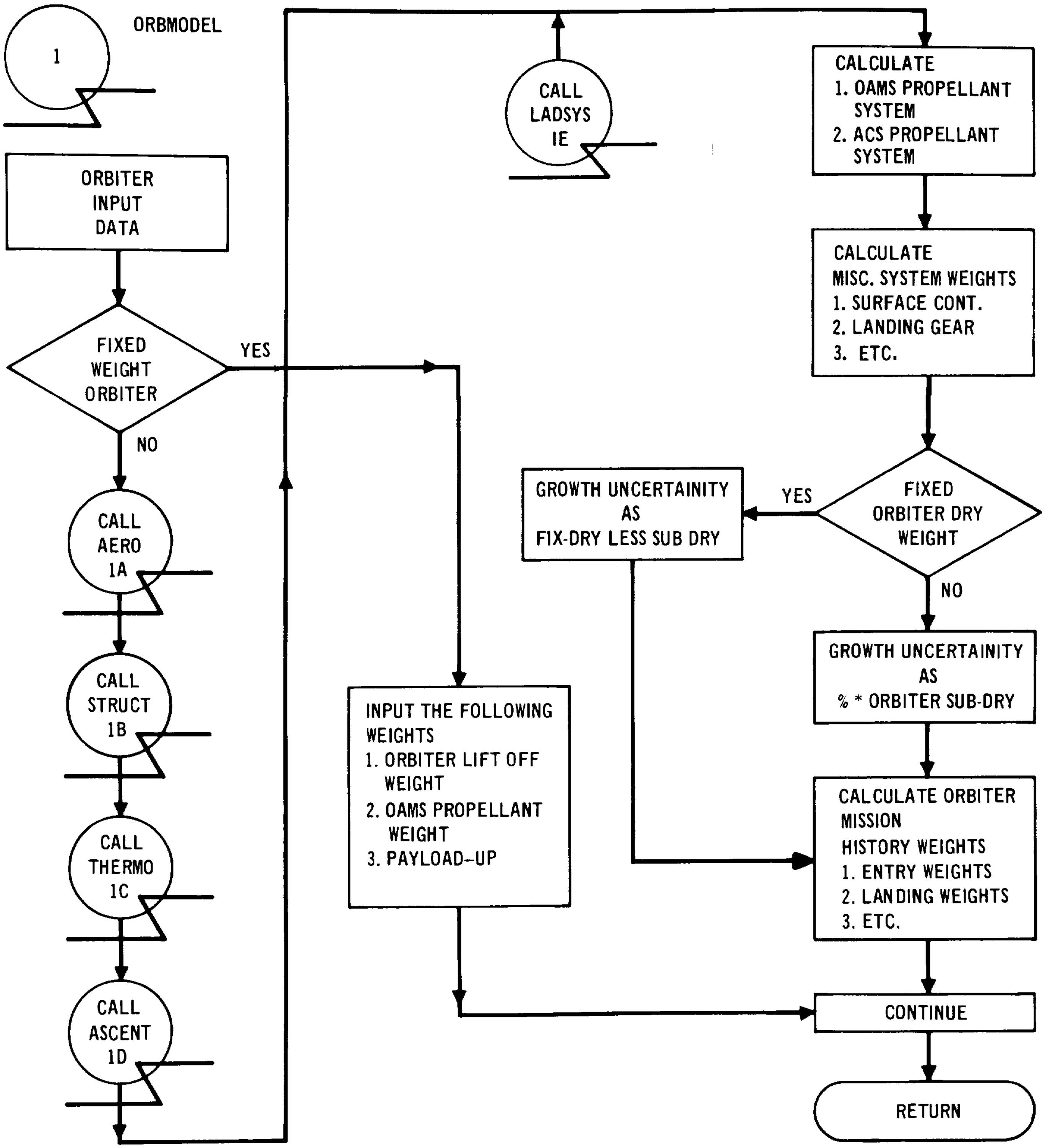

FIGURE 3–1 FLOW DIAGRAM

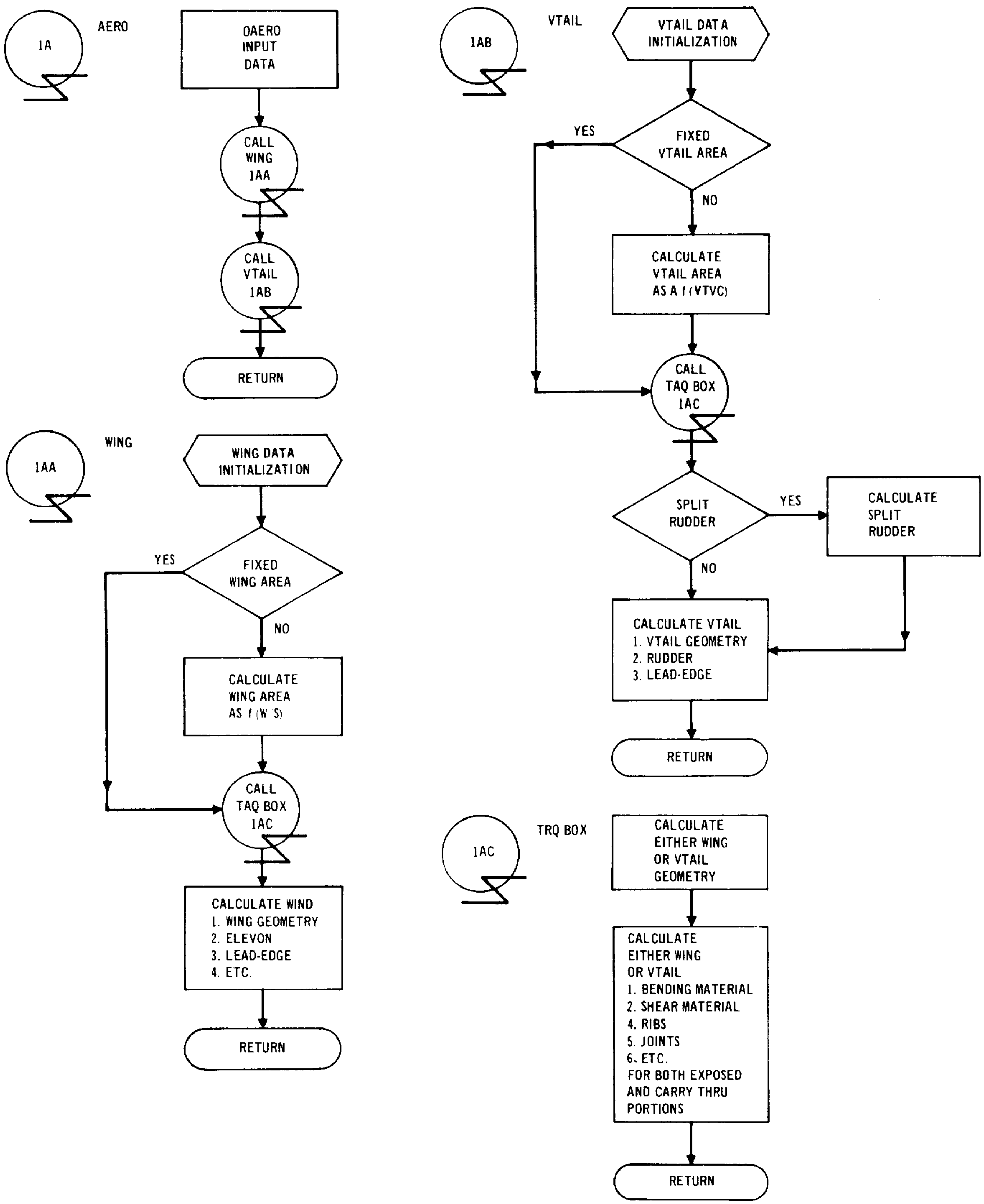

FIGURE 3-1 FLOW DIAGRAM (Continued)

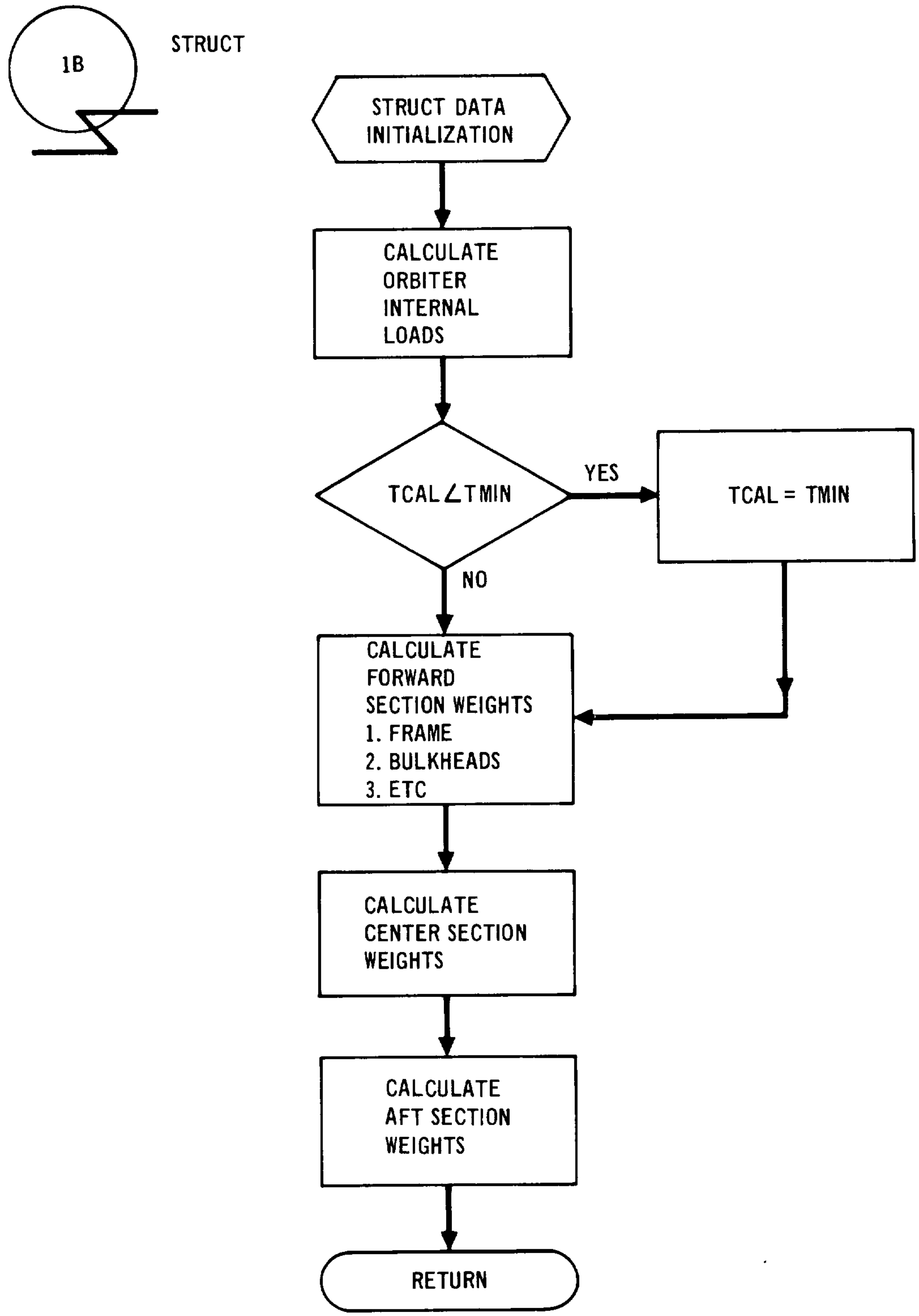

FIGURE 3–1 FLOW DIAGRAM (Continued)

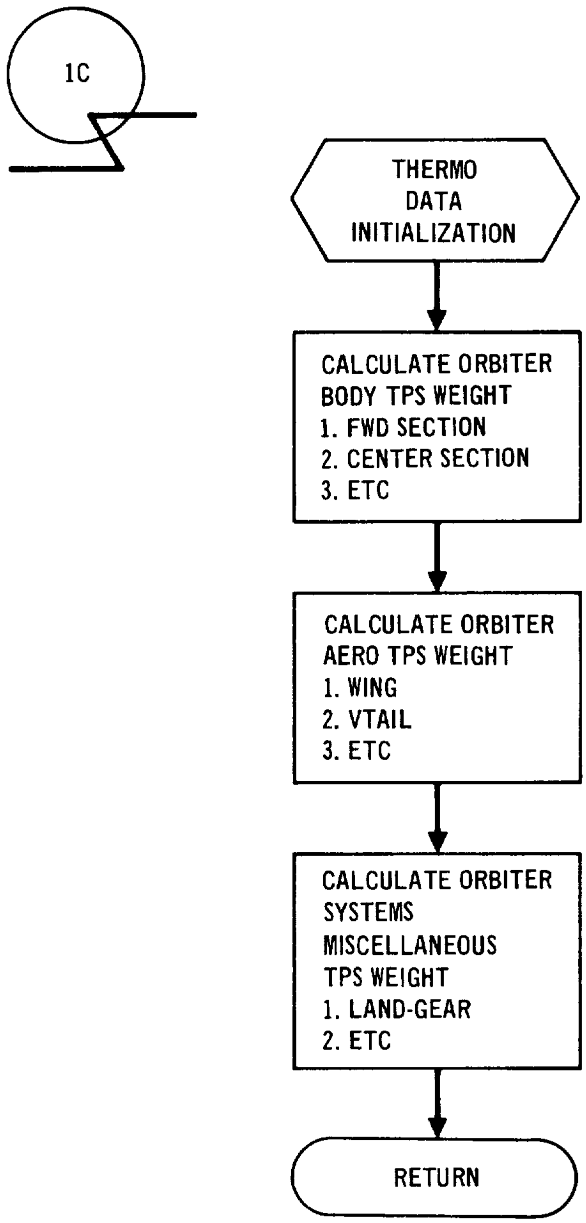

FIGURE 3–1 FLOW DIAGRAM (Continued)

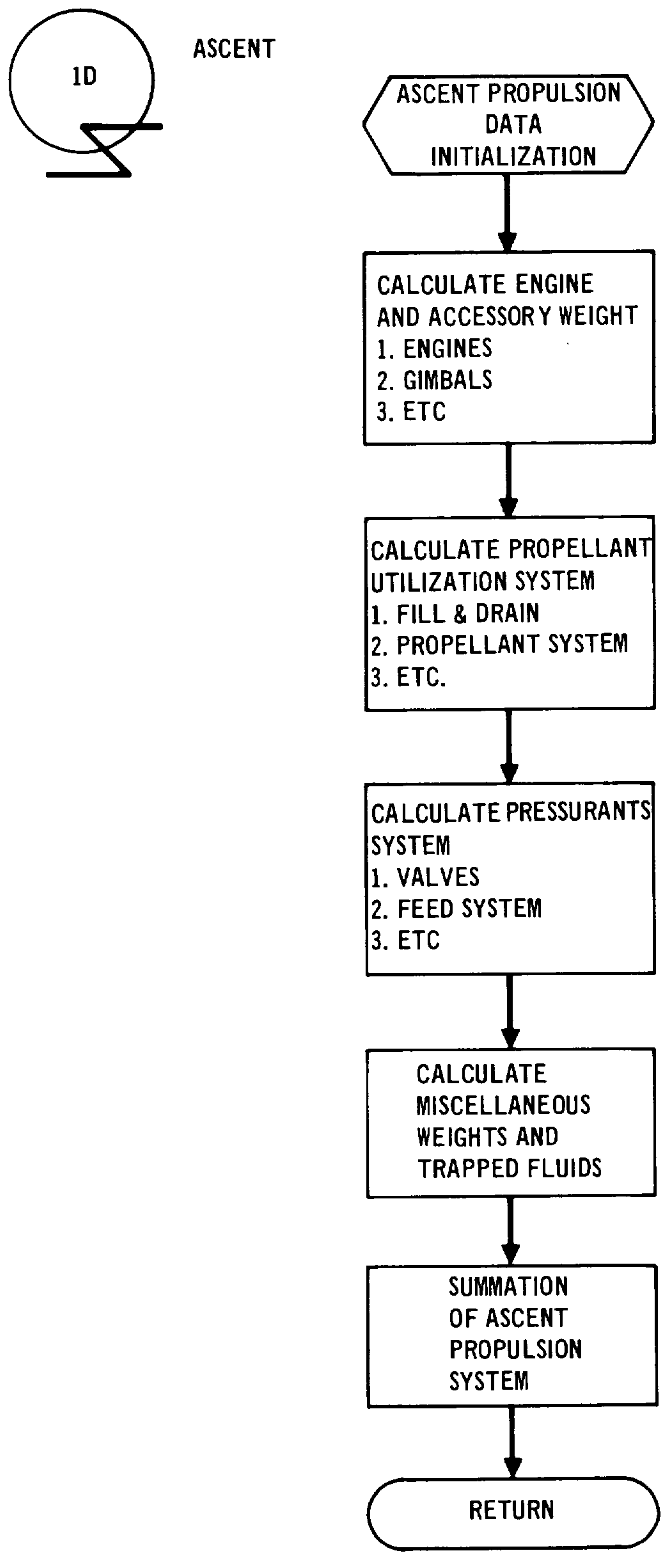

FIGURE 3–1 FLOW DIAGRAM (Continued)

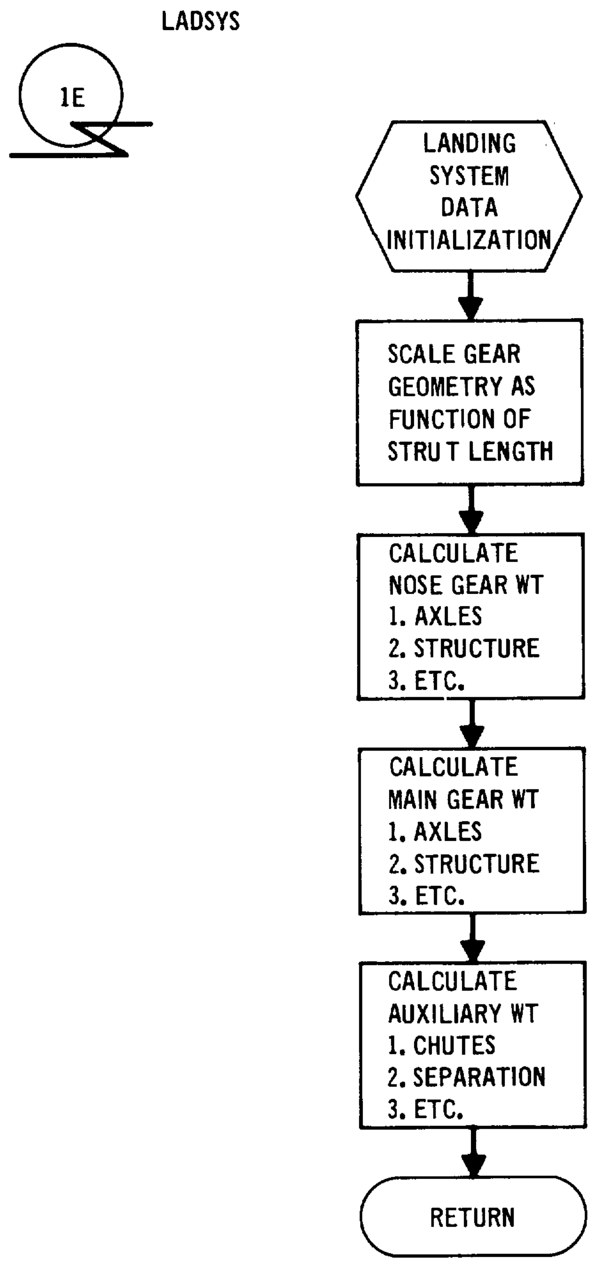

FIGURE 3–1 FLOW DIAGRAM (Continued)

COPY OSUB TO LP(K,NC)

```
$FIXED
C
C                         ORBITER SUBROUTINE
C
      SUBROUTINE ORBMODEL
C     THE ORBITER PROGRAM CONTAINS THE ANALYTICAL AND
C     EMPIRICAL WEIGHT ESTIMATION RELATIONSHIPS NECESSARY
C     TO COMPLETELY DEFINE THE VEHICLE. THESE RELATION-
C     SHIPS ARE COMBINED INTO SEPARATE MODELS,EACH MODEL
C     FULLY DESCRIBING A WEIGHT GROUP FROM THE NASA
C     FUNCTIONAL CODING.THE PRIMARY PURPOSE OF THE ORBITER
C     MODULE IS TO PROVIDE THE CAPABILITY OF ANALYZING AN
C     ITERATED VEHICLE TO DETERMINE PERFORMANCE TRADES AND TO
C     LEND DIRECTION TO THE OVERALL DESIGN EFFORT.
      IMPLICIT REAL(A-Z)
C
C     PERFORMANCE COMMON BLOCK
C
      COMMON/MAIN/ PROPB,PROPO,POT,BCANT,OCANT,OCANTY
     1,OCANTP,NOENGO,NOENGB,THBSL,THOSL,THOV,TOV,FLOWR
     2,TF,FTW,FIXHRD
     3,ISPOS,ISPOV,ISPOBS,ISPOBV,SCD,BTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THBTC,THOSLT,THOTC,ISPO,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVONC,DVB,DVTOTC,WASCD,OINWT,OLANWT
     7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
C
C     ORBITER OUTPUT COMMON BLOCK
C
      COMMON/ORBO/R1,R2,RL,WTAUX,WTACS,ACSENG,ACSSYS,WTACTK
     1,ACSMOD,WTOMS,OMSENG,PROPSY,WTOMTK,MODULE
     1,SURFC,PPWR,ELEC,HYDR,AVIONO,ECLSO,PPROV,OUNCWT
     2,ORBMIS,TARPRP,SURFK
     1,PERSON,ORESD,ORESV,PLOADU,PLOADD,ACSPRO,WOPROP,SUDLE
     2,FIXORO,FIXWOR,ODRYWT,ORIFL
     1,WWT,WSG,WBSTR,WTORBE,WTORBC,LEW,WTE,WAIL,WAS,WADR,WAH
     2,WAP,SPROV,PWINGK,TAIL,TSG,TBSTR,TTORDB,TLE,WRJD,WRS
     3,WRDR,WRH,WRP,PTAILK
     1,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36
     1,TOTTPS,TWGWT,WGWT,WGLEWT,TWT,TLEWT,BLOTPS
     2,BASEWT,IBTWT,PTPSCN,TTWT,BTPSWT,MCSWT,LDTWT
     3,PROWT,PPPC,PHYC,SCWT
     1,TAPROP,ENGPAC,ENG,TVC,CONTR,PRPUTL,PROSYS
     2,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC
     1,LNDCK,NG1,NG2,NG3,NGEAR,MG1,MG2,MG3,MGEAR,AX1,AX2
     2,AX3,AXGEAR,LNDCKK
C
C     EXTERNAL TANK OUTPUT COMMON BLOCK
C
      COMMON/EXTO/BODGRP,TOTPS,FWDTK,FAIRT,FWDBLF,FCCTPS
     1,CONSCT,TPSIN,CYLSCT,ACYDM,AFTBLF,WINT,PROSY,AFTNK
     2,FEDSYS,FWDBLA,PRSVNT,AFTCYL,SUMP,AFTBLA,PNPJ,TWINT
     3,NOSFAR,AVIONT,JMBPNL,WRETRO,TUNNEL,MISCT,BAFF,SUBDRY
     4,GU,DRYWT,RESIDT,UNDRAN,FEEDTR,PRSURT,FBIAS,INERT
     5,GROSSW,TLAMB,OTRAP,EXTL,EXTD,BLKHD,EXTHB,EXTHH,SIMPTK
```

```
C
C     STRUCTURE INPUT DATA COMMON BLOCK
C
      COMMON/STRD/ABI,HTV,LTV,HB,LB,LI,NX,NZ,FS
     1,HF,HL,X,K1,SFW,VC,PC,Q,SND,LNG
     2,FAL,TMIN,RHOL,RHOS,TAUS,EF,LFS,DELP,FAB,RHOB
     3,TAUB,RHOF,K2,SW,K7,K3,K4,ACD,FAF,SAW,FAPB,TAUPB
     4,DFAC,ET,RHOTP,RHOPB,K6
C
C     THERMO(TPS) INPUT DATA COMMON BLOCK
C
      COMMON/THERD/NCTPS,NCA,FWDTPS,FWDA,CTTPS,CTA,CSTPS
     2,CSA,CBTPS,CBA,ATTPS,ATA,ASTPS,ASA,ABTPS,ABA,BASTPS
     3,BASA,TPSCON,WGTPS,WGPLE,WLETPS,TLTPS,TLPLE
     4,TLETPS,MCSTPS,MCSA,WACON,TACON,IBA,IBTPS,IBC,LDA,LDTPS
     5,PROA,PROTPS,PROC,PPC,HYC,SCA,SCTPS,SWI,SWC,WSI
C
C     ASCENT PROPULSION INPUT DATA COMMON BLOCK
C
      COMMON/ASPD/PROP,SPI,HHEAD,OHEAD,HULL,OULL,FTU,RHO
     1,MATL,HCLEN,OCLEN,HELEN,OELEN,CPLGI
C
C     LANDING AND DOCKING INPUT DATA COMMON BLOCK
C
      COMMON/LDGD/LGFTU,LGVSL,LGLC,LGLS,LGDIA,BRCF
C
C     ORBITER (ORB) INPUT DATA BLOCK
C
      NAMELIST ABI,HTV,LTV,HB,LB,LI,NX,NZ,FS
     1,HF,HL,X,K1,SFW,VC,PC,Q,SND,LNG
     3,FAL,TMIN,RHOL,RHOS,TAUS,EF,LFS,DELP,FAB,RHOB
     4,TAUB,RHOF,K2,SW,K7,K3,K4,ACD,FAF,SAW,FAPB,TAUPB
     5,DFAC,ET,RHOTP,RHOPB,K6
     6,NCTPS,NCA,FWDTPS,FWDA,CTTPS,CTA,CSTPS,CSA,CBTPS
     7,CBA,ATTPS,ATA,ASTPS,ASA,ABTPS,ABA,BASTPS,BASA,TPSCON
     8,WGTPS,WGPLE,WLETPS,TLTPS,TLPLE,TLETPS
     8,MCSTPS,MCSA,WACON,TACON,IBA,IBTPS,IBC,LDA,LDTPS,PROA
     9,PROTPS,PROC,PPC,HYC,SCA,SCTPS,SWI,SWC,WSI
     4,PROP,SPI,HHEAD,OHEAD,HULL,OULL,FTU,RHO,MATL,HCLEN,OCLEN
     5,HELEN,OELEN,CPLGI
     1,DENSF,DENSO,RHOT,FTUT
     1,PRESOM,RHOP,FTUP,OMSENG,PROPSY,MODULE,PRESF,PRESO
     2,ACSPRO,ACSDEN,ACSPRS,ACSENG,ACSSYS,ACSMOD
     3,ORBMIS,FIXDWT,OUNC1,PERSON,ORESD,ORESV,PLOADU,PLOADD
     4,FIXORB,FIXWOR,PPWR,HYDRK,ELECK,AVIONO,ECLSO
     5,PPROV,ORIFL,TABPRO,SURFK,OLLPLO,WOPROP
     6,LGFTU,LGVSL,LGLC,LGLS,LGDIA,BRCF,AX2,AX3,LNDDKK
C
      IF(P.EQ.2.0) GO TO 2
      INPUT(1)
    2 IF(FIXWOR.GT.0.0.OR.FIXORB.GT.0.0) GO TO 400
      LDO=0
      LMG=LGLC
      CALL AERO(WSG,WBCT,WTHETA,WNZ,WB,SAIL,SRUD,VB,PV,WESG
     1,TSG,SURFI,LDO,LMG
     2,WWT,WBSTR,WTORBE,WTORBC,LEW,WTE,WAIL,WAS,WADR,WAH,WAP
     3,PWINGK,GPROV
     1,TAIL,TBSTR,TTORQB,TLE,WRUD,WRS,WROR,WRH,WRP,PTAILK)
      CALL STRUCT(WTORBC,VB,WBCT,WTHETA,WNZ,WB,SAIL,SRUD,PV
     &,P1,R2,RL
     1,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
```

```
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36)
      CALL THERMO(WSG,WESG,TSG,TOTTPS,TW3WT,WGWT,WGLEWT,TWT,TLEWT
     1,PLOTPS,BASEWT,IBTWT,PTPSCN,TTWT,BTPSWT,MCSWT,LDTWT
     2,PROWT,PPPC,PHYC,SCWT)
      CALL ASCENT(TAPROP,ENGPAC,ENG,TVC,CONTR,PRPUTL
     1,PROSYS,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC)
      CALL LADSYS(LNDDK,NG1,NG2,NG3,NGEAR,MG1,MG2,MG3
     1,MGEAR,AX1,AX2,AX3,AXGEAR,LNDDKK)
C     MISCELLANEOUS SUBSYSTEM WEIGHT CALCULATIONS
      K=1.0
      IF(SURFI.GT.0.0) K=3.0
      SURFC=1060.+3.45*SAIL+K*(360.+1.67*SRUD)+SURFK
      PPWR=PPWR
      HYDR=2264.*((WSG+K*TSG)/4525.)+HYDRK
      ELEC=ELECK+2805.+1840.*(LI-X)/747.
      AVION=AVION
      ECLS=ECLS
      PPROV=PPROV
      TABPRO=TABPRO
C
C     OMS SYSTEM CALCULATIONS
C
      WTPROP=OLOWLO*(EXP(OMSDVT/(32.174*OMSISP))-1.)
      WOPROP=OLOWLO*(EXP(OMSDVP/(32.174*OMSISP))-1.)
      WFUEL=WTPROP/(1.+OMR)
      WOX=OMR*WFUEL
      VFUEL=(WFUEL/DENSF)*1.15
      VOX=(WOX/DENSO)*1.15
      FTANK=3./2.*RHOT*PRESF*VFUEL/FTUT*1728.*1.28
      OTANK=3./2.*RHOT*PRESO/FTUT*VOX*1728.*1.28
      VPRES=(PRESO*VOX+PRESF*VFUEL)/PRESOM*1.47
      PRESTK=3./2.*RHOP*PRESOM*VPRES/FTUP*1728.*1.28
      WTOMTK=FTANK+OTANK+PRESTK
      WTOMS=WTOMTK+OMSENG+PROPSY+MODULE
C
C     ACS SYSTEM CALCULATIONS
C
      ACSVOL=ACSPRO/ACSDEN*1.15
      ACSTNK=3./2.*RHOT*ACSPRS*ACSVOL/FTUT*1728.*1.28
      VPTNK=ACSVOL*ACSPRS/PRESOM*1.47
      PTNK=3./2.*RHOT*PRESOM*VPTNK/FTUP*1728.*1.28
      WTACTK=(ACSTNK+PTNK)*1.25
      WTACS=WTACTK+ACSSYS+ACSENG+ACSMOD
      WTAUX=WTOMS+WTACS
      ORBMIS=ORBMIS
C
C     ORBITER MISSION HISTORY
C
      SUBDRY=WWT+TAIL+G37+TOTTPS+SURFC+TAPROP+WTAUX+ORBMIS
     1+PPWR+HYDR+ELEC+AVION+ECLS+PPROV+LNDDK+TABPRO
      SUDLE=SUBDRY-ENG
      IF(FIXDWT.GT.0.0) OUNCWT=FIXDWT-SUBDRY
      IF(FIXDWT.GT.0.0) GO TO 50
      OUNCWT=SUDLE*OUNC1
   50 ODRYWT=SUBDRY+OUNCWT
      OINWT=ODRYWT+PERSON+ORESD+PLOADU
      OLANWT=OINWT-PLOADU+PLOADD+ORESV
      OINJWT=OLANWT+ACSPRO+WOPROP-ORESV-PLOADD+PLOADU+ORIFL
      IF(P.EQ.0.0) OTRAP=1000.
      OLOWT=OINJWT+OTRAP
```

```
      ØLØWLØ=ØLØWT-WØPRØP
      ØLLPLØ=ØLØWT-WØPRØP-PLØADJ
      GØ TØ 500
C
C     FIXED WEIGHT ØRBITER CALCULATIØNS
C
  400 ØLLPLØ=ØLLPLØ
      WØPRØP=WØPRØP
      PLØADJ=PLØADJ
      ØLØWT=ØLLPLØ+PLØADJ+WØPRØP
  500 CØNTINUE
      RETURN
      END
      SUBRØUTINE AERØ(WSG,WBCT,WTHETA,WNZ,WB,SAIL,SRUD,VB
     1,PV,WESG,TSG,SURFI,LDG,LMG
     2,WWT,WBSTR,WTØRBE,WTØRBC,LEW,WTE,WAIL,WAS,WADR,WAH,WAP
     3,PWINGK,GPRØV
     1,TAIL,TBSTR,TTØRØB,TLE,WRUD,WRS,WRDR,WRH,WRP,PTAILK)
      REAL LAMB,LH,M,K,NZ,LE,KEAS,LAMBP,MP,KP,LEW
     1,KMPP,KMC,MPP,LDG,LMG
      DIMENSIØN  AR(2),SG(2),LAMB(2),TØCR(2),TØCT(2)
     1,BCT(2),THETA(2),NZ(2),DELP(2),LH(2),PTØXC(2)
     2,PTØXE(2),CB(2),RHØ(2),FA(2),CS(2),TAU(2),TEMP(2)
     3,UWW(2),CSR(2),TMIN(2),ULE(2),WLE(2),CLE(2)
     4,EMØDJ(2),WC1(2),WC2(2),CM1(2),BLP1(2),BLP2(2),BCM1(2)
      CØMMØN/RWRC/ AR,SG,LAMB,TØCR,TØCT,BCT,THETA,NZ,DELP,LH
     1,PTØXC,PTØXE,CB,RHØ,FA,CS,TAU,TEMP,UWW,B,SEXP,SL,RBM
     3,EMØDJ,WS,PKR,WBREL,WC1,WC2,CM1,BLP1,BLP2,BCM1
     2,WLE,ULE,CSR,TMIN,WANG(22),TANS(22),SL1,SL2,CLE
      CØMMØN/RWRW/KEAS,AILP,AICP,UWAIL,WWØS,WINGK,TLDG,SMGDR,TLMG
      CØMMØN/RWRVT/RDC,RUDUL,URS,VTVC,LVT,SPRUD,TAILK
      CØMMØN/MAIN/ PRØPB,PRØPØ,ØØT,BCANT,ØCANT,ØCANTY
     1,ØCANTP,NØENGB,NØENGØ,THØSL,THØSL,THØV,TØV,FLØWR
     2,TF,FTW,FIXHRD
     3,ISPØS,ISPØV,ISPØØS,ISPØØV,SCD,ØTW
     4,H,DVCØRR,INC,STAØV,DVCØN,DVCNST
     5,REL,THØTC,THØSLT,THØTC,ISPØ,ISPB,PRØPØT
     6,PRØPØ1,PRØPØ2,FW(2),DVØNC,DVB,DVTØTC,WØSCD,ØINWT,ØLANWT
     7,ØINJWT,ØLØWT,ØGLØW,GLØW,TØTAL,S,P,MATCH,TLSSR,FØRP
     8,ØHØLD,ØLLPLØ,ØLØWLØ,ØMSISP,ØMSDVT,ØMSDVP,ØMR
     9,LØNGP,TØWØ,TØWB,SENS,GRØW,MINGLW
C
C     ØRBITER AERØ-SURFACE (ØAERØ) INPUT DATA BLØCK
C
      NAMELIST
     1 AR,SG,LAMB,TØCR,TØCT,BCT,THETA,NZ,DELP,LH,PTØXC
     2,PTØXE,CB,RHØ,FA,CS,TAU,TEMP,UWW,CSR,TMIN
     3,ULE,KEAS,AICP,UWAIL,CLE,AILP
     4,RDC,RUDUL,URS,WLE
     5,WWØS,VTVC,LVT,SPRUD,WINGK,TAILK
     6,EMØDJ,WC1,WC2,CM1,BLP1,BLP2,BCM1,SMGDR
C
      TLDG=LDG
      TLMG=LMG
      IF(P.EQ.2.0) GØ TØ 20
      INPUT(1)
   20 CALL WING(WSG,WBCT,J,ØLANWT,SAIL,LEW,WAS,WADR,WAH,WAP,WAIL
     1,WTE,WWT,WBSTR,GPRØV)
      WNZ=NZ(J)
      WB=B
      WTHETA=THETA(J)
```

```
      WSG=SG(J)
      WBCT=BCT(J)
      WESG=SEXP
      WTORBE=WANS(20)
      WTORBC=WANS(21)
      PWINGK=WINGK
      CALL VTAIL(J,BLANWT,WSG,WB,SRUD,TLE,WRS,WRDR,WRH
     1,WRP,TAIL,VB,P,TBSTR,WRUD)
      VB=B
      PV=SL
      TSG=SG(J)
      SURFI=SPRUD
      TTORQB=TANS(22)
      PTAILK=TAILK
      RETURN
      END
      SUBROUTINE WING(WSG,WBCT,J,BLANWT,SAIL,LEW,WAS,WADR,WAH,WAP
     1,WAIL,WTE,WWT,WBSTR,SPROV)
      REAL LAMB,LH,M,K,NZ,LE,KEAS,LAMBP,MP,KP,LEW
     1,KMPP,KMC,MPP,LDG,LMG
      DIMENSION  AR(2),SG(2),LAMB(2),TOCR(2),TOCT(2)
     1,BCT(2),THETA(2),NZ(2),DELP(2),LH(2),PTBXC(2)
     2,PTBXE(2),CB(2),RHO(2),FA(2),CS(2),TAU(2),TEMP(2)
     3,UWW(2),CSR(2),TMIN(2),ULE(2),WLE(2),CLE(2)
     4,EMODU(2),WC1(2),WC2(2),CM1(2),BLP1(2),BLP2(2),BCM1(2)
      COMMON/RWRC/ AR,SG,LAMB,TOCR,TOCT,BCT,THETA,NZ,DELP,LH
     1,PTBXC,PTBXE,CB,RHO,FA,CS,TAU,TEMP,UWW,B,SEXP,SL,RBM
     3,EMODU,WSWPKR,WBREL,WC1,WC2,CM1,BLP1,BLP2,BCM1
     2,WLE,ULE,CSR,TMIN,WANS(22),TANS(22),SL1,SL2,CLE
      COMMON/RWRW/KEAS,AILP,AICP,UWAIL,WWBS,WINGK,TLDG,SMGDR,TLMG
      J=1
      IF(WWBS.GT.0.0) SG(J)=BLANWT/WWBS
      CALL TORBOX(J,BLANWT)
      LDG=TLDG
      LMG=TLMG
      BSC=BCT(J)/B
      CR=(2.*SG(J))/(B*(1.+LAMB(J)))
      BE=B-BCT(J)
      STBE=PTBXE(J)*SEXP
      CF=CR*(1.-BSC*(1.-LAMB(J)))
      CRCF=CR/CF
      SAIL=AILP*BE*AICP*CF/2.*(2.-(1.-LAMB(J))*CRCF*(1.-BSC)*(AILP))
      COST=COS(THETA(J)/57.2958)
      CR=(2.*SAIL)/(BE*(1.+LAMB(J)))
      CT=CR*LAMB(J)
      TR=CR*TOCR(J)
      TT=CT*TOCT(J)
      CMR=(CR+CT)/2.
      SRUD=SAIL
      RHM=SRUD*144.*KEAS*12.*CMR*.5*.001
      HLTR=12.*(TR+TT)/2.
      HLLR=BE/COST
      WAS=UWAIL*SAIL
      WADR=(CMR*(RHM/HLTR)**.75)*2.
      WAH=(.40*HLLR*RHM**.2)*2.
      WAP=.25*(WAS+WADR+WAH)
      WAIL=WAS+WADR+WAH+WAP
      SLE=CLE(J)*SEXP
      WLET=2.2*SLE+WLE(J)
      LEW=WLET*ULE(J)
      FMG=BLANWT*350000./215115.
```

```
      GPRBV=.43*LDG**.3*SMGDR+.077*(.001*FMG*LMG)**.9
      STE=SEXP-STBE-SAIL-SLE
      WTE=1.87*STE*(.001*BLANWT*NZ(J)/SG(J))**.2
      IF(SL2.GT.SL1) WTE=1.87*STE*(.001*DELP(J))**.2
      WWT=WANS(20)+LEW+WAIL+WTE+WINGK+GPRBV
      WBSTR=WANS(20)+LEW+WTE
      RETURN
      END
      SUBROUTINE VTAIL(J,BLANWT,WSG,WB,SRUD,TLE,WRS,WRDR
     1,WRH,WRP,TAIL,VB,P,TBSTR,WRUD)
      REAL LAMB,LH,M,K,NZ,LE,KEAS,LAMBP,MP,KP,LEW
     1,KMPP,KMC,MPP,LDG,LMG
      DIMENSION  AR(2),SG(2),LAMB(2),TBCR(2),TBCT(2)
     1,BCT(2),THETA(2),NZ(2),DELP(2),LH(2),PTBXC(2)
     2,PTBXE(2),CB(2),RHB(2),FA(2),CS(2),TAU(2),TEMP(2)
     3,UWW(2),CSR(2),TMIN(2),ULE(2),WLE(2),CLE(2)
     4,EMBDU(2),WC1(2),WC2(2),CM1(2),BLP1(2),BLP2(2),BCM1(2)
      COMMON/RWRC/ AR,SG,LAMB,TBCR,TBCT,BCT,THETA,NZ,DELP,LH
     1,PTBXC,PTBXE,CB,RHB,FA,CS,TAU,TEMP,UWW,B,SEXP,SL,RBM
     3,EMBDU,WSWPKR,WBREL,WC1,WC2,CM1,BLP1,BLP2,BCM1
     2,WLE,ULE,CSR,TMIN,WANS(22),TANS(22),SL1,SL2,CLE
      COMMON/RWRVT/RDC,RUDJL,URS,VTVC,LVT,SPRUD,TAILK
      J=2
      IF(VTVC.GT.0.0) SG(J)=(WSG*WB*VTVC)/(LVT/12.)
      CALL TRDBOX(J,BLANWT)
      COST=COS(THETA(J)/57.2958)
      CR=(2.*SG(J))/(B*(1.+LAMB(J)))
      CT=CR*LAMB(J)
      IF(SPRUD.GT.0.0.AND.P.GT.0.0) GO TO 5
      IF(SPRUD.GT.0.0) TTBCR=TBCR(J)*.5
      IF(SPRUD.GT.0.0) TTBCT=TBCT(J)*.5
    5 TR=CR*TTBCR
      TT=CT*TTBCT
      BE=B-BCT(J)
      CMR=RDC*(CR+CT)/2.
      SRUD=RDC*SG(J)
      RHM=SRUD*144.*RUDJL*12.*CMR*.5*.001
      HLTR=12.*(TR+TT)/2.
      HLLR=BE/COST
      WRS=URS*SRUD
      WRDR=.46*CMR*(RHM/HLTR)**.75
      WRH=.40*HLLR*RHM**.2
      IF(SPRUD.EQ.0.0) GO TO 10
      WRS=WRS*2.
      WRDR=WRDR*2.
      WRH=WRH*2.
   10 WRP=.25*(WRS+WRDR+WRH)
      WRUD=WRS+WRDR+WRH+WRP
      WLET=2.2*CLE(J)*SG(J)+WLE(J)
      TLE=ULE(J)*WLET
      TAIL=TANS(22)+WRUD+TLE+TAILK
      TBSTR=TANS(22)+TLE
      RETURN
      END
      SUBROUTINE TRDBOX(J,BLANWT)
      REAL LAMB,LH,M,K,NZ,LE,KEAS,LAMBP,MP,KP,LEW
     1,KMPP,KMC,MPP,LDG,LMG
      DIMENSION G(22)
      DIMENSION  AR(2),SG(2),LAMB(2),TBCR(2),TBCT(2)
     1,BCT(2),THETA(2),NZ(2),DELP(2),LH(2),PTBXC(2)
     2,PTBXE(2),CB(2),RHB(2),FA(2),CS(2),TAU(2),TEMP(2)
```

```
     3,UWW(2),CSR(2),TMIN(2),ULE(2),WLE(2),CLE(2)
     4,EMBDJ(2),WC1(2),WC2(2),CM1(2),BLP1(2),BLP2(2),BCM1(2)
      COMMON/RWBC/ AR,SG,LAMB,TBCR,TBCT,BCT,THETA,NZ,DELP,LH
     1,PTBXC,PTBXE,CB,RHB,FA,CS,TAJ,TEMP,UWW,B,SEXP,SL,RBM
     3,EMBDJ,WSWPKR,WBREL,WC1,WC2,CM1,BLP1,BLP2,BCM1
     2,WLE,ULE,CSR,TMIN,WANS(22),TANS(22),SL1,SL2,CLE
      COST=COS(THETA(J)/57.2958)
      B=(AR(J)*SG(J))**.5
      CR=(2.*SG(J))/(B*(1.+LAMB(J)))
      CT=CR*LAMB(J)
      CF=CR*(1.-(1.-LAMB(J))*BCT(J)/B)
      TR=TBCR(J)*12.*CR
      TT=TBCT(J)*12.*CT
      IF(LAMB(J).EQ.0.) TT=TBCT(J)
      M=TT/TR
      TF=TR*(1.+(M-1.)*BCT(J)/B)
      MP=TT/TF
      RBSPC=TF*(.8+.2*MP)
      IF(RBSPC.LT.12.) RBSPC=12.
      IF (MP.GT..99) MP=.99
      LAMBP=LAMB(J)/(1.-(1.-LAMB(J))*BCT(J)/B)
       KP=3.*LAMBP/(1.+LAMBP)*((1.-3.*MP)*(1.-MP)-2.*MP**2*ALOG(MP))
      KP=KP/(1.-MP)**3
      KP=KP+(1.-LAMBP)/(1.+LAMBP)*((2.-7.*MP+11.*MP**2)
     5*(1.-MP)+6.*MP**3*ALOG(MP))/3./(1.-MP)**4
      SCT=BCT(J)*(CR+CF)/2.
      SEXP=SG(J)-SCT
      STBC=PTBXC(J)*SCT
      STBE=PTBXE(J)*SEXP
      WW=UWW(J)*SG(J)
      BE=B-BCT(J)
      TANT=SIN(THETA(J)/57.2958)/COST
      TANLE=TANT+2.*(1.-LAMB(J))/(AR(J)*(1.+LAMB(J)))
      TANTE=TANLE-4.*(1.-LAMB(J))/(AR(J)*(1.+LAMB(J)))
      ANGLE=57.2958*ATAN(TANLE)
      ETAWNG=(.04+AR(J)*(.0049+.000045*ANGLE))*LAMB(J)
      ETAWNG=ETAWNG-.05*(LAMB(J)-.4)**2
      ETAWNG=ETAWNG+.41*(1.+.00033*ANGLE)-(60.-ANGLE)/3000.
      F=3.395-5.*ETAWNG
      GEE=AR(J)*TANTE*(-.01484)
      H=(AR(J)-4.)*(1.+3.5*TANTE)*.003
      IF(AR(J).LT.4.) H=0.0
      CBARCL=F+GEE+H
      ETAF=BCT(J)/B
      FF=F*(1.-ETAF**2)**.5+(20.*ETAWNG-8.488)*ETAF**2
     1*(1.-ETAF**2)**.5
      GEEF=GEE*(1.-6.666*ETAF+7.316*ETAF**2)*(1.-ETAF**2)**.5
      HF=H*(1.-14.5*ETAF**2+21.*ETAF**4)*(1.-ETAF**2)**.5
      CBARF=FF+GEEF+HF
      SLRATI=(CBARCL+CBARF)*ETAF/2.
      ETAEXP=(ETAWNG-SLRATI*ETAF*.5-(1.-SLRATI)*ETAF)
     1/(1.-SLRATI)
      ETAEXP=ETAEXP*B/BE
      ETAJNF=(2.*LAMBP+1.)/(3.*(LAMBP+1.))
      RLCP=ETAEXP/ETAJNF
      SLTOT=((OLANWT-.5*WW)*NZ(J)+LH(J))
      SL1=SLTOT*(1.-SLRATI)
      SL2=DELP(J)*SEXP
      SL=SL1
      IF (SL2.GT.SL1) SL=SL2
      RBM=RLCP*(SL/6.)*BE/(2.*COST)*(2.*LAMBP+1.)/(LAMBP+1.)
```

```
      IF (J.EQ.2) RBM=4.0*RBM
      TCB=CB(J)*RBSPC/20.
      PF=RBM/(TF*CF*PTBXE(J)*.8)
      TCBVE=TCB/(RHB(J)*144.)+PF/FA(J)
      RRNN=2.*PF**2*RBSPC/(EMBDU(J)*TCBVE*TF*.8)
      TRIBF=TCB/(RHB(J)*144.)+RRNN/FA(J)
      TRSTF=67.5*RBM*12./(3.1416**2*EMBDU(J)*PTBXE(J)*CF*12.*RBSPC)
      IF(TRSTF.GT.TRIBF) TRIBF=TRSTF
      TRIBT=TCB/(RHB(J)*144.)
      WRIBF=RHB(J)*TRIBF*TF*CF*PTBXE(J)*12.
      WRIBT=RHB(J)*TRIBT*TT*CT*PTBXE(J)*12.
      WSWPKR=.058*(ABS(SL*.001*SIN(THETA(J)/57.2958)*BE/
     1(COST*TF/1.2)))**.92
      G(1)=2.*CB(J)*STBE
      G(2)=2.*CB(J)*STBC
      G(3)=RHB(J)*KP*BE*RBM*(1.+LAMBP)/(FA(J)*.8*TF*(2.*LAMBP+1.))*144.
      G(4)=RHB(J)*RBM*2.*COST*BCT(J)*144./(FA(J)*.8*TF)
      TC=TR*(1.-(1.-M)*BLP1(J)*2./BE)
      MPP=TC/TF
      KMPP=2.*(1./(1.-MPP)+MPP*ALOG(MPP)/(1.-MPP)**2)
      KMC=ALOG(MPP)/(MPP-1.)
      BL1=(BLP1(J)-BCT(J)/2.)/COST
      WREL1=RHB(J)*WC1(J)*NZ(J)*BL1**2*144.*KMPP/(FA(J)*.8*TF)
      WREL1=WREL1+RHB(J)*WC1(J)*NZ(J)*BL1*144.*COST*BCT(J)/
     1(FA(J)*.8*TF)
      WREL1=WREL1+RHB(J)*WC1(J)*NZ(J)*BL1*12./TAU(J)
      TC=TR*(1.-(1.-M)*BLP2(J)*2./BE)
      MPP=TC/TF
      KMPP=2.*(1./(1.-MPP)+MPP*ALOG(MPP)/(1.-MPP)**2)
      BL2=(BLP2(J)-BCT(J)/2.)/COST
      WREL2=RHB(J)*WC2(J)*NZ(J)*BL2**2*144.*KMPP/(FA(J)*.8*TF)
      WREL2=WREL2+RHB(J)*WC2(J)*NZ(J)*BL2*144.*COST*BCT(J)/
     1(FA(J)*.8*TF)
      WREL2=WREL2+RHB(J)*WC2(J)*NZ(J)*BL2*12./TAU(J)
      WCM=4.*RHB(J)*CM1(J)*BCM1(J)*144.*KMC/(FA(J)*.8*TF)
      WBREL=(-WREL1-WREL2+WCM)
      G(5)=12.*RHB(J)*BE/COST*(2.*CS(J)*(.8*TF*(1.+MP)/2.)**2
     1+TMIN(J)*(.8*TF*(1.-MP)))
      G(6)=2.*CB(J)*.8*TF*BCT(J)/12.
      G(7)=2.*RHB(J)*RBM*12./TAU(J)
      IF(J.EQ.2) G(7)=G(7)*.5
       G(8)=(WRIBF+WRIBT)/(2.*RBSPC*COST)*BE*12.
      G(9)=WRIBF/(RBSPC)*BCT(J)*12.
      G(10)=(G(1)+G(3)+G(5)+G(7)+G(8))*0.1
      G(11)=(G(2)+G(4)+G(6)+G(9))*0.1
      G(12)=.14*STBE
      G(13)=.14*STBC
      G(14)=.10*G(3)+.20*(G(7)+G(8))
      G(15)=.10*G(4)+G(9)*.2
      G(16)=G(1)+G(3)+G(5)+G(7)+G(8)+G(10)+G(12)+G(14)
     1+WSWPKR+WBREL
      G(17)=G(2)+G(4)+G(6)+G(9)+G(11)+G(13)+G(15)
      G(18)=.25*G(16)
      G(19)=.25*G(17)
      G(20)=G(16)+G(18)
      G(21)=G(17)+G(19)
      G(22)=G(20)+G(21)
      DO 200 IK=1,22
      IF(J.EQ.1) WANS(IK)=G(IK)
      IF(J.EQ.2) TANS(IK)=G(IK)
  200 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE STRUCT (WTOBBC,VB,WBCT,WTHETA,WNZ,WB,SAIL,SRUD
     1,PV,R1,R2,RL,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36)
      IMPLICIT REAL(A-Z)
      COMMON/MAIN/ PROPB,PROPO,RPT,BCANT,OCANT,BCANTY
     1,OCANTP,NOENGB,NOENGO,THBSL,THOSL,THBV,TOV,FLOWR
     2,TF,FTW,FIXHRD
     3,ISPBS,ISPBV,ISPOS,ISPOBV,SCD,OTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THBTC,THOSLT,THOTC,ISPB,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVBNC,DVB,DVTOTC,WBSCD,OINWT,OLANWT
     7,OINJWT,BLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
      COMMON/STRD/
     1 AOI,HTV,LTV,HO,LO,LI,NX,NZ,FS
     2,HF,HL,X
     3,K1,SFW,VC,PC,Q,SND,LNG
     4,FAL,TMIN,RHOL,RHOS,TAUS
     5,FF,LFS,DELP,FAB,RHOB,TAUB
     6,RHOF,K2,SW,K7
     7,K3,K4,ACD,FAF
     8,SAW,FAPB,TAUPB
     9,DFAC,ET,RHOTP,RHOPB
     5,K6
C     MOMENT CALCULATIONS DUE TO INTERSTAGE REACTIONS
      SMAR1=(FS*OLOWT*NX*HO)+(OLOWT*NZ*(LI-LO)*FS)
     1-(FS*TOV*NOENGO*COS(AOI/57.2958)*HTV)
     2+(FS*TOV*NOENGO*SIN(AOI/57.2958)*(LI+LTV))
      SMAR2=(FS*TOV*NOENGO*COS(AOI/57.2958)*HTV)
     1-(TOV*NOENGO*SIN(AOI/57.2958)*LTV*FS)
     2-(OLOWT*NX*HO*FS)+(OLOWT*NZ*LO*FS)
      R2=SMAR1/LI
      R1=SMAR2/LI
      RL=TOV*NOENGO*FS*COS(AOI/57.2958)-OLOWT*NX*FS
C     SHEAR CALCULATIONS
      SR1TL=R1
      SLTR2=R1-OLOWT*NZ*FS
C     MOMENT CALCULATIONS
      MR1TL=R1*X
      MLTR2=R1*(LI-LO)+OLOWT*NX*HO*FS
     1+(R1-OLOWT*NZ*FS)*LO
C     TORQUE CALCULATIONS
      TR2TR1=PV*(.4*VB*12.+HF-(HL*HL/(2.*HL+WBCT*12.)))
C     FWD SECTION WEIGHT CALCULATIONS
      G1=K1*SFW
      G2=3.08*VC**.78*(1.+2.*PC)**.35
      WFW=5800.
      RHOW=.08
      GW=2.*((.75*3.*PC*144.*SW)/(6.*WFW*2.61))**.5*SW*RHOW*144.
      GS=.5*(SW/6.)**.5*288.
      G3=GW+GS
      G4=.35*Q**.3*SND
      ENG=OLANWT*119000./215115.
      G5=.039*(ENG*.001*LNG)**.9
      G6=G4+G5
      G7=K7
      G8=G1+G2+G3+G6
```

```
      G9=G8+G7
C     CENTER SECTIØN WEIGHT CALCULATIØNS
      PLX=MR1TL/HL
      PLY=MLTR2/HL
      AX=PLX/FAL
      AY=PLY/FAL
      G10=(AX+AY)*1.28*RHØL*(LI-X)
      TBYT=(TR2TR1/(WBCT*12.*HL))*(1./TAUS)
      IF(TBYT.LE.TMIN) TBYT=TMIN
      TBY=(SLTR2/HL)*(1./TAUS)
      IF(TBY.LE.TBYT) TBY=TBYT
      TBX=(SR1TL/HL)*(1./TAUS)
      IF(TBX.LE.TMIN) TBX=TMIN
      TBAVG=(TBY+TBX)/2.
      G11=TBAVG*(LI-X)*(2.*HL+WBCT*12.)*RHØS*1.28
C     CENTER SECTIØN FRAME CALCULATIØN
      CF=1./16000.
      DF=6.0
      XCSEC=(6.0*CF*MLTR2*(WBCT*12.)**2.)/(LFS*EF*DF**2.)
      WBM=XCSEC*(WBCT*12.+2.*HF)*RHØF/LFS
      SXCSEC=(3.*DELP*HL*HL*LFS)/(DF*FAF*8.)
      BXCSEC=(144*DELP*WBCT*WBCT*LFS)/(4.*DF*FAF)
      WBP=(SXCSEC*2.*HL+BXCSEC*WBCT*12.)*RHØF/LFS
      IF(WBP.LE.WBM) WBP=WBM
      G12=WBP*(LI-X)*1.28
C     CENTER SECTIØN BULKHEAD CALCULATIØN
      G13=RHØB*((R1/(2.*FAB))*(4.*HF+2.*WBCT*12.)
     1+(R1*WBCT*12./(2.*TAUB)))*1.28
      G14=RHØB*((R2/(2.*FAB))*(4.*HF+2.*WBCT)
     1+(R2*WBCT/(2.*TAUB)))*1.28
      G15=G13+G14
      G16=K2*300.
C     CENTER SECTIØN WING PRØVISIØN CALCULATIØN
      G17=.F*(PLANWT*.001*WNZ)*(WB/CØS(WTHETA/57.2958))**.01
      G18=G10+G11+G12+G15+G16+G17+WTØRBC
C     CENTER SECTIØN DØØR CALCULATIØN
      G19=1.585*ACD
      G20=1.08*(LI-X)*K3
      G21=660.
      G22=G20+G21
      G23=K4
      G24=G18+G19+G22+G23
C     CØVER CALCULATIØNS
C     CØVER SHELL CALCULATIØN
      G25=SAW*TBAVG*RHØS*144.*1.28*1.3
C     CØVER FRAME CALCULATIØN
      G26=220.*WBP*(2.*HF+2.*WBCT*12.)/(2.*HL+WBCT*12.)*1.28*1.3
C     CØVER LØNGERØN CALCULATIØN
      G27=110.*AY*RHØL*2.*1.28*1.3
      G28=G25+G26+G27
C     THRUST STRUCTURE CALCULATIØNS
C     THRUST PØST
      LE=(LTV**2+HTV**2)**.5
      FTP=ET*.00395
      G29=((TØV*NØENGØ)/FTP)*LE*RHØTP*DFAC*2.
C     THRUST GIMBAL PLANE BULKHEAD
      G30=RHØPB*((TØV*NØENGØ*SIN(AØI/57.2958)/(2.*FAPB)
     1*DFAC*FS)*(4.*HF+2.*WBCT*12.)+(TØV*NØENGØ*SIN(AØI/57.2958)
     2*WBCT*12./(2.*TAUPB)))*2.
      G31=NØENGØ*200.
      G32=G29+G30+G31
```

```
      G33=K6
      G34=6.7*(PV*.001)**.6
      G35=G25+G26+G27+G32+G34
      G36=G35+G33
      G37=G9+G24+G36
      RETURN
      END
      SUBROUTINE THERMO(WSG,WESG,TSG,TOTTPS,TWGWT,WGWT,WGLEWT,TWT,TLEWT
     1,BLBTPS,BASEWT,IBTWT,BTPSCN,TTWT,BTPSWT,MCSWT,LDTWT
     2,PROWT,PPPC,PHYC,SCWT)
      IMPLICIT REAL(A-Z)
      COMMON/MAIN/PROPB,PROPO,BBT,BCANT,OCANT,OCANTY
     1,OCANTP,NOENGB,NOENGO,THBSL,THOSL,THOV,TOV,FLOWR
     2,TF,FTW,FIXHRD
     3,ISPBS,ISPBV,ISPOBS,ISPOBV,SCD,BTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THBTC,THBSLT,THOTC,ISPO,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVONC,DVB,DVTOTC,WOSCD,OINWT,OLANWT
     7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
      COMMON/THERD/NCTPS,NCA,FWDTPS,FWDA,CTTPS,CTA,CSTPS
     1,CSA,CBTPS,CBA,ATTPS,ATA,ASTPS,ASA,ABTPS,ABA,BASTPS
     2,BASA,TPSCON,WGTPS,WGPLE,WLETPS,TLTPS,TLPLE
     3,TLETPS,MCSTPS,MCSA,WACON,TACON,IBA,IBTPS,IBC,LDA
     4,LDTPS,PROA,PROTPS,PROC,PPC,HYC,SCA,SCTPS,SWI,SWC,WSI
      SWC=WSG
C     W/S CORRECTION
      WSC=OLANWT/(SWI+SWC)
      DUWT=(WSC/WSI)**.125
C     BODY TPS WEIGHT
      NCWT=NCTPS*NCA*DUWT
      FWDWT=FWDTPS*FWDA*DUWT
      CTWT=CTTPS*CTA*DUWT
      CSWT=CSTPS*CSA*DUWT
      CBWT=CBTPS*CBA*DUWT
      CTOTA=CTA+CSA+CBA
      CTOTWT=CTWT+CSWT+CBWT
      ATWT=ATTPS*ATA*DUWT
      ASWT=ASTPS*ASA*DUWT
      ABWT=ABTPS*ABA*DUWT
      ATOTA=ATA+ASA+ABA
      ATOTWT=ATWT+ASWT+ABWT
      BLBTPS=NCWT+FWDWT+CTOTWT+ATOTWT
      BASEWT=BASTPS*BASA*DUWT
      IBTWT=IBA*IBTPS+IBC
      BTPSWT=NCWT+FWDWT+CTOTWT+ATOTWT+BASEWT+TPSCON+IBTWT
C     WING TPS WEIGHT
      WGWA=WESG*WACON
      WGLEA=WGWA*WGPLE
      WGTPSA=WGWA-WGLEA
      WGWT=WGTPSA*WGTPS*DUWT
      WGLEWT=WGLEA*WLETPS*DUWT
      TWGWT=WGWT+WGLEWT
C     TAIL TPS WEIGHT
      TWA=TSG*TACON
      TLEA=TWA*TLPLE
      TTPSA=TWA-TLEA
      TWT=TTPSA*TLTPS*DUWT
      TLEWT=TLEA*TLETPS*DUWT
      TTWT=TWT+TLEWT
```

```
      MCSWT=MCSTPS*MCSA*DUWT
      LDTWT=LDA*LDTPS
      PRØWT=PRØA*PRØTPS+PRØC
      SCWT=SCA*SCTPS
      TØTTPS=BTPSWT+TWGWT+TTWT+MCSWT
     1+LDTWT+PRØWT+PPC+HYC+SCWT
      PTPSCN=TPSCØN
      PPPC=PPC
      PHYC=HYC
      RETURN
      END
      SUBRØUTINE ASCENT(TAPRØP,ENGPAC,ENG,TVC,CØNTR,PRPUTL
     1,PRØSYS,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC)
      REAL MISC,MISCF,JAC,NØENGØ
      CØMMØN/MAIN/ PRØPB,PRØPØ,HØT,BCANT,ØCANT,ØCANTY
     1,ØCANTP,NØENGB,NØENGØ,THØSL,THØSL,THØV,TØV,FLØWR
     2,TF,FTW,FIXHRD
     3,ISPBS,ISPBV,ISPØBS,ISPØBV,SCD,BTW
     4,H,DVCØRR,INC,STAGV,DVCØN,DVCNST
     5,REL,THBTC,THØSLT,THØTC,ISPØ,ISPB,PRØPØT
     6,PRØPØ1,PRØPØ2,FW(2),DVØNC,DVB,DVTØTC,WØSCD,ØINWT,ØLANWT
     7,ØINJWT,ØLØWT,ØGLØW,GLØW,TØTAL,S,P,MATCH,TLSSR,FPRP
     8,ØHØLD,ØLLPLØ,ØLØWLØ,ØMSISP,ØMSDVT,ØMSDVP,ØMR
     9,LØNGP,TØWØ,TØWB,SENS,GRØW,MINGLW
      CØMMØN/ASPD/PØGØ,SPI,HHEAD,ØHEAD,HULL,ØULL,FTU,RHØ
     1,MATL,HCLEN,ØCLEN,HELEN,ØELEN,CPLGI
      BENGNØ=3.0
      BAENG=6326.
      ESLP=1.225
      BETHST=472000.
      BPUTL=10.
      BFAD=773.
      BPRES=1097.
      BCHIL=133.
      BRECIR=885.
      BDIAIN=12.
      BPØGØ=100.
      BDIAD=17.
      GES=3.
      SUPTF=25.
      HDØØR=50.
      ØDØØR=50.
      MISCF=10.
C     CALCULATE MAIN ENGINE WT.
      ECST=BAENG/BETHST**ESLP
      ENGU=ECST*TØV**ESLP
      ENG=NØENGØ*ENGU
C     CALCULATE TVC WT.
      TVCU=.000422*TØV+208.
      TVC=TVCU*NØENGØ
C     CALCULATE IGNITIØN AND CØNTRØL WT.
      CØNTR=377.+67.*NØENGØ+PØGØ*NØENGØ*53.34
C     CALCULATE PRØPELLANT UTILIZATIØN SYS WT.
      PRPUTL=BPUTL
C      CALCULATE FILL AND DRAIN WT.
      FAD=BFAD
C     CALCULATE PRESSURIZATIØN SYS WT.
      PRES=BPRES*((TØV*NØENGØ)/(BETHST*BENGNØ))**.5
C      CALCULATE CHILLDØWN DUMP SYS WT.
      CHIL=(BCHIL/BENGNØ)*(TØV/BETHST)**.5*NØENGØ
C     CALCULATE RECIRC SYS WT.
```

```
      RECIR=BRECIR*((TBV*NBENGB)/(BETHST*BENGNB))**.5*SPI
      IF(SPI.GT.0.) CHIL=0.
C     CALCULATE PRE VALVE WT.
      DIAIN=BDIAIN*(TBV/BETHST)**.5
      VALVE=1.582*DIAIN**1.78
      BPPV=NBENGB*(VALVE+PBGB*BPBGB)
      HPPV=NBENGB*VALVE
C     CALCULATE EXT TANK DISCONNECT WT.
      DIAD=BDIAD*((TBV*NBENGB)/(BETHST*BENGNB))**.5
      DIADR=(.125*DIAD**2.)**.5
      DIADB=SPI*(.5*DIAD**2.)**.5
      IF(SPI.EQ.0.) DIADB=DIAD
      H2DV=1.582*DIAD**1.78
      B2DV=SPI*2.*1.582*DIADB**1.78
      IF(SPI.EQ.0.) B2DV=H2DV
C     CALCULATE FEED DUCT WT.
      HPRES=HULL+HHEAD*GES*4.4/1728.
      BPRES=BULL+BHEAD*GES*71./1728.
      THD=HPRES*DIAD/FTU
      THDM=0.
      IF(MATL.EQ.1) THDM=.002*DIAD+.008
      IF(MATL.EQ.2) THDM=.003*DIAD+.010
      IF(MATL.EQ.3) THDM=.003*DIAD+.030
      IF(MATL.EQ.4) THDM=.002*DIAD+.024
      IF(THDM.GT.THD) THD=THDM
      TBD=BPRES*DIADB/FTU
      TBDM=0.
      IF(MATL.EQ.1) TBDM=.002*DIADB+.008
      IF(MATL.EQ.2) TBDM=.003*DIADB+.010
      IF(MATL.EQ.3) TBDM=.003*DIADB+.030
      IF(MATL.EQ.4) TBDM=.002*DIADB+.024
      IF(TBDM.GT.TBD) TBD=TBDM
      THE=HPRES*DIAIN/FTU
      TBE=BPRES*DIAIN/FTU
      TEM=0.
      IF(MATL.EQ.1) TEM=.002*DIAIN+.008
      IF(MATL.EQ.2) TEM=.003*DIAIN+.010
      IF(MATL.EQ.3) TEM=.003*DIAIN+.030
      IF(MATL.EQ.4) TEM=.002*DIAIN+.024
      IF(THE.LT.TEM) THE=TEM
      IF(TBE.LT.TEM) TBE=TEM
      HDUCT=HCLEN*3.1416*DIAD*THD*RHB
     &+HELEN*NBENGB*3.1416*DIAIN*THE*RHB
      JAC=HCLEN*(3.1416/2.)*(2.+DIAD)*3.1416*.012
     &*.286+HELEN*NBENGB*3.1416/2.*(2.+DIAIN)*3.1416
     &*.012*.286+(HCLEN*(1.+DIAD)*3.1416*1.
     &+HELEN*NBENGB*(1.+DIAD)*3.1416*1.)*5./1728.
      HDUCT=HDUCT+JAC
      BDUCT=BCLEN*3.1416*DIADB*TBD*RHB*(1.+SPI)
     &+BELEN*NBENGB*3.1416*DIAIN*TBE*RHB
      Y=.012
      Z=8.98
      IF(CPLGI.EQ.1.) Y=.126
      IF(CPLGI.EQ.1.) Z=6.24
      HCBUPL=2.*NBENGB*(Y*DIAIN+Z*.193*DIAIN)
     &+NBENGB*(Y*DIAIN+Z*(.286+RHB)*DIAIN/2.)
     &+NBENGB*(Y*DIAIN+Z*RHB*DIAIN)
     &+(Y*DIAD+Z*(.1+RHB)*DIAD/2.)
      X=(Y*DIAIN+Z*(.286+RHB)*DIAIN/2.)*NBENGB
      IF(RHB.GT..28) HCBUPL=HCBUPL-X
      BCBUPL=2.*NBENGB*(Y*DIAIN+Z*.193*DIAIN)
```

```
          &+NOENGO*(Y*DIAIN+Z*(.286+RHO)*DIAIN/2.)
          &+NOENGO*(Y*DIAIN+Z*RHO*DIAIN)
          &+(1.+SPI)*(Y*DIADO+Z*(.1+RHO)*DIADO/2.)
           X=(Y*DIAIN+Z*(.286+RHO)*DIAIN/2.)*NOENGO
           IF(RHO.GT..28) OCOUPL=OCOUPL-X
           X=INT(HCLEN/300.)
           HBELOS=X*.04178*DIAD**2.1+NOENGO
          &*(.04178*DIAIN**2.1+2.*.01854*DIAIN**2.8)
           X=INT(OCLEN/300.)
           OBELOS=X*.1475*DIADO**2.05+NOENGO
          &*(.1475*DIAIN**2.05+2.*.03451*DIAIN**2.86)
           HSUPT=.01*SUPTF*(HCOUPL+H2PV+H2DV+HDUCT+HBELOS)
           OSUPT=.01*SUPTF*(OCOUPL+O2PV+O2DV+ODUCT+OBELOS)
           HDR=HDOOR
           ODR=ODOOR
           O2FD=OCOUPL+ODUCT+OBELOS+OSUPT+ODR
           H2FD=HCOUPL+HDUCT+HBELOS+HSUPT+HDR
      C    CALCULATE MISC WT.
           MISC=.01*MISCF*(TVC+CONTR+PRPUTL+FAD
          &+PRES+CHIL+RECIR+O2PV+H2PV+H2DV+O2DV
          &+O2FD+H2FD)
      C    SUM DRY WT.
           ATOT=MISC/(.01*MISCF)+MISC+ENG
      C    MAIN ASCENT OUTPUT
           ENGPAC=ENG+TVC+CONTR+PRPUTL
           PREVAL=O2PV+H2PV
           FEEDS=O2FD+H2FD
           DISC=O2DV+H2DV
           PROSYS=FAD+PRES+CHIL+PREVAL+FEEDS+DISC+MISC
           TAPROP=ENGPAC+PROSYS
           RETURN
           END
           SUBROUTINE LADSYS(LNDDK,NG1,NG2,NG3,NGEAR,MG1,MG2,MG3
          1,MGEAR,AX1,AX2,AX3,AXGEAR,LNDDKK)
           IMPLICIT REAL(A-Z)
           COMMON/MAIN/ PROP3,PROPO,BOT,BCANT,OCANT,OCANTY
          1,OCANTP,NOENGB,NOENGO,THBSL,THOSL,THOV,TOV,FLOWR
          2,TF,FTW,FIXHRD
          3,ISPBS,ISPBV,ISPOBS,ISPOBV,SCD,BTW
          4,H,DVCORR,INC,STAGV,DVCON,DVCNST
          5,REL,THBTC,THBSLT,THOTC,ISPO,ISPB,PROPOT
          6,PROPO1,PROPO2,FW(2),DVONC,DVB,DVTOTC,WOSCD,OINWT,OLANWT
          7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
          8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
          9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
           COMMON/LDGD/ LGFTU,LGVSL,LGLC,LGLS,LGDIA,BRCF
           MGR=OLANWT*235000./215000.
           NGR=OLANWT*79000./215000.
           PHI=0.
           THETA=45.
           THETA=THETA*3.14159/180.
           PHI=PHI*3.14159/180.
           NUMNT=2.
           NUMW=4.
           L1=LGLC
           L2=.2*LGLC
           L3=.5*LGLS
           L4=.5*LGLC
           L5=25.5
           L6=9.
           L8=L4*TAN(PHI)
```

```
L7=L6+L3/TAN(THETA)-L8
D=9.
BRKWT = (OLANWT/128.8)*((1.6878*LGVSL)**2.)/BRCF
SP=NGR
TWTT = SP*.00687/2.
TWH=SP/(266.666667*2.)
AXLES = TWH*.44226
VR=.5*OLANWT*1.4
DR=.4*OLANWT*1.4
VI = .4*VR
VB = .6 * VR
VC = (1./L7)*(VB*(L5+L8)-VI*(L5-L8))
SC = VC/TAN(THETA)
VA = (1./L1)*(LGLS*DR-((VR+VC)*(L1/2.)))
VB = VR+VC+VA
MAA=.5*L1*(VA+VB)-L4*DR
AC = 1.26*((MAA/(.85*LGFTU))**(2.0/3.0))
RC = SQRT(AC/.596902604165)
WC = AC*(LGLC-L2)*.283
BRACE = .1*WC
B = 2.*RC-9.*SIN(PHI)
BETA = ATAN((L2+L4)/L1)
LA = (.5*L1-RC)/(COS(BETA)*COS(PHI))
MSIDE = (VB*COS(PHI)+SC*SIN(PHI))*COS(BETA)*LA
MFRONT = (VB*SIN(PHI)-SC*COS(PHI))*LA
D1 = (729.-(54.*MSIDE)/(LGFTU*(2.*RC-9.*SIN(PHI))))**(1./3.)
D4 = 6.*MFRONT/(4.*LGFTU*RC*RC)
AF = (D-(D1-D4))*3+(D1-D4)
WFA = 1.05*AF*LA*.283*.5
MSIDEP = ((VB*COS(PHI)+SC*SIN(PHI))*COS(BETA)-DR*SIN(BETA))*LA
D1P = (729.-(54.*MSIDEP)/(LGFTU*(2.*RC-9.*SIN(PHI))))**(1./3.)
AA = (D-(D1P-D4))*8+(D1P-D4)
WAA = 1.05*AA*LA*.283*.5
MP = DR*(LGLS-LGLC)
AP = 1.26*(MP/LGFTU)**(2./3.)
WP = 1.5*AP*LGLC*.283
TWSSC = WAA+WFA+WC+BRACE
ATF = .06*TWSSC**1.1
WT = TWSSC + WP+BRKWT+TWTT+AXLES+ATF+TWH
MCONTM = .225*(WT**.95)
WT = WT + MCONTM
THETA2 = THETA*180./3.14159
PHI2 = PHI*180./3.14159
SPN = NGR/NUMNT
WNT = SPN*.006875
TWNT = WNT*NUMNT
WHN = SPN/266.66667
AXLN = WHN*.44226
STN = TWSSC*.40
ATFN = .06*TWSSC**1.1
WTNG = TWNT+WHN+AXLN+STN+ATFN
NCONTN = (WTNG**.95)*.850
WTNG = WTNG+NCONTN
WC = WC*1.25
WFA = WFA*1.25
WAA = WAA * 1.25
BRACE = BRACE * 1.25
TWSSC = TWSSC * 1.25
AXLES=AXLES*1.25
WP = WP * 1.25
MCONTM = MCONTM*1.25
```

```
      ATF = ATF*1.25
      BRKWT = BRKWT * 1.25
      TWTT = TWTT*1.25
      TWH = TWH*1.25
      WT = TWSSC+WP+BRKWT + TWTT+AXLES+ATF+TWH+MCONTM
      TWNT = TWNT**.80
      WHN = WHN**.80
      AXLN = AXLN**.80
      STN = STN**.80
      ATFN = ATFN**.80
      NCONTN = NCONTN**.80
      WTNG = TWNT+WHN+AXLN+STN+ATFN+NCONTN
      NG1=TWNT+WHN
      NG2=STN+AXLN+ATFN
      NG3=NCONTN
      NGEAR=NG1+NG2+NG3
      MG1=(TWH+TWTT+BRKWT)*2.
      MG2=(TWSSC+AXLES+ATF+WP)*2.
      MG3=(MCONTM)*2.
      MGEAR=MG1+MG2+MG3
      AX1=1.5*(1.82*(.0074*(.07528/(2.*32.2)*(1.6878*L3VSL)**2)**.57
     &*(LGDIA*LGDIA+3.*LGDIA)+10.))
      AX2=AX2
      AX3=AX3
      AXGEAR=AX1+AX2+AX3
      LNDOK=NGEAR+MGEAR+AXGEAR+LNDOKK
      RETURN
      END
```

4. EXTERNAL TANK MODULE

This module contains the analytical and empirical weight estimation relationships necessary to completely define the external tank. The basic sizing logic consists of three basic general arrangement options and three separate iteration techniques, i.e., solve for specific tank dimensions as a function of volume requirements with either input of fixed length, fixed diameter, or fixed L/D. Design features, such as separate and common bulkhead and an alternate forward section design are also included. A LOX aft option, which simply uses the generalized baseline LOX forward method, setting mixture ratio to its inverse and switching the hydrogen and oxygen densities, is also available.

The external tank module includes a design loads model which considers ullage and head pressure, interstage reactions, and axial load factors.

A multistation analysis method is included, whereby a number of body station cuts are examined to determine the effective unit load and corresponding material thickness required for pure unstiffened Monocoque structure. Alternate material allowables may be input to handle variations in design temperature and other candidate construction techniques. The resultant material thicknesses are integrated over the total body area using the dimensional data from the sizing routine, to determine the total sidewall weight. The bulkheads are sized to their representative loads, i.e., internal or external pressure, and meridional and hoop forces. Splice rings and attachment structure are treated as discrete items, with major attention given to the redistributions of point loads and manufacturing processes such as welding.

The external tank thermal protection system is based on detailed MDAC point design data with input unit weights for alternate design concepts.

Other external tank subsystems are expressed as either input constants for such systems as avionics, or simplified sizing equations, where, for example, plumbing weight is a function of engine flow rate and overall tank length/diameter.

Figure 4-1 is a flow diagram of the External Tank Module, followed by a detail listing of the program.

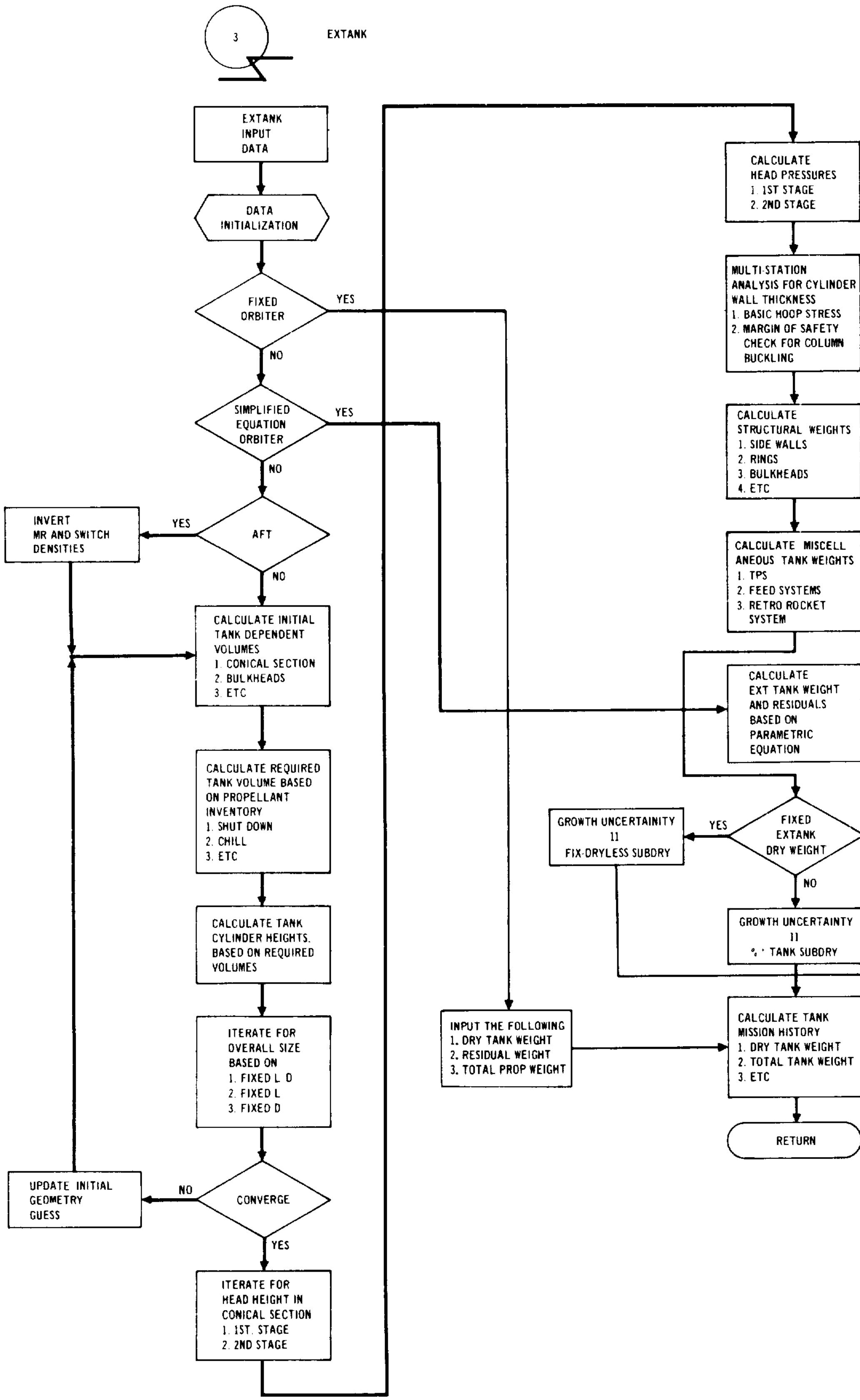

FIGURE 4–1 FLOW DIAGRAM

```
COPY ESUB TO LP(K,NC)
$FIXED
C
C                          EXTERNAL TANK SUBROUTINE
C
      SUBROUTINE EXTANK
C     THIS PROGRAM COMPUTES EXTERNAL (LOX/LH2) TANK DIMENSIONS
C     AND WEIGHT KNOWING MIXTURE RATIO,USEABLE PROPELLANT
C     LOAD AND EITHER REQUIRED TANK L/D,LENGTH OR DIAMETER.
      IMPLICIT REAL(A-Z)
      INTEGER I
      DIMENSION HFX(9),
     &OHCT(9),WLDF(9),WLDO(9),
     &WF(9),WO(9),FHCT(9),HOX(9),TFT(9),TO(9),MF(9),MO(9)
C
C     PERFORMANCE COMMON BLOCK
C
      COMMON/MAIN/ PROPB,PROPO,BOT,BCANT,OCANT,OCANTY
     1,OCANTP,NOENGB,NOENGO,THBSL,THOSL,THOV,TOV,FLOWR
     2,TF,FTW,FIXHRD
     3,ISPBS,ISPBV,ISPOBS,ISPOBV,SCD,BTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THBTC,THBSLT,THOTC,ISPO,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVONC,DVB,DVTOTC,WOSCD,OINWT,OLANWT
     7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
C
C     SRM OUTPUT COMMON BLOCK
C
      COMMON/SRMO/BOOWT,BDRYWT,BGLOW,LAMB,PWOI
     1,BASSRM,WCASE,WJOINT,WNOZZ,WTTER,WINST,WIGN,BSRMC
     2,SRMISS,PWFS,PWASLS,PWAS,PWNF,PWTN,PWAV,WNCTPS,SRMIC
     3,WRECOV,PWPAR,PWPI,PWRR,PWRP,PWWR,SRMRC
     4,UNCERT,EXPINS,BOSLUN,SRML,SRMD
     5,PGROSS,PBOOWT,PPROPB,PDRYWT,PBOSLU
     6,FIXBOO,SIMPBO
C
C     ORBITER OUTPUT COMMON BLOCK
C
      COMMON/ORBO/R1,R2,RL,WTAUX,WTACS,ACSENG,ACSSYS,WTACTK
     1,ACSMOD,WTOMS,OMSENG,PROPSY,WTOMTK,MODULE
     1,SURFC,PPWR,ELEC,HYDR,AVIONO,ECLSS,PPROV,OUNCWT
     2,ORBMIS,TABPRP,SURFK
     1,PERSON,ORESD,ORESV,PLOADU,PLOADD,ACSPRO,WOPROP,SUDLE
     2,FIXORB,FIXWOR,ODRYWT,ORIFL
     1,WWT,WSG,WBSTR,WTORBE,WTORBC,LEW,WTE,WAIL,WAS,WADR,WAH
     2,WAP,GPROV,PWINGK,TAIL,TSG,TBSTR,TTORGB,TLE,WRUD,WRS
     3,WRDR,WRH,WRP,PTAILK
     1,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36
     1,TOTTPS,TWGWT,WGWT,WGLEWT,TWT,TLEWT,BLBTPS
     2,BASEWT,IBTWT,PTPSCN,TTWT,BTPSWT,MCSWT,LDTWT
     3,PROWT,PPPC,PHYC,SCWT
     1,TAPROP,ENGPAC,ENG,TVC,CONTR,PRPUTL,PROSYS
     2,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC
     1,LNDDK,NG1,NG2,NG3,NGEAR,MG1,MG2,MG3,MGEAR,AX1,AX2
     2,AX3,AXGEAR,LNDDKK
C
```

```
 C      EXTERNAL TANK 8UTPUT C8MM8N BL8CK
 C
        C8MM8N/EXT8/8P8GRP,T8TPS,FWDTK,FAIRT,FWDBLF,FCCTPS
       1,C8NSCT,TPSIN,CYLSCT,SCYDM,AFTBLF,WINT,PR8SY,AFTNK
       2,FEDSYS,FWDBLA,PRSVNT,AFTCYL,SUMP,AFTBLA,PNPJ,TWINT
       3,N8SEAR,AVI8NT,JMBPNL,WRETR8,TUNNEL,MISCT,BAFF,SJBDRY
       4,GU,DRYWT,RESIDT,UNDRAN,FEEDTR,PRSJRT,FBIAS,INERT
       5,GR8SSW,TLAMB,8TRAP,EXTL,EXTD,BLKHD,EXTH8,EXTHH,SIMPTK
 C
 C      EXTERNAL TANK (EXT) INPUT DATA BL8CK
 C
        NAMELIST
       & DI,LI,LD,NR,ND,THETA,HHI,MRI,UPER8,LA,F8PRES,8PRES
       &,FUPRES,8UPRES,LF,DF,LC8N,BLKHD,BX,K,UPERF,
       &8HPIAS,FS,NXL,NXS,FTU,E,RH8,TMIN,
       &NCTPS,UCTPS,LCTPS,CYTPS,INTPS,DMTPS,FIXDWT,GJP,
       &RETDV,RETISP,AVI8NT,MISCT,AFT,HRI,RX1,RX2,RXL
       &,SIMPTK,INERT,DRYWT,RESIDT,PR8MIN
 C
        IF(P.EQ.2.0) G8 T8 1
        INPUT(1)
        L=LI
     1  IF(FIX8R8.GT.0.0) G8 T8 400
        IF(SIMPTK.GT.0.0) G8 T8 500
        IF(PR8P8T.LE.PR8MIN.AND.DF.EQ.0.0)  G8 T8 600
        IF(AFT.NE.0.) G8 T8 11
        G8 T8 12
 C      SET L8X AFT(INVERT MR & SWITCH DENSITIES)
    11  MR=1./MRI
        XFUPR=8UPRES
        XF8PR=8PRES
        X8UPR=FUPRES
        X8PR=F8PRES
        XPER8=UPERF
        XPERF=UPER8
        FDEN=71.
        8DEN=4.4
        G8 T8 13
 C      BASELINE L8X F8RWARD
    12  FDEN=4.4
        8DEN=71.
        MR=MRI
 C      INITIALIZE DIMENSI8NS DF=FIXED DIAMETER,
 C      LF=FIXED LENGTH,LD=FIXED L/D,LI,DI,HHI ARE
 C      INITIAL GUESSES
    13  HH=HHI
        FUEL=PR8P8T/(1.+MR)
        8XID=FUEL*MR
        IF(LF.EQ.0.0) G8 T8 5
        TV8L=FUEL/FDEN+8XID/8DEN
        D=((TV8L*1728.)/(.7854*LF))**.5
     5  IF(DF.GT.0.0) D=DF
        L=LI
        IF(LF.GT.0.0) L=LF
    10  IF(LD.GT.0.) D=L/LD
        D=D-2.*K
 C      CALCULATE DEPENDENT DIMENSI8NS,THETA IS FWD
 C      C8NE ANGLE INPUTTED IN DEGREES
    15  IF(BLKHD.GT.2.) G8 T8 16
        ND=2.*NR*C8S(THETA/57.2958)
    16  HC=(.5*D-.5*ND)/TAN(THETA/57.2958)
```

```
      R=D/3.**.5
      HR=HRI*R
      IF(BLKHD.GT.2.) GO TO 17
      HCO=HC*.1364/SIN(THETA/57.2958)
      OD=D-2.*HCO*TAN(THETA/57.2958)
      OR=OD/3.**.5
      HOR=.5*OR
      HN=NR-.5*(4.*NR**2-ND**2)**.5
C     CALCULATE DEPENDENT VOLUMES
      VOLF=.2618*HCO*(D**2+D*OD+OD**2)
      VOLG=1.0472*HOR**2*(3.*OR-HOR)
      GO TO 18
   17 VOLF=.2618*HC*(D**2+D*ND+ND**2)
   18 VOLA=1.0472*HR**2*(3.*R-HR)
      VOLC=.7854*D**2*HR-VOLA
      IF(BLKHD.GT.1.0) VOLC=VOLA
      VOLD=VOLA
C     CALCULATE PROPELLANT INVENTORY
      OFLOWX=FLOWR/(1.+MR)
      OFLOWY=FLOWR*MR/(1.+MR)
C     FUEL INVENTORY
      FSTART=3.255*OFLOWX
      IF(OX.NE.1.0) FSTART=.84*OFLOWX
      FSHUT=.1965*OFLOWX
      FEEDF=.00189*D*OFLOWX
      FCHILL=.01355*OFLOWX
      IF(OX.NE.1.0) FCHILL=.0435*OFLOWX
      FENG=.16941*OFLOWX
      FBIAS=.0025*FUEL
      IF(HBIAS.GT.0.) FBIAS=HBIAS
      FDRAIN=100.
      FBELOW=FEEDF+FCHILL+FENG
      DISPVF=.0000952*(HH-HR)*OFLOWY
      IF(BLKHD.GT.1.0) DISPVF=0.
      IF(AFT.EQ.0.0) GO TO 35
      UPERF=XPERF
      FUPRES=XFUPR
      FOPRES=XFOPR
      FPRESS=.001326*FUEL*FUPRES/18.
      FRES=.00457*FUEL*(UPERF+LA-2.)*FOPRES/30.
      GO TO 36
   35 FPRESS=.003635*FUEL*FUPRES/40.
      FRES=.02667*FUEL*(UPERF+LA-2.)*FOPRES/40.
C     NOMINAL FUEL LOAD
   36 NOMFUL=FUEL+FSTART+FSHUT+FEEDF+FCHILL+FENG+FBIAS+FDRAIN
     &+FPRESS
C     FUEL LOADING ALLOWANCE
      FALLOW=NOMFUL*(LA-1.)
      MAXFUL=NOMFUL+FALLOW
C     MAXIMUM FUEL IN TANK
      MAXFIT=MAXFUL-FBELOW
C     FUEL VOLUME
      FUVOL=MAXFIT/FDEN
C     ADD FUEL ULLAGE VOLUME
      TFVFL=FUVOL*UPERF
C     ADD VOL DISPLACED BY LOX LINE IF COMMON BLKHD DES
      TFVIT=TFVFL+DISPVF
C     OXYGEN INVENTORY
      OSTART=2.7912*OFLOWY
      IF(OX.NE.1.0) OSTART=.382*OFLOWY
      OSHUT=.0565*OFLOWY
```

```
      FEEDØ=.3456*ØFLØWY
      ØCHILL=.05044*ØFLØWY
      IF(3X.NE.1.0) ØCHILL=.1025*ØFLØWY
      ØENG=.40358*ØFLØWY
      ØDRAIN=0.
      IF(BLKHD.GT.1.) ØDRAIN=300.
      ØLINE=.0046*(HH-HR)*ØFLØWY
      IF(BLKHD.GT.1.0) ØLINE=.0046*(HH+.5*D+LCØN)*ØFLØWY
      ØBELØW=ØLINE+FEEDØ+ØCHILL+ØENG
      IF(AFT.EQ.0.0) GØ TØ 37
      UPERØ=XPERØ
      ØUPRES=XØUPR
      ØPRES=XØPR
      ØPRESS=.003635*ØXID*ØUPRES/40.
      ØRES=.02667*ØXID*(UPERØ+LA-2.)*ØPRES/40.
      GØ TØ 38
   37 ØPRESS=.001326*ØXID*ØUPRES/15.
      ØRES=.00457*ØXID*(UPERØ+LA-2.)*ØPRES/30.
C     NØMINAL ØXYGEN LØAD
   38 NØMØXL=ØXID+ØSTART+ØSHUT+FEEDØ+ØCHILL+ØENG+ØPRESS+ØDRAIN
C     ØXYGEN LØADING ALLØWANCE
      ØALLØW=NØMØXL*(LA-1.)
      MAXØXL=NØMØXL+ØALLØW
C     MAXIMUM ØXYGEN IN TANK
      MAXØIT=MAXØXL-ØBELØW
C     ØXYGEN VØLUME
      ØXVØL=MAXØIT/ØDEN
C     ADD ØXYGEN ULLAGE VØLUME
      TØVFL=ØXVØL*UPERØ
C     FUEL TANK HEIGHT AS FUNCTIØN ØF TØTAL FUEL TANK VØL
C     REQUIRED LESS PREVIØUSLY CALC DEPENDENT VØLUMES
      HH=(TFVIT*1728.-VØLA-VØLC)/(.7854*D**2)+HR
      IF(HH.LT.HR) HH=HR
      IF(BLKHD.GT.2.) VØLG=0.
C     ØXY TANK CYL HEIGHT AS FUNCTIØN ØF TØT ØXY TANK VØL
C     REQUIRED LESS PREV CALC DEP VØLS;IF CØNE IS BIG
C     ENØUGH CYL HEIGHT (HØ) IS SET EQUAL TØ ZERØ.
      HØ=(TØVFL*1728.-VØLD-VØLF-VØLG)/(.7854*D**2)
      IF(HØ.LT.0.) HØ=0.
      IF(BLKHD.GT.2.) HN=0.
C     CALCULATE ØVERALL TANK LENGTH
      L1=HN+HC+HH+HR+HØ
      IF(BLKHD.GT.1.0) L1=L1+HR+LCØN
C     ITERATE TØ REQUIRED DIMENSIØNAL CØNSTRAINTS
      IF(ABS(L1-L).LT.1.) GØ TØ 20
      IF(DF.GT.0.) L=L1
      IF(DF.GT.0.) GØ TØ 15
      IF(LF.GT.0.) D=D*(L1/LF)
      IF(LD.GT.0.) L=(L1+L)/2.
      GØ TØ 10
   20 L=L1
C     CALCULATE PRØP INVENTØRY SUBTØTALS
      UNDRAN=ØDRAIN+FDRAIN
      FEEDTR=FEEDF*.3333+FEEDØ*.2418
      PRSURT=FPRESS+FRES+ØPRESS+ØRES
      RESIDT=UNDRAN+FEEDTR+PRSURT+FBIAS
C     CALC RESULTING VØLS FØR CHECK AGAINST REQD VØLS
      ØTRAP=FEEDØ+ØENG+ØCHILL+FEEDF+FENG+FCHILL-FEEDTR
      ØXVLF=(VØLD+VØLF+VØLG+HØ*.7854*D**2)/1728.
      FUVLF=(VØLA+VØLC+(HH-HR)*.7854*D**2)/1728.
C     CALCULATE LIFT-ØFF ØXYGEN HEAD HEIGHT;IS HEAD ABØVE
```

```
C     OR BELOW CONE/CYL CONTOUR BREAK
      HO1=(OXVOL*1728.-VOLA)/(.7854*D**2)
      IF(HO1.GT.HO) GO TO 21
      GO TO 24
C     IF YES ITERATE FOR HEAD HEIGHT IN CONICAL SECTION
21    HOH1=HO1-HO
      OXVOL1=OXVOL*1728.-VOLA-HO*.7854*D**2
22    D1=D-2.*HOH1*TAN(THETA/57.2958)
      VO1=.2618*HOH1*(D**2+D*D1+D1**2)
      IF(ABS(OXVOL1-VO1).LT.100.) GO TO 23
      HOH1=(OXVOL1*HOH1)/VO1
      GO TO 22
23    HOH=HOH1+HO
      GO TO 25
24    HOH=HO1
C     CALCULATE LIFT-OFF FUEL HEAD HEIGHT
25    IF(HH.EQ.HR) GO TO 51
      HFH=(VOLA-1728.*FUVOL)/(1728.*DISPVF/(HH-HR)-.7854*D**2)
      GO TO 52
51    HFH=(VOLA-1728.*FUVOL)/(-.7854*D**2)
C     OXY ULT ULLAGE PRESSURE
52    PULLO=FS*OPRES
C     FUEL ULT ULLAGE PRESSURE
      PULLF=FS*FOPRES
C     OXY AFT DOME ULT LIFT-OFF HEAD PRESSURE
      OHDM1=(HOH+HR)*ODEN*NXL*FS/1728.
C     FUEL AFT DOME ULT LIFT-OFF HEAD PRESSURE
      FHDM1=(HFH+HR)*FDEN*NXL*FS/1728.
C     2ND STAGE FUEL LOAD
      FUEL2=PROPO2/(1.+MR)
C     2ND STAGE OXYGEN LOAD
      OXID2=FUEL2*MR
      XFUL2=FUEL2+MAXFIT-FUEL-FSTART-FCHILL
      XOXD2=OXID2+MAXOIT-OXID-OSTART-OCHILL
C     2ND STAGE FUEL VOLUME
      VXF2=XFUL2/FDEN
C     2ND STAGE OXYGEN VOLUME
      VXO2=XOXD2/ODEN
C     2ND STAGE FUEL HEAD HEIGHT
      IF(HH.EQ.HR) GO TO 53
      HFH2=(VOLA-1728.*VXF2)/(1728.*DISPVF/(HH-HR)-.7854*D**2)
      GO TO 54
53    HFH2=(VOLA-1728.*VXF2)/(-.7854*D**2)
54    IF(HFH2.LE.0.0) HFH2=0.0
C     CALCULATE 2ND STAGE OXYGEN HEAD HEIGHT..IS HEAD
C     ABOVE OR BELOW CONE/CYL CONTOUR BREAK
      HO2=(VXO2*1728.-VOLA)/(.7854*D**2)
      IF(HO2.GT.HO) GO TO 26
      GO TO 29
C     IF YES ITERATE FOR HEAD HEIGHT IN CONICAL SECTION
26    HOHX=HO2-HO
      OXVOL2=VXO2*1728.-VOLA-HO*.7854*D**2
27    D2=D-2.*HOHX*TAN(THETA/57.2958)
      VO2=.2618*HOHX*(D**2+D*D2+D2**2)
      IF(ABS(OXVOL2-VO2).LT.100.) GO TO 28
      HOHX=(OXVOL2*HOHX)/VO2
      GO TO 27
28    HOH2=HOHX+HO
      GO TO 50
29    HOH2=HO2
C     OXY AFT DOME ULT 2ND STAGE HEAD PRESSURE
```

```
   50 OHDM2=(HOH2+HR)*ODEN*NXS*FS/1728.
C     FUEL AFT DOME ULT 2ND STAGE HEAD PRESSURE
      FHDM2=(HFH2+HR)*FDEN*NXS*FS/1728.
C     ULT OXY AFT DOME ULLAGE+HEAD PRES EITHER LIFT-
C     OFF OR 2ND STAGE WHICH EVER IS GREATER
      PODOM=PULLO+OHDM1
      IF(OHDM2.GT.OHDM1) PODOM=PULLO+OHDM2
C     ULT FUEL AFT DOME ULLAGE+HEAD PRES EITHER LIFT-
C     OFF OR 2ND STAGE WHICH EVER IS GREATER
      PFDOM=PULLF+FHDM1
      IF(FHDM2.GT.FHDM1) PFDOM=PULLF+FHDM2
C     INITIALLY SET WT'S & T'S TO ZERO BEFORE BEGINNING
C     MULTI-STATION ANALYSIS
      WFX=0.
      WOX=0.
      WLDFX=0.
      WLDOX=0.
      TAF=0.
      TAO=0.
C     MULTI-STATION ANALYSIS (FUEL TANK REFERS TO AFT TANK
C     AND OXY TANK REFERS TO FWD TANK; THE REVERSE IS LITERALLY
C     TRUE FOR THE LOX AFT OPTION BECAUSE OF LINE 18.5
C     AT BEGINNING OF PROGRAM; THE FWD AND AFT TANKS ARE
C     THEREFORE ANALIZED IDENTICALLY IN THE EVENT THAT THE FWD
C     TNK IS LARGE AND THE AFT TNK IS SMALL AND VICE VERSA)
      DO 84 I=1,9
      HF=HH-HR
C     SET STATION LOCATIONS FOR FUEL TANK CYL WALL ANALYSIS
      HFX(I)=.125*(I-1)*HF
C     CALCULATE FUEL TNK CYL WALL HEAD PRES @ HFX(I)
C     ABOVE CYL BASE BY COMPARING WITH LIFT-OFF AND
C     2ND STAGE HEAD HEIGHTS AS CALCULATED PREVIOUSLY
      FHC2=(HFH2-HFX(I))*FDEN*NXS*FS/1728.
      FHC1=(HFH-HFX(I))*FDEN*NXL*FS/1728.
      IF(FHC1.LT.FHC2) GO TO 59
      FHCX=FHC1
      GO TO 60
   59 FHCX=FHC2
   60 IF(FHCX.LE.0.) FHCX=0.
C     ADD FUEL ULLAGE PRESS TO RESULTING HEAD PRESSURE
      FHCT(I)=FHCX+PULLF
C     SET STATION LOCATIONS FOR OXY TANK CYL WALL ANALYSIS
      HOX(I)=.125*(I-1)*HO
C     CALCULATE OXY TNK CYL WALL HEAD PRES @ HOX(I)
C     ABOVE CYL BASE BY COMPARING WITH LIFT-OFF AND
C     2ND STAGE HEAD HEIGHTS AS CALCULATED PREVIOUSLY
      OHC2=(HOH2-HOX(I))*ODEN*NXS*FS/1728.
      OHC1=(HOH-HOX(I))*ODEN*NXL*FS/1728.
      IF(OHC1.LT.OHC2) GO TO 63
      OHCX=OHC1
      GO TO 64
   63 OHCX=OHC2
   64 IF(OHCX.LE.0.) OHCX=0.
C     ADD OXY ULLAGE PRESS TO RESULTING HEAD PRESSURE
      OHCT(I)=OHCX+PULLO
C     CALCULATE FUEL TANK BENDING MOMENTS
      MF(I)=0.
C     CALCULATE OXYGEN TANK BENDING MOMENTS
      MO(I)=0.
C     CALCULATE FUEL TNK THICKNESS REQUIRED DUE TO ULT PRES
      TFT(I)=FHCT(I)*D*.5/FTU
```

```
C      CALCULATE OXY TNK THICKNESS REQUIRED DUE TO ULT PRES
       TO(I)=OHCT(I)*D*.5/FTU
C      CHECK FUEL TNK THICKNESS & SET = OR GREATER THAN TMIN
       IF(TFT(I).LT.TMIN)  TFT(I)=TMIN
C      CHECK OXY TNK THICKNESS & SET = OR GREATER THAN TMIN
       IF(TO(I).LT.TMIN)  TO(I)=TMIN
C      CALCULATE ULT LIFT- OFF & 2ND STAGE AXIAL LOAD USE
C      WHICH EVER IS GREATER
       AL1=MAXOIT*FS*NXL
       AL2=XOXO2*FS*NXS
       AL=AL1
       IF(AL2.GT.AL1) AL=AL2
C      CHECK MARGIN OF SAFETY FOR COLUMN BUCKLING FOR B.M.'S
C      AND AXIAL LOAD WITH INTERNAL PRESSURE STABILIZATION
       FROT=.5*D/TFT(I)
       FLOR=(L-HC-HR-HV-HFX(I))/(.5*D)
       RXT=FROT
       LXR=FLOR
       TX=TFT(I)
       TPRES=FUPRES
       MX=MF(I)
       GO TO 70
    69 RXT=OROT
       LXR=OLOR
       TX=TO(I)
       TPRES=OUPRES
       MX=MO(I)
       AL=0.
C      CURVE FIT OF BRUHN'S GRAPHS
    70 IF(LXR.GT.1.0) GO TO 73
       A=7.34127+.269862*LXR
       IF(LXR.GT..26) GO TO 71
       F=7.26403+7.93752*LXR
       GO TO 79
    71 IF(LXR.GT..5) GO TO 72
       F=9.34345
       GO TO 79
    72 F=9.63181-.59176*LXR
       GO TO 79
    73 IF(LXR.GT.4.0) GO TO 74
       A=8.30526-.671566*LXR
       IF(LXR.LT.2.0) GO TO 72
    75 F=8.88855-.212134*LXR
       GO TO 79
    74 IF(LXR.GT.6.0) GO TO 76
       A=5.61685
       GO TO 75
    76 IF(LXR.GT.16.0) GO TO 77
       A=6.47239-.144808*LXR
       IF(LXR.LT.8.0) GO TO 75
    78 F=8.25156-.133275*LXR
       GO TO 79
    77 A=5.14063-.061365*LXR
       GO TO 78
    79 B=LXR/(-.624947*LXR-.007487)
       G=LXR/(-.626078*LXR-.008016)
C      CRITICAL STRESS CALCULATION
    80 FCRE=A*RXT**B
       FCR=FCRE*E
       FBRE=F*RXT**G
       FBR=FBRE*E
```

```
C     CØRRECT FØR INTERNAL PRESSURE STABILIZATIØN
      PXE=TPRES*RXT**2/E
      DFLAX=PXE/(4.41405*PXE+.603563)
      DLFCR=DFLAX*E*TX/.5*D
      FPT=TPRES*.5*D/(2.*TX)
      PA=(FCR+DLFCR+FPT)*2.98451*D*TX
      DELBM=.376601*PXE**.210492
      DLFBM=DELBM*E*TX/(.5*D)
      MA=(FBR+DLFBM+FPT)*.74612*TX*D**2
C     CALCULATE MARGIN ØF SAFETY AND INCREASE T IF INADEQUATE
      RCB=AL/PA+MX/MA
      IF(RCB.LT.1.0) GØ TØ 81
      TX=TX+.001
      RXT=.5*D/TX
      GØ TØ 80
   81 IF(FLØR.EQ.LXR) GØ TØ 82
      IF(ØLØR.EQ.LXR) GØ TØ 83
   82 TFT(I)=TX
C     SUM FUEL TANK T'S FØR AVERAGE CALCULATIØN
      TAF=TAF+TFT(I)
      ØRØT=.5*D/TØ(I)
      ØLØR=(HØ-HØX(I))/(.5*D)
      GØ TØ 69
   83 TØ(I)=TX
C     SUM ØXY TANK T'S FØR AVERAGE CALCULATIØN
      TAØ=TAØ+TØ(I)
C     CALC FUEL TNK CYL WALL WT ADD .005 TØ T FØR MATL TØLERANCE
      WF(I)=(TFT(I)+.005)*3.14159*D*.125*HF*RHØ
      WFX=WFX+WF(I)
C     CALC FUEL TNK CIRCUMFERENTIAL WELD @ 3.5  WIDE
      WLDF(I)=(TFT(I)+.005)*3.14159*D*3.5*RHØ
      WLDFX=WLDFX+WLDF(I)
C     CALC ØXY TNK CYL WALL WT ADD .005 TØ T FØR MATL TØLERANCE
      WØ(I)=(TØ(I)+.005)*3.14159*D*.125*HØ*RHØ
      WØX=WØX+WØ(I)
C     CALC ØXY TNK CIRCUMFERENTIAL WELDS @ 3.5  WIDE
      WLDØ(I)=(TØ(I)+.005)*3.14159*D*3.5*RHØ
      WLDØX=WLDØX+WLDØ(I)
   84 CØNTINUE
C     DELETE MID CIRC WELDS IF TNK IS SHØRT I.E. LØX AFT/FWD
      IF(HØ.GT.HH) GØ TØ 85
      WLDØT=WLDØ(1)+WLDØ(9)
      WLDFT=WLDFX
      GØ TØ 86
   85 WLDØT=WLDØX
      WLDFT=WLDF(1)+WLDF(9)
C     RING MINIMUM GAUGE
   86 AMIN=14.*TMIN
C     AVERAGE FUEL TANK T
      AVFT=TAF/9.
C     AVERAGE ØXYGEN TANK T
      AVØT=TAØ/9.
C     CALC NØ. ØF LØNGITUDINAL WELDS BASED ØN 156  WIDE SHEETS
      CIR=3.14159*D
      DØ 87 I=1,100
      SX=156.*I
      IF(SX.GT.CIR) GØ TØ 88
   87 CØNTINUE
   88 SEG=SX/156.
C     CALCULATE INTER-TANK SECTIØN WEIGHT
      TINT=TØ(1)
```

```
      IF(TØ(1).LT.TFT(9)) TINT=TFT(9)
      WINT=(3.14159*D+3.5*SEG)*(TINT+.005)*(2.*HR+LCØN)*RHØ
      IF(BLKHD.EQ.1.0) WINT=0.
C     CALCULATE FUEL TNK LØNGITUDINAL WELD WEIGHT
      LWLDF=HF*SEG*3.5*(AVFT+.005)*RHØ
      IF(BLKHD.EQ.1.0) LWLDF=HH*SEG*3.5*(AVFT+.005)*RHØ
C     CALCULATE ØXY TNK LØNGITUDINAL WELD WEIGHT
      LWLDØ=HØ*SEG*3.5*(AVØT+.005)*RHØ
C     AFT FUEL BLKHD THICKNESS
      TAFB=PFDØM*R/(2.*FTU)
      IF(TAFB.LT.TMIN)  TAFB=TMIN
C     AFT FUEL BLKHD RING AREA SIZED FØR ELASTIC STABILITY
      NAFB=(PFDØM*R/2.)*SIN(30./57.2958)
      AAFB=(NAFB*(.5*D)**3/(3.*E))**.5
      IF(AAFB.LT.AMIN) AAFB=AMIN
C     AFT ØXY BLKHD THICKNESS
      TAØB=PØDØM*R/(2.*FTU)
      IF(TAØB.LT.TMIN)  TAØB=TMIN
C     AFT ØXY BLKHD RING AREA SIZED FØR ELASTIC STABILITY
      NAØB=(PØDØM*R/2.)*SIN(30./57.2958)
      AAØB=(NAØB*(.5*D)**3/(3.*E))**.5
      IF(AAØB.LT.AMIN) AAØB=AMIN
C     FWD FUEL BLKHD THICKNESS
      TFFB=PULLF*R/(2.*FTU)
      IF(TFFB.LT.TMIN)  TFFB=TMIN
C     FWD FUEL BLKHD RING AREA SIZED FØR ELASTIC STABILITY
      NFFB=(PULLF*R/2.)*SIN(30./57.2958)
      AFFB=(NFFB*(.5*D)**3/(3.*E))**.5
      IF(AFFB.LT.AMIN) AFFB=AMIN
C     WEIGHT ØF AFT ØXY BLKHD RING AND WELD
      WAØB=(TAØB+.005)*2.*3.14159*R*HR*RHØ
      WAØR=3.14159*D*RHØ*AAØB
      WAØW=RHØ*3.5*(TAØB+.005)*3.14159*(.333*R*SEG+D)
C     WEIGHT ØF AFT FUEL BLKHD RING AND WELD
      WAFB=(TAFB+.005)*2.*3.14159*R*HR*RHØ
      WAFR=3.14159*D*RHØ*AAFB
      WAFW=RHØ*3.5*(TAFB+.005)*3.14159*(.333*R*SEG+D)
C     WEIGHT ØF FWD FUEL BLKHD RING AND WELD
      WFFB=(TFFB+.005)*2.*3.14159*R*HR*RHØ
      WFFR=3.14159*D*RHØ*AFFB
      WFFW=RHØ*3.5*(TFFB+.005)*3.14159*(.333*R*SEG+D)
      IF(BLKHD.EQ.1.0) GØ TØ 90
      GØ TØ 91
C     CØMMØN BLKHD T-BAR BASED ØN REVERSE FUEL ULLAGE
C     PRESSURE AND ISØ-GRID MATL.
   90 TCFB=(.000614+(.0000206*FØPRES))*R
C     WEIGHT ØF CØMMØN FUEL BLKHD AND WELD
      WCFB=TCFB*2.*3.14159*R*HR*RHØ
      WCFW=RHØ*3.5*TCFB*3.14159*(.333*R*SEG+D)
C     CHECK WITH AFT ØXY BLKHD TENSIØN LØAD REQMTS
      IF(WCFB.LT.WAØB) WCFB=WAØB
      IF(WCFW.LT.WAØW) WCFW=WAØW
      WFFB=WCFB
      WFFW=WCFW
      WAØW=0.
      WAØB=0.
      WFFR=WAØR
      WAØR=0.
C     FWD CØNE AND FAIRING ANALYSIS
   91 IF(BLKHD.GT.2.0) GØ TØ 92
C     UPR CØNE WALL THICKNESS
```

```
      TUCN=PULLB*BR/FTU
      IF(TUCN.LT.TMIN)  TUCN=TMIN
C     FWD OXY BLKHD THICKNESS
      TFOB=PULLB*BR/(2.*FTU)
      IF(TFOB.LT.TMIN)  TFOB=TMIN
C     FWD OXY BLKHD RING AREA SIZED FOR ELASTIC STABILITY
      NFOR=ABS((PULLB/2.)*(BR*SIN(30./57.2958)-.5*BD*TAN(THETA/
     &57.2958)))
      AFOR=(NFOR*(.5*BD)**3/(3.*E))**.5
      IF(AFOR.LT.AMIN) AFOR=AMIN
C     WEIGHT OF FWD OXY BLKHD RING AND WELD
      WFOR=3.14159*BD*RHO*AFOR
      WFOB=(TFOB+.005)*2.*3.14159*BR*HOR*RHO
      WFOW=RHO*3.5*(TFOB+.005)*3.14159*(.333*BR*SEG+BD)
C     WEIGHT OF FWD FAIRING USING MINIMUM GAUGE PLUS WELDS
      WFAIR=RHO*(TMIN+.005)*3.14159*(2.*NR*HN+(.5*BD+.5*ND)*
     &((HC-HCO)**2+(.5*BD-.5*ND)**2)**.5)
      WFAW=RHO*3.5*(TMIN+.005)*(3.14159*ND*SEG*(HC-HCO)/
     &COS(THETA/57.2958))
C     WEIGHT OF FWD CONE OXY TANK WALL AND WELDS
      WCON=RHO*3.14159*.5*((TUCN+.005)*BD+(TO(9)+.005)*D)*HCO/
     &COS(THETA/57.2958)
      WCONW=RHO*3.5*(((TUCN+TO(9))*.5+.005)*SEG*HCO/COS(THETA/
     &57.2958)+3.14159*(TUCN+.005)*BD)
      GO TO 93
C     ALTERNATE FWD SECTION WITHOUT FAIRING
C     FWD OXY BLKHD THICKNESS
   92 TFOB=PULLB*.5*ND/(2.*FTU)
      IF(TFOB.LT.TMIN)  TFOB=TMIN
C     UPR CONE WALL THICKNESS
      TUCN=PULLB*.5*ND/FTU
      IF(TUCN.LT.TMIN)  TUCN=TMIN
C     FWD OXY BLKHD RING AREA SIZED FOR ELASTIC STABILITY
      NFOR=ABS((PULLB/2.)*(.5*ND*SIN(30./57.2958)-.5*ND*TAN(
     &THETA/57.2958)))
      AFOR=(NFOR*(.5*ND)**3/(3.*E))**.5
      IF(AFOR.LT.AMIN) AFOR=AMIN
C     WEIGHT OF FWD OXY BLKHD RING AND WELD
      WFOR=3.14159*ND*RHO*AFOR
      WFOB=RHO*(TFOB+.005)*2.*3.14159*(.5*ND)**2
      WFOW=RHO*3.5*(TFOB+.005)*3.14159*ND
      WFAIR=0.
      WFAW=0.
C     WEIGHT OF FWD CONE OXY TANK WALL AND WELDS
      WCON=RHO*3.14159*.5*((TUCN+.005)*ND+(TO(9)+.005)*D)*HC/
     &COS(THETA/57.2958)
      WCONW=RHO*3.5*(((TUCN+TO(9))*.5+.005)*SEG*HC/COS(THETA/
     &57.2958)+3.14159*(TUCN+.005)*ND)
C     FWD CONE/CYL RING AREA SIZED FOR ELASTIC STABILITY
   93 NCCR=9HCT(9)*(.5*D)*TAN(THETA/57.2958)/2.
      ACCR=(NCCR*(.5*D)**3/(3.*E))**.5
      IF(ACCR.LT.AMIN) ACCR=AMIN
C     WEIGHT OF FWD CONE/CYL RING AND WELDS
      WCCR=3.14159*D*RHO*ACCR
      WCCW=3.14159*D*RHO*(TO(9)+.005)*3.5
C     WEIGHT OF MISC STR COMPONENTS(UMBILICAL PANELS,
C     TUNNEL,AND BAFFLES
      UMPANL=300.
      TUNNEL=200.*L/1694.
      BAFF=D*(HB+HCO)/343.
      IF(BLKHD.GT.2.) BAFF=D*(HB+HC)/343.
```

```
            IF(HO.GT.HH) BAFF=D*HF/343.
C           STRUCTURAL WEIGHT SUBTOTALS
            FWDBLF=WFOR+WFOB+WFOW
            CONSCT=WCON+WCONW+WCCR+WCCW
            CYLSCT=WBX+WLDOT+LWLDO
            AFTBLF=WABB+WABR+WABW
            FWDTK=FWDBLF+CONSCT+CYLSCT+AFTBLF
            FWDBLA=WFFB+WFFR+WFFW
            AFTCYL=WFX+WLDFT+LWLDF
            AFTBLA=WAFB+WAFR+WAFW
            AFTNK=FWDBLA+AFTCYL+AFTBLA
            NOSFAR=WFAIR+WFAW
C           SET D=D.D.,RE-DEFINE L/D
            D=D+2.*K
            LOD=L/D
C           CALCULATE INDUCED ENVIRN PROT WITH 10% NON OPT
C           INTER TANK TPS
            TPSIN=(2.*HR+LCON)*3.14159*D*INTPS/144.
C           NOSE FAIRING TPS
            TPSFA=3.14159*.5*(UCTPS*ND+LCTPS*BD)*(HC-HCB)/
           &(COS(THETA/57.2958)*144.)
C           NOSE CAP TPS
            TPSNC=NCTPS*3.14159*2.*NR*HN/144.
            IF(HO.GT.HH) GO TO 97
C           LOX FWD TPS
C           AFT DOME TPS
            TPSDM=DMTPS*2.*3.14159*R*HR/144.
C           AFT CYLINDER TPS
            TPSCY=3.14159*D*HF*CYTPS/144.
            TPSCN=0.
            TPSCJ=0.
            IF(BLKHD.EQ.1.) GO TO 94
            GO TO 95
C           AFT CYLINDER TPS FOR COMMON BLKHD CASE
         94 TPSCY=3.14159*D*HH*CYTPS/144.
            TPSIN=0.
         95 IF(BLKHD.GT.2.) GO TO 96
            GO TO 101
C           LOX AFT TPS
C           FWD CONE TPS
         96 TPSFA=0.
            TPSNC=0.
            GO TO 101
         97 TPSCN=3.14159*.5*(LCTPS*BD+CYTPS*D)*HCB/
           5(COS(THETA/57.2958)*144.)
C           FWD CYLINDER TPS
            TPSCJ=3.14159*CYTPS*D*HO/144.
            TPSDM=0.
            TPSCY=0.
            IF(BLKHD.EQ.1.) GO TO 98
            GO TO 99
         98 TPSIN=0.
         99 IF(BLKHD.GT.2.) GO TO 100
            GO TO 101
C           ALTERNATE FWD SECTION WITHOUT FAIRING
        100 TPSFA=0.
            TPSNC=0.
C           FWD CONE TPS
            TPSCN=3.14159*.5*(UCTPS*ND+LCTPS*D)*HC/
           5(COS(THETA/57.2958)*144.)
C           TPS WEIGHT SUBTOTALS WITH 10% NON-OPTIMUM
```

```
  101 FAIRT=TPSFA+TPSNC
      FCCTPS=TPSCN+TPSCJ
      ACYOM=TPSOM+TPSCY
      TOTPS=TPSIN+FAIRT+FCCTPS+ACYOM
C     INTERSTAGE ANALYSIS
C     LB=SPACE BETWEEN BOOSTER ATTACH POINTS
      LB=HH+LCON+HR
      IF(RX1.GT.0.0) R1=RX1
      IF(RX2.GT.0.0) R2=RX2
      IF(RXL.GT.0.0) RL=RXL
      IF(BLKHD.EQ.1.) LB=HH
C      BOOSTER INDUCED MOMENT AND REACTIONS
      IF(HO.GT.HH) LB=HO+HH
      M6=(THBSLT/NOENGB)*COS(BCANT/57.2958)*78.
     5-(THBSLT/NOENGB)*SIN(BCANT/57.2958)*27.
      R6=(M6/LB)*FS
      R5=((THBSLT/NOENGB)*SIN(BCANT/57.2958)-R6)*FS
      R8=((THBSLT/NOENGB)*COS(BCANT/57.2958)-BGLOW/NOENGB)*FS
C     GENERALIZED WT EQ FOR RING CAPS,WEBS AND BEAM
C     CAPS,WEBS(.94*FTU KNOCK DOWN FACTOR FOR COMP STAB
C     .3125*FTU FOR SHEAR ALLOWABLE)
      WIT=(RHO*1.10/(2.*FTU))*(3.14159*(D/.94+40./.3125)
     5+D**2/(40.*.94)+D/.3125)
C     WEIGHT  OF BOOSTER ATTACH RINGS-CHECK AGAINST BLKHD
C     RINGS  AND ADD INTERSTAGE BEEF UP REQMTS
      W6=WIT*NOENGB*R6*1.01
      W5=WIT*NOENGB*R5*.614
C     A DD SEPARATE ORBITER ATTACH RINGS BASED ON ORBITER
C     REACTION  LOADS FROM ORBITER MODULE
      W2=WIT*R2*.716
      W1=WIT*R1*2.055
C     ADD  SPECIAL INCREMENTS FOR SWAY BRACES DOUBLERS
C     D RAG LINKS AND FITTINGS ADD 10% STR NON OPT
      WSL=45.
      ASB=(R1*12.*(.5*D*.5774)**2/(E*3.14159**2))**.5
      WSB=ASB*.5*D*.5774*RHO*2.*1.34*1.05
      WSF=8.284*(R6*3./(.3125*FTU))**1.5*2.*RHO*1.10
      WFDUB=150.*.5*3.14159*D*RHO*.071*1.10
      WDRAG=.5*100.*RHO*1.10*(NOENGB*R8+RL)/(.94*FTU)
      WADUB=100.*.5*3.14159*D*RHO*.055*1.10
      WOLL=R2*39.*2.*RHO*1.34*1.10/FTU
      WSLS=35.
      WSLF=55.
      WBLL=R5*1.5*78.*4.*RHO*1.34*1.10/FTU
C     TOTAL INTERSTAGE WEIGHT
      TWINT=W6+W5+W2+W1+WSL+WSB+WSF+WFDUB+WDRAG+WADUB
     5+WOLL+WSLS+WSLF+WBLL
C     TOTAL STRUCTURE WEIGHT
      BODSRP=FWDTK+WINT+AFTNK+NOSFAR+TWINT+UMBPNL+TUNNEL+BAFF
C     PROPELLANT SYSTEMS-BASED ON DETAIL POINT DESIGN WITH
C     SCALLING LAWS FOR ORBITER ENGINE FLOW RATE,MIXTURE
C     RATIO AND TANK DIMENSIONS
      IF(BLKHD.GT.1.0) GO TO 103
C     COMMON BLKHD INTERNAL LOX LINE
      OXVENT=564.
      HVENT=427.
      PNEU=25.5+30.5*L/1694.
      IF(HO.GT.HH) GO TO 102
C     LOX FORWARD
      OXPRES=122.+109.*L/1694.
      OXFEED=(19.5+.01222*HF)*OFLOWY**.5
```

```
      HPRES=61.+83.*HF/898.
      HFEED=(31.59+.03176*D)*OFLOWX**.5
      RX=.5*(7.+.33*OFLOWY**.5)
      GO TO 105
C     LOX AFT
  102 OXPRES=122.+109.*HF/1694.
      OXFEED=(19.5+.01222*.5*D)*OFLOWX**.5
      HPRES=61.+83.*L/898.
      HFEED=(31.59+.03176*HF)*OFLOWY**.5
      RX=.5*(7.+.76*OFLOWY**.5)
      GO TO 105
C     SEPARATE BLKHDS EXTERNAL LOX LINES
  103 PNEU=25.5+18.5*L/1470.
      WJCKT=0.
      RX=0.
      OXVENT=522.
      HCIR=68.+100.*D/314.
      HVENT=622.
      IF(HO.GT.HH) GO TO 104
C     LOX FORWARD
      OXPRES=128.+132.*L/1470.
      OXFEED=(18.1+.01465*(HH+LCON))*OFLOWY**.5
      HPRES=69.+115.*HH/703.
      HFEED=(34.21+.01333*D)*OFLOWX**.5
      GO TO 106
C     LOX AFT
  104 OXPRES=128.+132.*HH/1470.
      OXFEED=(18.1+.01465*.5*D)*OFLOWX**.5
      HPRES=69.+115.*L/703.
      HFEED=(34.21+.01333*(HH+LCON))*OFLOWY**.5
      GO TO 106
C     COMMON BLKHD INTERNAL LOX LINE JACKET UNDER EXTERNAL PRES
  105 TTOP=(FHCT(9)*10.*RX**1.5/(E*.88157))**.4
      IF(TTOP.LT.TMIN)  TTOP=TMIN
      TBOT=(FHCT(1)*10.*RX**1.5/(E*.88157))**.4
      IF(TBOT.LT.TMIN)  TBOT=TMIN
      WJACK=(.5*(TTOP+TBOT)+.005)*3.14159*2.*RX*1.25
     &*RHO*(HF-100.)
      TUPR=(FHCT(9)*10.*(RX+3.)**1.5/(E*.88157))**.4
      IF(TUPR.LT.TMIN)  TUPR=TMIN
      TLWR=(FHCT(1)*10.*(RX+3.)**1.5/(E*.88157))**.4
      IF(TLWR.LT.TMIN)  TLWR=TMIN
      WEND=(TUPR+TLWR+.01)*50.*3.14159*(RX+3.)*2.
     &*1.25*RHO
      WJCKT=1.10*(WJACK+WEND)
      HCIR=0.
C     MISC PROP SYS COMPONENTS;SUMP,PUSYS
  106 SUMP=220.
      PUSYS=185.
C     PROPELLANT SYSTEMS SUBTOTALS
      FEDSYS=OXFEED+HFEED+WJCKT+HCIR
      PRSVNT=HPRES+HVENT+OXPRES+OXVENT
      PNPU=PNEU+PUSYS
      PROSY=FEDSYS+PRSVNT+PNPU+SUMP
C     ESTIMATED TANK INERT WT FOR RETRO SYS CALCULATION
      XDRY=(BODGRP+TOTPS+PROSY+AVIONT+MISCT)*(1.+GJP)
      XINERT=XDRY+RESIDT
C     RETRO ROCKET SIZING
      DELX=EXP(RETDV/(32.17*RETISP))-1.
      WRPROP=XINERT*DELX/(1.-(1.+GJP)*.365*DELX)
      WRETRO=1.365*WRPROP
```

```
C       TANK SUBTOTAL DRY WEIGHT
        SUBDRY=BODGRP+TOTPS+PROSY+AVIONT+MISCT+WRETRO
C       GROWTH/UNCERTAINTY AS PERCENT OR DELTA TO A FIXED
C       DRY WEIGHT
        GU=SUBDRY*GUP
        IF(FIXDWT.GT.0.) GU=FIXDWT-SUBDRY
C       TANK TOTAL DRY WEIGHT
        DRYWT=SUBDRY+GU
C       TANK TOTAL INERT WEIGHT
        INERT=DRYWT+RESIDT
C       EXTERNAL TANK GROSS WEIGHT
        GROSSW=INERT+PROPOT
C       EXTERNAL TANK MASS FRACTION
        TLAMB=PROPOT/GROSSW
        EXTL=L
        EXTD=D
        EXTHO=HO
        EXTHH=HH
        GO TO 700
C
C        FIXED WEIGHT EXTERNAL TANK CALCULATIONS
C
  400   DRYWT=DRYWT
        RESIDT=RESIDT
        INERT=DRYWT+RESIDT
        GROSSW=INERT+PROPOT
        TLAMB=PROPOT/GROSSW
        GO TO 700
C
C        EXTERNAL TANK SIMPLIFIED EQUATION CALCULATIONS
C
  500   RATPRO=PROPO2/PROPOT
        IF(RATPRO.LT..50) RATPRO=.50
        CFB=213.777+10.690*RATPRO
        CFA=-31957.6-2444.00*RATPRO
        CFC=.0478364+.0214184*RATPRO
        CFD=-.000104847-.00001502*RATPRO
        CFF=CFC+CFD*DF
        CFE=CFA+CFB*DF
        DRYWT=CFE+CFF*PROPOT
        IF(PROPOT.LT.1000000.) DRYWT=DRYWT*(1.11520-.0000001452*PROPOT)
        IF(PROPOT.GT.1000000.) DRYWT=DRYWT*(.908317+.0000000567*PROPOT)
        DRYWT=DRYWT/1.075
        GU=GUP*DRYWT
        DRYWT=DRYWT+GU
        RESIDT=2206.+.001972*PROPOT
        INERT=DRYWT+RESIDT
        GROSSW=INERT+PROPOT
        TLAMB=PROPOT/GROSSW
        GO TO 700
C
C        FIXED LAMBDA EXTERNAL TANK CALCULATIONS
C
  600   DRYWT=PROPOT*((1.0-.954)/.954)
        RESIDT=2206.+.001972*PROPOT
        INERT=DRYWT+RESIDT
        GROSSW=INERT+PROPOT
        TLAMB=PROPOT/GROSSW
  700   CONTINUE
        RETURN
        END
```

5. SOLID ROCKET MOTOR BOOSTER MODULE

This module contains the analytical and empirical weight estimation relationships necessary to completely define the solid rocket motor (SRM) booster system. The NASA weight report and design data, coupled with a three-view drawing of the SRM, supplies all inputs necessary to analyze the configuration. Here again, it is important to note that the velocity correlation coefficients described in Option 1 must be calculated or known before this option can be executed. The primary purpose of this option is to provide the capability of optimizing the SRM by inputting a diameter and iterating on propellant load and engine characteristics. The iteration calculates SRM burnout weight and dry weight, which, in turn, modifies retro and parachute system weights, which, in turn, is rippled through the other weights. This entire module continues the iteration until a completely balanced system exists. This option also has the capability of inputting constants, which allows the user to input weight changes without modifying the program. The SRM option contains four distinct modes of operation which are as follows:

1. iterative analysis
2. iterative analysis (fixed dry weight)
3. simplified equation
4. fixed booster.

Figure 5-1 is the flow diagram of the SRM Module, followed by a detail listing of the program.

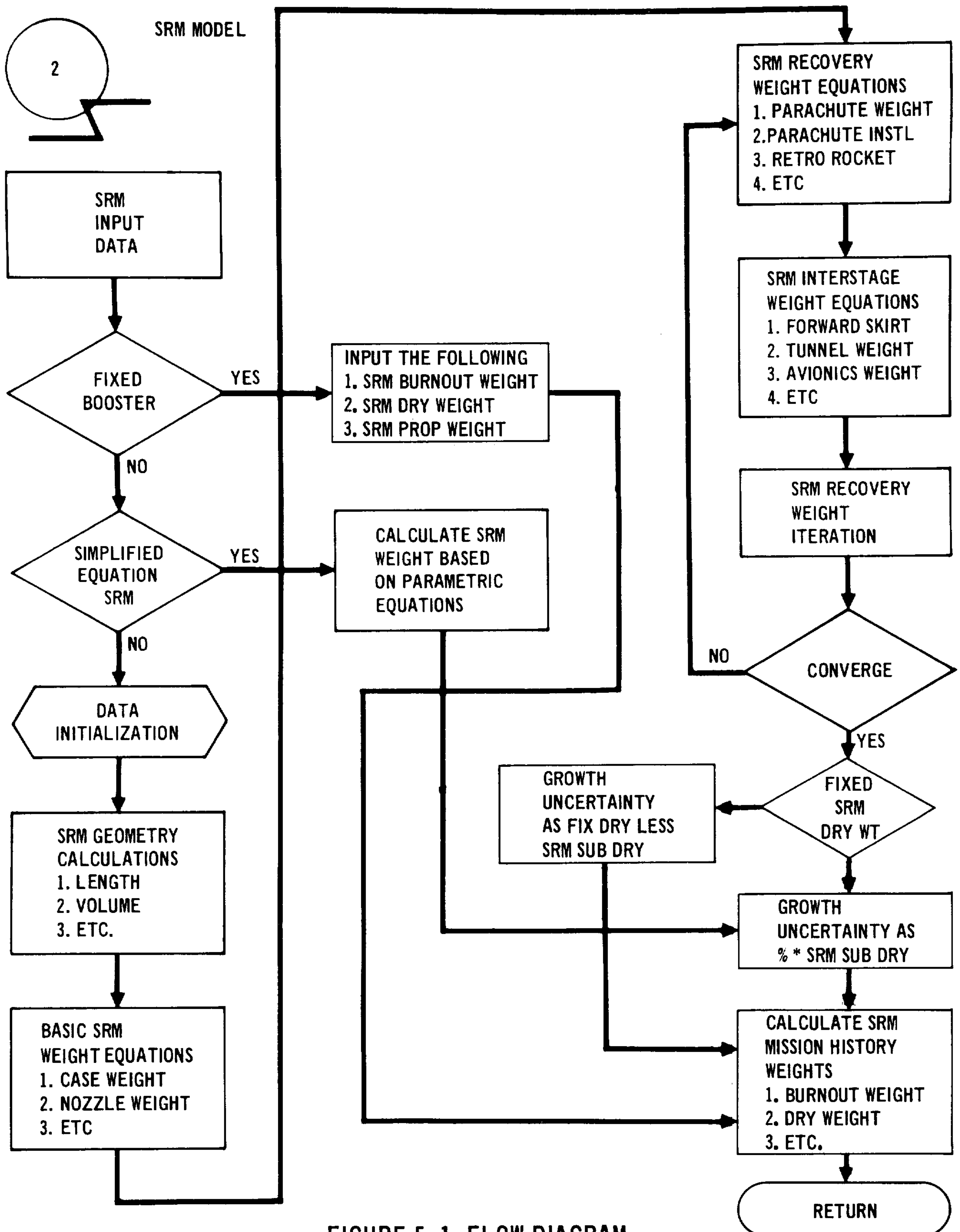

FIGURE 5–1 FLOW DIAGRAM

FIGURE 5–1

```
COPY RSUB TO LP(K,NC)
@FIXED
C
C                             SRM SUBROUTINE
C
      SUBROUTINE SRMMODEL
C     THE PRIMARY PURPOSE OF THIS PROGRAM IS TO PROVIDE THE
C     CAPABILITY OF OPTIMIZING THE SRM BY INPUTTING A
C     DIAMETER AND ITERATING ON PROPELLANT LOAD AND ENGINE
C     CHARACTERISTICS.THE ITERATION CALCULATES SRM BURNOUT
C     WEIGHT AND DRY WEIGHT,WHICH, IN TURN,MODIFIES RETRO
C     AND PARACHUTE SYSTEM WEIGHTS,WHICH,IN TURN, IS RIPPLIED
C     THROUGH THE OTHER WEIGHTS.THIS OPTION ALSO HAS THE
C     CAPABILITY OF INPUTTING CONSTANTS,WHICH ALLOWS THE
C     USER TO INPUT WEIGHT CHANGES WITHOUT MODIFYING THE
C     PROGRAM.
      IMPLICIT REAL(A-Z)
      DIMENSION FID(4)
C
C     PERFORMANCE COMMON BLOCK
C
      COMMON/MAIN/ PROPB,PROPO,FBT,BCANT,OCANT,OCANTY
     1,OCANTP,NOENGB,NOENGO,THBSL,THOSL,THOV,TOV,FLOWR
     2,TF,FTW,FIXHRD
     3,ISPBS,ISPBV,ISPOBS,ISPOBV,SCD,BTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THBTC,THRSLT,THOTC,ISPO,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVONC,DVB,DVTOTC,WOSCD,OINWT,OLANWT
     7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
C
C     SRM OUTPUT COMMON BLOCK
C
      COMMON/SRMO/BBOWT,BDRYWT,BGLOW,LAMB,PWOI
     1,BASSRM,WCASE,WJOINT,WNOZZ,WTTER,WINST,WIGN,BSRMC
     2,SRMISS,PWFS,PWASLS,PWAS,PWNF,PWTN,PWAV,WNCTPS,SRMIC
     3,WRECOV,PWPAR,PWPI,PWRR,RWRP,PWWR,SRMRC
     4,UNCERT,EXPINS,BOSLUN,SRML,SRMD
     5,PGROSS,PBBOWT,PPROPB,PDRYWT,PBOSLU
     6,FIXBOO,SIMPBO
C
C     SRM (SRM) INPUT DATA BLOCK
C
      NAMELIST RHOP,DIA,MIAX,MAAX,MEOP,FS,FTU,AT,RHOM
     1,INT,NP,NJ,WNOZ,NER,AP,CF,PC,NDHA,TC,TDES,VRI
     2,VSD,AADE,LF,RUNC1,FIXDWT,RRISP,WEI,BSRMC,SRMIC,SRMRC
     3,FIXBOO,BBOWT,BDRYWT,SIMPBO
C
      IF(P.EQ.2.0) GO TO 2
      INPUT(1)
    2 IF(FIXBOO.GT.0.0) GO TO 60
      IF(SIMPBO.GT.0.0) GO TO 70
      WRECOV=10000.
      SRMISS=10000.
C
C     BASIC SRM WEIGHT
C
C     GEOMETRY EQUATIONS
      PWOI=PROPB/(1.005*NOENGB)
```

```
      THBSLS=THBSLT/NOENGB
      VPC=(PWOI/RHOP)-(.292674*DIA*DIA*DIA*(MIAX/MAAX))
      TCYL=(MEOP*DIA*FS/(2.0*FTU))+INT
      ACS=(3.14159/4.0)*(DIA-TCYL)*(DIA-TCYL)
      LTH=VPC/(ACS-AP)
      VCASE=(3.14159/6.0)*DIA*DIA*DIA*(MIAX/MAAX)
     1+(3.14159/4.0)*DIA*DIA*(VPC/(ACS-1.3*AT))
      PLE=PWOI/(RHOP*VCASE)
C     CASE WEIGHT
      CASE1=RHOM*(PWOI/(RHOP*NP))**.95
      CASE2=MEOP**.7*FS**.7/FTU**.9
      CASE3=(LTH/DIA)**((.163*MIAX/MAAX)-.114)
      CASE4=(MIAX/MAAX)**.315
      WCASE=11.16*(CASE1*CASE2*(CASE3/CASE4))**1.013
C     CASE JOINT WEIGHT
      WJOINT=7.7*DIA*DIA*(MEOP*FS*NJ/FTU)
C     NOZZLE WEIGHT
C     FIXED NOZZLE
      NOZZ1=RBT**.6*THBSLS**1.2*NER**.7
      NOZZ2=CF**1.2*PC**.8*(TAN(NDHA/57.29578))**.4
      IF(WNOZ.GT.0.0) GO TO 10
      WNOZZ=.003505*(NOZZ1/NOZZ2)**.916
      GO TO 20
C     GIMBALLED NOZZLE
   10 KGN=2.11/NER**.115
      WNOZZ=.003505*KGN*(NOZZ1/NOZZ2)**.916
C     THRUST TERMINATION WEIGHT
   20 WTTER=.03518*(THBSLS/(CF*PC))**1.45
C     INSULATION WEIGHT
C     CASE
      INSC1=(PWOI/(RHOP*NP))**.80
      INSC2=RBT**.5*PC**.117*(TC/1000.)*(TC/1000.)
      INSC3=(LTH/DIA)**.1*TDES**.2
      WINSC=.000602*(INSC1*(INSC2/INSC3))**.86
C     JOINT
      DP=((AP/AT)*(4.0*THBSLS/(3.14159*CF*PC)))**.5
      WUI=.00054*(DIA/DP)*DIA*RBT
      WII=.000178*(DIA-DP)*((2.*DIA+DP)*RBT+80.*DIA)
      IF(WEI.GT.0.0) GO TO 33
      WINSJ=2.0*WUI*NJ
      GO TO 35
   33 WINSJ=(WUI+WII)*NJ
   35 WINST=WINSC+WINSJ
C     IGNITER WEIGHT
      WIGN=.23658*DIA**1.45
      EXPINS=-.005*PWOI
      BASSRM=WCASE+WJOINT+WNOZZ+WTTER+WINST+WIGN+BSRMC
      PROSLU=BASSRM+SRMISS+WRECOV
      UNCERT=PROSLU*RUNC1
      IF(FIXDWT.GT.0.0) UNCERT=FIXDWT-PROSLU
      BBOWT=BASSRM+SRMISS+WRECOV+UNCERT+EXPINS
      BLOWT=BBOWT+PROPB-EXPINS
   30 TBBOWT=BBOWT
C
C     RECOVERY SYSTEM WEIGHT
C
C     PARACHUTE WEIGHT
      WPAR=175.0+498.0*(BBOWT/(VRI*VRI))
      PWPAR=WPAR
C     PARACHUTE INST WEIGHT
      WPI=.6916*WPAR
```

```
      PWPI=WPI
C     RETRO ROCKET WEIGHT
      WRR=.0819*BBOWT*(VRI-VSO)/RRISP
      PWRR=WRR
C     PROPELLANT WEIGHT
      WRP=.675*WRR
      PWRP=WRP
C     WATER RECOVERY HDW WEIGHT
      WWR=350.
      PWWR=WWR
C
C     SRM BODY ADAPTER WEIGHT
C
C     FORWARD SKIRT
      WFS=13.65*DIA
      PWFS=WFS
C     AFT SKIRT/LAUNCH STRUCTURE WEIGHT
      ASLS1=(BGLOW+(2.0*BLOWT))/4.0
      ASLS2=.65*THOSL*COS(AAOE/57.292578)*(LF+(DIA/2.))
      WASLS=.00464*(ASLS1+(ASLS2/DIA))
      PWASLS=WASLS
C     ATTACH SEP/STRUCTURE WEIGHT
      WAS=352.0*THOSLS/BBOWT
      PWAS=WAS
C     NOSE FAIRING
      WNF=.0607*DIA*DIA
      PWNF=WNF
C     TUNNEL WEIGHT
      WTN=.114*LTH
      PWTN=WTN
C     AVIONICS WEIGHT
      WAV=182.0
      PWAV=WAV
C     TPS WEIGHT
      WNCTPS=.018*DIA*DIA
      SRMISS=WFS+WASLS+WAS+WNF+WTN+WAV+WNCTPS+SRMIC
      WRECOV=WPAR+WPI+WRR+WRP+WWR+SRMRC
      PBOSLU=BASSRM+SRMISS+WRECOV
      UNCERT=PBOSLU*BUNC1
      IF(FIXOWT.GT.0.0) UNCERT=FIXOWT-PBOSLU
      BBOWT=BASSRM+SRMISS+WRECOV+UNCERT+EXPINS
      BLOWT=BBOWT+PROPB-EXPINS
      BGLOW=BLOWT
      IF(ABS(TBBOWT-BBOWT).LE.10.) GO TO 50
      GO TO 30
   50 PBBOWT=BBOWT
      PPROPB=PROPB/NOENGB
      PDRYWT=PBBOWT-EXPINS
      PGROSS=PBBOWT+PPROPB
      BBOWT=BBOWT*NOENGB
      BOSLUN=BBOWT-(UNCERT*NOENGB)
      BDRYWT=BBOWT-(EXPINS*NOENGB)
      BGLOW=BBOWT+PROPB
      PWBI=PWOI*NOENGB
      LAMB=PROPB/BGLOW
      SRML=LTH
      SRMD=DIA
      GO TO 80
C
C     FIXED BOOSTER CALCULATIONS
C
```

```
   60 BBOWT=BBOWT
      BDRYWT=BDRYWT
      BGLOW=PROPB+BBOWT
      PWOI=PROPB/1.005
      LAMB=PROPB/BGLOW
      GO TO 80
C
C     SIMPLIFIED BOOSTER EQUATIONS
C
   70 PWOI=PROPB/(1.005*NOENGB)
      THBSLS=THBSLT/NOENGB
      EXPINS=-.005*PWOI
      COEFSE=29653.3+(PWOI*.0773338)
      BBOWT=COEFSE+(THBSLS*.01887863)
      BBOWT=BBOWT/1.035
      UNCERT=BBOWT*BUNC1
      BBOWT=BBOWT+UNCERT
      BBOWT=BBOWT*NOENGB
      BDRYWT=BBOWT-(EXPINS*NOENGB)
      BGLOW=BBOWT+PROPB
      PWOI=PWOI*NOENGB
      LAMB=PROPB/BGLOW
   80  CONTINUE
      RETURN
      END
```

6. TRAJECTORY MODULE

The trajectory module contains the curve fits of an optimized trajectory, established by MDAC during the Phase B Shuttle program. These curve fit equations determine the total required velocity by defining the velocity losses. This is accomplished in several distinct steps:

a. The ideal required velocity is determined as a function of first stage velocity, first and second stage thrust to weight ratios and the ascent drag parameter.
b. The velocity losses attributable to the launch site altitude and the required mission inclination.
c. A delta velocity correction factor which allows the curve fit equations to translate through the defined losses of an analyzed point design.

The equations are empirical relationships derived from parametric ascent trajectory shaping studies, and are intended to be used for ideal staging velocities in the range of 8,000 to 12,000 ft/sec.

Ascent losses have been shown to be a strong function of thrust/weight at lift-off (T/W_1), and thrust/weight immediately after staging, (T/W_2). Other significant correlation factors in the velocity loss equation are staging velocity (V_S), first stage drag parameter (W/SC_D), and launch site altitude (H). All of the above parameters are self-explanatory with the exception of SC_D. The SC_D value used is between Mach numbers 1.2 and 1.5.

The velocity losses were curve fit for ease of interpolation when used for sizing studies. The coefficients of the multivariate, polynominal fit were evaluated by a least-squares technique. Each coefficient of the initial polynominal was tested for significance, and the least significant term was eliminated. This procedure was repeated until a minimum term polynominal was determined which had accuracy essentially equal to the original. The accuracy of the curve fit was then improved by conditioning the independent variables with natural logarithmic functions. However, if new data is curve fit, other functions may be more appropriate.

The curve fits are predicated on limiting values of the thrust to weight ratios and the accent drag coefficient. These limits are:

a. First stage thrust to weight is less than 1.60 or greater than 1.18.
b. Second stage thrust to weight is less than 2.0 or greater than 0.70.
c. First stage drag parameter is less than 12,000 or greater than 1,000.

If these limits are exceeded, the program selects the applicable limiting parameter and outputs a warning that the results are outside the bounds of the curve fit equation and the validity is questionable.

Figure 6-1 is a flow diagram of the Trajectory Module, followed by a detail listing of the program.

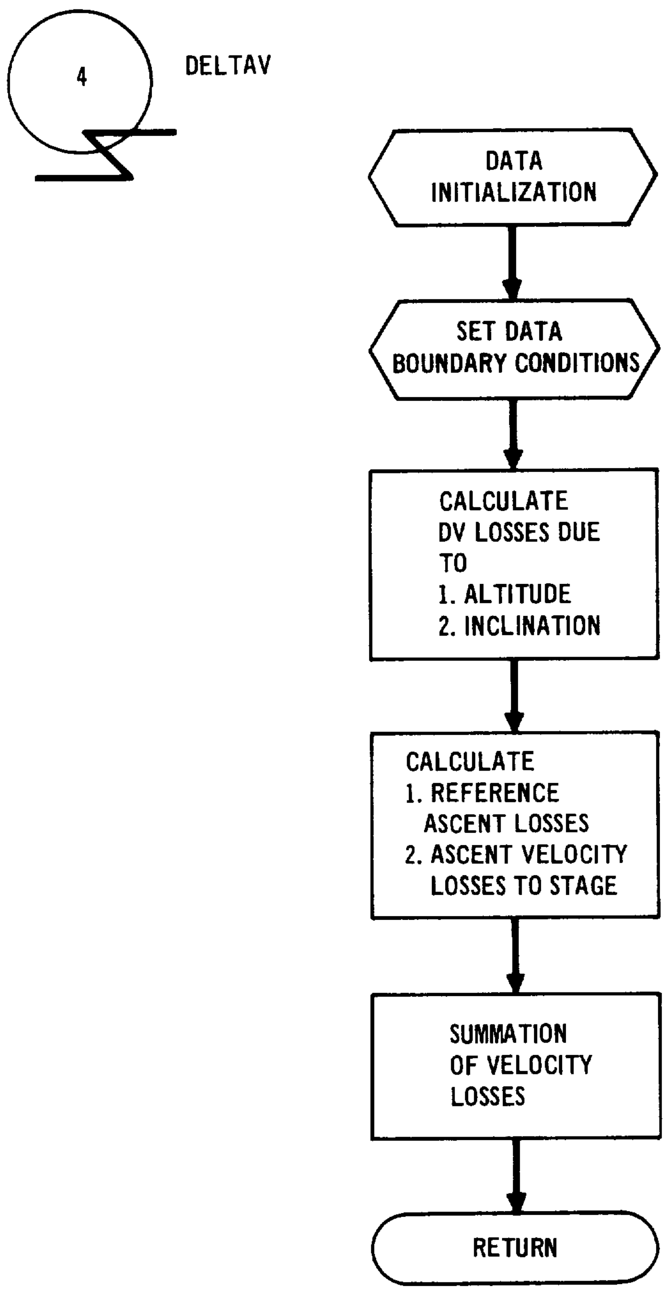

FIGURE 6-1 FLOW DIAGRAM

```
COPY DVSUB TO LP(K,NC)
$FIXED
C
C                            DELTAV SUBROUTINE
C
      SUBROUTINE DELTAV
C     THIS PROGRAM CONTAINS THE CURVE FITS OF AN OPTIMIZED
C     TRAJECTORY,ESTABLISHED BY MDAC DURING THE PHASE B SHUTTLE
C     PROGRAM.THESE CURVE FIT EQUATIONS DETERMINE THE TOTAL
C     REQUIRED VELOCITY BY DEFINING THE VELOCITY LOSSES.
C     THIS IS ACCOMPLISHED IN SEVERAL DISTINCT STEPS:
C       1.THE IDEAL REQUIRED VELOCITY IS DETERMINED AS A
C         FUNCTION OF FIRST STAGE VELOCITY,FIRST AND SECOND
C         STAGE THRUST TO WEIGHT RATIOS AND THE ASCENT DRAG
C         PARAMETERS.
C       2.THE VELOCITY LOSSES ATTRIBUTABLE TO THE LAUNCH SITE
C         ALTITUDE AND THE REQUIRED MISSION INCLINATION.
C       3.A DELTA VELOCITY CORRECTION FACTOR WHICH ALLOWS THE
C         CURVE FIT EQUATIONS TO TRANSLATE THROUGH THE DEFINED
C         LOSSES OF AN ANALYZED POINT DESIGN.
C     THE EQUATIONS ARE EMPIRICAL RELATIONSHIPS DERIVED FROM
C     PARAMETRIC ASCENT TRAJECTORY SHAPING STUDIES,AND ARE
C     INTENDED TO BE USED FOR IDEAL STAGING VELOCITIES IN THE
C     RANGE OF 8,000. TO 12,000. FT/SEC.
      IMPLICIT REAL(A-Z)
C
C      PERFORMANCE COMMON BLOCK
C
      COMMON/MAIN/ PROPB,PROPO,OBT,OCANT,OCANT,OCANTY
     1,OCANTP,NOENGB,NOENGO,THBSL,THOSL,THOV,TOV,FLOWR
     2,TF,FIW,FIXHRD
     3,ISPBS,ISPBV,ISPOBS,ISPOBV,SCD,BTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THBTC,THBSLT,THOTC,ISPO,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVONC,DVB,DVTOTC,WOSCD,OINWT,OLANWT
     7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
C
C     DELTAV OUTPUT COMMON BLOCK
C
      COMMON/DVO/DVT,DVON,DVR,DVBR,DVBRP,X2,X3
     1,DVFPR,TOTLSS,DVONR,DVALT
C
C     THE CURVE FITS ARE PREDICATED ON LIMITING VALUES OF THE
C     THRUST TO WEIGHT RATIOS AND ASCENT DRAG COEFFICIENT.
C     THESE LIMITS ARE AS FOLLOWS:
C        1.FIRST STAGE THRUST TO WEIGHT IS LESS THAN 1.60 OR
C          GREATER THAN 1.18.
C        2.SECOND STAGE THRUST TO WEIGHT IS LESS THAN 2.0 OR
C          GREATER THAN 0.70.
C        3.FIRST STAGE DRAG PARAMETER IS LESS THAN 12,000. OR
C          GREATER THAN 1,000.
      IF(FW(1).LE.1.18) FW(1)=1.18
      IF(FW(1).GE.1.60) FW(1)=1.60
      IF(FW(2).LE..70) FW(2)=.70
      IF(FW(2).GE.2.0) FW(2)=2.0
      IF(WOSCD.LE.1000.) WOSCD=1000.
      IF(WOSCD.GE.12000.) WOSCD=12000.
C     ALTITUDE VELOCITY LOSS CALCULATIONS
```

```
      DVALT=(.327+6.723E-2*H-3.6191E-6*H*H+2.2222E-10*H*
     5H*H)*(-1.0)
      IF(H.EQ.0.0) DVALT=0.0
      DVCORR=DVCORR
C     MISSION INCLINATION VELOCITY LOSS CALCULATIONS
      IF(INC.GT.90.0) GO TO 5
      X1=24279.5+5.160*INC+0.14668*INC*INC
      GO TO 6
    5 X1=21903.2+57.9*INC-0.14668*INC*INC
    6 X2=-43.596-0.3961*INC+0.00977*INC*INC
      IF(INC.EQ.90.0) X2=0.0
      X3A=ALOG(DVB)
      X3B=ALOG(FW(1))
      X3C=ALOG(FW(2))
      X3D=ALOG(WOSCD)
C     REFERENCE ASCENT VELOCITY LOSSES UP TO STAGING
      X3=20626.0-4240.8*X3B-2631.0*X3C-347.4*X3A*X3A
     5+5358.7*X3C*X3C+28.5*X3A*X3A*X3A-16228D.0*X3B*X3B*X3B
     5-6548.7*X3C*X3C*X3C-705.9*X3A*X3A*X3B+18842.0
     5*X3B*X3B*X3A+14088.47-2930.7156*X3D+148.6349*X3D*X3D
      DVR=X1+X2+X3+DVALT+DVCORR
      TOTLSS=X3+DVCORR
      DVT=DVR*(1.+DVCON)
      DVBN=DVT-DVB
      DVFPR=DVT-DVR
C     ASCENT VELOCITY LOSSES UP TO STAGING
    7 X4=DVCNST-1152757.0*X3A+518528.7*X3B+131715.3
     5*X3A*X3A-284199.2*X3B*X3B-91690.6*X3A*X3B-631.23
     5*X3A*X3D-4343.76*X3A*X3A*X3A+3725.35*X3A*X3A*X3B
     5+16.96*X3A*X3A*X3D+31700.56*X3B*X3B*X3A+26.20*X3D
     5*X3D*X3A+47.44-566.27*X3C+521.99*X3C*X3C
     5-5729.4+1486.12*X3D-95.874*X3D*X3D
      DVBR=X4/1000.
      DVBNR=DVBN
      DVBRP=DVB-X4
      RETURN
      END
```

7. FIXED HARDWARE MODULE

This optional performance module determines the payload capability for a given configuration and physical characteristics through the use of the rocket equation (ΔV = gISP ln (MASS FRACTION)). In this option, the configuration weights and propellant loadings, as well as the ascent engine characteristics, are input into ESPER. To complete the analysis, the configuration velocity losses ($\Delta V_{CONFIG} = \Delta V_{ORB} + \Delta V_{SRM} + \Delta V_{LOSSES}$) are determined from an empirical relationship derived from parametric-ascent trajectory shaping studies. These velocity losses were curve fit for ease of interpolation when running ESPER. When the configuration velocity correlation constaints needed for the interpolation routine (DVCORR and DVCNST) are known, the velocity losses are calculated by ESPER. This option gives the user an invaluable tool by which the user can measure the impact on payload due to changes to a FIXED HARDWARE DESIGN. (For example, what is the change to the deliverable payload due to a 1-sec increase in orbiter ascent ISP.) Generally, however, these velocity correlation constants (DVCORR and DVCNST) are not known, but the staging velocity and the total losses are usually readily available. A matching routine, based on a simplified newtonian iteration technique, is provided in the FIXED HARDWARE OPTION that will internally modify the existing velocity loss curve fits. This routine solves for the correct DVCORR and DVCNST that will satisfy the total losses and staging velocity constraints. Since DVCORR and DVCNST are required input parameters, this solution serves a dual purpose for the user, it not only allows ESPER to compute the velocity losses, but it also opens the door to the other options offerred by ESPER.

Figure 7-1 is a flow diagram of the Fixed Hardware option followed by a detail listing of the program.

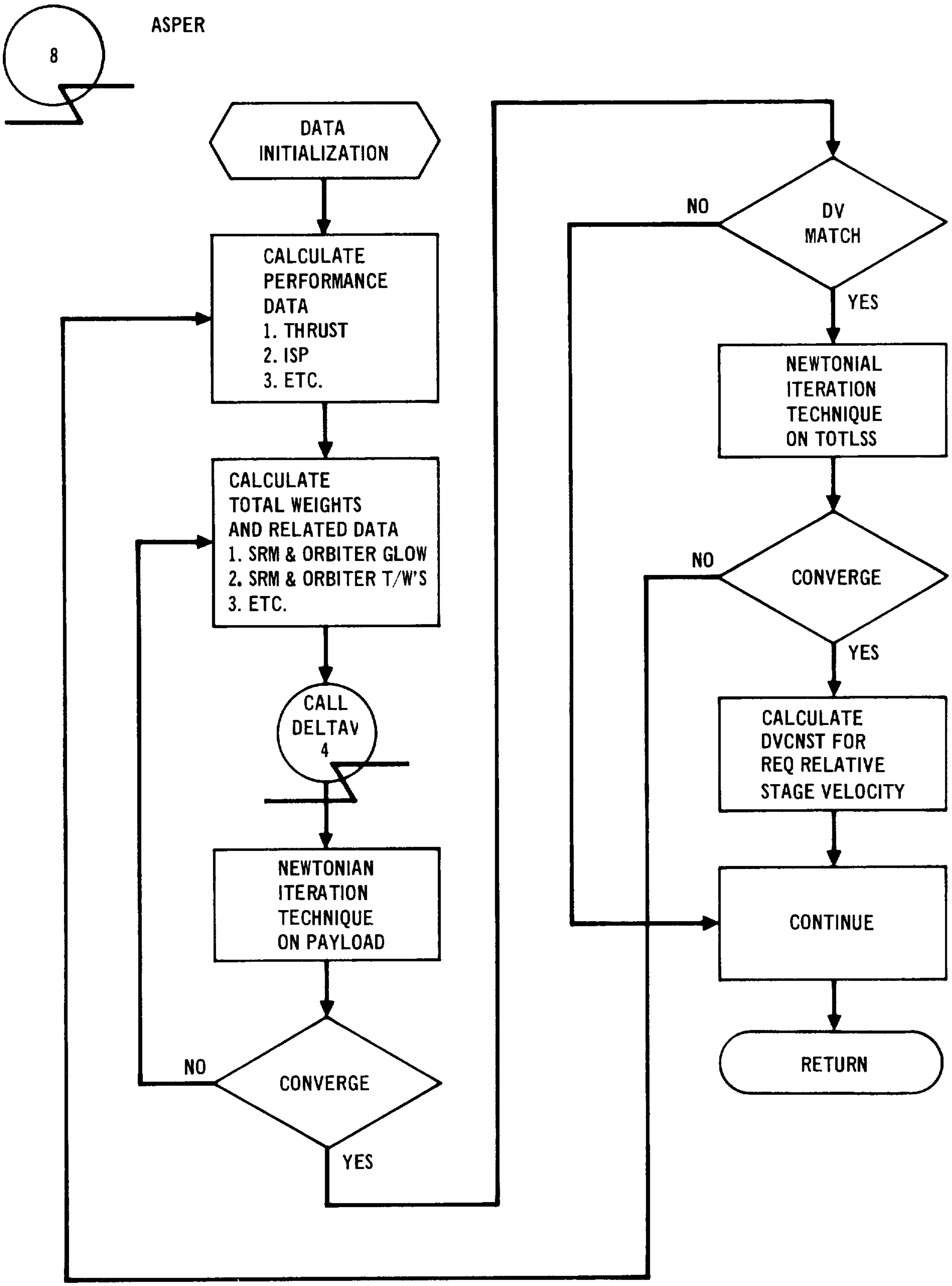

FIGURE 7–1 FLOW DIAGRAM

```
CØPY FHSUB TØ LP(K,NC)
      ØFIXED
C
C                           FIXED HARDWARE SUBRØUTINE
C
      SUBRØUTINE ASPER
C     THIS PRØGRAM DETERMINES THE PAYLØAD CAPABILITY FØR A
C     GIVEN CØNFIGURATIØN AND PHYSICAL CHARACTERISTICS
C     THRØUGH THE USE ØF THE RØCKET EQUATIØN.IN THIS ØPTIØN
C     THE CØNFIGURATIØN WEIGHTS AND PRØPELLANT LØADINGS,AS
C     WELL AS THE ASCENT ENGINE CHARACTERISTICS,ARE INPUT
C     INTØ ESPER.TØ CØMPLETE THE ANALYSIS,THE CØNFIGURATIØN
C     VELØCITY LØSSES ARE DETERMINED FRØM AN EMPIRICAL
C     RELATIØNSHIP DERIVED FRØM PARAMETRIC-ASCENT TRAJECTØRY
C     STUDIES.
      DIMENSIØN  RAT(2),PAY(2),DVM(2),RATM(2)
      IMPLICIT REAL(A-Z)
C
C     PERFØRMANCE CØMMØN BLØCK
C
      CØMMØN/MAIN/PRØPB,PRØPØ,ØBT,ØCANT,ØCANT,ØCANTY
     1,ØCANTP,NØENGØ,NØENGØ,THØSL,THØSL,THØV,TØV,FLØWR
     2,TF,FTW,FIXHRD
     3,ISPØS,ISPØV,ISPØØS,ISPØØV,SCD,ØTW
     4,H,DVCØRR,INC,STAGV,DVCØN,DVCNST
     5,PEL,THØTC,THØSLT,THØTC,ISPØ,ISPØ,PRØPØT
     6,PRØPØ1,PRØPØ2,FW(2),DVØNC,DVØ,DVTØTC,WØSCD,ØINWT,ØLANWT
     7,ØINJWT,ØLØWT,ØGLØW,GLØW,TØTAL,S,P,MATCH,TLSSR,FPRP
     8,ØHØLD,ØLLPLØ,ØLØWLØ,ØMSISP,ØMSDVT,ØMSDVP,ØMR
     9,LØNGP,TØWØ,TØWØ,SENS,GRØW,MINGLW
C
C     SRM ØUTPUT CØMMØN BLØCK
C
      CØMMØN/SRMØ/ØRØWT,ØDRYWT,ØØLØW,LAMØ,PWØI
     1,MASSRM,WCASE,WJØINT,WNØZZ,WTTER,WINST,WIGN,ØSRMC
     2,SRMISS,PWFS,PWASLS,PWAS,PWNF,PWTN,PWAV,WNCTPS,SRMIC
     3,WRECØV,PWPAR,PWPI,PWRR,PWRP,PWWR,SRMRC
     4,UNCERT,EXPINS,ØØSLUN,SRML,SRMD
     5,PGRØSS,PWØØWT,PPRØPØ,PDRYWT,PØØSLJ
     6,FIXØØØ,SIMPØØ
C
C     ØRØITER ØUTPUT CØMMØN BLØCK
C
      CØMMØN/ØRØØ/R1,R2,RL,WTAJX,WTACS,ACSENG,ACSSYS,WTACTK
     1,ACSMØD,WTØMS,ØMSENG,PRØPSY,WTØMTK,MØDULE
     1,SURFC,PPWP,ELEC,HYDR,AVIØNØ,ECLSØ,PPRØV,ØUNCWT
     2,ØPØMIS,TAØPRØ,SURFK
     1,PERSØN,ØPESD,ØRESV,PLØADU,PLØADD,ACSPRØ,WØPRØP,SUDLE
     2,FIXØRØ,FIXWØR,ØDRYWT,ØRIFL
     1,WWT,WSG,WØSTR,WTØRØE,WTØRØC,LEW,WTE,WAIL,WAS,WADR,WAH
     2,WAP,SPRØV,PWINGK,TAIL,TSG,TØSTR,TTØRQØ,TLE,WRJD,WRS
     3,WRDR,WRH,WRP,PTAILK
     1,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36
     1,TØTTPS,TWGWT,WGWT,WØLEWT,TWT,TLEWT,ØLØTPS
     2,ØASEWT,IØTWT,PTPSCN,TTWT,ØTPSWT,MCSWT,LDTWT
     3,PRØWT,PPPC,PHYC,SCWT
     1,TAPRØP,ENGPAC,ENG,TVC,CØNTR,PRPUTL,PRØSYS
     2,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC
```

```
     1,LNDDK,NG1,NG2,NG3,NGEAR,MG1,MG2,MG3,MGEAR,AX1,AX2
     2,AX3,AXGEAR,LNDDKK
C
C     EXTERNAL TANK OUTPUT COMMON BLOCK
C
      COMMON/EXTO/BODGRP,TOTPS,FWDTK,FAIRT,FWDBLF,FCCTPS
     1,CONSCT,TPSIN,CYLSCT,ACYDM,AFTBLF,WINT,PROSY,AFTNK
     2,FEDSYS,FWDBLA,PRSVNT,AFTCYL,SUMP,AFTBLA,PNPJ,TWINT
     3,NOSEAR,AVIONT,UMBPNL,WRETRO,TUNNEL,MISCT,BAFF,SUBDRY
     4,GU,DRYWT,RESIDT,UNDRAN,FEEDTR,PRSURT,FBIAS,INERT
     5,GROSSW,TLAMB,STRAP,EXTL,EXTD,BLKHD,EXTHB,EXTHH,SIMPTK
C
C     DELTAV OUTPUT COMMON BLOCK
C
      COMMON/DVO/DVT,DVON,DVR,DVBR,DVBRP,X2,X3
     1,DVEPR,TOTLSS,DVONR,DVALT
C
      T=0.0
      M=0.0
      N=0.0
      WOPROP=5000.
 1    PLOADJ=1000.
      RAT(2)=0.0
      PAY(2)=0.0
      PAY(1)=PLOADJ
      THBSLC=THBSL*COS(BCANT/57.2958)
      THOTC=THOV*NOENGO
      THBTC=(THBSLC*NBENGB)+(NOENGO*THOSL)
      THBSLT=THBSL*NBENGB
      BBT=(PROPB*ISPBS)/THBSLT
      IF(TF.NE.0.0) BBT=BBT/TF
      PROPO1=BBT*FLOWR*NOENGO
      PROPBT=PROPB+PROPO1
      ISPB=(PROPB*ISPBV+PROPO1*ISPOV)/PROPBT
      ISPO=ISPOV
      PROPOT=PROPO+PROPO1
      PROPO1=PROPB1
      PROPO2=PROPB
 10   OLOWT=OLLPLB+WOPROP+PLOADU
      OTEMP=EXP(OMSDVP/(32.174*OMSISP))
      WOPROP=OLOWT*(1.-(1./OTEMP))
      OLOWT=OLLPLD+WOPROP+PLOADU
      INERT=DRYWT+RESIDT
      OGLOW=OLOWT+PROPOT+INERT
      BGLOW=PROPB+BOOWT
      GLOW=OGLOW+BGLOW
      WOSCD=GLOW/SCD
      TOTAL=GLOW+BHOLD
      LAMB=PROPB/BGLOW
      TLAMB=PROPOT/(INERT+PROPOT)
      PWO1=PROPB/1.005
      TOWB=THBTC/GLOW
      TOWO=THOTC/(OGLOW-PROPO1)
      FW(1)=TOWB
      FW(2)=TOWO
      DVB=32.174*ISPB*ALOG(GLOW/(GLOW-PROPBT))
      DVONC=32.174*ISPO*ALOG((OGLOW-PROPO1)/
     1(OGLOW-PROPOT))
      DVTOTC=DVB+DVONC
      CALL DELTAV
      RATIOS=DVT/DVTOTC
```

```
      RAT(1)=RATIO
      IF(ABS(DVT-DVTOTC).LT..1) GO TO 100
      PLOADJ=PAY(1)+(PAY(1)-PAY(2))*
     1(1.-RAT(1))/(RAT(1)-RAT(2))
      PAY(2)=PAY(1)
      RAT(2)=RAT(1)
      PAY(1)=PLOADJ
      GO TO 10
  100 IF(N.EQ.1.0) GO TO 200
C
C     DELTAV CORRELATION COEFFICIENT MATCHING ROUTINE
C
      IF(MATCH.EQ.0.0) GO TO 200
      IF(T.GT.0.0) GO TO 130
      RATM(2)=0.0
      DVM(2)=0.0
      DVM(1)=DVCOPR
      RATIOM=TLSSR/TOTLSS
      IF(TOTLSS.GT.TLSSR) RATIOM=TOTLSS/TLSSR
      IF(TOTLSS.GT.TLSSR) M=1.0
      GO TO 140
  130 RATIOM=TLSSR/TOTLSS
      IF(M.EQ.1.0) RATIOM=TOTLSS/TLSSR
  140 RATM(1)=RATIOM
      IF(ABS(TOTLSS-TLSSR).LT..1) GO TO 150
      DVCORR=DVM(1)+(DVM(1)-DVM(2))
     1*(1.-RATM(1))/(RATM(1)-RATM(2))
      DVM(2)=DVM(1)
      RATM(2)=RATM(1)
      DVM(1)=DVCORR
      T=T+1.0
      GO TO 1
  150 DVORR=STAGV
      DIFFRV=DVORR-DVORP
      IF(DIFFRV.LE.0.0) GO TO 170
      DVCNST=DVCNST-DIFFRV
      GO TO 190
  170 DVCNST=DVCNST+DIFFRV
  190 N=1.0
      GO TO 1
  200 CONTINUE
      RETURN
      END
```

8. SENSITIVITIES MODULE

The Fixed Hardware Sensitivities Module was developed for the specific purpose of automating the task of assessing vehicle sensitivities. Theoretically, any sensitivity evaluation can be made by simply making two back-to-back runs through ESPER with the individual sensitivity element adjusted by the desired increment for the second case, and simply subtracting the resulting payloads and/or gross liftoff weights, and dividing by this increment.

Since this option is a part of ESPER, it utilizes the same ascent performance logic and ascent velocity equations. Thus, any output case from ESPER can take advantage of this option. The basic equations within this part of ESPER, however, are modified to contain discrete sensitivity increments (i.e., delta booster inert weight, delta Orbiter inert weight, delta booster ISP, etc.).

The first input case is treated as a fixed hardware configuration and initializes and varies the basic configuration performance capability. The various sensitivity increments are applied one by one, and each sensitivity is calculated separately against the initial case. To save the user the time and money involved in inputting, these sensitivity increments have been fixed in the program. Since this option can be run with any ESPER option, the user need only set SENS equal to 1.0.

Figure 8-1 is a flow diagram of the Sensitivities Module followed by a detail listing of the program.

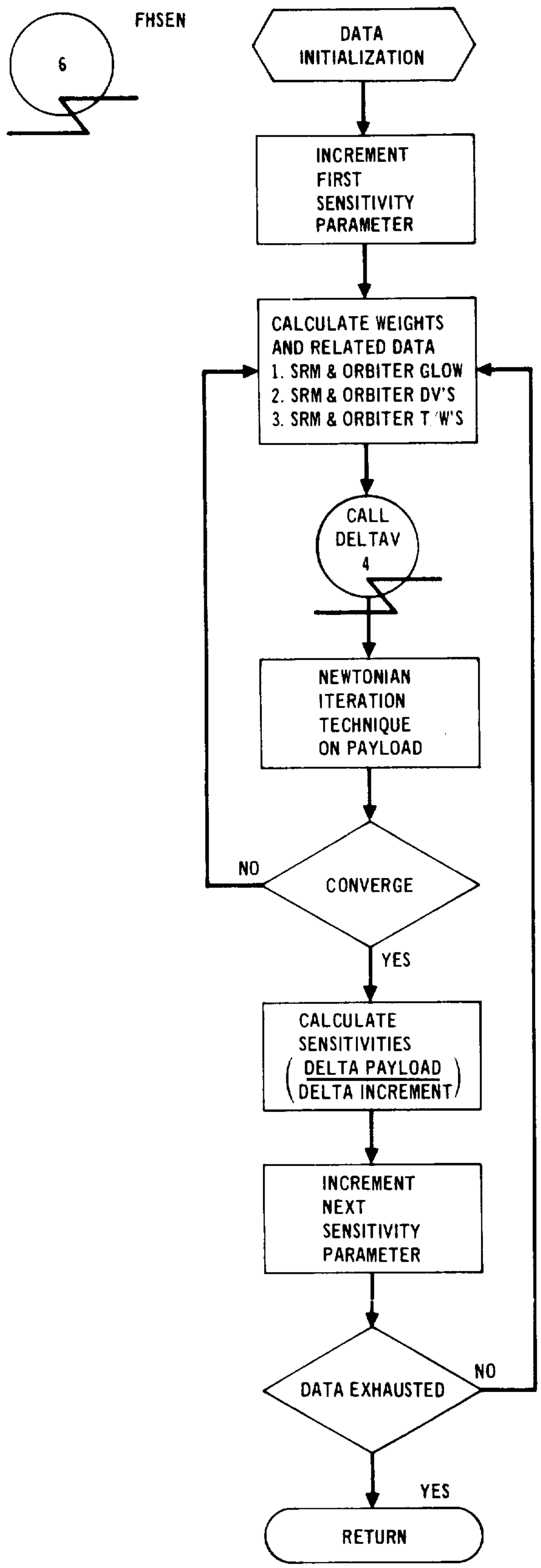

FIGURE 8–1 FLOW DIAGRAM

```
COPY SSUH TO LP(K,NC)
      $FIXED
C
C                                SENSITIVITY SUBROUTINE
C
      SUBROUTINE FHSEN
C     THIS PROGRAM WAS DEVELOPED FOR THE SPECIFIC PURPOSE OF
C     AUTOMATING THE TASK OF ASSESSING VEHICLE SENSITIVITIES.
C     THEORETICALLY,ANY SENSITIVITY EVALUATION CAN BE MADE
C     BY SIMPLY MAKING TWO BACK-TO-BACK RUNS THROUGH ESPER WITH
C     THE INDIVIDUAL SENSITIVITY ELEMENT ADJUSTED BY THE DESIRED
C     INCREMENT FOR THE SECOND CASE,AND SIMPLY SUBTRACTING THE
C     RESULTING PAYLOADS AND/OR LIFTOFF WEIGHTS,AND DIVIDING BY
C     THIS INCREMENT.THESE SENSITIVITY INCREMENTS HAVE BEEN
C     FIXED IN THE PROGRAM.
      IMPLICIT REAL (A-Z)
      DIMENSION RAT(2),PAY(2)
C
C     PERFORMANCE COMMON BLOCK
C
      COMMON/MAIN/ PROPA,PROPB,BRT,BCANT,OCANT,OCANTY
     1,OCANTP,NOENGB,NOENGO,THBSL,THOSL,THOV,TBV,FLOWR
     2,TF,FTW,FIXHRD
     3,ISPBS,ISPBV,ISPOS,ISPOV,SCD,BTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THBTC,THBSLT,THOTC,ISPO,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVONC,DVO,DVTOTC,WOSCD,OINWT,OLANWT
     7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
C
C     SRM OUTPUT COMMON BLOCK
C
      COMMON/SRMO/BOOWT,BDRYWT,BGLOW,XLAMB,PWBI
     1,BASSRM,WCASE,WJOINT,WNOZZ,WTTER,WINST,WIGN,BSRMC
     2,SRMISS,PWFS,PWASLS,PWAS,PWNF,PWTN,PWAV,WNCTPS,SRMIC
     3,WRECOV,PWPAR,PWPI,PWRR,PWRP,PWWR,SRMRC
     4,UNCERT,EXPINS,BOSLUN,SRML,SRMD
     5,PSROSS,PBOOWT,PPROPB,PDRYWT,PBOSLO
     6,FIXBOO,SIMPBO
C
C     ORBITER OUTPUT COMMON BLOCK
C
      COMMON/ORBO/R1,R2,RL,WTAJX,WTACS,ACSENG,ACSSYS,WTACTK
     1,ACSMOD,WTOMS,OMSENG,PROPSY,WTOMTK,MODULE
     1,SURFC,POWR,ELEC,HYDR,AVIONO,ECLSO,PPROV,OUNCWT
     2,ORBMIS,TABPRP,SURFK
     1,PERSON,ORESD,ORESV,PLOADU,PLOADD,ACSPRO,WOPROP,SUDLE
     2,FIXORB,FIXWOR,ODRYWT,ORIFL
     1,WWT,WSG,WBSTR,WTORBE,WTORBC,LEW,WTE,WAIL,WAS,WADR,WAH
     2,WAP,GPROV,PWINGK,TAIL,TSG,TBSTR,TTORGB,TLE,WRUD,WRS
     3,WPDR,WRH,WRP,PTAILK
     1,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36
     1,TOTTPS,TAGWT,WGWT,WGLEWT,IWT,TLEWT,BLBTPS
     2,BASEWT,IBTWT,PTPSCN,TTWT,BTPSWT,MCSWT,LDTWT
     3,PROWT,PPPC,PHYC,SCWT
     1,TAPROP,ENGPAC,ENG,TVC,CONTR,PRPUTL,PROSYS
     2,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC
```

```
     1,LNDOCK,NG1,NG2,NG3,NGEAR,MG1,MG2,MG3,MGEAR,AX1,AX2
     2,AX3,AXGEAR,LNDOCK
C
C     EXTERNAL TANK OUTPUT COMMON BLOCK
C
      COMMON/EXTO/BODGRP,TOTPS,FWDTK,FAIRT,FWDBLF,FCCTPS
     1,CONSCT,TPSIN,CYLSCT,ACYDM,AFTBLF,WINT,PROSY,AFTNK
     2,FEDSYS,FWDBLA,PRSVNT,AFTCYL,SUMP,AFTBLA,PNPJ,TWINT
     3,NOSFAR,AVIONI,UMBPNL,WRETRO,TUNNEL,MISCT,BAFF,SUBDRY
     4,GU,DRYWT,RESIDT,UNDRAN,FEEDTR,PRSURT,FBIAS,INERT
     5,GROSSW,TLAMB,STRAP,EXTL,EXTD,BLKHD,EXTHB,EXTHH,SIMPTK
C
C     DELTAV OUTPUT COMMON BLOCK
C
      COMMON/DVO/DVT,DVON,DVR,DVBR,DVBRP,X2,X3
     1,DVFPR,TOTLSS,DVONR,DVALT
C
      PAYL=PLOADJ
      PAYLX=PAYL
      DVCORX=DVCORR
      DVTX=DVT
      DVMX=OMSDVP
      LAMBX=TLAMB
      GLOWX=GLOW
      CORELP=OLLPLO+WOPROP
      DELCOR=10000.
      DELBWT=0.
      DELBIS=0.
      DELOIS=0.
      DELTHB=0.
      DELTHO=0.
      DELPRO=0.
      DELPRB=0.
      DELLAM=0.
      DELBOO=0.
      DPROI=0.
      DPRBI=0.
      DELPRB=0.
      DELVT=0.
      DELUGO=0.
      DELUGT=0.
      DELUGB=0.
      DLTBIS=0.
      DLTOIS=0.
      XYZ=0.
    1 PAYL=1000.
      RAT(2)=0.0
      PAY(2)=0.0
      PAY(1)=PAYL
      BLAM=(PROPB/(BBOWT+PROPB))+DELBOO
      BBOWTX=PROPB*(1.+DPRBI/100.)*(1.-BLAM)/BLAM
      BBODEL=BBOWTX-BBOWT
      IF(DELBOO.EQ.0.0.AND.DPRBI.EQ.0.0) BBODEL=0.0
      FLOWB=THBSL*NOENGB/ISPBS
      NTHRB=FLOWB*(ISPBS+DLTBIS)
      DLTATB=NTHRB-THBSL*NOENGB
      IF(DLTBIS.EQ.0.0) DLTATB=0.0
      THRBT=THBSL*NOENGB*(1.+DELTHB/100.)+DLTATB
      CTHRB=THRBT*COS(BCANT/57.2958)
      BBT=(((PROPB+DELPRB+DPRBI*PROPB/100.)*(ISPBS+DELBIS
     5+DLTBIS))/THRBT)/TF
```

```
      XLAMB=LAMBX+DELLAM
      NTNKIN=(1.-XLAMB)*PRØPØT*(1.+DPRØI/100.)/XLAMB
      DLTNK=NTNKIN-INERT
      DTKDRY=DLTNK-RESIDT*DPRØI/100.
      IF(DELLAM.EQ.0.0.AND.DPRØI.EQ.0.0) DLTNK=0.0
      UFLØWØ=TØV/ISPØBV
      UNTHRØ=UFLØWØ*(ISPØBV+DLTØIS)
      DLTATØ=UNTHRØ-TØV
      IF(DLTØIS.EQ.0.0) DLTATØ=0.0
      UTHRØ=TØV*(1.+DELTHØ/100.)+DLTATØ
      IF(NØENGØ.EQ.3.0) CTHØV=(UTHRØ+UTHRØ*2.*CØS(ØCANTY
     &/57.2958))/NØENGØ
      IF(NØENGØ.NE.3.0) CTHØV=UTHRØ*CØS(ØCANTY/57.2958)
      CTHØSL=CTHØV*CØS(ØCANTP/57.2958)*(ISPØBS/ISPØBV)
      CTHRØT=CTHØV*NØENGØ
      THRØT=UTHRØ*NØENGØ
      THR1=CTHRØ+CTHØSL*NØENGØ
      PRØPØ1=BØT*THRØT/(ISPØBV+DELØIS+DLTØIS)
      PRØPØ2=PRØPØT*(1.+DPRØI/100.)-PRØPØ1+DELPRØ
   10 GLØW2=CØRELP+PAYL+INERT+DLTNK+PRØPØ2+DELCØR
      GLØW=GLØW2+PRØPØ1+BØØWT+PRØPB+DELBWT+DELPRB+BØØDEL
     &+PRØPB*DPRBI/100.
      ISP1=((ISPBV+DELBIS+DLTBIS)*(PRØPB+DELPRB+DPRBI*
     &PRØPB/100.)+(ISPØBV+DELØIS+DLTØIS)*PRØPØ1)/
     &(PRØPB+DELPRB+DPRBI*PRØPB/100.+PRØPØ1)
      DVB=32.174*ISP1*ALØG(GLØW/(GLØW-PRØPB-PRØPØ1-DELPRB
     &-DPRBI*PRØPB/100.))
      DVØNC=32.174*(ISPØBV+DELØIS+DLTØIS)*ALØG(GLØW2/
     &(GLØW2-PRØPØ2))
      DVTØTC=DVB+DVØNC
      TW1=THR1/GLØW
      TW2=CTHRØT/GLØW2
      FW(1)=TW1
      FW(2)=TW2
      ØGLØW=GLØW2+PRØPØ1
      BGLØW=GLØW-ØGLØW
      DVØB=32.174*BMSISP*ALØG((CØRELP+PAYLX)/(CØRELP+PAYLX-
     &(WBPRØP+DELPRB)))
      WØSCD=GLØW/SCD
      DVCØRR=DVCØRX+DELVT
      CALL DELTAV(H,DVCØRR,INC,DVCØN,DVCNST,FW,WØSCD,DVB)
      RATIØØ=DVT/DVTØTC
      RAT(1)=RATIØØ
      IF(ABS(DVT-DVTØTC).LT..1) GØ TØ 100
      PAYL=PAY(1)+(PAY(1)-PAY(2))*
     1(1.-RAT(1))/(RAT(1)-RAT(2))
      PAY(2)=PAY(1)
      RAT(2)=RAT(1)
      PAY(1)=PAYL
      GØ TØ 10
  100 CALL SENSIV (PAYLX,PAYL,DVTX,DVMX,DVB3,LAMBX,DELCØR
     1,DELBWT,DELBIS,DELØIS,DELTHB,DELTHØ,XYZ,DELPRØ,DELPRB
     2,DELLAM,DELB30,DPRØI,DPRBI,DELVT,DELPR3,DELU30
     3,DELUST,DELU3B,GLØWX,DLTØIS,DLTBIS,DTKDRY,XLAMB)
      IF(XYZ.EQ.1.) GØ TØ 999
      GØ TØ 1
  999 CØNTINUE
      RETURN
      END
      SUBRØUTINE SENSIV (PAYLX,PAYL,DVTX,DVMX,DVB3,LAMBX
     1,DELCØR,DELBWT,DELBIS,DELØIS,DELTHØ,DELTHB,XYZ
```

```
     2,DELPRO,DELPRB,DELLAM,DELBOO,DPROI,DPRBI,DELVT
     3,DELPRB,DELUOR,DELUGT,DELUOB,GLOWX,DLTBIS,DLTOIS
     4,DTKDRY,XLAMB)
      IMPLICIT REAL (A-Z)
      COMMON/MAIN/ PROPB,PROPO,BOT,BCANT,OCANT,OCANTY
     1,OCANTP,NHENGB,NOENGO,THBSL,THOSL,THOV,TOV,FLOWR
     2,TF,FTW,FIXHRD
     3,ISPBS,ISPBV,ISPOBS,ISPOBV,SCD,BTW
     4,H,DVCORR,INC,STAGV,DVCON,DVCNST
     5,REL,THBTC,THPSLT,THOTC,ISPO,ISPB,PROPOT
     6,PROPO1,PROPO2,FW(2),DVONC,DVB,DVTOTC,WBSCD,OINWT,OLANWT
     7,OINJWT,OLOWT,OGLOW,GLOW,TOTAL,S,P,MATCH,TLSSR,FPRP
     8,OHOLD,OLLPLO,OLOWLO,OMSISP,OMSDVT,OMSDVP,OMR
     9,LONGP,TOWO,TOWB,SENS,GROW,MINGLW
      COMMON/SRMO/BROWT,BDRYWT,BGLOW,LAMB,PWOI
     1,RASSRM,WCASE,WJOINT,WNOZZ,WTTER,WINST,WIGN,BSRMC
     2,SRMISS,PWFS,PWASLS,PWAS,PWNF,PWTN,PWAV,WNCTPS,SRMIC
     3,WRECOV,PWPAR,PWPI,PWRR,PWRP,PWWR,SRMRC
     4,UNCERT,EXPINS,BOSLUN,SRML,SRMD
     5,PGROSS,PBROWT,PPROPB,PDRYWT,PBOSLU
     6,FIXBOO,SIMPBO
      COMMON/ORBO/R1,R2,RL,WTAJX,WTACS,ACSENG,ACSSYS,WTACTK
     1,ACSMOD,WTOMS,OMSENG,PROPSY,WTOMTK,MODULE
     1,SURFC,PPWR,ELEC,HYDR,AVIONO,ECLSO,PPROV,OUNCWT
     2,ORBMIS,TAOPRO,SURFK
     1,PERSON,ORESD,ORESV,PLOADU,PLOADD,ACSPRO,WOPROP,SUDLE
     2,FIXORB,FIXWOF,ODRYWT,ORIFL
     1,WWT,WSG,WBSTR,WTORBE,WTORBC,LEW,WTE,WAIL,WAS,WADR,WAH
     2,WAP,GPROV,PWINGK,TAIL,TSG,TOSTR,TTORQB,TLE,WRUD,WRS
     3,WRDR,WRH,WRP,PTAILK
     1,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36
     1,TOTTPS,TWGWT,WGWT,WGLEWT,TWT,TLEWT,BLBTPS
     2,BASEWT,IBTWT,PTPSCN,TTWT,BTPSWT,MCSWT,LDTWT
     3,PROWT,PPPC,PHYC,SCWT
     1,TAPROP,ENGPAC,ENG,TVC,CONTR,PRPUTL,PROSYS
     2,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC
     1,LNDDK,NG1,NG2,NG3,NGEAR,MG1,MG2,MG3,MGEAR,AX1,AX2
     2,AX3,AXGEAR,LNDDKK
      COMMON/EXTO/BODGRP,TOTPS,FWDTK,FAIRT,FWDBLF,FCCTPS
     1,CONSCT,TPSIN,CYLSCT,ACYDM,AFTBLF,WINT,PROSY,AFTNK
     2,FEDSYS,FWDBLA,PRSVNT,AFTCYL,SUMP,AFTBLA,PNPJ,TWINT
     3,NOSFAR,AVIONT,JMOPNL,WRETRO,TUNNEL,MISCT,BAFF,SUBDRY
     4,GU,DRYWT,RESIDT,UNDRAN,FEEDTR,PRSURT,FBIAS,INERT
     5,GROSSW,TLAMB,OTRAP,EXTL,EXTD,BLKHD,EXTHO,EXTHH,SIMPTK
      COMMON/DVO/DVT,DVON,DVR,DVBR,DVBRP,X2,X3
     1,DVFPR,TOTLSS,DVONR,DVALT
      DELPAY=PAYL-PAYLX
      IF(DELCOR.EQ.0.0) GO TO 10
      IF(DELCOR.LT.0.0) GO TO 9
      WRITE(108,400)
      SNCORP=DELPAY/DELCOR
      WRITE(108,410) DELCOR,DELPAY,SNCORP
      DELCOR=-10000.
      GO TO 43
    9 SNCORN=DELPAY/DELCOR
      WRITE(108,410) DELCOR,DELPAY,SNCORN
      SNCORA=(SNCORP+SNCORN)/2.
      WRITE(108,500) SNCORA
      DELCOR=0.
```

```
      DELBWT=20000.
      GO TO 43
   10 IF(DELBWT.EQ.0.0) GO TO 12
      IF(DELBWT.LT.0.0) GO TO 11
      SNBWTP=DELPAY/DELBWT
      WRITE(108,411) DELBWT,DELPAY,SNBWTP
      DELBWT=-20000.
      GO TO 43
   11 SNBWTN=DELPAY/DELBWT
      WRITE(108,411) DELBWT,DELPAY,SNBWTN
      SNBWTA=(SNBWTP+SNBWTN)/2.
      WRITE(108,501) SNBWTA
      DELBWT=0.
      DELBIS=3.
      GO TO 43
   12 IF(DELBIS.EQ.0.0) GO TO 14
      IF(DELBIS.LT.0.0) GO TO 13
      SNBISP=DELPAY/DELBIS
      WRITE(108,412) DELBIS,DELPAY,SNBISP
      DELBIS=-3.
      GO TO 43
   13 SNBISN=DELPAY/DELBIS
      WRITE(108,412) DELBIS,DELPAY,SNBISN
      SNBISA=(SNBISP+SNBISN)/2.
      WRITE(108,500) SNBISA
      DELBIS=0.
      DEL8IS=3.
      GO TO 43
   14 IF(DEL8IS.EQ.0.0) GO TO 39
      IF(DEL8IS.LT.0.0) GO TO 15
      SN8ISP=DELPAY/DEL8IS
      WRITE(108,413) DEL8IS,DELPAY,SN8ISP
      DEL8IS=-3.
      GO TO 43
   15 SN8ISN=DELPAY/DEL8IS
      WRITE(108,413) DEL8IS,DELPAY,SN8ISN
      SN8ISA=(SN8ISP+SN8ISN)/2.
      WRITE(108,500) SN8ISA
      DEL8IS=0.
      DLTBIS=3.
      GO TO 43
   39 IF(DLTBIS.EQ.0.0) GO TO 41
      IF(DLTBIS.LT.0.0) GO TO 40
      STBISP=DELPAY/DLTBIS
      WRITE(108,435) DLTBIS,DELPAY,STBISP
      DLTBIS=-3.
      GO TO 43
   40 STBISN=DELPAY/DLTBIS
      WRITE(108,435) DLTBIS,DELPAY,STBISN
      STBISA=(STBISP+STBISN)/2.
      WRITE(108,500) STBISA
      DLTBIS=0.
      DLT8IS=3.
      GO TO 43
   41 IF(DLT8IS.EQ.0.0) GO TO 16
      IF(DLT8IS.LT.0.0) GO TO 42
      ST8ISP=DELPAY/DLT8IS
      WRITE(108,436) DLT8IS,DELPAY,ST8ISP
      DLT8IS=-3.
      GO TO 43
   42 ST8ISN=DELPAY/DLT8IS
```

```
      WRITE(108,436) DLTØIS,DELPAY,STØISN
      STØISA=(STØISP+STØISN)/2.
      WRITE(108,500) STØISA
      DLTØIS=0.
      DELTHB=2.
      GO TO 43
   16 IF(DELTHB.EQ.0.0) GO TO 18
      IF(DELTHB.LT.0.0) GO TO 17
      SNTHBP=DELPAY/DELTHB
      WRITE(108,414) DELTHB,DELPAY,SNTHBP
      DELTHB=-2.
      GO TO 43
   17 SNTHBN=DELPAY/DELTHB
      WRITE(108,414) DELTHB,DELPAY,SNTHBN
      SNTHBA=(SNTHBP+SNTHBN)/2.
      WRITE(108,500) SNTHBA
      DELTHB=0.
      DELTHØ=3.
      GO TO 43
   18 IF(DELTHØ.EQ.0.0) GO TO 20
      IF(DELTHØ.LT.0.0) GO TO 19
      SNTHØP=DELPAY/DELTHØ
      WRITE(108,415) DELTHØ,DELPAY,SNTHØP
      DELTHØ=-3.
      GO TO 43
   19 SNTHØN=DELPAY/DELTHØ
      WRITE(108,415) DELTHØ,DELPAY,SNTHØN
      SNTHØA=(SNTHØP+SNTHØN)/2.
      WRITE(108,500) SNTHØA
      DELTHØ=0.
      DELPRØ=10000.
      GO TO 43
   20 IF(DELPRØ.EQ.0.0) GO TO 22
      IF(DELPRØ.LT.0.0) GO TO 21
      SNPRØP=DELPAY/DELPRØ
      WRITE(108,416) DELPRØ,DELPAY,SNPRØP
      DELPRØ=-10000.
      GO TO 43
   21 SNPRØN=DELPAY/DELPRØ
      WRITE(108,416) DELPRØ,DELPAY,SNPRØN
      SNPRØA=(SNPRØP+SNPRØN)/2.
      WRITE(108,501) SNPRØA
      DELPRØ=0.
      DELPRB=10000.
      GO TO 43
   22 IF(DELPRB.EQ.0.0) GO TO 24
      IF(DELPRB.LT.0.0) GO TO 23
      SNPRBP=DELPAY/DELPRB
      WRITE(108,417) DELPRB,DELPAY,SNPRBP
      DELPRB=-10000.
      GO TO 43
   23 SNPRBN=DELPAY/DELPRB
      WRITE(108,417) DELPRB,DELPAY,SNPRBN
      SNPRBA=(SNPRBP+SNPRBN)/2.
      WRITE(108,501) SNPRBA
      DELPRB=0.
      DELLAM=.01
      GO TO 43
   24 IF(DELLAM.EQ.0.0) GO TO 26
      IF(DELLAM.LT.0.0) GO TO 25
      SNLAMP=DELPAY
```

```
      WRITE(108,418) DELLAM,DELPAY,SNLAMP
      DELLAM=-.01
      GO TO 43
25    SNLAMN=-DELPAY
      WRITE(108,418) DELLAM,DELPAY,SNLAMN
      SNLAMA=(SNLAMP+SNLAMN)/2.
      WRITE(108,500) SNLAMA
      DELLAM=0.
      DELB99=.01
      GO TO 43
26    IF(DELB99.EQ.0.0) GO TO 28
      IF(DELB99.LT.0.0) GO TO 27
      SNB99P=DELPAY
      WRITE(108,419) DELB99,DELPAY,SNB99P
      DELB99=-.01
      GO TO 43
27    SNB99N=-DELPAY
      WRITE(108,419) DELB99,DELPAY,SNB99N
      SNB99A=(SNB99P+SNB99N)/2.
      WRITE(108,500) SNB99A
      DELB99=0.
      DPR9I=10.
      GO TO 43
28    IF(DPR9I.EQ.0.0) GO TO 30
      IF(DPR9I.LT.0.0) GO TO 29
      SNP9IP=DELPAY/DPR9I
      DELGLW=GLOW-GLOWX
      DGOP=DELGLW/DELPAY
      P=SIGN(1.,DGOP)
      DLPR9=PR9P9I*DPR9I/100.
      DPDP=DLPR9/DELPAY
      DWDP=DGOP*P
      DTDP=DTKDRY/DELPAY
      WRITE(108,420) DPR9I,DELPAY,SNP9IP
      DPR9I=-10.
      GO TO 43
29    SNP9IN=DELPAY/DPR9I
      WRITE(108,420) DPR9I,DELPAY,SNP9IN
      SNP9IA=(SNP9IP+SNP9IN)/2.
      WRITE(108,500) SNP9IA
      DPR9I=0.
      DPR8I=10.
      GO TO 43
30    IF(DPR8I.EQ.0.0) GO TO 32
      IF(DPR8I.LT.0.0) GO TO 31
      SNP8P=DELPAY/DPR8I
      WRITE(108,421) DPR8I,DELPAY,SNP8P
      DPR8I=-10.
      GO TO 43
31    SNP8N=DELPAY/DPR8I
      WRITE(108,421) DPR8I,DELPAY,SNP8N
      SNP8A=(SNP8P+SNP8N)/2.
      WRITE(108,500) SNP8A
      DPR8I=0.
      DELVT=100.
      GO TO 43
32    IF(DELVT.EQ.0.0) GO TO 34
      DLVTX=DVT-DVTX
      IF(DELVT.LT.0.0) GO TO 33
      SNVTP=DELPAY/DLVTX
      WRITE(108,422) DLVTX,DELPAY,SNVTP
```

```
      DELVT=-100.
      GO TO 43
   33 SNVTN=DELPAY/DLVTX
      WRITE(108,422) DLVTX,DELPAY,SNVTN
      SNVTA=(SNVTP+SNVTN)/2.
      WRITE(108,500) SNVTA
      DELVT=0.
      DELPR3=1000.
      GO TO 43
   34 IF(DELPR3.EQ.0.0) GO TO 36
      DLVM=DVO3-DVMX
      DELPAY=-DELPR3
      IF(DELPR3.LT.0.0) GO TO 35
      SNPR3P=DELPAY/DLVM
      WRITE(108,423) DLVM,DELPAY,SNPR3P
      DELPR3=-1000.
      GO TO 43
   35 SNPR3N=DELPAY/DLVM
      WRITE(108,423) DLVM,DELPAY,SNPR3N
      SNPR3A=(SNPR3P+SNPR3N)/2.
      WRITE(108,500) SNPR3A
      DELPR3=0.
      IF(FIXBOO.GT.0.0.OR.SIMPBO.GT.0.0
     1.OR.FIXORB.GT.0.0.OR.FIXWBR.GT.0.0
     2.OR.SIMPTK.GT.0.0.OR.FIXHRD.GT.0.0) GO TO 44
      DELJGO=.01*SJDLE
      GO TO 43
   36 IF(DELUGO.EQ.0.0) GO TO 37
      SNJGO=-DELJGO
      WRITE(108,450) DELUGO,SNJGO
      DELJGO=0.
      DELJGT=.01*SJBDRY
      GO TO 43
   37 IF(DELJGT.EQ.0.0) GO TO 38
      SNUGT=-DELJGT
      WRITE(108,451) DELUGT,SNJGT
      DELJGT=0.
      DELJGB=.01*BOSLJN
      GO TO 43
   38 DELPAY=-DELUGB*SNBWTN
      SNJGB=-DELPAY
      WRITE(108,452) DELPAY,SNJGB
C     GLOW SENSITIVITIES WITH STRETCHED HB TANK
   44 DIOW=-(DWDP*SNCORP-P)
      DIBW=-(DWDP*SNBWTP-P)
      DISO=DWDP*SNOISN
      DISB=DWDP*SNBISN
      DTHO=DWDP*SNTHON
      DTHB=DWDP*SNTHBN
      DLVT=-(DWDP*SNVTP)
      DLVM=-(DWDP*SNPR3P)
      WRITE(108,425)
      WRITE(108,426) DGDP,DIOW
      WRITE(108,427) DISO,DISB
      WRITE(108,428) DTHO,DTHB
      WRITE(108,429) DLVT
      WRITE(108,430) DLVM
      WRITE(108,431) DIOW
      WRITE(108,432) DPDP
      WRITE(108,434) DTDP
      XYZ=1.
```

```
43    CONTINUE
      RETURN
  400 FORMAT('1',///24X,'VEHICLE SENSITIVITIES'//
     &9X,'ITEM',8X,'INCREMENT',6X,'DELTA PAYLOAD',
     &7X,'SENSITIVITY'/)
  410 FORMAT(1X,'ORBITER INERT WT',3X,F10.2,
     &' LB',F13.1,' LB',F13.1,' LB/LB')
  411 FORMAT(1X,'BOOSTER INERT WT',3X,F10.2,
     &' LB',F13.1,' LB',F13.5,' LB/LB')
  412 FORMAT(1X,'BSTR ISP(CONST THRUST)',F7.2,
     &' SEC',F12.1,' LB',F13.1,' LB/SEC')
  413 FORMAT(1X,'ORBT ISP(CONST THRUST)',F7.2,
     &' SEC',F12.1,' LB',F13.1,' LB/SEC')
  414 FORMAT(1X,'BOOSTER THRUST',5X,F10.2,
     &' % ',F13.1,' LB',F13.1,' LB/%')
  415 FORMAT(1X,'ORBITER THRUST',5X,F10.2,
     &' % ',F13.1,' LB',F13.1,' LB/%')
  416 FORMAT(1X,'ORBITER PROPELLANT',1X,F10.2,
     &' LB',F13.1,' LB',F13.5,' LB/LB')
  417 FORMAT(1X,'BOOSTER PROPELLANT',1X,F10.2,
     &' LB',F13.1,' LB',F13.5,' LB/LB')
  418 FORMAT(1X,'ORBITER TANK LAMBDA',F10.2,
     &' LB/LB',F10.1,' LB',F13.1,' LB/.01')
  419 FORMAT(1X,'BOOSTER LAMBDA',5X,F10.2,
     &' LB/LB',F10.1,' LB',F13.1,' LB/.01')
  420 FORMAT(1X,'ORBITER PROP+INERT WT',F8.2,
     &' % ',F13.1,' LB',F13.1,' LB/%')
  421 FORMAT(1X,'BOOSTER PROP+INERT WT',F8.2,
     &' % ',F13.1,' LB',F13.1,' LB/%')
  422 FORMAT(1X,'TOTAL DELTA V',6X,F10.2,
     &' FPS',F12.1,' LB',F13.1,' LB/FPS')
  423 FORMAT(1X,'ORBITER MANUEV DELTA V',F7.2,
     &' FPS',F12.1,' LB',F13.1,' LB/FPS')
  425 FORMAT('1',///26X,'GLOW SENSITIVITIES'///
     &40X,'ORBITER/TANK',5X,'BOOSTER'///
     &5X,'+ DELTA GLOW (LB) DUE TO:'/)
  426 FORMAT(7X,'+ DELTA INERT WT (LB)',15X,F6.1,9X,F6.1/)
  427 FORMAT(7X,'- DELTA ISP (SEC)',17X,F7.0,8X,F7.0/)
  428 FORMAT(7X,'- DELTA THRUST (%)',16X,F7.0,8X,F7.0/)
  429 FORMAT(7X,'+ DELTA V TOTAL (FPS)',13X,F7.0,10X,
     &'----'/)
  430 FORMAT(7X,'+ DELTA V MANUEVER (FPS)',10X,F7.0,10X,
     &'----'/)
  431 FORMAT(5X,'DELTA GLOW/DELTA PAYLOAD (LB/LB)',13X,F6.1/)
  432 FORMAT(5X,'DELTA HO-PROP/DELTA PAYLOAD',18X,F6.1/)
  434 FORMAT(5X,'DELTA HO-TANK(DRY)/DELTA PAYLOAD',13X,F6.3/)
  435 FORMAT(1X,'BOOSTER ISP(CONST FLOW)',F6.2,
     &' SEC',F12.1,' LB',F13.1,' LB/SEC')
  436 FORMAT(1X,'ORBITER ISP(CONST FLOW)',F6.2,
     &' SEC',F12.1,' LB',F13.1,' LB/SEC')
  450 FORMAT(1X,'ORBITER GROWTH/UNCER.',2X,
     &'= 1.',3X,'% ',F13.1,' LB',F13.1,' LB/%')
  451 FORMAT(1X,'HO-TANK GROWTH/UNCER.',2X,
     &'= 1.',3X,'% ',F13.1,' LB',F13.1,' LB/%')
  452 FORMAT(1X,'BOOSTER GROWTH/UNCER.',2X,
     &'= 1.',3X,'% ',F13.1,' LB',F13.1,' LB/%')
  500 FORMAT(49X,'AVG',F10.1)
  501 FORMAT(49X,'AVG',F10.5)
      END
```

9. OUTPUT MODULE

The output module contains two basic options, either a simplified printout or a detailed printout.

The simplified weight statement supplies the user with total vehicle weights and performance parameters, such as, TW's, GLOW's and ΔV's.

The detailed weight statement exists in the form of the NASA Phase B functional weight grouping. This coding gives a direct line-by-line comparison of ESPER's data with the weight status report of the mainline Shuttle program.

Combinations of these two mainline options are numerous. For example, if the user is running a fixed SRM and wishes to see details on the Orbiter, or if the user is running a fixed Orbiter and wishes to see details on the SRM, etc.). Figure 9-1 is a flow diagram of the Output Module followed by a detailed listing of the program.

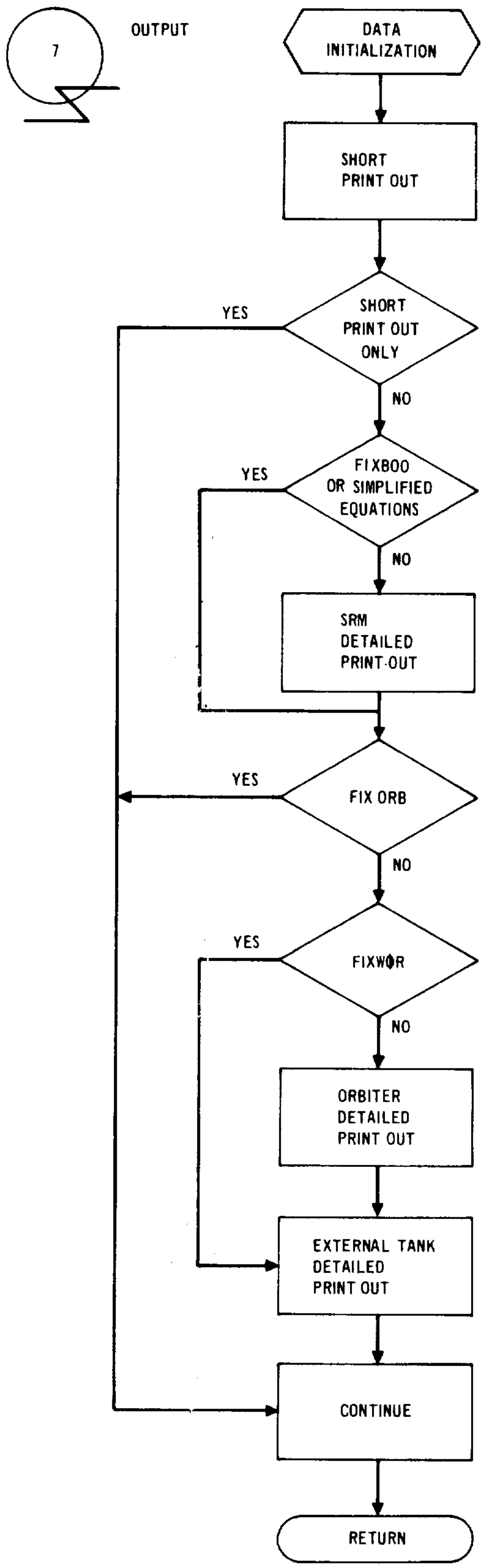

FIGURE 9–1 FLOW DIAGRAM

```
CØPY ØPSUB TØ LP(K,NC)
*FIXED
C
C                          ØUTPUT SUBRØUTINE
C
      SUBRØUTINE ØUTPUT
C     THE PRIMARY PURPØSE ØF THIS PRØGRAM IS TØ FØRMAT THE
C     ESPER ØUTPUT.THE BASIC ØUTPUT IS EITHER A SIMPLIFIED ØR
C     DETAILED WEIGHT STATEMENT.THE SIMPLIFIED WEIGHT STATEMENT
C     SUPPLIES THE USER WITH TØTAL VEHICLE WEIGHTS AND
C     PERFØRMANCE PARAMETERS,SUCH AS,TW,GLØW AND DELTA V.
C     THE DETAILED WEIGHT STATEMENT EXISTS IN THE FØRM ØF THE
C     NASA PHASE B FUNCTIONAL WEIGHT GRØUPING.
      IMPLICIT REAL (A-Z)
C
C     PERFØRMANCE CØMMØN BLØCK
C
      CØMMØN/MAIN/ PRØPB,PRØPØ,RØT,BCANT,ØCANT,ØCANTY
     1,ØCANTP,NØENGØ,NØENGØ,THØSL,THØSL,THØV,TØV,FLØWR
     2,TF,FTW,FIXHRD
     3,ISPØS,ISPØV,ISPØØS,ISPØØV,SCD,ØTW
     4,H,DVCØRR,INC,STAGV,DVCØN,DVCNST
     5,REL,THØTC,THØSLT,THØTC,ISPØ,ISPØ,PRØPØT
     6,PRØPØ1,PRØPØ2,FW(2),DVØNC,DVØ,DVTØTC,WØSCD,ØINWT,ØLANWT
     7,ØINJWT,ØLØWT,ØGLØW,GLØW,TØTAL,S,P,MATCH,TLSSR,FPRP
     8,ØHØLD,ØLLPLØ,ØLØWLØ,ØMSISP,ØMSDVT,ØMSDVP,ØMR
     9,LØNGP,TØWØ,TØWØ,SENS,GRØW,MINGLW
C
C     SRM ØUTPUT CØMMØN BLØCK
C
      CØMMØN/SRMØ/ØØØWT,ØDRYWT,ØGLØW,LAMØ,PWØI
     1,MASSRM,WCASE,WJØINT,WNØZZ,WITER,WINST,WIGN,ØSRMC
     2,SRMISS,PWFS,PWASLS,PWAS,PWNF,PWTN,PWAV,WNCTPS,SRMIC
     3,WRECØV,PWPAR,PWPI,PWRR,PWRP,PWWR,SRMRC
     4,UNCERT,EXPINS,ØØSLUN,SRML,SRMD
     5,PGRØSS,PØØØWT,PPRØPB,PDRYWT,PRØSLJ
     6,FIXØØØ,SIMPØØ
C
C     ØRBITER ØUTPUT CØMMØN BLØCK
C
      CØMMØN/ØRØØ/R1,R2,RL,WTAJX,WTACS,ACSENG,ACSSYS,WTACTK
     1,ACSMØD,WTØMS,ØMSENG,PRØPSY,WTØMTK,MØDULE
     1,SURFC,PPWR,ELEC,HYDR,AVIØNØ,ECLSØ,PPRØV,ØUNCWT
     2,ØRØMIS,TAØPRØ,SURFK
     1,PERSØN,ØFESD,ØRESV,PLØADU,PLØADD,ACSPRØ,WØPRØP,SUDLE
     2,FIXØRØ,FIXWØR,ØDRYWT,ØRIFL
     1,WWT,WSG,WØSTR,WTØRØE,WTØRØC,LEW,WTE,WAIL,WAS,WADR,WAH
     2,WAP,GPRØV,PWINGK,TAIL,TSG,TØSTR,TTØRØB,TLE,WRUD,WRS
     3,WRDR,WRH,WRP,PTAILK
     1,G37,G1,G2,G3,G6,G7,G8,G9,G10,G11,G12,G15,G16
     2,G17,G18,G19,G22,G23,G24,G25,G26,G27,G32,G33,G34
     3,G35,G36
     1,TØTTPS,TWGWT,WGWT,WGLEWT,TWT,TLEWT,BLØTPS
     2,BASEWT,IBTWT,PTPSCN,TTWT,BTPSWT,MCSWT,LDTWT
     3,PRØWT,PPPC,PHYC,SCWT
     1,TAPRØP,ENGPAC,ENG,TVC,CØNTR,PRPUTL,PRØSYS
     2,FAD,PRES,CHIL,PREVAL,FEEDS,DISC,MISC
     1,LNDDK,NG1,N32,NG3,NGEAR,M31,M32,M33,MGEAR,AX1,AX2
     2,AX3,AXGEAR,LNDDKK
C
```

```
C      EXTERNAL TANK OUTPUT COMMON BLOCK
C
       COMMON/EXTO/BODGRP,TOTPS,FWDTK,FAIRT,FWDBLF,ECCTPS
      1,CONSCT,TPSIN,CYLSCT,ACYDM,AFTBLF,WINT,PROSY,AFTNK
      2,FEDSYS,FWDBLA,PRSVNT,AFTCYL,SUMP,AFTBLA,PNPJ,TWINT
      3,NOSFAR,AVIONT,JMBPNL,WRETRO,TUNNEL,MISCT,BAFF,SJODRY
      4,GJ,DRYWT,RESIDT,UNDRAN,FEEDTR,PRSJRT,FBIAS,INERT
      5,GROSSW,TLAMB,OTRAP,EXTL,EXTD,BLKHD,EXTHO,EXTHH,SIMPTK
C
C      DELTAV OUTPUT COMMON BLOCK
C
       COMMON/DVO/DVT,DVON,DVR,DVBR,DVBRP,X2,X3
      1,DVFPR,TOTLSS,DVONR,DVALT
C
C      SIMPLIFIED WEIGHT STATEMENT
C
       WRITE(108,100)
       IF(FIXHRD.GT.0.0)  WRITE(108,120)
       IF(FIXHRD.GT.0.0) GO TO 9
       IF(MINGLW.GT.0.0) WRITE(108,125)
       IF(MINGLW.GT.0.0) GO TO 9
       IF(FIXBRH.GT.0.0) WRITE(108,130)
       IF(FIXWBR.GT.0.0) WRITE(108,131)
       IF(FIXHOB.GT.0.0) WRITE(108,140)
       IF(SIMPHB.GT.0.0) WRITE(108,142)
       IF(SIMPTK.GT.0.0) WRITE(108,143)
       IF(FTW.GT.0.0) GO TO 8
       WRITE(108,150)
       GO TO 9
 8     WRITE(108,160)
 9     WRITE(108,110)
       WRITE(108,115)
       IF(ABS(DVONR-DVONC).GT.1.) GO TO 10
       GO TO 20
 10    WRITE(108,190)
       WRITE(108,200)
       WRITE(108,210)
 20    IF(TOWB.LT..7.OR.TOWB.GT.2.0.OR.TOWB.LT.1.18
      $.OR.TOWB.GT.1.6)  WRITE(108,220)
       IF(DVBRP.LT.3000..OR.DVBRP.GT.9000.) WRITE(108,230)
       WRITE(108,240) INC
       WRITE(108,250) TOTLSS
       WRITE(108,260) DVCORR,DVCNST
       WRITE(108,270) X3
       WRITE(108,280) X2
       WRITE(108,290) DVALT
       WRITE(108,300) WOSCD,SCD
       WRITE(108,310)
       WRITE(108,320) BCANT
       WRITE(108,330) BOOWT
       WRITE(108,340) BDRYWT
       WRITE(108,360) PROPB,PWBI
       WRITE(108,350) LAMB
       WRITE(108,370) ISPBV,ISPOBV,ISPB
       WRITE(108,390) THBTC,THBSLT
       WRITE(108,410) BBT
       WRITE(108,420)
       WRITE(108,430) OCANTY,OCANTP
       WRITE(108,460) OLLPLO
       WRITE(108,440) PLOADJ
       WRITE(108,470) WOPROP
```

```
          WRITE(108,800)
          WRITE(108,480) DRYWT
          WRITE(108,760) RESIDT
          WRITE(108,530)
          WRITE(108,540) PRØPØT,PRØPØ1,PRØPØ2
          WRITE(108,350) TLAMB
          WRITE(108,610)  ISPØ
          WRITE(108,620)  THØTC,TØV,NØENGØ
          WRITE(108,810) FLØWR
          WRITE(108,630)  ØGLØW
          WRITE(108,640)  BGLØW
          WRITE(108,650) GLØW
          IF(ØHØLD.GT.0.0)  WRITE(108,660)  TØTAL,ØHØLD
          WRITE(108,670)  TØWB
          WRITE(108,680)  TØWØ
          WRITE(108,690)  DVB,DVFPR
          WRITE(108,700)  DVBRP,FPRP
          WRITE(108,710)  DVR
          WRITE(108,720)  DVCØN,DVT
          IF(FIXØRB.GT.0.) WRITE(108,730)  DVCØN,DVTØTC
          WRITE(108,740)  DVØNR
          WRITE(108,750) DVØNC
    810   FØRMAT(6X,'FLØW RATE',9X,F12.2)
    800   FØRMAT(/4X,'EXT TANK PARAMETERS')
    470   FØRMAT(6X,'ØAMS PRØP WEIGHT',2X,F12.2)
    760   FØRMAT(6X,'EXT. TANK RESID.',2X,F12.2)
    430   FØRMAT(6X,'(CANT ANGLE YAW=',F4.1,')',/,6X,
         1'(CANT ANGLE PITCH=',F4.1,')')
    115   FØRMAT(21X,'BØØSTER - SØLID RØCKET MØTØR')
    320   FØRMAT(6X,'(CANT ANGLE=',F4.1,')')
    130   FØRMAT(29X,'FIXED ØRBITER')
    131   FØRMAT(27X,'FIXED WT. ØRBITER')
    300   FØRMAT(24X,'W/SCD',12X,F7.1,2X,'SCD',2X,F7.1)
    230   FØRMAT(/15X,'********** WARNING LØØK AT DVBRP **********')
    460   FØRMAT(6X,'LIFT ØFF-PAY-ØAMS',1X,F12.2)
    440   FØRMAT(6X,'PAYLØAD',11X,F12.2)
    120   FØRMAT(25X,'FIXED HARDWARE  (ASPER)')
    125   FØRMAT(23X,'MINIMUM GLØW VEHICLE')
    370   FØRMAT(6X,'ISP AV (',F5.1,',',F5.1,')',4X,F6.2)
    380   FØRMAT(6X,'SEA LEVEL THRUST',2X,F12.2)
    140   FØRMAT(29X,'FIXED BØØSTER')
    142   FØRMAT(27X,'SIMPLIFIED BØØSTER')
    143   FØRMAT(27X,'SIMPLIFIED EXTANK')
    150   FØRMAT(27X,'FIXED THR BØØSTER')
    160   FØRMAT(27X,'FIXED T/W BØØSTER')
    100   FØRMAT('1',/,29X,'PARALLEL BURN')
    110   FØRMAT(15X,'PARAMETRIC STUDY EXTERNAL (H2-02) ØRBITER')
    190   FØRMAT(/26X,'NØ  VALID SØLUTIØN')
    200   FØRMAT(26X,'ØRBITER TØØ LARGE')
    210   FØRMAT(21X,'FØR THE PARAMETERS SELECTED')
    220   FØRMAT(/15X,'********** WARNING  LØØK AT T/W **********')
    240   FØRMAT(/17X,'INCLINATIØN  ØF ØRBITER',F7.2,'  DEGREES')
    250   FØRMAT(/24X,'DV TØTAL LØSSES',1X,'(',F7.1,')')
    280   FØRMAT(/24X,'DV INC',11X,F7.1)
    290   FØRMAT(24X,'DV ALT',11X,F7.1)
    260   FØRMAT(24X,'DV CØRRECTIØN',4X,F7.1,2X,'DVCNST',2X,F9.1)
    270   FØRMAT(24X,'DV CURVE LØSSES',2X,F7.1)
    310   FØRMAT(/4X,'BØØSTER  PARAMETERS')
    330   FØRMAT(6X,'BURN ØUT WEIGHT',3X,F12.2)
    340   FØRMAT(6X'DRY WEIGHT',8X,F12.2)
    360   FØRMAT(6X,'PRØPELLANT WEIGHT',1X,F12.2,2X,'PRØPB WØI',1X,F12.2)
```

```
 350     FORMAT(3X,'LAMBDA=',F5.4)
 390   FORMAT(6X,'SEA LEVEL THRUST',2X,F12.2
      &,2X,'S.L. T 399',F12.2)
 410   FORMAT(6X,'BURN TIME',9X,F12.2)
 420   FORMAT(/4X,'ORBITER  PARAMETERS')
 480   FORMAT(6X,'EXT. TANK WT (DRY)',F12.2)
 530   FORMAT(6X,'PROPELLANT WEIGHT',4X,'SRB PROP',4X,
      &'1STSTAG PROP',2X,'2NDSTAG PROP')
 540   FORMAT(24X,F12.2,2X,F12.2,2X,F12.2)
 560   FORMAT(6X,'3STG OAMS ASC PROP',F12.2)
 610   FORMAT(6X,'ISP',15X,F12.2)
 620   FORMAT(6X,'VACUUM THRUST',5X,F12.2,2X,'THRUST/ENG'
      &,F8.0,2X,'# ENG',F4.0)
 630   FORMAT(/4X,'GGLOW=',2X,F12.3)
 640   FORMAT(4X,'BGLOW=',2X,F12.3)
 650   FORMAT(4X,'GLOW =',2X,F12.3)
 660   FORMAT(4X,'TOTAL=',2X,F12.3,2X,
      &'(INCLUDES HOLD DOWN PROP=',F7.1,')')
 670   FORMAT(4X,'T/W1 =',2X,F12.3)
 680   FORMAT(4X,'T/W2 =',2X,F12.3)
 690   FORMAT(/4X,'IDEAL STAGING VELOCITY',5X,F12.2
      &,3X,'F.P.R. DELTA V',F9.2)
 700   FORMAT(4X,'REAL STAGING VELOCITY',6X,F12.2
      &,3X,'F.P.R. PROP',3X,F9.2)
 710   FORMAT(4X,'NOMINAL REQUIRED VELOCITY',2X,F12.2)
 720   FORMAT(4X,'TOTAL VELOCITY(',F5.4,'% FPR)',1X,F12.2)
 730   FORMAT(4X,'TOTAL VEL(CAL)(',F5.4,'% FPR)',1X,F12.2)
 740   FORMAT(4X,'ORBITER VELOCITY(CALC)',5X,F12.2)
 750   FORMAT(4X,'ORBITER VELOCITY(ACT)',6X,F12.2)
 490   FORMAT(6X,'(COMMON BULKHEAD)')
 500   FORMAT(6X,'(SEPARATE BULKHEAD)')
 510   FORMAT(6X,'ALT. SEP. BULKHEAD)')
       IF(LONGP.EQ.0.0) GO TO 9999
       IF(FIX899.GT.0.0.OR.SIMPBB.GT.0.0
      1.AND.LONGP.GT.0.0) GO TO 30
       WRITE(108,1000) SRML,SRMD
       WRITE(108,1010) BASSRM
       WRITE(108,1020) WCASE,WJOINT,WNOZZ,WTTER,WINST,WIGN,BSRMC
       WRITE(108,1030) WRECOV
       WRITE(108,1040) PWPAR,PWPI,PWRR,PWRP,PWWR,SRMRC
       WRITE(108,1050) SRMISS
       WRITE(108,1060) PWFS,PWASLS,PWAS,PWNF,PWTN,PWAV,WNCTPS,SRMIC
       WRITE(108,1065)
       WRITE(108,1070) PBOSLJ,UNCERT
       WRITE(108,1065)
       WRITE(108,1080) PDRYWT,EXPINS
       WRITE(108,1065)
       WRITE(108,1090) PBBOWT,PPROPB
       WRITE(108,1065)
       WRITE(108,1100) PGROSS
       WRITE(108,1110) LAMB
       WRITE(108,1120) NOENGB
 C
 C     DETAILED SRM WEIGHT STATEMENT
 C
 1000 FORMAT('1',15X,'SOLID ROCKET MOTOR (SRM) WEIGHT SUMMARY'
     1,/,30X,'(PER SRM)',//,28X,'LENGTH=',2X,F5.0,//,28X,
     2'DIAMETER=',F5.0)
 1010 FORMAT(/,6X,'BASIC SRM WEIGHT',16X,'(',F9.0,')')
 1020 FORMAT(9X,'CASE WEIGHT',19X,F9.0,
     1/,9X,'JOINT WEIGHT',18X,F9.0,
```

```
     2/,9X,'NOZZLE WEIGHT',17X,F9.0,
     3/,9X,'THRUST TERM WT.',15X,F9.0,
     4/,9X,'INSULATION WEIGHT',13X,F9.0,
     5/,9X,'IGNITER WEIGHT',16X,F9.0,
     6/,9X,'BASIC SRM WT.CON.',13X,F9.0)
1030 FORMAT(6X,'SRM RECOVERY WEIGHT',13X,'(',F9.0,')')
1040 FORMAT(9X,'PARACHUTE WEIGHT',14X,F9.0,
     1/,9X,'PARACHUTE INSTAL.',13X,F9.0,
     2/,9X,'RETRO ROCKET',18X,F9.0,
     3/,9X,'PROPELLANT WEIGHT',13X,F9.0,
     4/,9X,'WATER REC. HWD.',15X,F9.0,
     5/,9X,'SRM REC.WT.CONST.',13X,F9.0)
1050 FORMAT(6X,'SRM INTERSTAGE STRU.',12X,'(',F9.0,')')
1060 FORMAT(9X,'FORWARD SKIRT',17X,F9.0,
     1/,9X,'AFT SKIRT STRUCT',14X,F9.0,
     2/,9X,'ATTACH/SEP STRUCT',13X,F9.0,
     3/,9X,'NOSE FAIRING',18X,F9.0,
     4/,9X,'TUNNEL WEIGHT',17X,F9.0,
     5/,9X,'AVIONICS WEIGHT',15X,F9.0,
     6/,9X,'TPS WEIGHT',20X,F9.0,
     7/,9X,'SRM INTERS.CONST.',13X,F9.0)
1065 FORMAT(38X,'- - - - - - -')
1070 FORMAT(6X,'SUBTOTAL DRY WEIGHT',14X,F9.0,
     1/,6X,'GROWTH UNCERTAINTY',15X,F9.0)
1080 FORMAT(6X,'DRY WEIGHT',23X,F9.0,
     1/,6X,'EXPENDABLE PROP.',17X,F9.0)
1090 FORMAT(6X,'BURN OUT WEIGHT',18X,F9.0,
     1/,6X,'USABLE PROP WEIGHT',15X,F9.0)
1100 FORMAT(6X,'TOTAL GROSS WEIGHT',14X,'(',F9.0,')')
1110 FORMAT(/,6X,'LAMBDA=WPROP/WGROSS=',F6.5)
1120 FORMAT(6X,'TOTAL  NO  OF SRMS =',F2.0)
C
C     DETAILED ORBITER WEIGHT STATEMENT
C
  30  IF(FIXORB.GT.0.0.AND.LONGP.GT.0.0) GO TO 9999
      IF(FIXWOR.GT.0.0.AND.LONGP.GT.0.0) GO TO 40
      WRITE(108,2000)
      WRITE(108,2010) WSG,WWT
      WRITE(108,2020) WBSTR,WTORBE,LEW,WTE,GPROV,GPROV
     1,WAIL,WAS,WADR,WAH,WAP,PWINGK
      WRITE(108,2030) TSG,TAIL
      WRITE(108,2040) TBSTR,TTORQB,TLE,WRUD,WRS,WRDR,WRH
     1,WRP,PTAILK
      WRITE(108,2050) G37
      WRITE(108,2060)
      WRITE(108,2070) G1,G11,G25,G10,G27,G12,G26,G15,G2,G3
     1,G6,G16,WTOR30,G17,G32,G34,G8,G18,G35,G19,G22,G7,G23,G33
     2,G9,G24,G36
      WRITE(108,2080) TOTTPS
      WRITE(108,2090) TWGWT,WGWT,WGLEWT,TTWT,TWT,TLEWT,BTPSWT
     1,BLBTPS,BASEWT,IBTWT,PTPSCN,MCSWT,LDTWT,PROWT,PPPC,PHYC
     2,SCWT
      WRITE(108,2100) LNDDK
      WRITE(108,2110) NGEAR,NG1,NG2,NG3,MGEAR,MG1,MG2,MG3
     1,AXGEAR,AX1,AX2,AX3,LNDDKK
      WRITE(108,2120) TAPROP
      WRITE(108,2130) ENGPAC,ENG,TVC,CONTR,PRPUTL,PROSYS,FAD
     1,PRES,CHIL,PREVAL,FEEDS,DISC,MISC
      WRITE(108,2140) TABPRO
      WRITE(108,2150) WTAUX
      WRITE(108,2160) WTACS,ACSENG,ACSSYS,WTACTK,ACSMOD,WTOMS
```

```
      1,OMSENG,PROPSY,WTOMTK,MODULE
       WRITE(108,2170) PPWR
       WRITE(108,2180) ELEC
       WRITE(108,2190) HYDR
       WRITE(108,2200) SURFC
       WRITE(108,2210) AVIONO
       WRITE(108,2220) ECLSS
       WRITE(108,2230) PPROV
       WRITE(108,2235) ORBMIS
       WRITE(108,2240) OUNCWT,ODRYWT
       WRITE(108,2245)
       WRITE(108,2250) ODRYWT,PERSON,ORESO,PLOADU,OINWT,ORESV
      1,PRIFL,ACSPRO,WOPROP,OTRAP,OLBWT,PLBADD,OLANWT,PLBADU,OINJWT
       WRITE(108,2255) R1,R2,RL
 2000 FORMAT('1',24X,'ORBITER WEIGHT SUMMARY')
 2010 FORMAT(/,4X,'WING  GROUP(AREA=',F5.0,')',23X,
      1'(',F9.0,')')
 2020 FORMAT(6X,'BASIC STRUCTURE',18X,F9.0,
      1/,9X,'TORQUE BOX EXPOSE',2X,F9.0,
      3/,9X,'LEADING EDGE',7X,F9.0,
      4/,9X,'TRAILING EDGE',6X,F9.0,
      5/,6X,'SECONDARY STRUCTURE',14X,F9.0,
      6/,9X,'M.L.G. PROVISIONS',2X,F9.0,
      7/,6X,'CONTROL SURFACE',18X,F9.0,
      8/,9X,'SHELL',14X,F9.0,
      9/,9X,'DRIVE RIB',10X,F9.0,
      A/,9X,'HINGE',14X,F9.0,
      5/,9X,'ATTACH',13X,F9.0,
      A/,6X,'WING WEIGHT CONSTANT',13X,F9.0)
 2030 FORMAT(4X,'TAIL GROUP(AREA=',F5.0,')',23X,
      1'(',F9.0,')')
 2040 FORMAT(6X,'BASIC STRUCTURE',18X,F9.0,
      1/,9X,'TORQUE BOX',9X,F9.0,
      2/,9X,'LEADING EDGE',7X,F9.0,
      3/,6X,'CONTROL SURFACE',18X,F9.0,
      4/,9X,'SHELL',14X,F9.0,
      5/,9X,'DRIVE RIB',10X,F9.0,
      6/,9X,'HINGE',14X,F9.0,
      7/,9X,'ATTACH',13X,F9.0,
      8/,6X,'TAIL WEIGHT CONSTANT',13X,F9.0)
 2050 FORMAT(4X,'BODY GROUP',35X,'(',F9.0,')')
 2060 FORMAT(31X,'FWD',8X,'CTR',8X,'AFT')
 2070 FORMAT(6X,'BASIC STRUCTURE',
      1/,9X,'SIDEWALLS',10X,F9.0,2X,F9.0,1X,F9.0,
      2/,9X,'LONGERONS',21X,F9.0,1X,F9.0,
      3/,9X,'FRAMES',24X,F9.0,1X,F9.0,
      4/,9X,'BULKHEADS',21X,F9.0,
      5/,9X,'CREW CPT. PROV.',4X,F9.0,
      6/,9X,'WINDSHIELD PROV.',3X,F9.0,
      7/,9X,'NOSE WHL.WEL PROV',2X,F9.0,
      8/,9X,'PAYLOAD REACTION',14X,F9.0,
      A/,9X,'WING CARRY THRU.',14X,F9.0,
      A/,9X,'WING SHEAR PROV.',14X,F9.0,
      1/,9X,'THRUST STRUCTURE',25X,F9.0,
      2/,9X,'TAIL PROV.',31X,F9.0,
      3/,6X,'SUB TOTAL',13X,F9.0,2X,F9.0,2X,F9.0,
      4/,6X,'SECONDARY STRUCTURE',
      5/,9X,'CARGO DOOR SHELL',14X,F9.0,
      6/,9X,'CARGO DOOR MECH.',14X,F9.0,
      7/,6X,'MISCELLANEOUS WTS.',4X,F9.0,2X,F9.0,2X,F9.0,
      8/,6X,'TOTAL',17X,F9.0,2X,F9.0,2X,F9.0)
```

```
2080 FORMAT(4X,'INDUCED ENVIRON. PROT.',23X,
    1'(',F9.0,')')
2090 FORMAT(6X,'WING',29X,F9.0,
    1/,9X,'SURFACE PANELS',5X,F9.0,
    2/,9X,'LEADING EDGE',7X,F9.0,
    3/,6X,'TAIL',29X,F9.0,
    4/,9X,'SURFACE PANELS',5X,F9.0,
    5/,9X,'LEADING EDGE',7X,F9.0,
    6/,6X,'BODY',29X,F9.0,
    7/,9X,'BODY PANELS',8X,F9.0,
    8/,9X,'BASE',15X,F9.0,
    9/,9X,'INTERNAL TPS',7X,F9.0,
    $/,9X,'BODY CNST TPS WT.',2X,F9.0,
    1/,6X,'MIS CONT. SURFACE',16X,F9.0,
    2/,6X,'LAND + DOCKING',19X,F9.0,
    3/,6X,'PROPULSION',23X,F9.0,
    4/,6X,'PRIME POWER',22X,F9.0,
    5/,6X,'HYDRAULICS',23X,F9.0,
    6/,6X,'SURFACE CONTROLS',17X,F9.0)
2100 FORMAT('1',3X,'LANDING & DOCKING',28X,'(',F9.0,')')
2110 FORMAT(6X,'NOSE GEAR',24X,F9.0,
    1/,9X,'ROLL GEAR',10X,F9.0,
    2/,9X,'STRUCTURE',10X,F9.0,
    3/,9X,'CONTROLS',11X,F9.0,
    4/,6X,'MAIN GEAR',24X,F9.0,
    5/,9X,'ROLL GEAR',10X,F9.0,
    6/,9X,'STRUCTURE',10X,F9.0,
    7/,9X,'CONTROLS',11X,F9.0,
    8/,6X,'AUXILIARY SYSTEMS',16X,F9.0,
    9/,9X,'DECELERATION SYS',3X,F9.0,
    1/,9X,'SEPARATION SYS',5X,F9.0,
    2/,9X,'HANDLING & MANIP',3X,F9.0,
    3/,6X,'MISCELLANEOUS',20X,F9.0)
2120 FORMAT(4X,'PROPULSION MAIN ASCENT',23X,'(',F9.0,')')
2130 FORMAT(6X,'ENGINES+ACESSORIES',15X,F9.0,
    1/,9X,'ENGINES',12X,F9.0,
    2/,9X,'GIMBAL SYSTEM',6X,F9.0,
    3/,9X,'CONTROLS',11X,F9.0,
    4/,9X,'PROPELLANT UTILIZ',2X,F9.0,
    5/,6X,'PROPELLANT SYSTEM',16X,F9.0,
    6/,9X,'FILL & DRAIN',7X,F9.0,
    7/,9X,'PRESSURIZATION',5X,F9.0,
    8/,9X,'CHILL DUMP LINES',3X,F9.0,
    9/,9X,'PRE VALVES',9X,F9.0,
    $/,9X,'FEED SYSTEM',8X,F9.0,
    1/,9X,'DISCONNECTS',8X,F9.0,
    2/,9X,'MISCELLANEOUS',6X,F9.0)
2140 FORMAT(4X,'PROPULSION AIR BREATH',24X,'(',F9.0,')')
2150 FORMAT(4X,'PROPULSION AUXILIARY',25X,
    1'(',F9.0,')')
2160 FORMAT(6X,'ACS SYSTEM',23X,F9.0,
    1/,9X,'THRUSTERS',10X,F9.0,
    2/,9X,'PROP. SYSTEM',7X,F9.0,
    3/,9X,'TANK',15X,F9.0,
    4/,9X,'MODULE',13X,F9.0,
    5/,6X,'OAMS SYSTEM',22X,F9.0,
    6/,9X,'THRUSTERS',10X,F9.0,
    7/,9X,'PROP. SYSTEM',7X,F9.0,
    8/,9X,'TANK',15X,F9.0,
    9/,9X,'MODULE',13X,F9.0)
2170 FORMAT(4X,'PRIME POWER',34X,'(',F9.0,')')
```

```
2180 FORMAT(4X,'ELECTRICAL',35X,'(',F9.0,')')
2190 FORMAT(4X,'HYDRAULIC',36X,'(',F9.0,')')
2200 FORMAT(4X,'SURFACE CONTROLS',29X,'(',F9.0,')')
2210 FORMAT(4X,'AVIONICS',37X,'(',F9.0,')')
2220 FORMAT(4X,'ENVIRONMENTAL CONTROL',24X,'(',F9.0,')')
2230 FORMAT(4X,'PERSONNEL PROVISIONS',25X,'(',F9.0,')')
2235 FORMAT(4X,'MISCELLANEOUS',32X,'(',F9.0,')')
2240 FORMAT(4X,'GROWTH/UNCERTAINTY',27X,'(',F9.0,')'
    1,/,49X,'- - - - - -',/,4X,
    2'DRY WEIGHT',34X,'((',F9.0,'))')
2245 FORMAT(//,'1',24X,'ORBITER MISSION HISTORY')
2250 FORMAT(/,4X,'DRY WEIGHT',13X,'(',F9.0,')',
    1/,6X,'PERSONNEL',13X,F9.0,
    2/,6X,'ORB RESD PROP WT.',5X,F9.0,
    3/,6X,'PAYLOAD UP',12X,F9.0,
    4/,4X,'INERT WEIGHT',11X,'(',F9.0,')',
    5/,6X,'ORB RESV PROP WT.',5X,F9.0,
    6/,6X,'ORB INFLIGHT LOSSES',3X,F9.0,
    7/,6X,'ACS PROP WT',11X,F9.0,
    8/,6X,'OAMS PROP WT',10X,F9.0,
    9/,6X,'ORB TRAPED PROP WT',4X,F9.0,
    &/,4X,'GROSS WT(ORB-ONLY)',5X,'(',F9.0,')',
    1//,6X,'(LAND WT PAY=',F6.0,')',
    2/,4X,'LANDING WEIGHT',9X,'(',F9.0,')',
    3/,6X,'(INJE WT PAY=',F6.0,')',
    4/,4X,'INJECTED WEIGHT',8X,'(',F9.0,')')
2255 FORMAT(//,6X,'INTER STAGE REACTIONS'
    1,/,3X,'R1=',1X,F9.0
    2,/,3X,'R2=',1X,F9.0
    3,/,3X,'RL=',1X,F9.0)
40    IF(SIMPTK.GT.0.0.AND.LONGP.GT.0.0) GO TO 9999
C
C     DETAILED EXTERNAL TANK WEIGHT STATEMENT
C
      WRITE(108,3000)
      IF(BLKHD.EQ.1.) GO TO 45
      GO TO 50
45    IF(EXTHO.GT.EXTHH) GO TO 55
      WRITE(108,3010)
      GO TO 60
50    IF(BLKHD.EQ.3.) GO TO 65
      GO TO 70
65    WRITE(108,3050)
70    IF(EXTHO.GT.EXTHH) GO TO 75
      WRITE(108,3030)
      GO TO 60
55    WRITE(108,3020)
      GO TO 60
75    WRITE(108,3040)
60    WRITE(108,3060) EXTL,EXTD
      WRITE(108,3070)BODGRP,TOTPS,FWDTK,FAIRT,FWDBLF,FCCTPS,
     &CONSCT,TPSIN,CYLSCT,ACYDM,AFTBLF,WINT,PROSY,AFTNK,
     &FEDSYS,FWDBLA,PRSVNT
      WRITE(108,3080)AFTCYL,SUMP,AFTBLA,PNPU,TWINT,NOSFAR,
     &AVIONT,UMBPNL,WRETRO,TUNNEL,MISCT,BAFF,SUBDRY
      WRITE(108,3090)SUBDRY,GU,DRYWT,RESIDT,UNDRAN,FEEDTR,
     &PRSURT,FBIAS,INERT,PROPOT,GROSSW,TLAMB
3000 FORMAT('1',18X,'EXTERNAL  TANK WEIGHT SUMMARY')
3010 FORMAT(20X,'COMMON  BULKHEAD-LOX FWD')
3020 FORMAT(20X,'COMMON  BULKHEAD-LOX AFT')
3030 FORMAT(20X,'SEPARATE  BULKHEAD-LOX FWD')
```

```
 3040 FØRMAT(20X,'SEPARATE  BULKHEAD-LØX AFT')
 3050 FØRMAT(10X,'ALTERNATE  FWD SECTIØN(WITHØUT NØSE FAIRING)')
 3060 FØRMAT(/,25X,'LENGTH=',2X,F5.0,//,25X,'DIAMETER=',F5.0)
 3070 FØRMAT(/23X,'WEIGHT',26X,'WEIGHT'/24X,'-LB.',28X,'-LB.'//
     &2X,'BØDY GRØUP',9X,'|',F8.0,'  ',
     &2X,'IND. ENVIRN. PRØT. |',F8.0,'  '/
     &3X,'FWD TANK',10X,'(',F8.0,' )',
     &3X,'NØSE FAIRING',7X,F8.0/
     &4X,'FWD BULKHEAD',6X,F8.0,
     &5X,'FWD CØNE & CYL.',4X,F8.0/
     &4X,'CØNICAL SECTIØN   ',F8.0,
     &5X,'INTER TANK',9X,F8.0/
     &4X,'CYLINDRICAL SECT. ',F8.0,
     &5X,'AFT CYL & DØME',5X,F8.0/
     &4X,'AFT BULKHEAD',6X,F8.0/
     &3X,'INTER TANK SECT.  (',F8.0,' )',
     &2X,'PRØPELLANT SYSTEMS |',F8.0,'  '/
     &3X,'AFT TANK',10X,'(',F8.0,' )',
     &3X,'FEED SYSTEM',8X,F8.0/
     &4X,'FWD BULKHEAD',6X,F8.0,
     &5X,'PRES. AND VENT',5X,F8.0)
 3080 FØRMAT(4X,'CYLINDRICAL SECT. ',F8.0,
     &5X,'SUMPS & VØRTEX CTL ',F8.0/
     &4X,'AFT BULKHEAD',6X,F8.0,
     &5X,'PNEUMATIC & PU SYS ',F8.0/
     &3X,'SRB/BSTR/TANK ATT.(',F8.0,' )'/
     &3X,'NØSE FAIRING',6X,'(',F8.0,' )',
     &2X,'AVIØNICS',11X,'|',F8.0,'  '/
     &3X,'UMBILICAL PANEL   (',F8.0,' )',
     &2X,'DEØRBIT SYSTEM',5X,'|',F8.0,'  '/
     &3X,'TUNNEL',12X,'(',F8.0,' )',
     &2X,'MISCELLANEØUS',6X,'|',F8.0,'  '/
     &3X,'BAFFLES-LØX',7X,'(',F8.0,' )',
     &' ----------------------------------'/
     &34X,'SUBTØTAL DRY WEIGHT',F9.0/)
 3090 FØRMAT(19X,'SUBTØTAL  DRY WEIGHT',F9.0//
     &19X,'GRØWTH/UNCERTAINTY |',F8.0,'  '/
     &18X,'----------------------------------'/
     &19X,'DRY WEIGHT',9X,F9.0//
     &19X,'RESIDUAL PRØPELLANT|',F8.0,'  '/
     &20X,'TANK UNDRAINABLE',3X,F8.0/
     &20X,'FEEDLINE TRAPPED',3X,F8.0/
     &20X,'PRESSURANT',9X,F8.0/
     &20X,'PU BIAS',12X,F8.0/
     &18X,'----------------------------------'/
     &19X,'INERT WEIGHT',7X,F9.0//
     &19X,'USABLE PRØPELLANT |',F9.0,'  '/
     &18X,'----------------------------------'/
     &19X,'TØTAL GRØSS WEIGHT ',F9.0//
     &19X,'LAMBDA=WPRØP/WGRØSS=',F6.4)
 9999 CØNTINUE
      RETURN
      END
```