AFFDL-TR-79-3105
VOLUME II, PART II

EASY-ACLS DYNAMIC ANALYSIS

VOLUME II Component Computer Programs

M. K. Wahi
G. S. Duleba
P. R. Perkins

BOEING MILITARY AIRPLANE DEVELOPMENT
BOEING AEROSPACE COMPANY
P.O. BOX 3999, SEATTLE, WA 98124

SEPTEMBER 1979

TECHNICAL REPORT AFFDL-TR-3105 Volume II, Part II

Final Report for Period April 1977 to June 1979

Approved for public release; distribution unlimited.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

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

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

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

DAVID L. FISCHER, 1LT, USAF
Project Engineer

HOWELL K. BREWER
Chief, Mechanical Branch
Vehicle Equipment Division

FOR THE COMMANDER

A. B. NUTT
Director, Vehicle Equipment Division
Air Force Flight Dynamics Laboratories/FEM

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

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

AIR FORCE/56780/14 January 1980 — 100

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFFDL-TR-79-3105, Vol. II, Part II
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): EASY ACLS DYNAMIC ANALYSIS Volume II Component Computer Programs
5. TYPE OF REPORT & PERIOD COVERED: FINAL APRIL 1977 - JUNE 1979
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): M. K. WAHI, G. S. DULEBA, P. R. PERKINS
8. CONTRACT OR GRANT NUMBER(s): F33615-77-C-3054
9. PERFORMING ORGANIZATION NAME AND ADDRESS: BOEING MILITARY AIRPLANE DEVELOPMENT, BOEING AEROSPACE COMPANY, P.O. BOX 3999, SEATTLE, WA 98124
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project 2402, Task 240201, Work Unit 24020112
11. CONTROLLING OFFICE NAME AND ADDRESS: AIR FORCE FLIGHT DYNAMICS LABORATORY (AFFDL/FEM), AIR FORCE WRIGHT AERONAUTICAL LABORATORIES, WRIGHT-PATTERSON AFB, OH 45433
12. REPORT DATE: September 1979
13. NUMBER OF PAGES: 393
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Approved For Public Release; Distribution Unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): EASY, Air Cushion Landing Systems, FORTRAN, Functions, Listings, Subroutines
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

This volume contains a detailed description of the component subroutines and of other standard functions/subroutines which are used in the EASY dynamic analysis program. Macro-flow charts and some micro-flowcharts and listings are included. Sample output is included where appropriate.

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This report presents results of work conducted by the Boeing Company, Seattle, Washington, under Air Force Contract F33615-77-C-3054, "Application of the EASY Dynamic Program to the Analysis of Air Cushion Systems on Aircraft", during the period from 15 April 1977 to 1 June 1979. This contract was conducted under the sponsorship of the Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio with Mr. Peters Skele and Lt D. L. Fischer as project engineers.

This report is comprised of three volumes.

Volume I - Component Mathematical Models
Volume II - Component Computer Programs
(Parts I and II)
Volume III - Description of Simulations

In addition, a User's Manual (Reference 1) has been written to provide a concise reference for day to day usage.

The results presented were developed by the Boeing Aerospace Company. The program managers were A. J. P. Lloyd, H. H. Straub and J. R. Kilner. The principal investigators were Mr. K. Wahi, G. S. Duleba, J. R. Kilner and P. R. Perkins.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

LIST OF TABLES (Concluded)

SECTION VI

SUBROUTINE LISTINGS

This section contains the listings of all the EASY ACLS subroutines. The listings are ordered alphabetically according to subroutine name.

Table 128: LISTING FOR SUBROUTINE AB

```
CAB
      SUBROUTINE AB(ABL,XYZ,DSM,IAL,REL,ZTR,
     1  FXT,FYT,FZT,TXT,TYT,TZT,PTR,PTRD,IPTR,PTL,PTLD,IPTL,
     2 VTR,VTRD,IVTR,VTL,VTLD,IVTL,WAR,WAL,CPT,
     3 ROL,PIT,YAW,X,ALT,U,V,W,PA,VU,EPC,WTR,TTR,WTL,TTL,ANE,
     4 ANSET,ANPTS,BST,WLT,CD1,CD2,CDA,BSCG,WLCG,TAU,P,Q,R,
     5 AMODE,ANR,DL,H,DMP)
C
C   VERSION 1.                               APRIL 10,1978
C
C   PURPOSE - MODEL THE PNEUMATIC, GEOMETRIC AND RESULTING
C           - DYNAMIC RESPONSE OF A INELASTIC AIR BAG
C           - SUSPENSION SYSTEM.
C
C   METHOD  - SEE AIR BAG DOCUMENTATION, VOL I, FINAL REPORT
C
      DIMENSION A(6),B(6),LO(6),XA(12),YA(12),ZA(12),D(12),S(12),
     *    IS(12),AP(12),LP(12),LH(12),MU(12),REL(1)
      DIMENSION XBA(12),YBA(12),ZBA(12),GA(6),GB(6)
      DIMENSION ABL(1),XYZ(1),DSM(1),IAL(1)
      DIMENSION XMU(4),ZTR(1)
C
      REAL LO,LP,LH,MU,L3,L1,IAL
C
      DIMENSION AZO(6),AYO(10,10,6),AL1(10,10,6),AL3(10,10,6),
     *  AAS(10,10,6),BYO(10,10,6),BL1(10,10,6),BL3(10,10,6),
     *  BAS(10,10,6)
      COMMON/CIO/IREAD,IWRITE,IDIAG
      COMMON/CTIME/TIME
      COMMON/CXDOT/XD(1)
      COMMON/CDIFS/JST,KIN,TP
C
      DATA RG,NA,TEST2/53.34,10,0./
C
C
C   CALL SEQUENCE
C
C ***** INPUT TABLES OR DATA ARRAYS *****
C     A,B,LO          -ARRAYS OF AIR BAG ELEMENT DIMENSIONS; ATTACH
C                     -POINT SPACING,ATTACH POINT HEIGHT AND
C                     -MEMBRANE CIRCUMFERENCE RESPECTIVELY, INCHES
C     GA,GB           -ARRAYS OF ANGULAR POSITION OF FUSELAGE
C                     -CONSTRAINTS ON MEMBRANE SHAPE
C
C     XA,YA,ZA        -ARRAYS OF COORDINATES OF AIR BAG ELEMENT
C                     -ATTACH POINT, INCHES
C
C     D               -ARRAY OF ELEMENT WIDTH, INCHES
C     S               -ARRAY OF ELEMENT SCALING FACTORS
C     MU              -ARRAY OF ELEMENT COEFFICIENTS OF FRICTION
C                     -IN X AND Y AXIS RESPECTIVELY
C
C     IS              -ARRAY OF PARAMETER SET NUMBERS ASSOCIATED
C                     -WITH EACH ELEMENT
C     AP              -ARRAY OF ORIFICE AREA PER UNIT AREA
C                     -(OR POROSITY) OF BAG SURFACE
```

```
C     LP                -ARRAY OF CIRCUMFERENTIAL DISTANCE FROM ATTACH
C                       -POINT B TO BEGINNING OF PERFORATIONS, INCHES
C     LH                -ARRAY OF WIDTH OF PERFORATED AREA, INCHES
C
C     REL               -RELIEF VALVE AREA OPENING AS A FUNCTION OF
C                       -BAG PRESSURE ,ONE DIM.TABLE,SQ.IN VS PSIG
C
C    ***** OUTPUTS *****
C     FXT,FYT,FZT       -X,Y,Z AXIS,AXIAL,LATERAL AND VERTICAL FORCE
C                       -SUMMATION TERMS, LBS
C     TXT,TYT,TZT       -X,Y,Z AXIS SUMMATION TERMS FOR ROLL,PITCH,
C                       -AND YAW MOMENTS, FT-LB.
C     PTR,PTRD,IPTR -RIGHT AIR BAG PRESSURE,RATE,INT CONTROL, PSIA
C     PTL,PTLD,IPTL -LEFT AIR BAG PRESSURE,RATE,INT CONTROL, PSIA
C     VTR,VTRD,IVTR -RIGHT AIR BAG VOLUME,RATE,INT CONTROL, CU FT
C     VTL,VTLD,IVTL -LEFT AIR BAG VOLUME,RATE,INT CONTROL, CU FT
C     WAR               -RIGHT AIR FLOW RATE,AIR BAG TO ATMOSPHERE, LB/MIN
C     WAL               -LEFT AIR FLOW RATE,AIR BAG TO ATMOSPHERE, LB/MIN
C
C    *****  INPUTS  *****
C     ROL,PIT,YAW       -ROLL,PITCH,YAW EULER ANGLES, DEG
C     X,ALT             -X,Z EARTH AXIS POSITIONS
C     U,V,W             -X,Y,Z BODY AXIS LINEAR VELOCITIES, FT/SEC
C     PA                -AMBIENT PRESSURE, PSIA
C     VU                -BREAK POINT IN MU-VELOCITY CURVE, IN/SEC
C     WTR               -SUPPLY RIGHT AIR FLOW RATE TO AIR BAG, LB/MIN
C     TTR               -TEMPERATURE OF WTR AIR, DEGR
C     WTL               -SUPPLY LEFT AIR FLOW RATE TO AIR BAG, LB/MIN
C     TTL               -TEMPERATURE OF WTL AIR, DEGR
C     NE(ANE)           -NUMBER OF AIR BAG ELEMENTS
C                       -IF ANE.LT.0 MODEL IS SYMMETRIC ABOUT ROLL AXIS
C     NSET              -NUMBER OF ELEMENT SHAPE PARAMETER SETS
C     CDA               - DISCHARGE COEFF. FOR FLOW THROUGH
C                       -RELIEF VALVE
C     NPTS              -NO. OF ELEMENTS IN A ROW OR COLUMN IN THE
C                       - PARAMETER SET
C     BST,WLT           -BODY STATION AND WATER LINE OF AIR BAG AXIS, INCHES
C     CD1               -ORIFICE DISCHARGE COEFFICIENT FOR FREE
C                       -PORTION OF AIR BAG
C     CD2               -ORIFICE DISCHARGE COEFFICIENT FOR AIR BAG AREA
C                       -IN CONTACT WITH THE GROUND
C     BSCG,WLCG         -BODY STATION AND WATER LINE OF C.G., INCHES
C     TAU               -TIME CONSTANT FOR AIR BAG VOLUME
C                       -RATE OF CHANGE, SEC
C     P,Q,R             -X,Y,Z BODY AXIS ANGULAR VELOCITIES,DEG/SEC
C     AMODE,ANR,DL,H-TERRAIN MODEL PARAMETERS,SEE FUNCTION TERRA
C     DMP               -DAMPING COEFFICIENT AS A FUNCTION OF
C                       -FLATTENED AREA, LB-SEC/IN./SQ IN.
C     EPC               -PRINT CONTROL,  EPC=1. PRINT ELEMENT VARIABLE
C                       -VALUES EVERY PRINT INTERVAL
C
C *** CONSTANTS ***
C     RG=53.34          -GAS CONSTANT FOR AIR, FT-LB/LB/DEGR
C
C    WRITTEN BY    J.R.KILNER
C
```

```
C   INITIAL CALCULATIONS AND ELEMENT SECTION PROPERTY
C   PARAMETER DATA CALCULATED AT TIME=0
C   IF INPUT DATA ARE UPDATED
C
      CALL SECOND(CPT)
      IF(TIME.NE.0.)GOTO 11
C
      NSET=ANSET
      NE=ABS(ANE)
      NPTS=ANPTS
      NSET2=5*NSET+3
      TEST=0.
      DO 4 I=4,NSET2
4     TEST=TEST+ABL(I)
C
      NE3=NE*3+3
      NE4=NE*4+3
      DO 6 I=4,NE3
6     TEST=TEST+XYZ(I)+DSM(I)
      DO 7 I=4,NE4
7     TEST=TEST+IAL(I)
C
      IF(TEST.EQ.TEST2)GO TO 11
      TEST2=TEST
C
      MS=2
      IF(ANE.LT.0.)MS=1
      CNT=CNTL=CNTF=0.
C
      DO 10 I=1,NSET
      I5=I*5
      A(I)=ABL(I5-1)
      B(I)=ABL(I5)
      LO(I)=ABL(I5+1)
      GA(I)=ABL(I5+2)
      GB(I)=ABL(I5+3)
 10   CONTINUE
C
      DO 20 I=1,NE
      I3=I*3
      I4=I*4
      XA(I)=XYZ(I3+1)
      YA(I)=XYZ(I3+2)
      ZA(I)=XYZ(I3+3)
C
      D(I)=DSM(I3+1)
      S(I)=DSM(I3+2)
      MU(I)=DSM(I3+3)
C
      IS(I)=IAL(I4)
      AP(I)=IAL(I4+1)
      LP(I)=IAL(I4+2)
      LH(I)=IAL(I4+3)
C
      XBA(I)= BSCG-BST+XA(I)
      YBA(I)= YA(I)
```

```
      ZBA(I)= WLCG-WLT+ZA(I)
 20   CONTINUE
C
      XMU(1)= 0.
      XMU(2)= VU
      XMU(3)= 0.
      XMU(4)= 1.
C
C
      WRITE(IWRITE,6000)
      WRITE(IWRITE,6002)(I,XA(I),YA(I),ZA(I),D(I),S(I),
     *  MU(I),IS(I),AP(I),LP(I),LH(I),I=1,NE)
6000  FORMAT(1H1,34H***** AIR BAG PARAMETER DATA  ****,14(5H*****)/
     * 3X,31H*** FOR TWIN SYMMETRIC BAGS ***///
     * 37H  ELEMENT       XA          YA          ZA,
     * 8X,*D       S*,
     * 6X,* MU     IS     AP          LP      LH*/)
6002  FORMAT(4X,I2,2X,3F10.2,F10.2,2F8.3,I4,F9.5,2F7.1/)
C
      CALL ICB(NSET,NPTS,A,B,L0,GA,GB,DMU,
     *          AZ0,AY0,AL1,AL3,AAS,BY0,BL1,BL3,BAS)
C
 11   CONTINUE
C
      IF(EPC.EQ.1.)CALL VPRINB(0,I,M,MS,NE,ZGAP,Z0,Y0,L1,L3,AS,UTY,
     *          FFXBAR,FFYBAR,FDBAR,FTBAR,CNT,CNTL,CNTF)
      CNT=CNT+1.
C
C  INITIALIZATION FOR LOOP ITERATION
C
      FXT =0.
      FYT =0.
      FZT =0.
      TXT =0.
      TYT =0.
      TZT =0.
C
      CR=COS(.01745*ROL)
      CP=COS(.01745*PIT)
      CY=COS(.01745*YAW)
      SR=SIN(.01745*ROL)
      SP=SIN(.01745*PIT)
      SY=SIN(.01745*YAW)
C
      CPCY=CP*CY
      CRSY=CR*SY
      SRSPCY=SR*SP*CY
      SYSR=SY*SR
      CYCRSP=CY*CR*SP
      CPSR=CP*SR
      CRCP=CR*CP
      SPSR=SP*SR
      SPCR=SP*CR
C
      P1=P*.01745
      Q1=Q*.01745
```

```
      R1=R*.01745
C
      U1=U*12.
      V1=V*12.
      W1=W*12.
C
C **  M=1  RIGHT AIR BAG
C **  M=2  LEFT  AIR BAG
C **  I= AIR BAG ELEMENT NUMBER  **
C      E=+1 IF M=1
C      E=-1 IF M=2
C
      DO 31 M=1,MS
      E=1.
      PT=PTR
      IF(M.EQ.1)GOTO 22
      E=-1.
      PT=PTL
C
22    VTS=0.
      AH1=0.
      AH2=0.
C
      DO 30  I=1,NE
C
C ** FS REFERS TO FREE SHAPE VALUES
C ** SUBSCRIPT U INDICATES UNSCALED VARIABLE
C
      ISI=IS(I)
      ZOFSU=AZO(ISI)
      ZOFS =S(I)*ZOFSU
      YOFS =S(I)*AYO(1,1,ISI)
C
C
      XBT=XBA(I)
      YBT=(YBA(I)+YOFS)*E
      ZBT=ZBA(I)+ZOFS
C
C  DETERMINE X AND Z POSITION OF POINT T IN EARTH COORDINATES
C
      XET=X*12.+XBT*CPCY+YBT*(SRSPCY-CRSY)+ZBT*(SYSR+CYCRSP)
      ZET=-ALT*12.-XBT*SP+YBT*CPSR+ZBT*CRCP
C
C  DETERMINE TERRAIN ELEVATION AT POINT T
C
      ZEG =TERRA(XET,AMODE,ANR,DL,H,ZTR)
C
C CALCULATE BAG-GROUND GAP HEIGHT.
C  NEGATIVE GAP IMPLIES A LOADED BAG ELEMENT
C
      ZGAP=-ZEG-ZET
      ZO=ZOFS+ZGAP
      IF(ZGAP.GT.0.) GO TO 44
C
C  LOADED SHAPES
C
```

```
      CNTL=CNTL+1.
C
C  DETERMINE VELOCITY OF POINT T RELATIVE TO EARTH
C  WITH X AND Y COMPONENTS ORIENTED TO BODY AXIS
C
      ZBT=ZBA(I)+ZO
C
      XBTD= ZBT*Q1-YBT*R1+U1
      YBTD=-ZBT*P1+XBT*R1+V1
      ZBTD= YBT*P1-XBT*Q1+W1
C
      XTD2=XBTD*CP+YBTD*SPSR+ZBTD*SPCR
      YTD2=YBTD*CR-ZBTD*SR
      ZTD=-XBTD*SP+YBTD*CPSR+ZBTD*CRCP
C
      VET=SQRT(XTD2*XTD2+YTD2*YTD2)
C
C  CALCULATE ELEMENT  FRICTION COEFFICIENTS
C
      IF(VET.EQ.0.)GOTO 24
      UTO=MU(I)*TBLU1(VET,XMU,XMU(3),1,-2)
      UTX=UTO*XTD2/VET
      UTY=UTO*YTD2/VET
      GOTO 26
24    UTX=0.
      UTY=0.
26    CONTINUE
C
C  CALCULATE ELEMENT DIMENSIONS AND AREAS FOR A
C  LOADED SHAPE
C
      ZOU=ZO/S(I)
      UT=E*UTY
      IF(UT.LT.0.)GOTO 33
      YO=S(I)*TBL2(UT,ZOU,ZOFSU,AYO,DMU,NPTS,IS(I),NA)
      L1=S(I)*TBL2(UT,ZOU,ZOFSU,AL1,DMU,NPTS,IS(I),NA)
      L3=S(I)*TBL2(UT,ZOU,ZOFSU,AL3,DMU,NPTS,IS(I),NA)
      AS=S(I)*S(I)*TBL2(UT,ZOU,ZOFSU,AAS,DMU,NPTS,IS(I),NA)
      GOTO 34
C
33    YO=S(I)*TBL2(UT,ZOU,ZOFSU,BYO,DMU,NPTS,IS(I),NA)
      L1=S(I)*TBL2(UT,ZOU,ZOFSU,BL1,DMU,NPTS,IS(I),NA)
      L3=S(I)*TBL2(UT,ZOU,ZOFSU,BL3,DMU,NPTS,IS(I),NA)
      AS=S(I)*S(I)*TBL2(UT,ZOU,ZOFSU,BAS,DMU,NPTS,IS(I),NA)
34    CONTINUE
C
C  AIR BAG GROUND REACTION
C
      AT=D(I)*L3
      FTBAR=(PT-PA)*AT
C
C  CALCULATE FRICTION FORCES
C
      FFXBAR=-UTX*FTBAR
      FFYBAR=-UTY*FTBAR
C
```

```
C   ** CALCULATE ELEMENT DAMPING FORCE
C
      FDBAR=DMP*AT*ZTD
C  ** CALCULATE FORCES AND MOMENTS
C
C
      FXT=FXT+FFXBAR
      FYT=FYT+FFYBAR
      FZT=FZT-FTBAR-FDBAR
      YBT=(YBA(I)+Y0+.5*L3)*E
      TXT=TXT+(-(FTBAR+FDBAR)*YBT-FFYBAR*ZBT)*.08333
      TYT=TYT+((FTBAR+FDBAR)*XBT+FFXBAR*ZBT)*.08333
      TZT=TZT+(FFYBAR*XBT-FFXBAR*YBT)*.08333
      GO TO 66
C
C   CALCULATE ELEMENT DIMENSION AND AREAS FOR FREE SHAPE
C
44    CONTINUE
      CNTF=CNTF+1.
      Y0=AYO(1,1,ISI)
      L1=AL1(1,1,ISI)
      L3=0.
      AS=AAS(1,1,ISI)
      UTY=0.
      FFXBAR=0.
      FFYBAR=0.
      FDBAR=0.
      FTBAR=0.
 66   CONTINUE
C
C   CALCULATE ELEMENT VOLUME
C
      VTS=VTS+D(I)*AS*.0005787
C
      IF(AP(I).NE.0.) CALL PERFB(ZGAP,L1,L3,LP(I),LH(I),
     *      D(I),AP(I),AH1,AH2)
C
      IF(EPC.EQ.1.)CALL VPRINB(1,I,M,MS,NE,ZGAP,Z0,Y0,L1,L3,AS,UTY,
     *      FFXBAR,FFYBAR,FDBAR,FTBAR,CNT,CNTL,CNTF)
 30   CONTINUE
C
      VTSL=VTS
      AH1L=AH1
      AH2L=AH2
      IF(M.EQ.2)GOTO 32
      VTSR=VTS
      AH1R=AH1
      AH2R=AH2
32    CONTINUE
31    CONTINUE
C
C  ** CALCULATE FLOW RATES, BAG VOLUME RATES OF CHANGE AND
C  ** BAG PRESSURE RATES OF CHANGE
C
C     RIGHT AIR BAG
C
```

Table 128: LISTING FOR SUBROUTINE AB (CONCLUDED)

```
      N=REL(2)
      AREL = TBLU1(PTR-PA,REL(4),REL(N+4),1,-N)
      CATA=CD1*AH1R+.6667*CD2*AH2R+CDA*AREL
      IF(IVTR.NE.0) VTRD=(VTSR-VTR)/TAU
      CALL FNFLOW(PTR,PA,TTR,CATA,1.,FN,WAR)
      IF(IPTR.NE.0)PTRD=(.0001389*RG*TTR*(WTR-WAR)-1.2*PTR*VTRD)/VTR
C
C     LEFT AIR BAG
C
C ** TEST FOR SYMMETRIC MODEL
C
      IF(ANE.LT.0.)GOTO 55
C
      N=REL(2)
      AREL = TBLU1(PTL-PA,REL(4),REL(N+4),1,-N)
      CATA=CD1*AH1L+.6667*CD2*AH2L+CDA*AREL
      IF(IVTL.NE.0) VTLD=(VTSL-VTL)/TAU
      CALL FNFLOW(PTL,PA,TTL,CATA,1.,FN,WAL)
      IF(IPTL.NE.0)PTLD=(.0001389*RG*TTL*(WTL-WAL)-1.2*PTL*VTLD)/VTL
      RETURN
C
C ** SYMMETRIC MODEL
C
55    FXT=2.*FXT
      FYT=0.
      FZT=2.*FZT
      TXT=0.
      TYT=2.*TYT
      TZT=0.
C
      IF(IVTL.NE.0)VTLD=VTRD
      WAL=WAR
      IF(IPTL.NE.0)PTLD=PTRD
C
      RETURN
      END
```

Table 129: LISTING FOR SUBROUTINE AC

```
CAC
      SUBROUTINE AC(CLT,CDT,CMT,CYT,CLBT,CNBT,CL,CD,CM,CY,CLB,CNB,AL,BE)
C
C  VERSION 1.  JUNE 1977
C
C  PURPOSE    STANDARD JINDIVIK AERODYNAMIC DATA - REF. AERO TECH MEMO
C             222. USED FOR TASK 2 TRIM AND DYNAMIC FLIGHT CONDITIONS.
C
C  METHOD     AERODYNAMIC COEF ARE INTERPOLATED FROM TABLES.
C
C  CALL SEQUENCE
C  *****  TABLES   *****
C    CLT  -COEF OF LIFT VS. ANGLE OF ATTACK
C    CDT  -COEF OF DRAG VS. COEF OF LIFT
C    CMT  -COEF OF PITCH MOMENT VS. COEF OF LIFT
C    CYT  -COEF OF SIDE FORCE VS. ANGLE OF SIDE SLIP
C    CLBT -DERIVITIVE,ROLLING MOMENT DUE TO SIDE SLIP VS ANGLE OF ATTACK
C    CNBT -DERIVITIVE,YAWING MOMENT DUE TO SIDE SLIP VS ANGLE OF ATTACK
C  *****  OUTPUTS *****
C    CL   -COEF OF LIFT
C    CD   -COEF OF DRAG FORCE
C    CM   -COEF OF PITCH MOMENT
C    CY   -COEF OF SIDE FORCE
C    CLB  -ROLL MOMENT DERIVITIVE
C    CNB  -YAW  MOMENT DERIVITIVE
C  *****  INPUTS   *****
C    AL   -ANGLE OF ATTACK
C    BE   -SIDE SLIP ANGLE
C
C    WRITTEN BY JOHN MC AVOY AND MAHINDER WAHI
C
      DIMENSION CLT(1),CDT(1),CMT(1),CYT(1),CLBT(1),CNBT(1)
C**INTERPOLATE CL
      NCLT=CLT(2)
      CL=TBLU1(AL,CLT(4),CLT(NCLT+4),1,-NCLT)
C**INTERPOLATE CD
      NCDT=CDT(2)
      CD=TBLU1(CL,CDT(4),CDT(NCDT+4),1,-NCDT)
C**INTERPOLATE CM
      NCMT=CMT(2)
      CM=TBLU1(CL,CMT(4),CMT(NCMT+4),1,-NCMT)
C**INTERPOLATE CY
      NCYT=CYT(2)
      CY=TBLU1(BE,CYT(4),CYT(NCYT+4),1,-NCYT)
C**INTERPOLATE CLB
      NCLBT=CLBT(2)
      CLB=TBLU1(AL,CLBT(4),CLBT(NCLBT+4),1,-NCLBT)
C**INTERPOLATE CNB
      NCNBT=CNBT(2)
      CNB=TBLU1(AL,CNBT(4),CNBT(NCNBT+4),1,-NCNBT)
      CL=-CL
      CD=-CD
      RETURN
      END
```

Table 130: LISTING FOR SUBROUTINE AF

```
CAF
      SUBROUTINE AF(FO,COD,C1,C2,C3,C4,C5)
C
C    PURPOSE - TO SIMULATE ANALYTICAL FUNCTIONS
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  -  ADAM LLOYD                   LATEST REVISION    FEB 76
C
C
C    LIMITATIONS -  NONE
C
C
C    INPUT/OUTPUT LIST
C
C    FO          OUTPUT VARIABLE                         ANY     OUTPUT VAR
C    COD         CODE IDENTIFYING ANALYTICAL FUNCTION    ---     INPUT  PARAM
C                CODE=    FO=
C                1        C1+C2*SIN(C3*TIME+C4)
C                2        C1+C2*COS(C3*TIME+C4)
C                3        C1+C2*EXP(-C5*TIME)*SIN(C3*TIME+C4)
C                4        C1+C2*EXP(-C5*TIME)*COS(C3*TIME+C4)
C                5        C1+C2*TIME
C                6        C1+C2*EXP(-C5*TIME)
C    C1          CONSTANT INPUTS FOR ABOVE EQNS          ---     INPUT  PARAM
C    C2          CONSTANT INPUTS FOR ABOVE EQNS          ---     INPUT  PARAM
C    C3          CONSTANT INPUTS FOR ABOVE EQNS          ---     INPUT  PARAM
C    C4          CONSTANT INPUTS FOR ABOVE EQNS          ---     INPUT  PARAM
C    C5          CONSTANT INPUTS FOR ABOVE EQNS          ---     INPUT  PARAM
      COMMON/CTIME/TIME
      NCODE=COD
      GO TO (10,20,30,40,50,60)NCODE
 10   FO=C1+C2*SIN(C3*TIME+C4)
      GO TO 100
 20   FO=C1+C2*COS(C3*TIME+C4)
      GO TO 100
 30   FO=C1+C2*EXP(-C5*TIME)*SIN(C3*TIME+C4)
      GO TO 100
 40   FO=C1+C2*EXP(-C5*TIME)*COS(C3*TIME+C4)
      GO TO 100
 50   FO=C1+C2*TIME
      GO TO 100
 60   FO=C1+C2*EXP(-C5*TIME)
 100  RETURN
      END
```

Table 131: LISTING FOR SUBROUTINE AMACH

```
CAMACH
      FUNCTION AMACH(P,T,A,W,SH)
C
C    PURPOSE - TO CALCULATE MACH NUMBER IN AN ELEMENT OF KNOWN AREA
C              WITH MOIST AIR
C
C
C    METHOD  - CALCULATES F(M)=(W*SQRT(T)/(A*P))*SQRT(R/(GAMMA*32.2))
C                    =M/(1.+(GAMMA-1.)*M*M/2.)**(GAMMA+1.)/2.*(GAMMA-1.)
C              F(M) IS ONLY SLIGHTLY DEPENDENT ON THE RATIO OF SPECIFIC
C              HEATS (GAMMA). HENCE THE ARRAYS OF F(M) VERSUS MACH NUMBER
C              ASSUME GAMMA=1.4
C
C
C    WRITTEN BY  -   ADAM LLOYD          LATEST REVISION      NOV 75
C
C
C    LIMITATIONS -  NONE
C
C
C    INPUT/OUTPUT LIST
C
C    AMACH       MACH NUMBER                          ---          OUTPUT
C    P           PRESSURE                             PSIA         INPUT
C    T           TEMPERATURE                          DEGR         INPUT
C    A           AREA                                 IN2          INPUT
C    W           FLOW RATE                            LB/MIN       INPUT
C    SH          SPECIFIC HUMIDITY                    LB/LB        INPUT
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      DIMENSION AM1(26),AM2(10)
C    DATA ARRAYS AM GIVE MACH NUMBER AS A FUNCTION OF F(M)
C    ARRAY AM1 - INCREMENT IN FM = .02
C    ARRAY AM2 - INCREMENT IN FM = .01
      DATA AM1 /
     1    .000      ,.020      ,.040      ,.060      ,.080      ,.101      ,
     2    .122      ,.143      ,.164      ,.185      ,.206      ,.227      ,
     3    .248      ,.271      ,.296      ,.319      ,.343      ,.368      ,
     4    .394      ,.421      ,.449      ,.481      ,.513      ,.548      ,
     5    .586      ,.632      /
      DATA AM2 /
     1    .632      ,.653      ,.678      ,.705      ,.735      ,.772      ,
     2    .812      ,.875      ,1.0       ,1.0       /
C    CALCULATE FLUID PROPERTIES
      CP=SHCP(T,SH)
      R=(53.3+85.7*SH)/(1.+SH)
      GAMMA=1.+R/(CP*778.-R)
C    CALCULATE F(M)
      C1=ABS(R*T/(GAMMA*32.2))
      C1=AMAX1(C1,.01)
      FM=ABS(W*SQRT(C1))/(60.*A*P)
C    CALCULATE MACH NUMBER
C    TABLES DO NOT PERMIT EXTRAPOLATION
      IF(FM.GT.0.5)GO TO 10
      X1=FM/.02 + 1.
      X2=AMAX1(X1,1.)
```

Table 131: LISTING FOR SUBROUTINE AMACH (CONCLUDED)

```
      I=X2
      I=MAXO(I,1)
      AMACH=(X2-I)*(AM1(I+1)-AM1(I)) + AM1(I)
      GO TO 100
 10   X1=(FM-.50)/.01 + 1.
      X2=AMIN1(X1,10.)
      I=X2
      I=MINO(I,9)
      AMACH=(X2-I)*(AM2(I+1)-AM2(I)) + AM2(I)
      IF(X1.LE.10.)GO TO 100
C   WARNING DIAGNOSTIC IF VALID RANGE OF INDEPENDENT VARIABLE EXCEEDED
C   TEST FOR DIAGNOSTIC PRINT OUT
      IF(IERR.NE.1)GO TO 100
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,33HNON FATAL ERROR CALLED FROM AMACH/
     1 10X,45HVARIABLE EXCEEDS UPPER LIMIT OF DATA ARRAY AM)
 100  RETURN
      END
```

Table 132: LISTING FOR SUBROUTINE AP

```
CAP
      SUBROUTINE AP(ELE,ETA,ETAD,IETA,PGR,PGRD,IPGR,X1,X1D,IX1,
     1 X2,X2D,IX2,ELD,GAM,GAD,A1,A2,A3,A4,
     2 A5,A6,A7,A8,PGL,PIT,PITD,GL1,GL2)
C            SIMULATED BY SATURATION FUNCTION WITH DEAD BAND
C  VERSION 2.                     REVISED OCT 3,1977
C
C  PURPOSE   SIMULATION OF JINDIVIK AUTOPILOT FOR PITCH CONTROL
C
C  METHOD    EQUATIONS AND TRANSFER FUNCTIONS SUPPLIED BY AUSTRALIAN GAF.
C            AN APPENDIX-AERODYNAMIC DERIVITIVES AND AUTOPILOT/SERVO
C            TRANSFER FUNCTIONS FOR MG 203A  SHORT SPAN JINDIVIK FITTED
C            WITH MG 7/8 WING TIP FUEL PODS. DATE 1973.
C            PITCH CONTROL IS SIMULATED. GROUND BASED PITCH CONTROL IS
C
C  CALL SEQUENCE
C
C  *****  OUTPUTS  *****
C    ELE             ELEVATOR ANGLE,DEG
C    ETA,ETAD,IETA   ELEVATOR SERVO ANGLE(DEG),RATE AND INTEGRATION CONTROL
C    ELD             ELEVATOR SERVO DEMAND ANGLE(DEG)
C    PGR,PGRD,IPGR   PITCH GYRO REF ANGLE(DEG),RATE,INTEGRATION CONTROL
C    X1,X1D,IX1      TRANSFER FUNCTION INTERMEDIATE STATE PARAMETER
C    X2,X2D,IX2      TRANSFER FUNCTION INTERMEDIATE STATE PARAMETER
C  *****  INPUTS  *****
C    GAM             VERTICAL FLIGHT PATH ANGLE IN DEG
C    GAD             DEMANDED FLIGHT PATH ANGLE IN DEG
C    A1,A2           POS AND NEG LIMIT OF DEAD BAND IN DEG
C    A3,A4           POS AND NEG FLIGHT PATH ANGLE FOR CONTINUOUS BEEP IN C
C    A5,A6           MAX AND MIN PITCH GYRO REFERENCE ANGLE RATES IN DEG
C    A7,A8           SATURATION SLOPES
C    PGL             PITCH GYRO REFERENCE ANGLE LIMIT
C    PIT,PITD        AIRCRAFT PITCH ANGLE AND PITCH ANGLE RATE
C    GL1             GAIN FOR REF ANGLE(PITCH) LIMITS
C    GL2             GAIN FOR INTEGRATION LIMITS ON ELE SERVO ANGLE
C
C**WRITTEN BY J.J.MCAVOY AND M.K.WAHI           SEPT 1977
C
C**PITCH ATTITUDE CONTROL
C***LANDING APPROACH CONDITIONS
C        PITCH=1.0 DEG.
C        ENGINE SPEED=7000 RPM
C        DESCENT RATE=-2.0 DEG.
C        PITCH RATE DEMAND=2.0 DEG/SEC
C***LEVEL FLIGHT CONDITIONS
C        PITCH=1.0 DEG
C        PITCH RATE DEMAND=4.0 DEG/SEC
C**GROUND CONTROL OF PITCH TO MAINTAIN VERTICAL FLIGHT PATH ANGLE(BEEPING)
C**SATURATION FUNCTION WITH DEAD BAND GIVES PITCH GYRO REF ANGLE RATE
C**SATURATION FUNCTION SHIFT FOR NON ZERO VERTICAL FLIGHT PATH ANGLE
C
      GSF=GAM-GAD
      CALL SB(PGF,GSF,A1,A2,A3,A4,A5,A6,A7,A8)
C**INTEGRATE WITH LIMITS GYRO PITCH REFERENCE ANGLE RATE
      IF(IPGR.NE.0)PGRD=PGF + GL1*AMIN1(0.,PGL-PGR)
      IF(IPGR.NE.0)PGRD=PGF + GL1*AMAX1(0.,-PGL-PGR)
```

Table 132: LISTING FOR SUBROUTINE AP (CONCLUDED)

```
C**ELEVATOR DEMAND FUNCTION
      ELD=0.28*(PITD-PGRD)+0.533*(PIT-PGR)
C**ELEVATOR SERVO FUNCTION INTEGRATION
      IF(IX1.NE.0)X1D=191.6*(ELD-ETA)
      IF (IX2.NE.0)X2D=X1+61.3*(ELD-97.7*ETA)
      IF(IETA.NE.0)ETAD=X2-9.64*ETA + GL2*AMIN1(0.,22.5-ETA)
      IF(IETA.NE.0)ETAD=X2-9.64*ETA + GL2*AMAX1(0.,-15.-ETA)
      ELE=ETA/1.5
      RETURN
      END
```

Table 133: LISTING FOR SUBROUTINE AR

```
CAR
      SUBROUTINE AR(AIL,ZET,ZETD,IZET,RGR,RGRD,IRGR,R1,R1D,IR1,
     1 R2,R2D,IR2,X1,X1D,IX1,X2,X2D,IX2,AID,AZI,CRS,DC,ROL,ROLD,R,
     2 GL1,GL2)
C  VERSION 2.                       REVISED OCT 3,1977
C
C  PURPOSE   SIMULATION OF JINDIVIK AUTOPILOT FOR ROLL CONTROL
C
C  METHOD    EQUATIONS AND TRANSFER FUNCTIONS SUPPLIED BY AUSTRALIAN GAF.
C            AN APPENDIX-AERODYNAMIC DERIVITIVES AND AUTOPILOT/SERVO
C            TRANSFER FUNCTIONS FOR MK 203A SHORT SPAN JINDIVIK FITTED
C            WITH MK 7/8 WING TIP FUEL PODS. DATE 1973
C  CALL SEQUENCE
C  *****  OUTPUTS  *****
C    AIL              AILERON DEFLECTION,DEG
C    ZET,ZETD,IZET    AILERON SERVO DEFLECTION(DEG),RATE,INTEGRATION CONTROL
C    RGR,RGRD,IRGR    ROLL GYRO REFERENCE ANGLE(DEG),RATE,INTEGRATION CONTROL
C    AID               AILERON SERVO DEMAND ANGLE(DEG) FROM AUTOPILOT
C    R1,R1D,IR1        SERVO MOTOR PARAMETER,RATE,INT CONTROL
C    R2,R2D,IR2        SERVO MOTOR PARAMETER,RATE,INT CONTROL
C    X1,X1D,IX1        SERVO MOTOR INTERMEDIATE STATE,RATE,INT CONTROL
C    X2,X2D,IX2        SERVO MOTOR INTERMEDIATE STATE,RATE,INT CONTROL
C  *****  INPUTS  *****
C    AZI               AIRCRAFT AZIMUTH(+ CLOCKWISE,DEG)
C    CRS               DEMANDED AIRCRAFT COURSE(+ CLOCKWISE,DEG)
C    DC                ALLOWABLE COURSE ERROR(+-DEG)
C    ROL               AIRCRAFT ROLL ANGLE(DEG)
C    ROLD              AIRCRAFT ROLL ANGLE RATE(DEG/SEC)
C    R                 AIRCRAFT YAW RATE, DEG/SEC
C    GL1               GAIN FOR GYRO REF ANGLE LIMITS
C    GL2               GAIN FOR INTEGRATION LIMITS ON AILERON SERVO ANGLE
C
C   WRITTEN BY    J.J.MCAVOY AND M.K.WAHI          SEPT 1977
C**AIRCRAFT COURSE CONTROL BY GROUND COMMANDS.
C***GROUND CONTROL COMMAND LEFT OR RIGHT TURN WHICH CAUSES ROLL GYRO REF
C   ANGLE TO ROTATE AT 10 DEG/SEC, GYRO REF ANGLE IS LIMITED TO + 30 DEG.
C   STRAIGHT COMMAND CAUSES GYRO REF ANGLE TO DEROTATE AT 10 DEG/SEC
C   UNTIL REF ANGLE IS ZERO.
      TRGRD=0.0
      IF(AZI.GT.CRS+DC) TRGRD=-10.0
      IF(AZI.LT.CRS-DC) TRGRD=10.0
      IF(AZI.LT.DC.AND.RGR.LT.0.0) TRGRD=10.0
      IF(AZI.GT.-DC.AND.RGR.GT.0.0)TRGRD=-10.0
      IF(IRGR.NE.0)RGRD=TRGRD + GL1*AMIN1(0.,30.-RGR)
      IF(IRGR.NE.0)RGRD=TRGRD + GL1*AMAX1(0.,-30.-RGR)
C**AUTOPILOT AILERON DEMAND FUNCTION
      IF(IR1.NE.0)R1D=R
      IF(IR2.NE.0)R2D=R1
      AID=.196*(ROLD-RGRD)+.42*(ROL-RGR)+0.2*R+0.35*R1+0.0082*R2
C**AILERON SERVO FUNCTION INTEGRATION
      IF(IX1.NE.0)X1D=191.6*(AID-ZET)
      IF(IX2.NE.0)X2D=X1+61.3*(AID-97.7*ZET)
      IF(IZET.NE.0)ZETD=X2-9.64*ZET + GL2*AMIN1(0.,24.-ZET)
      IF(IZET.NE.0)ZETD=X2-9.64*ZET + GL2*AMAX1(0.,-24.-ZET)
      AIL = ZET/3.
      RETURN
```

Table 134: LISTING FOR SUBROUTINE AS

```
CAS
      SUBROUTINE AS(ET,
     1 FX,FY,FZ,TX,TY,TZ,G1R,G1RD,IG1R,G2R,G2RD,IG2R,G1L,G1LD,IG1L,
     2 G2L,G2LD,IG2L,TR,TL,CIL,THL,
     3 ROL,PIT,YAW,X,Y,ALT,XD,YD,
     4 BSCG,WLCG,BSH,WLH,ALH,YS,YM,HC,EC,DNC,AC,CSI,
     5 DNT,THK,WDT,TPO,RO,ADR,DMP,VO)
C
C   VERSION 1.                 MAY 1978
C
C   PURPOSE - MODEL THE DYNAMIC RESPONSE OF A WATER TWISTER TYPE OF
C           - ARRESTING SYSTEM COMPOSED OF A STEEL CABLE PENDANT,
C           - NYLON TAPE AND WATER TWISTER ENERGY ABSORBER.
C
C   METHOD  - SEE ARRESTING SYSTEM MODEL DOCUMENTATION, VOL I, FINAL REPORT
C
      DIMENSION ET(23),EUL(3,3)
      REAL LH2,INTT
      COMMON/CTIME/TIME/CIO/IREAD,IWRITE,IDIAG
      COMMON/COVRLY/INST
      DATA TEST2,ZBG/0.,10./
C
C   CALL SEQUENCE
C
C **** INPUT TABLES ****
C
C     ET              -TAPE STRESS AS A FUNCTION OF STRAIN
C                     -ONE DIM. TABLE, LBS/SQ IN VS IN/IN
C
C **** OUTPUTS ****
C
C     FX,FY,FZ        -HOOK FORCES APPLIED AT VEHICLE CG, BODY AXIS, LBS
C     TX,TY,TZ        -HOOK MOMENTS APPLIED AT VEHICLE CG, BODY AXIS, FT-LBS
C     G1R,G1RD,IG1R -RIGHT TAPE DRUM ANGULAR DISP,RATE,INT CONTROL, RAD.
C     G2R,G2RD,IG2R -RIGHT TAPE DRUM ANGULAR VEL ,RATE,INT CONTROL, RAD/SE
C     G1L,G1LD,IG1L -LEFT  TAPE DRUM ANGULAR DISP,RATE,INT CONTROL, RAD
C     G2L,G2LD,IG2L -LEFT  TAPE DRUM ANGULAR VEL ,RATE,INT CONTROL, RAD/SE
C
C     TR,TL           -RIGHT AND LEFT CABLE/TAPE TENSION, LBS
C     CIL             -HOOK TO CABLE IMPACT LOAD, LBS
C     THL             -TOTAL LOAD APPLIED TO HOOK, LBS
C
C **** INPUTS ****
C
C     ROL,PIT,YAW     -ROLL,PITCH,YAW EULER ANGLES, DEG
C     X,Y,ALT         -VEHICLE CG POSITON IN EARTH AXIS, FT
C     XD,YD           -VEHICLE CG VELOCITY IN EARTH AXIS, FT/SEC
C     BSCG,WLCG       -VEHICLE CG BODY STATION AND WATER LINE, INCHES
C     BSH,WLH         -HOOK PIVOT BODY STATION AND WATER LINE, INCHES
C     ALH             -HOOK ARM LENGTH, INCHES
C
C     YS              -RUNWAY SPAN BETWEEN SHEAVES, FT
C     YM              -TAPE DRUM TO SHEAVE DISTANCE, FT
C                     -(NOTE, YM.GT. 5 PERCENT YS )
C     HC              -INITIAL CABLE HEIGHT ABOVE RUNWAY, FT
C
```

```
C      EC                 -CABLE MODULUS OF ELASTICITY, LBS/ SQ IN
C      DNC                -CABLE WEIGHT DENSITY, LBS/CU IN
C      AC                 -CABLE CROSS SECTIONAL AREA, SQ IN
C      CSI                -INITIAL CABLE STRESS, LBS/SQ IN
C                         -USED ONLY FOR KINK WAVE ANGLE CALCULATION
C
C      DNT                -TAPE WEIGHT DENSITY, LBS/CU IN
C      THK                -TAPE THICKNESS, INCHES
C      WDT                -TAPE WIDTH, INCHES
C      TPO                -MAXIMUM TAPE PAYOUT, FT
C      RO                 -OUTSIDE TAPE DRUM RADIUS, IN
C      ADR                -DRUM INERTIA, LBS-SQ IN
C                         -(INCLUDE  ALL ROTATING MASS EXCEPT TAPE)
C      DMP                -WATER TWISTER V-SQ DAMPING COEFFICIENT,
C                         -IN-LBS/(RAD/SEC)**2
C      VO                 -VEHICLE SPEED DURING INITIAL CABLE PICKUP, FT/SEC
C                         -USED FOR KINK WAVE ANGLE CALCULATION ONLY
C
C  WRITTEN BY   J.R.KILNER
C
C **** INITIAL CONDITION CALCULATIONS ****
C
      IF(TIME.NE.0.)GOTO 11
      TEST=VO+BSCG+WLCG+BSH+WLH+ALH+EC+DNC+AC+CSI+DNT
     *     +THK+WDT+TPO+RO
      IF(TEST.EQ.TEST2)GOTO 11
      TEST2=TEST
C
      CALL KINK(VO,CSI,EC,DNC,PKW,C,CSTR)
C
      IT=0
      IM=0
      PKWDEG=PKW*57.3
      RI2=RO*RO-3.8197*TPO*THK
      IF(RI2.LT.0.)RI2=0.
      RI=SQRT(RI2)
      BH=(BSH-BSCG)/12.
      HH=(WLCG-WLH)/12.
      LH2=ALH/12.
      INTT=1.5708*DNT*WDT
      TRO=.1592*THK/RO
      APG=.373*AC*DNC
      RR=RO
      RL=RO
C
      ADRX=ADR*.001
      ECX=EC*1.E-6
      NT=ET(2)
C
      WRITE(IWRITE,6000)
6000  FORMAT(1H1,35H***** ARRESTING SYSTEM CONSTANTS **,16(5H*****)//)
      WRITE(IWRITE,6001)BSH,YS,AC,TPO,DMP,WLH,YM,DNC,DNT,ADRX,
     *                     ALH,HC,ECX,THK,RO,CSI,WDT,RI
6001  FORMAT(7X,*HOOK*,14X,*FIELD*,13X,*CABLE*,19X,*TAPE*,18X,
     * *ARRESTOR*//7X,*BSH=*,F5.1,* IN*,6X,*YS =*,F5.1,* FT*,6X,
     * *AC =*,F5.3,* SQ IN*,9X,*TPO=*,F5.0,* FT*,10X,*DMP=*,F6.2,
```

```
     * 16H IN-LBS/(R/S)**2/7X,*WLH=*,F5.1,* IN*,6X,*YM =*,F5.1,* FT*,
     * 6X,*DNC=*,F5.3,* LBS/CU IN*,5X,*DNT=*,F5.3,* LBS/CU IN   IDR=*,
     * F6.1,12HE3 LBS-IN**2/7X,*LH =*,F5.1,* IN*,6X,*HC =*,F5.2,* FT*,
     * 6X,*EC =*,F5.2,*E6 LBS/SQ IN   THK=*,F5.3,* IN*,10X,*RO =*,
     * F6.2,* IN*/43X,*ICS=*,F5.0,* LBS/SQ IN*,5X,*WDT=*,F5.2,* IN*,
     * 10X,*RI =*,F6.2,* IN*/)
      WRITE(IWRITE,6003)(ET(I+3),I=1,NT)
6003  FORMAT(7X,*TAPE ELASTICITY*//7X,*STRAIN =*,10F8.3)
      WRITE(IWRITE,6004)(ET(I+3+NT),I=1,NT)
6004  FORMAT(7X,*STRESS =*,10F8.0)
      WRITE(IWRITE,6005)VO,PKWDEG
6005  FORMAT(/7X,*VO =*,F6.1,* FT/SEC*//7X,*KINK WAVE ANGLE =*,F5.1///)
      IF(RI2.EQ.0.)WRITE(IWRITE,6002)
6002  FORMAT(4X,21H***** WARNING *****   ,*TOTAL TAPE PAYOUT CANNOT*,
     * * BE STORED ON DRUM AS DEFINED*///)
      WRITE(IWRITE,6007)
6007  FORMAT(1X,23(5H*****)///)
C
11    CONTINUE
C
C
      FX=FY=FZ=TX=TY=TZ=0.
      TR=TL=CIL=THL=0.
      AR=AL=0.
C
C  HOOK MISSED CABLE IF IM=1
C
      IF(IM.EQ.1)GOTO 88
C
C  BODY-TO-EARTH TRANSFORMATION COEFFICIENTS
C
      CR=COS(.01745*ROL)
      CP=COS(.01745*PIT)
      CY=COS(.01745*YAW)
      SR=SIN(.01745*ROL)
      SP=SIN(.01745*PIT)
      SY=SIN(.01745*YAW)
C
      EUL(1,1)=CP*CY
      EUL(2,1)=CP*SY
      EUL(3,1)=-SP
      EUL(1,2)=SR*SP*CY-CR*SY
      EUL(2,2)=SR*SP*SY+CR*CY
      EUL(3,2)=SR*CP
      EUL(1,3)=CR*SP*CY+SR*SY
      EUL(2,3)=CR*SP*SY-SR*CY
      EUL(3,3)=CR*CP
C
C  DETERMINE POSTIION OF POINT H IN EARTH AXIS
C
      XEH=X-BH*EUL(1,1)+HH*EUL(1,3)
      YEH=Y
      ZEH=-ALT-BH*EUL(3,1)+HH*EUL(3,3)
C
C  NO CABLE HOOKUP IF XEH.LT.0
C
```

```
      IF(XEH.LT.0.)GOTO 88
C
C  DETERMINE POSITION OF POINT H IN HOOK-CABLE AXIS
C
      ZL=-ZEH-HC
      XCH=SQRT(ZL*ZL+XEH*XEH)
      SINC= ZL/XCH
      COSC=XEH/XCH
      YCH=YEH
C
C  DETERMINE ANGLE DELTA
C
      AB=XEH*SY/CY+(ZL/CP-SP*(ZL*SP/CP+XEH/CY))*SR/EUL(2,2)
      GG2=XCH*XCH+AB*AB
      GG=SQRT(GG2)
      SIND= AB/GG
      COSD=XCH/GG
C
C  DETERMINE POSITION OF POINT P IN HOOK-CABLE AXIS
C
      XCP=XCH-LH2*COSD
      YCP=YCH-LH2*SIND
C
C  TEST FOR CONDITION WHERE HOOK PASSES OVER CABLE
C  AND FAILS TO HOOKUP
C
      IF(IT.EQ.1)GOTO 22
      IT=1
      YR=.5*YS-YCP
      YL=.5*YS+YCP
      IF(XCP.LE.0.)GOTO 22
      IM=1
      D=12.*XCP
      WRITE(IWRITE,6006)D,TIME,XEH
6006  FORMAT(////1X,11(6H******)//3X,28H**** WARNING ****  HOOK HAS ,
     * 16HMISSED CABLE BY ,F5.2,13H INCHES   ****//24X,*T=*,F5.3,4X,
     * *XEH=*,F5.3,* FT*//1X,11(6H******)////)
      GOTO 88
C
C  HOOK-CABLE ENGAGEMENT
C    DETERMINE TAPE PAYOUT
C
22    AYR=SQRT((.5*YS-YCP)*(.5*YS-YCP)+XCP*XCP)
      AYL=SQRT((.5*YS+YCP)*(.5*YS+YCP)+XCP*XCP)
      CPR=(.5*YS-YCP)/AYR
      CPL=(.5*YS+YCP)/AYL
      SPR=XCP/AYR
      SPL=XCP/AYL
      PR=ACOS(CPR)
      PL=ACOS(CPL)
C
      IF(PKW.GT.PR)GOTO 24
      AR=AYR-YR
      GOTO 26
24    SPR=SIN(PKW)
      CPR=COS(PKW)
```

```
      AR=(1.-CPR)*XCP/SPR+.5*YS-YCP-YR
C
26    IF(PKW.GT.PL)GOTO 28
      AL=AYL-YL
      GOTO 30
28    SPL=SIN(PKW)
      CPL=COS(PKW)
      AL=(1.-CPL)*XCP/SPL+.5*YS+YCP-YL
C
C  DETERMINE UNSTRAINED TAPE PAYOUT
C
30    ARU=.08333*RO*G1R*(1.-.5*TRO*G1R)
      ALU=.08333*RO*G1L*(1.-.5*TRO*G1L)
      RR=RO*(1.-TRO*G1R)
      RL=RO*(1.-TRO*G1L)
C
C  DETERMINE TAPE STRAIN AND TAPE/CABLE TENSILE LOAD
C
      UR=(AR-ARU)/(ARU+YM)
      UL=(AL-ALU)/(ALU+YM)
      TR=WDT*THK*TBLU1(UR,ET(4),ET(NT+4),1,-NT)
      TL=WDT*THK*TBLU1(UL,ET(4),ET(NT+4),1,-NT)
C
C  DETERMINE HOOK-CABLE IMPACT LOAD
C
      IF(PR.GE.PKW.AND.PL.GE.PKW)GOTO 40
      VS=XD*XD+YD*YD
      AV=YD/XD
C
      FR=FL=0.
      IF(PKW.GT.PR)FR=APG*VS*COS(PKW-AV)/SPR
      IF(PKW.GT.PL)FL=APG*VS*COS(PKW+AV)/SPL
      CIL=FR+FL
C
C     DETERMINE HOOK ANGLE RELATIVE TO BODY X AXIS
C
40    XEG=XEH+ZBG*EUL(1,3)
      YEG=YEH+ZBG*EUL(2,3)
      ZEG=ZEH+ZBG*EUL(3,3)
C
      FF2=XEG*XEG+(YEG-YEH+AB)*(YEG-YEH+AB)+(ZEG+HC)*(ZEG+HC)
      EE=ZBG
      COSH=(EE*EE+GG2-FF2)/(2.*EE*GG)
      SINH=SQRT(1.-COSH*COSH)
C
C  CALCULATE HOOK COMPONENT FORCES IN HOOK-CABLE AXIS
C
      FCX=-TR*SPR-TL*SPL-CIL
      FCY= TR*CPR-TL*CPL-CIL*AV
      THL=SQRT(FCX*FCX+FCY*FCY)
C
C  DETERMINE BODY FORCES AND MOMENTS AT VEHICLE CG
C
      FXEP= FCX*COSC
      FYEP= FCY
      FZEP=-FCX*SINC
```

```
C
      FX=FXEP*EUL(1,1)+FYEP*EUL(2,1)+FZEP*EUL(3,1)
      FY=FXEP*EUL(1,2)+FYEP*EUL(2,2)+FZEP*EUL(3,2)
      FZ=FXEP*EUL(1,3)+FYEP*EUL(2,3)+FZEP*EUL(3,3)
C
      BP=BH+LH2*SINH
      HP=HH+LH2*COSH
      TX=-HP*FY
      TY=HP*FX+BP*FZ
      TZ=-BP*FY
C
C  CALCULATE DRUM DISPLACEMENT AND VELOCITY RATES
C
88    RI4=RI**4
      DRR4=RR**4-RI4
      DRL4=RL**4-RI4
      VIR=386./(INTT*DRR4+ADR)
      VIL=386./(INTT*DRL4+ADR)
C
      IF(IG1R.NE.0)G1RD=G2R
      IF(IG2R.NE.0)G2RD=VIR*(-DMP*G2R*ABS(G2R)+RR*TR)
      IF(IG1L.NE.0)G1LD=G2L
      IF(IG2L.NE.0)G2LD=VIL*(-DMP*G2L*ABS(G2L)+RL*TL)
C
C
      RETURN
      END
```

Table 135: LISTING FOR SUBROUTINE CLRNCE

```
CCLRNCE
       SUBROUTINE CLRNCE
C CALCULATION OF TRUNK GROUND CLEARANCE FOR EACH SEGMENT
C
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
      ICLNS=ICLN
      COSCOS=COS(PHIE)*COS(THETAE)
C CALCULATE SEGMENT GAP
      DO 16 I=1,NSTOP
      YGH(I)=SL4(I)-YG(I)*COSCOS
C IF NEGATIVE SET GAP TO ZERO
      YGH(I)=AMAX1(YGH(I),0.0)
C TEST FOR HARD SURFACE CONTACT ON SEGMENT
C IF ANY SEGMENT CONTACTS HARD SURFACE WRITE ERROR ONCE
      IF(YGH(I).LE.0.0.AND.ICLN.EQ.0) WRITE(6,9001)
9001   FORMAT(5X,* HARD SURFACE CONTACT THIS STEP *)
      IF(YGH(I).LE.0.0) ICLN=ICLN+1
16    CONTINUE
      ICLN=ICLNS
         RETURN
        END
```

Table 136: LISTING FOR SUBROUTINE CDVCHP

```
CDVCHP
      SUBROUTINE CDVCHP
C SUBROUTINE TO CALCULATE DVCHP
      COMMON/DYNAMIC/TIME,FTIME,DTIME,IQ,NQ,DVCH,DVTK,VELX,DERY(13)
     1,DVCHP
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      VCHSS=VCH
      PRAT=(PCH+(PTK-PCH)*0.1)/PTK
      CALL HYCURV (PRAT,HX)
      HY=HYI*HX
      CALL TRUNK (ISHAPE)
      CALL SEGMNT(1)
      CALL COORDN
      CALL PROFILE
      CALL CLRNCE
      CALL SHAPE2
      DVCHP=(VCHSS-VCH)/((PCH/PTK)-PRAT)
      RETURN
      END
```

Table 137: LISTING FOR SUBROUTINE COORDN

```
CCOORDN
      SUBROUTINE COORDN
C THIS SUBROUTINE CALCULATES X AND Z COORDINATES OF THE GROUND
C POINT CORRESPONDING TO EACH SEGMENT, FOR A PARTICULAR ACLS
C ORIENTATION
C
      REAL L,L1,L2,LS,LP,MASS
      COMMON/VEHCL/MASS,AIX,AIZ,AIXY,AIYZ,AIZX,CC,GG,FF,AIFAN
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/BTERM/B11,B12,B13,B21,B22,B23,B31,B32,B33
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
C CALL BMATRIX FOR SPACIAL TRANSFORMATION
      CALL ROTATE
C
C DO LOOP OF ALL  SEGMENTS TO GROUND POSITION
      DO 10 I=1,NSTOP
      XCXCC=(XCX(I)-CC)
      ZCXFF=(ZCX(I)-FF)
C
C CALCULATE VECTOR DA FOR SEGMENT
      SL4(I)=(YCG+XCXCC*B12+ZCXFF*B32)/B22-GG
      SL4GG=(SL4(I)+GG)
C
C CALCULATE X-GROUND COORDINATE
      XG(I)=XCXCC*B11-SL4GG*B21+ZCXFF*B31+XCG
C
C CALCULATE Z-GROUND COORDINATE
      ZG(I)=XCXCC*B13-SL4GG*B23+ZCXFF*B33
   10 CONTINUE
      RETURN
      END
```

Table 138: LISTING FOR SUBROUTINE DL

```
CDL
      SUBROUTINE DL(FY,VD,TX,TZ,CYB,CYBD,CYP,CYR,CYDR,CYDA,KCY,
     1 CYTR,CYFS,KCYGE,KCYB,KCYBR,CLB,CLBD,CLP,CLR,CLDR,CLDA,KCL,
     2 CLTR,CLFS,KCLGE,KCLB,KCLBR,CNB,CNBD,CNP,CNR,CNDR,CNDA,
     3 CNTR,CNFS,KCNGE,KCNB,KCNBR,RUD,AIL,FSP,UD,WD,FYIN,TXIN,
     4 TZIN,AMASS,B,XAC,DIM,CAS,SAS,U,V,W,P,R,BETA,EV,VBAR,QBAR,RW)
      REAL KCY,KCYGE,KCYB,KCYBR,KCL,KCLGE,KCLB,KCLBR,KCNGE,KCNB,KCNBR
      REAL KCN
C   VERSION 3.                                          AUG 18 1977
C   PURPOSE   COMPUTE LATERAL-DIRECTIONAL FORCES AND MOMENTS
C   MEHTOD    USE LINEAR STABILITY DERIVATIVES TO COMPUTE SIDE
C             FORCE, Y AXIS ACCEL., AND ROLL AND YAW MOMENTS.
C             TRASFORM TO BODY AXES AND ADD IN EXTERNAL FORCES,TORQUES
C   CALL SEQUENCE
C   ******  OUTPUTS  ******
C     FY                   -Y BODY AXIS FORCE SUM, LBS
C     VD                   -Y BODY AXIS ACCELERATION, FT/SEC**2
C     TX,TZ                -X,Z BODY AXIS (ROLL,YAW) MOMENTS, FT-LBS
C   ******  INPUTS  *******
C   AERO-DERIVATIVES- - UNITS FOR DIMENSIONAL CASE
C   SIDE FORCE COEFFICIENTS
C     CYB,CYBD             -BETA AND BETA DOT COEFFICIENTS (NONDIM.)
C                          -V AND VD COEFFICIENTS (DIM.),LB-SEC/FT,
C                          -LB-SEC**2/FT
C     CYP,CYR,CYDR,CYDA -ROLL RATE,YAW RATE,RUDDER AND AILERON
C                          -COEFFICIENTS, LB-SEC/DEG, LB-SEC/DEG,
C                          -LB/DEG, LB/DEG
C     KCY                  -AEROELASTIC EFFECTS COEFFICIENT (MACH,ALT)
C     CYTR                 -TAKEOFF OR RECOVERY TRUNK COEFFICIENT(NONDIM)
C                          -V COEFFICIENT (DIM.),LB-SEC/FT
C     CYFS                 -FLIGHT SPOILER COEFFICIENT,LB/DEG
C     KCYGE                -GROUND EFFECT FACTOR ON CYB
C     KCYB                 -LARGE SIDE SLIP ANGLE FACTOR FOR CYB
C     KCYBR                -SIDE FORCE RUDDER EFFECTIVENESS PARAMETER FOR
C                          -LARGE SIDE SLIP ANGLES
C   ROLLING MOMENT COEFFICIENTS
C     CLB,CLBD             -BETA AND BETA DOT COEFFICIENTS (NONDIM.)
C                          -V AND VD COEFFICIENTS(DIM),LB-SEC, LB-SEC**2
C     CLP,CLR,CLDR,CLDA -ROLL RATE,YAW RATE,RUDDER AND AILERON
C                          -COEFFICIENTS, FT-LB-SEC/DEG, FT-LB-SEC/DEG,
C                          -FT-LB/DEG, FT-LB/DEG
C     KCL                  -AEROELASTIC EFFECTS COEFFICIENT (MASH,ALT)
C     CLTR                 -TAKEOFF/RECOVERY TRUNK COEFFICIENT(NONDIM.)
C                          -V COEFFICIENT (DIM.), LB-SEC
C     CLFS                 -FLIGHT SPOILER COEFFICIENT, FT-LB/DEG
C     KCLGE                -GROUND EFFECT FACTOR ON CLB
C     KCLB                 -LARGE SIDE SLIP ANGLE FACTOR FOR CLB
C     KCLBR                -RUDDER EFFECTIVENESS PARAMETER FOR LARGE
C                          -SIDE SLIP ANGLES
C   YAWING MOMENT COEFFICIENTS
C     CNB,CNBD             -BETA AND BETA DOT COEFFICIENTS (NONDIM.)
C                          -V AND VD COEFFICIENTS (DIM.),LB-SEC,LB-SEC**2
C     CNP,CNR,CNDR,CNDA -ROLL RATE,YAW RATE, RUDDER AND AILERON
C                          -COEFFICIENTS, FT-LB-SEC/DEG,FT-LB-SEC/DEG,
C                          -FT-LB/DEG,FT-LB/DEG
C     KCN                  -AEROELASTIC EFFECTS COEFFICIENT (MACH,ALT)
```

```
C      CNTR                -TAKEOFF/RECOVERY TRUNK COEFFICIENT(NONDIM.)
C                          -V COEFFIEICNT (DIM.), LB-SEC
C      CNFS                -FLIGHT SPOILER COEFICIENT, FT-LB/DEG
C      KCNGE               -GROUND EFFECT FACTOR ON CNB
C      KCNB                -LARGE SIDE SLIP ANGLE FACTOR FOR CNB
C      KCNBR               -RUDDER EFFECTIVENESS PARAMETER FOR LARGE
C                          -SIDE SLIP ANGLES
C    CONTROL SURFACES
C      RUD,AIL,FSP         -RUDDER,AILERON AND FLIGHT SPOILER
C                          -DEFLECTIONS, DEG
C    LONGITUDINAL ACCELERATIONS
C      UD,WD               -X,Z BODY AXIS ACCELERATIONS, FT/SEC**2
C    EXTERNAL FORCES AND MOMENTS
C      FYIN                -Y BODY AXIS FORCE, LBS
C      TXIN,TZIN           -X AND Z BODY AXIS MOMENTS, FT-LBS
C    CONSTANTS
C      AMASS               -RIGID BODY MASS, SLUGS
C      B                   -WING SPAN, FT
C      XAC                 -X AXIS LOCATION OF CENTER OF PRESSURE
C                          -FROM C.G., FT
C      DIM                 -INDICATOR FUNCTION FOR AERO COEFFICIENTS
C                                0= BODY AXIS, DIM.
C                                1= BODY AXIS, NONDIM.
C                                2= STABILITY AXIS, DIM.
C                                3= STABILITY AXIS, NONDIM.
C      CAS,SAS             -DIRECTION COSINES FOR STABILITY AXIS
C                          -OR BODY AXIS DEPENDING ON DIM.
C    STATES AND AERO-VARIABLES
C      U,V,W               -X,Y,Z BODY AXIS VELOCITIES, FT/SEC
C      P,R                 -X,Z BODY AXIS ANGULAR RATES, DEG/SEC
C      BETA                -SIDESLIP ANGLE, DEG
C      EV                  -Y BODY AXIS ACCELERATION TERM FOR VD,FT/SEC**2
C      VBAR                -TRUE AIRSPEED, FT/SEC
C      QBAR                -DYNAMIC PRESSURE TIMES REFERENCE AREA, LBS
C      RW                  -Y BODY AXIS ANGULAR RATE GUST, DEG/SEC
C
C    WRITTEN BY A.W.WARREN  AS COMPONENT LD IN FLIGHT CONTROLS
C                           LIBRARY.          SEPT. 1976
C
C    MODIFIED BY  MAHINDER WAHI                             MAY 2 1977
C
C                           INITIALIZATION
      DATA ISW,RPD /0,.01745329/
      IF(ISW.EQ.1) GO TO 10
      IF(CYB  .EQ. .99999) CYB  =0.
      IF(CYBD .EQ. .99999) CYBD =0.
      IF(CYP  .EQ. .99999) CYP  =0.
      IF(CYR  .EQ. .99999) CYR  =0.
      IF(CYDR .EQ. .99999) CYDR =0.
      IF(CYDA .EQ. .99999) CYDA =0.
      IF(CYTR .EQ. .99999) CYTR =0.
      IF(CYFS .EQ. .99999) CYFS =0.
      IF(KCYGE.EQ. .99999) KCYGE=1.
      IF(KCYB .EQ. .99999) KCYB =1.
      IF(KCYBR.EQ. .99999) KCYBR=1.
      IF(KCY  .EQ. .99999) KCY  =1.
```

```
C
      IF(CLB  .EQ. .99999) CLB  =0.
      IF(CLBD .EQ. .99999) CLBD =0.
      IF(CLP  .EQ. .99999) CLP  =0.
      IF(CLR  .EQ. .99999) CLR  =0.
      IF(CLDR .EQ. .99999) CLDR =0.
      IF(CLDA .EQ. .99999) CLDA =0.
      IF(CLTR .EQ. .99999) CLTR =0.
      IF(CLFS .EQ. .99999) CLFS =0.
      IF(KCLGE.EQ. .99999) KCLGE=1.
      IF(KCLB .EQ. .99999) KCLB =1.
      IF(KCLBR.EQ. .99999) KCLBR=1.
      IF(KCL  .EQ. .99999) KCL  =1.
C
      IF(CNB  .EQ. .99999) CNB  =0.
      IF(CNBD .EQ. .99999) CNBD =0.
      IF(CNP  .EQ. .99999) CNP  =0.
      IF(CNR  .EQ. .99999) CNR  =0.
      IF(CNDR .EQ. .99999) CNDR =0.
      IF(CNDA .EQ. .99999) CNDA =0.
      IF(CNTR .EQ. .99999) CNTR =0.
      IF(CNFS .EQ. .99999) CNFS =0.
      IF(KCNGE.EQ. .99999) KCNGE=1.
      IF(KCNB .EQ. .99999) KCNB =1.
      IF(KCNBR.EQ. .99999) KCNBR=1.
C
      IF(XAC  .EQ. .99999) XAC  =0.
      IF(UD   .EQ. .99999) UD=WD=0.
      IF(FYIN .EQ. .99999) FYIN=TXIN=TZIN=0.
      IF(RUD  .EQ. .99999) RUD  =0.
      IF(AIL  .EQ. .99999) AIL  =0.
      IF(FSP  .EQ. .99999) FSP  =0.
      ISW=1
C                       DIMENSIONAL FORCE AND MOMENT SOLUTION
C
   10 IF(DIM.EQ.1. .OR. DIM.EQ.3.) GO TO 20
C
C              SOLUTION OF IMPLICIT EQN FOR FY AND VD USING ASOL,EVP
      VWDOT= RW*VBAR*RPD
      ASOL= 1.- CYBD/AMASS
      EVP = EV + VWDOT
      FY= ((CYB*KCY + CYTR)*KCYGE*V + CYBD*EVP + CYP*P + CYR*R
     *      + CYDA*AIL + CYFS*FSP)*KCYB + CYDR*RUD*KCYBR
      FYAERO= FY/ASOL
      FY= (FY + FYIN)/ASOL
      VDP= FY/AMASS + EVP
      VD= VDP - VWDOT
C                    ROLL AND YAW MOMENT COMPUTATIONS
C
      KCN=KCY
      TXS= ((CLB*KCL + CLTR)*KCLGE*V + CLBD*EVP + CLP*P + CLR*R
     *       + CLDA*AIL + CLFS*FSP)*KCLB + CLDR*RUD*KCLBR
      TZS= ((CNB*KCN + CNTR)*KCNGE*V + CNBD*EVP + CNP*P + CNR*R
     *     + CNDA*AIL + CNFS*FSP)*KCNB +CNDR*RUD*KCNBR
      IF(DIM.EQ.2.) GO TO 40
C
```

Table 138: LISTING FOR SUBROUTINE DL (CONCLUDED)

```
C                    BODY AXIS TORQUES
   30 TX= TXS + TXIN
      TZ= TZS + TZIN + XAC*FYAERO
      RETURN
C                    STABILITY AXIS TORQUES
C
   40 TX = TXS*CAS - TZS*SAS + TXIN
      TZ = TXS*SAS + TZS*CAS + TZIN + XAC*FYAERO
      RETURN
C
C               NONDIMENSIONAL FORCE AND MOMENT SOLUTION
C
   20 BDIM= B/(VBAR+VBAR)
      BET1= BETA*RPD
      P1= P*RPD
      R1= R*RPD
      AIL1= AIL*RPD
      RUD1= RUD*RPD
      FSP1= FSP*RPD
C
C               SOLUTION OF IMPLICIT EQN FOR FY AND VD USING ASOL,DSOL
      DIV= (1. - BET1**2)/VBAR
      ASOL= 1. - QBAR*BDIM*CYBD*DIV/AMASS
      DSOL=EV*DIV - BET1*(U*UD+W*WD)/VBAR**2 + RW*RPD
      FY= QBAR*(((CYB*KCY + CYTR)*KCYGE*BET1 + BDIM*(CYBD*DSOL
     1             +CYP*P1 + CYR*R1) + CYDA*AIL1 + CYFS*FSP1)*KCYB
     2             +CYDR*RUD1*KCYBR)
      FYAERO= FY/ASOL
      FY= (FY+FYIN)/ASOL
      VD= FY/AMASS + EV
C               ROLL AND YAW MOMENT COMPUTATIONS
C
      KCN=KCY
      BETDOT= FY*DIV/AMASS + DSOL
      TXS=QBAR*B*(((CLB*KCL + CLTR)*KCLGE*BET1 + BDIM*(CLBD*BETDOT
     1               + CLP*P1 + CLR*R1) + CLDA*AIL1 + CLFS*FSP1)*KCLB
     2               + CLDR*RUD1*KCLBR)
      TZS= QBAR*B*(((CNB*KCN + CNTR)*KCNGE*BET1 + BDIM*(CNBD*BETDOT
     1               + CNP*P1 + CNR*R1) + CNDA*AIL1 + CNFS*FSP1)*KCNB
     2               + CNDR*RUD1*KCNBR)
      IF(DIM.EQ.1.) GO TO 30
      IF(DIM.EQ.3.) GO TO 40
      END
```

Table 139: LISTING FOR SUBROUTINE DS

```
CDS
      SUBROUTINE DS(U,UD,IU,V,VD,IV,W,WD,IW,P,PD,IP,Q,QD,IQ,
     1 R,RD,IR,ROL,ROLD,IROL,PIT,PITD,IPIT,YAW,YAWD,IYAW,
     2 XD,YD,Z,ZD,IZ,PDOT,QDOT,RDOT,ROD,PID,
     3 UDOT,VDOT,WDOT,TX,TY,TZ,XXI,YYI,ZZI,XZI)
C   VERSION 2.                      JULY 8 1977
C  PURPOSE:  SIX DEGREE OF FREEDOM RIGID BODY EQUATIONS OF MOTION
C  METHOD:  EULER ANGLES
C  CALL SEQUENCE:
C  *****  OUTPUTS  *****
C   LINEAR VELOCITIES  --  BODY AXES
C      U,UD,IU       - X AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C      V,VD,IV       - Y AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C      W,WD,IW       - Z AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C   ANGULAR VELOCITIES --  BODY AXES
C      P,PD,IP       - X AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C      Q,QD,IQ       - Y AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C      R,RD,IR       - Z AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C   EULER ANGLES  --  EARTH TO BODY - YAW,PITCH,ROLL
C      ROL,ROLD,IROL - ROLL ANGLE,RATE,INT CONTROL,DEG
C      PIT,PITD,IPIT - PITCH ANGLE,RATE,INT CONTROL, DEG
C      YAW,YAWD,IYAW - YAW ANGLE,RATE,INT CONTROL, DEG
C   POSITION  --  EARTH AXES
C      XD            - X AXIS LINEAR VELOCITY, FT/SEC
C      YD            - Y AXIS LINEAR VELOCITY,FT/SEC
C      Z,ZD,IZ       - -Z AXIS POSITION (ALT),VELOCITY,INT CONTOL, FT
C   ANGULAR ACCELERATION  --  BODY AXES
C      PDOT          - X AXIS ANGULAR ACCELERATION, DEG/SEC2
C      QDOT          - Y AXIS ANGULAR ACCELERATION, DEG/SEC2
C      RDOT          - Z AXIS ANGULAR ACCELERATION, DEG/SEC2
C ****************   INPUTS   *******************
C   LINEAR ACCELERATION  --  BODY AXES
C      UDOT          - X AXIS LINEAR ACCELERATION, FT/SEC2
C      VDOT          - Y AXIS LINEAR ACCELERATION, FT/SEC2
C      WDOT          - Z AXIS LINEAR ACCELERATION, FT/SEC2
C   MOMENTS
C      TX,TY,TZ      - X,Y,Z  AXIS TORQUES, FTLBS
C   MOMENTS OF INERTIA
C      XXI,YYI,ZZI - X,Y,Z AXIS MOMENTS OF INERTIA, SLUG-FT2
C      XZI           - PRODUCT OF INERTIA, SLUG-FT2
C     WRITTEN BY   J.D. BURROUGHS          MAY 1976
C    AS COMPONENT **SD** IN FLT CONTROLS LIBRARY
C
C   MODIFIED BY  M.K. WAHI           JULY 1977
C
      DATA RPD,DPR /.01745329,57.29578/
      CP=COS(PIT*RPD)
      SP=SIN(PIT*RPD)
      CR=COS(ROL*RPD)
      SR=SIN(ROL*RPD)
      P1= P*RPD
      Q1= Q*RPD
      R1= R*RPD
C ***************  LINEAR VELOCITY EQUATIONS  **************
      IF(IU.NE.0)UD=UDOT
      IF(IV.NE.0)VD= VDOT
```

```
      IF(IW.NE.0)WD= WDOT
C ***************  ANGULAR VELOCITY EQUATIONS  *************
      IF(XZI.NE.0..AND.XZI.NE. .99999) GO TO 100
      IF(IP.NE.0)PD=(TX-Q1*R1*(ZZI-YYI))/XXI*DPR
      IF(IQ.NE.0)QD=(TY-P1*R1*(XXI-ZZI))/YYI*DPR
      IF(IR.NE.0)RD=(TZ-Q1*P1*(YYI-XXI))/ZZI*DPR
      GO TO 160
100   IF(IQ.NE.0)QD=((TY-P1*R1*(XXI-ZZI)+(R1*R1-P1*P1)*XZI)/YYI)*DPR
      IF(IP+IR.EQ.0)GO TO 160
      TEM=ZZI/XZI
      DIV=XXI*TEM-XZI
      QR=Q1*R1
      PQ=P1*Q1
      SUM=ZZI-YYI+XXI
      IF(IP.NE.0)PD=((TX*TEM+TZ-QR*(TEM*(ZZI-YYI)+XZI)
     1 +PQ*SUM)/DIV)*DPR
      TEM=XXI/XZI
      IF(IR.NE.0)RD=(TX+TZ*TEM-QR*SUM+PQ*(TEM*(XXI-YYI)+
     1 XZI))/DIV*DPR
C ****************  EULER ANGLE EQUATIONS  *************
160   IF(IPIT.NE.0)PITD=Q*CR-R*SR
      PID = PITD
      IF(CP.NE.0.)PSID=(Q*SR+R*CR)/CP
      IF(IYAW.NE.0)YAWD=PSID
      IF(IROL.NE.0)ROLD=P+PSID*SP
      ROD = ROLD
C ***************  POSITION EQUATIONS  ***************
C -----       TEST IF X AND Y (LATITUDE AND LONGITUDE) ARE BOTH FROZEN
      SPSR=SP*SR
      SPCR=SP*CR
      CY=COS(YAW*RPD)
      SY=SIN(YAW*RPD)
      XD=CY*CP*U+(-SY*CR+CY*SPSR)*V+(SY*SR+CY*SPCR)*W
      YD=SY*CP*U+(CY*CR+SY*SPSR)*V+(-CY*SR+SY*SPCR)*W
200   IF(IZ.NE.0)ZD=SP*U-CP*SR*V-CP*CR*W
C ****************  ANGULAR ACCELERATIONS  (FOR OUTPUT PURPOSES ONLY) ****
      PDOT=PD
      QDOT=QD
      RDOT=RD
      RETURN
      END
```

Table 140: LISTING FOR SUBROUTINE DU

```
CDU
      SUBROUTINE DU(T2,W2,P1,P1DOT,IP1,T1,W1,P2,AK,AL,D,TAM,HO,FC)
C
C   PURPOSE - ANALYSIS OF DUCT, USING SIMPLIFIED MODEL WITH SINGLE STATE
C             VARIABLE
C
C
C   METHOD  - SEE SECTION 3.1.1 FOR DERIVATION OF EQUATIONS
C            FINAL REPORT AFFDL-TR-   VOLUME I
C       CONTRACT NO. F33615-76-C-3100,JULY 1977.
C       THE INPUTS TO EACH DUCT RELATED MODEL ARE THE
C       INLET FLOW RATE,INLET TEMPERATURE,AND OUTLET
C       PRESSURE,WHILE OUTPUTS ARE THE OUTLET FLOW RATE AND
C       TEMPERATURE AND INLET PRESSURE.THE INLET PRESSURE
C       IS TREATED AS A STATE VARIABLE.
C
C
C   LIMITATIONS - IF TEMPERATURE CHANGES ARE LARGE,ADDITION
C                 OF THERMAL NODES SHOULD BE CONSIDERED
C
C   WRITTEN BY   ADAM LLOYD  AS COMPONENT *DE* IN ECS LIBRARY
C                                                 NOV. 1975
C
C   MODIFIED BY  - MAHINDER WAHI                          AUGUST  1977
C
C
C   INPUT/OUTPUT LIST
C
C   T2         OUTLET TEMPERATURE(PORT NO 2)      DEGR      OUTPUT VAR
C   W2         OUTLET FLOW                        LB/MIN    OUTPUT VAR
C   P1         INLET  PRESSURE(PORT NO 1)         PSIA      OUTPUT STATE
C   P1DOT      INLET PRESSURE DERIVATIVE          PSIA/SEC  OUTPUT DERIV
C   IP1        INTEGRATOR CONTROL                 ---       PROGRAM VAR
C   T1         INLET TEMPERATURE                  DEGR      INPUT  VAR
C   W1         INLET FLOW                         LB/MIN    INPUT  VAR
C   P2         OUTLET PRESSURE(PORT NO 2)         PSIA      INPUT  VAR
C   AK         K FACTOR                           ---       INPUT  PARAM
C   AL         LENGTH                             FT        INPUT  PARAM
C   D          DIAMETER                           IN        INPUT  PARAM
C   TAM        EFFECTIVE LOCAL AMBIENT TEMP       DEGR      INPUT  PARAM
C   HO         EXTERNAL HEAT TRANSFER COEFFICIENT BTU/FT2   INPUT  PARAM
C              (BASED ON INTERNAL WETTED AREA)    HR DEGR
C   FC         FREQUENCY CONTROL ON P1. (FC.GE.1.)---       INPUT  PARAM
C              A VALUE OF FC GREATER THAN 1.
C              DECREASES FREQUENCY RESPONSE OF P1
C              CORRESPONDINGLY
C
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
C   CALCULATE GAS CONSTANT AND RATIO OF SPECIFIC HEATS, BASED ON
C   INLET TEMPERATURE
      CP=SHCP(T1,0.)
      R=53.3
      GAMMA=1.+R/(778.*CP-R)
      G1=1./(GAMMA-1.)
      G2=(GAMMA-1.)/2.
```

Table 140: LISTING FOR SUBROUTINE DU (CONCLUDED)

```
C     CALCULATE OUTLET FLOW BASED ON INLET TEMPERATURE
      CA=.785398*D*D
      CALL FNFLOW(P1,P2,T1,CA,AK,FN,W2)
C     CALCULATE OVERALL CONDUCTANCE (UA) BASED ON AVERAGE FLOW
      WBAR=(ABS(W1)+ABS(W2))/2.
      WBAR=AMAX1(WBAR,.01)
      HINT=HI(1,T1,T1,WBAR,0.,D,AL,0.)
C     THE INPUT VALUE OF THE EXTERNAL HEAT TRANSFER COEFFICIENT IS BASED
C     ON THE INTERNAL WETTED AREA. HENCE UA IS GIVEN BY
      UA=0.004363*D*AL*HINT*HO/(HINT+HO)
C     CONSTANT 0.004363=PI/(60.*12.)     UA IS IN BTU/MIN DEGR
C     CALCULATE OUTLET TEMPERATURE T2
      T2=TAM+(T1-TAM)/EXP(UA/(CP*WBAR))
C     WARNING DIAGNOSTIC IF TEMPERATURE CHANGE EXCEEDS 300 DEGR
      IF(ABS(T1-T2).LE.300.)GO TO 10
      IF(T1.GT.T2) T2=T1-300.
      IF(T2.GT.T1) T2=T1+300.
C     TEST FOR DIAGNOSTIC PRINT OUT
      IF(IERR.NE.1)GO TO 10
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,45HNON FATAL ERROR CALLED FROM DUCT COMPONENT DU/
     1 10X,35HTEMPERATURE CHANGE EXCEEDS 300 DEGR)
C     CALCULATE INLET PRESSURE BASED ON AVERAGE FLUID TEMPERATURE
 10   TBAR=(T1+T2)/2.
      PBAR=(P1+P2)/2.
C     CALCULATE MACH NUMBER BASED ON AVERAGE CONDITIONS
      AM=AMACH(PBAR,TBAR,CA,WBAR,0.)
      IF(IP1.NE.0)P1DOT=R*TBAR*(W1-W2)*(1.+G2*AM*AM)**G1/(60.*CA*AL*FC)
      RETURN
      END
```

Table 141: LISTING FOR SUBROUTINE DV

```
CDV
      SUBROUTINE DV(T2,W2,P1,P1DOT,IP1,T1,W1,P2,OPE,AL,D,DPO,
     1 TAM,HO,FC,VAL)
C
C    PURPOSE - ANALYSIS OF A VALVE (BUTTERFLY,GATE OR GLOBE) IN A DUCT
C
C
C    METHOD  - CALLS  VLX TO CALCULATE FLOW ACROSS VALVE. REMAINDER
C              OF ANALYSIS SIMILAR TO DUCT (DE). VALVE OPENING IS INPUT.
C
C
C    LIMITATIONS - AS FOR DUCT COMPONENT DE
C
C
C   WRITTEN BY  ADAM LLOYD AS COMPONENT *VD* IN ECS
C                          LIBRARY     NOV. 1975
C
C    MODIFIED BY  - MAHINDER WAHI                        AUGUST  1977
C
C
C    INPUT/OUTPUT LIST
C
C    T2          OUTLET TEMPERATURE(PORT NO 2)      DEGR      OUTPUT VAR
C    W2          OUTLET FLOW                        LB/MIN    OUTPUT VAR
C    P1          INLET  PRESSURE(PORT NO 1)         PSIA      OUTPUT STATE
C    P1DOT       INLET PRESSURE DERIVATIVE          PSIA/SEC  OUTPUT DERIV
C    IP1         INTEGRATOR CONTROL                 ---       PROGRAM VAR
C    T1          INLET TEMPERATURE                  DEGR      INPUT  VAR
C    W1          INLET FLOW                         LB/MIN    INPUT  VAR
C    P2          OUTLET PRESSURE(PORT NO 2)         PSIA      INPUT  VAR
C    OPE         VALVE OPENING                      ---       INPUT  PARAM
C                   VAL=1.   DEGREES OPEN
C                   VAL=2,3. FRACTIONAL OPENING
C                            (0.LE.OPEN.LE.1.)
C    AL          LENGTH                             FT        INPUT  PARAM
C    D           DIAMETER                           IN        INPUT  PARAM
C    DPO         POPPET DIAMETER                    IN        INPUT  VAR
C                (REQUIRED FOR GLOBE VALVES ONLY)
C    TAM         EFFECTIVE LOCAL AMBIENT TEMP       DEGR      INPUT  PARAM
C    HO          EXTERNAL HEAT TRANSFER COEFFICIENT BTU/FT2   INPUT  PARAM
C                (BASED ON INTERNAL WETTED AREA)    HR DEGR
C    FC          FREQUENCY CONTROL ON P1. (FC.GE.1.)---       INPUT  PARAM
C                A VALUE OF FC GREATER THAN 1.
C                DECREASES FREQUENCY RESPONSE OF P1
C                CORRESPONDINGLY
C    VAL         CODE IDENTIFYING TYPE OF VALVE     ---       INPUT  PARAM
C                =1.  BUTTERFLY VALVE
C                =2.  GATE VALVE
C                =3.  GLOBE VALVE
C
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
C    CALCULATE GAS CONSTANT AND RATIO OF SPECIFIC HEATS, BASED ON
C    INLET TEMPERATURE
      CP=SHCP(T1,0.)
      R=53.3
```

Table 141: LISTING FOR SUBROUTINE DV (CONCLUDED)

```
      GAMMA=1.+R/(778.*CP-R)
      G1=1./(GAMMA-1.)
      G2=(GAMMA-1.)/2.
C     CALCULATE OUTLET FLOW (FLOW THROUGH VALVE) USING SUBROUTINE VLX
      CA=.785398*D*D
      CALL VLX(P1,P2,T1,D,DPO,DPE,VAL,W2)
C     CALCULATE OVERALL CONDUCTANCE (UA) BASED ON AVERAGE FLOW
      WBAR=(ABS(W1)+ABS(W2))/2.
      WBAR=AMAX1(WBAR,.01)
      HINT=HI(1,T1,T1,WBAR,0.,D,AL,0.)
C     THE INPUT VALUE OF THE EXTERNAL HEAT TRANSFER COEFFICIENT IS BASED
C     ON THE INTERNAL WETTED AREA. HENCE UA IS GIVEN BY
      UA=0.004363*D*AL*HINT*HO/(HINT+HO)
C     CONSTANT 0.004363=PI/(60.*12.)     UA IS IN BTU/MIN DEGR
C     CALCULATE OUTLET TEMPERATURE T2
      T2=TAM+(T1-TAM)/EXP(UA/(CP*WBAR))
C     WARNING DIAGNOSTIC IF TEMPERATURE CHANGE EXCEEDS 300 DEGR
      IF(ABS(T1-T2).LE.300.)GO TO 10
      IF(T1.GT.T2) T2=T1-300.
      IF(T2.GT.T1) T2=T1+300.
C     TEST FOR DIAGNOSTIC PRINT OUT
      IF(IERR.NE.1)GO TO 10
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,45HNON FATAL ERROR CALLED FROM COMPONENT DV
     1 10X,35HTEMPERATURE CHANGE EXCEEDS 300 DEGR)
C     CALCULATE INLET PRESSURE BASED ON AVERAGE FLUID TEMPERATURE
 10   TBAR=(T1+T2)/2.
      PBAR=(P1+P2)/2.
C     CALCULATE MACH NUMBER BASED ON AVERAGE CONDITIONS
      AM=AMACH(PBAR,TBAR,CA,WBAR,0.)
      IF(IP1.NE.0)P1DOT=R*TBAR*(W1-W2)*(1.+G2*AM*AM)**G1/(60.*CA*AL*FC)
      RETURN
      END
```

Table 142: LISTING FOR SUBROUTINE DYNFAN

```
CDYNFAN
      SUBROUTINE DYNFAN(QFAN,PFAN)
C DYNAMIC FAN MODEL SUBROUTINE
C INPUT FLOW AND OUTPUT PRESSURE
      COMMON/ESTMD/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
       COMMON/COMPRS/AL0,AL1,AL2,AL3,AL4,G0,G1,G2,G3,G4,QP1
      COMMON/FMERR/FMC
      DATA Q1/0./
C COMPUTE POWER SERIES TERMS
      IF(FMC.GT.0.) Q1=QFAN
      Q2=Q1*Q1
      Q1=QFAN
      Q3=Q2*Q1
      Q4=Q3*Q1
      PFAN=G0+G1*Q1+G2*Q2+G3*Q3+G4*Q4
      IF(Q1.LT.QP2)PFAN=G0+G1*QP2+G2*QP2*QP2+G3*QP2*QP2*QP2+G4*(QP2)**4
     1+(QP2-Q1)*SLOPE
      RETURN
      END
```

Table 143: LISTING FOR SUBROUTINE EC

```
CEC
      SUBROUTINE EC(PR,WN2,TFT,TRT,TSR,TFN,TFP,TBT,TBP,TPO,
     1 PRO,PROD,IPRO,PRR,PRRD,IPRR,X2,X2D,IX2,X3,X3D,IX3,FX,
     2 FZ,TY,TH,FSP,FST,PPU,TPU,W2,T2,THT,C1,C2,C3,C4,C5,C6,C7,
     3 C8,TC1,ZTA,AMN,TC2,GAMX,GAMZ,X0,Z0,PAM,TAM,P2,FAN,BLD,FX1)
C
C  VERSION 4.                                REVISED SEPT 10,1977
C
C  PURPOSE   ENGINE TRANSIENTS MODEL
C
C  METHOD    FIRST ORDER LAG FOR SPINDOWN AND SECOND ORDER LAG FOR
C            SPINUP FOLLOWED BY BODY AXIS TRANSFORMATION.REVERSE
C            THRUST OPERATION ALSO INCLUDED.BLEED AND FAN AIR
C            PRESSURES AND TEMPERATURES ARE THEN CALCULATED AS
C            FUNCTIONS OF THRUST AND MACH NUMBER.
C
C  CALL SEQUENCE
C  *****  TABLES     *****
C    PR              -COMMAND PRESSURE RATIO AS A FUNCTION OF THRUST
C                     LEVER ANGLE(THT); ONE DIMENSIONAL TABLE
C    WN2             -NATURAL FREQUENCY SQUARED AS A FUNCTION OF
C                     ACTUAL ENGINE PRESSURE RATIO(PRO); ONE DIM. TBL
C    TFT             -ENGINE FORWARD THRUST AS A FUNCTION OF MACH NUMBER
C                     (AMN) AND ACTUAL PRESSURE RATIO(PRO); TWO DIM. TBL
C    TRT             -ENGINE REVERSE THRUST AS A FUNCTION OF MACH NUMBER
C                     (AMN) AND ACTUAL PRESSURE RATIO(PRO); TWO DIM. TBL
C    TSR             -ENGINE SPEED AS A FUNCTION OF MACH NO. AND
C                     INSTANTANEOUS THRUST (TWO DIMENSIONAL TABLE)
C    TFN             -TEMPERATURE RISE FAN-INLET TO OUTLET AS A
C                     FUNCTION OF CORRECTED ENG. SPEED(ONE DIM. TABLE)
C    TFP             -PRESSURE RATIO FAN OUTLET TO INLET AS A FUNCTION
C                     OF CORRECTED ENG. SPEED(ONE DIM. TABLE)
C    TBT             -TEMPERATURE RISE COMPRESSOR-INLET TO OUTLET AS A
C                     FUNCTION OF CORRECTED ENG. SPEED(ONE DIM. TABLE)
C    TBP             -PRESSURE RATIO COMPRESSOR OUTLET TO INLET AS A
C                     FUNCTION OF CORRECTED ENG. SPEED(ONE DIM. TABLE)
C    TPO            - CORRECTED BLEED FLOW RATE AS A FUNCTION OF
C                    ENGINE PORT PRESSURE DROP (ONE DIM. TABLE)
C
C  *****  OUTPUTS  *****
C    PRO,PROD,IPRO-ACTUAL ENGINE PRESSURE RATIO,RATE,INT CONTROL.
C    PRR,PRRD,IPRR-REVERSE ENGINE PRESSURE RATIO,RATE,INT CONTROL
C    X2,X2D,IX2      -INTERMEDIATE VALUE OF PROD,RATE,INT CONTROL
C    X3,X3D,IX3      -INTERMEDIATE VALUE OF PROD,RATE,INT CONTROL
C    FX,FZ           -X AND Z AXIS FORCES, LBS
C    TY              -Y AXIS TORQUE(PITCHING MOMENT), FT-LBS
C    TH              -ENGINE THRUST, LBS
C    FSP             -FAN STAGE DELIVERY PRESSURE (PSIA)
C    FST             -FAN STAGE DELIVERY TEMPERATURE (DEG RANKINE)
C    PPU             -BLEED PRESSURE UPSTREAM OF THE PORT (PSIA)
C    TPU             -BLEED TEMPERATURE UPSTREAM OF THE PORT (DEG RANKINE)
C    W2             - BLEED AIR FLOW RATE (LB/MIN) PORT NO 2
C    T2              -BLEED TEMPERATURE DOWNSTREAM OF THE PORT(DEG RANKINE)
C
C  *****  INPUTS   ******
C    THT             -THRUST LEVER ANGLE
```

```
C     C1              -POSITIVE DEADBAND ON THT
C     C2              -NEGATIVE DEADBAND ON THT
C     C3              -POSITIVE SATURATION INTERCEPT(.GT.C1)
C     C4              -NEGATIVE SATURATION INTERCEPT(.LT.C2)
C     C5              -POSITIVE SATURATION LIMIT ON THT
C     C6              -NEGATIVE SATURATION LIMIT ON THT
C     C7              -SATURATION SLOPE(MUST BE +VE)
C     C8              -SATURATION SLOPE(MUST BE +VE)
C     TC1              -ENGINE SPINDOWN TIME CONSTANT, SEC
C     ZTA             -DAMPING RATIO (SPINUP)
C     AMN             -MACH NUMBER
C     TC2              -THRUST REVERSERS TIME CONSTANT, SEC
C     GAMX,GAMZ       -X,Z DIRECTION COSINES
C     XO,ZO           -THRUST LOCATION COMPONENTS FROM C.G., FT
C     PAM             -AMBIENT PRESSURE (PSIA)
C     TAM             -AMBIENT TEMPERATURE (DEGREES RANKINE)
C     P2             - BLEED PRESSURE DOWNSTREAM OF THE PORT(PSIA)PORT NO 2
C     FAN             -INDICATOR FUNCTION FOR ENGINE FAN AIR CALCULATIONS
C                           0= TO BE INCLUDED
C                           1= TO BE EXCLUDED
C     BLD             -INDICATOR FUNCTION FOR ENGINE BLEED AIR CALCULATION
C                           0= TO BE INCLUDED
C                           1= TO BE EXCLUDED
C     FX1            -EXTERNAL FORCE X-AXIS(THRUST REDUCTION DUE TO
C                     USE OF A THRUSTER),LBS
C
C     WRITTEN BY - MAHINDER WAHI                                 APRIL 1977
C
      DIMENSION PR(1),WN2(1),TFT(1),TRT(1)
      DIMENSION  TSR(1),TFN(1),TFP(1),TBT(1),TBP(1),TPO(1)
C              INITIALIZATION
      IF(AMN.EQ. .99999) AMN=0.
      IF(PAM.EQ. .99999) PAM=14.7
      IF(TAM.EQ. .99999) TAM=459.
      IF(FX1.EQ. .99999) FX1=0.
C  PUT LIMITS WITH A DEAD ZONE ON THRUST LEVER ANGLE INPUT.
      CALL SB(THT,THT,C1,C2,C3,C4,C5,C6,C7,C8)
      IF(THT-0.)10,20,20
   10 ISW = 0
      GO TO 30
   20 ISW = 1
   30 CONTINUE
C  CALCULATE ENGINE COMMAND PRESSURE RATIO FROM TABLE LOOKUP ROUTINE
      NPR = PR(2)
      PRI = TBLU1(THT,PR(4),PR(NPR+4),1,-NPR)
      EPS = PRI-PRO
      EP1 = AMAX1(EPS,0.)
      EP2 = AMIN1(EPS,0.)
      IF(IX3.NE.0)X3D =(EP2-X3)/TC1
C  CALCULATE ENGINE SPINUP NATURAL FREQUENCY SQ. FROM TABLE LOOKUP ROUT.
      NWN = WN2(2)
      WNS = TBLU1(PRO,WN2(4),WN2(NWN+4),1,-NWN)
      EM1= EP1*WNS
      WN  = SQRT(WNS)
      IF(IX2.NE.0)X2D = EM1-2.*WN*ZTA*X2
      IF(IPRO.NE.0) PROD =X2+X3
```

```
      IF(ISW.EQ.0) GO TO 40
C   CALCULATE ENGINE FORWARD THRUST FROM STANDARD TABLE LOOKUP ROUTINES
      N1  = TFT(3)+4
      N2  = TFT(2)+TFT(3)+4
      N3  = TFT(2)
      N4  = TFT(3)
      TF  = TBLU2(AMN,PRO,TFT(N1),TFT(4),TFT(N2),1,1,-N3,-N4,N3,N4)
      GO TO 50
   40 IF(IPRR.NE.0) PRRD = (PRO-PRR)/TC2
C   CALCULATE ENGINE REVERSE THRUST FROM STANDARD TABLE LOOKUP ROUTINES
      N1  = TRT(3)+4
      N2  = TRT(2)+TRT(3)+4
      N3  = TRT(2)
      N4  = TRT(3)
      TR  = TBLU2(AMN,PRR,TRT(N1),TRT(4),TRT(N2),1,1,-N3,-N4,N3,N4)
   50 TH  = TR+TF+FX1
C   BODY AXIS TRANSOFRMATION
      FX  = TH*GAMX
      FZ  = TH*GAMZ
      TY  = ZO*FX-XO*FZ
C
      IF(BLD.NE.0. .AND. FAN.NE.0.) GO TO 70
C   RAM RISE/COMPRESSOR INLET CONDITIONS
      PT = PAM*(1+ .2*AMN*AMN)**3.5
      TT = TAM*(1+ .2*AMN*AMN)
C   CORRECTED ENGINE SPEED RATIO
      N1 = TSR(3)+4
      N2 = TSR(2)+TSR(3)+4
      N3 = TSR(2)
      N4 = TSR(3)
      SPD= TBLU2(AMN,TH,TSR(N1),TSR(4),TSR(N2),1,1,-N3,-N4,N3,N4)
      ENC= SPD*SQRT(519./TT)
      IF(FAN.NE.0.) GO TO 60
C    FAN STAGE DELIVERY TEMPERATURE AND PRESSURE
      NX = TFN(2)
      DTF= TBLU1(ENC,TFN(4),TFN(NX+4),1,-NX)
      FST= TT*(1.+DTF)
      NX = TFP(2)
      FPR= TBLU1(ENC,TFP(4),TFP(NX+4),1,-NX)
      FSP= PT*FPR
   60 CONTINUE
      IF(BLD.NE.0.) GO TO 70
C   BLEED AIR PRESSURE AND TEMPERATURE CALCULATIONS
C   PRESSURE AND TEMPERATURE UPSTREAM OF BLEED PORT
      NX = TBT(2)
      DT = TBLU1(ENC,TBT(4),TBT(NX+4),1,-NX)
      TPU= TT*(1.+DT)
      NX = TBP(2)
      CPR = TBLU1(ENC,TBP(4),TBP(NX+4),1,-NX)
      PPU = PT*CPR
      PRAT=PPU/P2
      NX = TPO(2)
      WCR= TBLU1(PRAT,TPO(4),TPO(NX+4),1,-NX)
C   FLOWRATE AND TEMPERATURE DOWNSTREAM OF BLEED PORT
      W2 = WCR*PPU/SQRT(TPU)
      T2= TPU
```

Table 143: LISTING FOR SUBROUTINE EC (CONCLUDED)

```
70 CONTINUE
   RETURN
   END
```

Table 144: LISTING FOR SUBROUTINE EJ

```
CEJ
      SUBROUTINE EJ(TAB,T3,W3,P1,P1DOT,IP1,T1,W1,T2,P2,P3,
     1 ANT,ANE,AK)
C
C  VERSION 2.  SEPT 1977
C
C   PURPOSE - ANALYSIS OF AN EJECTOR WITH CONVERGING-DIVERGING
C             NOZZLE AND SUBSONIC OR CHOKED FLOW CONDITIONS
C
C   METHOD  - USES A TWO DIEMNSIONAL INPUT TABLE OF FLOW RATIO (TOTAL/
C             PRIMARY) AS A FUNCTION OF THE TWO PRESSURE RATIOS (TOTAL/
C             SECONDARY AND PRIMARY/SECONDARY).FOR CHOKED THROAT FLOW
C             UPSTREAM PRESSURE IS COMPUTED TO MATCH FLOW.FOR SUBSONIC
C             FLOW THE EXIT PRESSURE(STATIC) EQUALS SECONDARY SUPPLY
C             PRESSURE.
C
C  CALL SEQUENCE
C  *****  TABLES  *****
C     TAB      -FLOW RATIO(TOTAL/PRIMARY) AS A FUNCTION OF THE TWO
C              -PRESSURE RATIOS(TOTAL/SECONDARY AND PRIMARY/SECONDARY)
C              -TWO DIMENSIONAL TABLE.
C  *****  OUTPUTS *****
C     T3       -OUTLET TEMPERATURE, DEG RANKINE(PORT NO 3)
C     W3       -TOTAL OUTLET FLOW, LB/MIN(PORT NO 3)
C     P1       -INLET PRESSURE PRIMARY AIR SOURCE, PSIA   )PORT
C     P1DOT    -INLET PRESSURE DERIVATIVE, PSIA/SEC       )NO
C     IP1      -INTEGRATOR CONTROL FOR P1                 )1
C  *****  INPUTS  *****
C     T1       -INLET TEMPERATURE PRIMARY SOURCE, DEG RANKINE
C     W1       -INLET FLOW RATE PRIMARY SOURCE, LBS/MIN(PORT NO 1)
C     T2       -INLET TEMPERATUREE SECONDARY SOURCE, DEG RANKINE
C     P2       -INLET PRESSURE SECONDARY AIR SOURCE, PSIA(PORT NO 2)
C     P3       -OUTLET PRESSURE, PSIA(PORT NO 3)
C     ANT      -NOZZLE THROAT AREA, SQFT
C     ANE      -NOZZLE EXIT AREA, SQFT
C     AK       -CONVERGENT-DIVERGENT NOZZLE DIFFUSER LOSS FACTOR
C              -(FOR CONVERGENT NOZZLE, INPUT AK=0., ANE=ANT)
C
C    WRITTEN BY MAHINDER WAHI AND ADAM LLOYD       JUNE 1977
C
      DIMENSION TAB(1)
      DATA GAMMA /1.400/
C    CHECK TO SEE IF CHOKING AT NOZZLE THROAT
      WCHO = 31.9*ANT*P1/SQRT(T1)
      IF(W1.LT.WCHO) GO TO 10
C    CHOKED FLOW - CALCULATE UPSTREAM PRESSURE TO MATCH FLOWS
      P1CAL = W1*SQRT(T1)/(31.9*ANT)
      IF(IP1.NE.0) P1DOT= (P1CAL-P1)/.01
      GO TO 100
C    FLOW IS NOT CHOKED - CALCULATE DYNAMIC HEAD AT THROAT
   10 AM = AMACH(P1,T1,ANT,W1,0.)
      PTS = P1/(1.+(GAMMA-1.)*AM*AM/2.)**(GAMMA/(GAMMA-1.))
      AQ = P1-PTS
      ALOSS = AQ*AK
C    CALCULATE TOTAL PRESSURE AT NOZZLE EXIT
      PE = P1-ALOSS
```

Table 144: LISTING FOR SUBROUTINE EJ (CONCLUDED)

```
      AME = AMACH(PE,T1,ANE,W1,0.)
      PESCAL = PE/(1.+(GAMMA-1.)*AME*AME/2.)**(GAMMA/(GAMMA-1.))
      PERR = P2-PESCAL
      IF(IP1.NE.0) P1DOT= PERR/.01
C
  100 N1= TAB(3)+4
      N2= TAB(2)+TAB(3) +4
      N3= TAB(2)
      N4= TAB(3)
      PRAT1= P3/P2
      PRAT2= P1/P2
      WRAT= TBLU2(PRAT1,PRAT2,TAB(N1),TAB(4),TAB(N2),1,1,-N3,-N4,N3,N4)
      C1=W1*SQRT(T1)*WRAT
      C2=C1/(W1*(T1-T2))
      B=2.*T2+1./(C2*C2)
      T3=(B+SQRT(B*B-4.*T2*T2))/2.
      W3=C1/SQRT(T3)
      RETURN
      END
```

Table 145: LISTING FOR SUBROUTINE ELAS

```
CELAS
      SUBROUTINE ELAS(NSIDE,NEND,SPH,STH,AA,B,LO,EPI,ETI,RA,NU,XR,PTM)
C
C
C     VERSION 2                              REVISED MARCH 1979
C
C     WRITTEN BY - GS DULEBA
C
C     PURPOSE - TO COMPUTE DATA ARRAYS FOR BOTH FREE AND LOADED
C               ACLS ELASTIC TRUNK SHAPES (FOR COMPONENT TS)
C
C     METHOD - SUBROUTINES ENDFS,ENDLS,SIDEFS AND SIDELS ARE CALLED
C              BY QNWT TO DETERMINE TRUNK CROSS-SECTIONAL SHAPES.
C              DATA ARRAYS ARE FILLED WITH CALCULATED TRUNK PARAMETERS.
C
C     OUTPUTS - DATA ARRAYS FOR VARIABLES YO,L1,L2,L3,VC,VS,E1,E3 AND ZO
C               ARE STORED IN COMMON/STRCH/.
C
C     INPUTS - SEE TS NOMENCLATURE FOR ARGUMENT LIST
C
C
      COMMON/ELAST/L2,NPHP,NTHP,NTS,YO,L1,PI,PO2,SR,E1,E3
      COMMON/CIO/IREAD,IWRITE,IDIAG
      COMMON/STRCH/EYO(2560),EL1(2560),EL3(2560),EVC(2560),EVS(2560),
     2 EE1(2560),EE3(2560),EL2(2560),AZO(512)
      DIMENSION AA(1),B(1),LO(1),X(7),R(7),P(150),AJ(7,7),BJ(7,9),IP(8)
      DIMENSION XR(1),SPT(7),SPR(7),PZL(5)
      DIMENSION EPI(1),ETI(1),RA(1),NU(1),SPH(1),STH(1)
      REAL L1,L2,LO,NU,L3
      EXTERNAL ENDFS
      EXTERNAL ENDLS
      EXTERNAL SIDEFS
      EXTERNAL SIDELS
      EQUIVALENCE (X(1),Q),(X(2),V),(X(4),A),(X(5),C),(X(7),L3)
      DATA PZL/1.,.8,.6,.4,.2/
C
      CALL SECOND(CPT)
      WRITE(6,5) CPT
 5    FORMAT(//,20X, F10.3,17H CPU SECONDS ****)
      PI=3.14159265
      PO2=PI/2.
C
      NPR=8
      NZ=5
      NPT=8
      ANZ=NZ
C     *************
      DPT=PTM/(NPT+1)
      DPR=1./NPR
      IP(1)=50
C
      NTHP=STH(2)
      NTHS=STH(3)
      NPHP=SPH(2)
      NPHS=SPH(3)
      DO 10 I=1,NPHP
```

```
 10    P(11+I)=SPH(NPHS+I+3)
       NW=11+2*NPHP
       DO 12 I=1,NTHP
 12    P(NW+I)= STH(NTHS+I+3)
       NPA=11+NPHP
       NPB=11+2*NPHP+NTHP
       NTS=NPA+NPHP+1
C
C    CALCULATE SHAPES FOR TRUNK END SECTIONS
C
       P(1)=0.
       IF(NEND.EQ.0) GO TO 501
       DO 500 I=1,NEND
       SR=XR(I)
       P(2)=AA(I)
       P(3)=B(I)
       P(4)=LO(I)
       P(7)=EPI(I)
       P(8)=ETI(I)
       P(9)=NU(I)
       P(10)=RA(I)
C
       DO 15 IQ=1,NPHP
 15    P(NPA+IQ)=SPH(3+NPHS+NPHP*I+IQ)
       DO 16 IQ=1,NTHP
 16    P(NPB+IQ)=STH(3+NTHS+NTHP*I+IQ)
C
C    SOLVE FOR FREE SHAPE PARAMETER DATA
C
       JQ=0
       DO 20 J=1,NPT
       P(6)=DPT*J
       X(1)=.3*P(4)
       X(2)=X(1)+1.
       X(3)=2.5
       X(4)=4.*X(1)
       X(5)=1.5
       JQ=0
       DO 20 K=1,NPR
       PR=(NPR-K)*DPR
       P(5)=P(6)*(1.-PR)
C
C    END ELEMENT FREE SHAPE CALCULATIONS
C
       NY=0
       GO TO 102
 101   X(1)=.3*P(4)
       X(2)=X(1)+1.
       X(3)=2.5
       X(4)=2.*X(1)
       X(5)=1.5
       NY=1
       JQ=0
 102   CALL QNWT(X,5,7,ENDFS,P,.001,IP,JQ,R,RMS,AJ,BJ)
       IF(RMS.LE.0.001) GO TO 19
       IF(NY.EQ.0) GO TO 101
```

```
      WRITE(6,18) RMS,I,J,K,I
 18   FORMAT(/,2X,F8.3,3I5,5X,30HCONVERGENCE HAS FAILED - ENDFS,
     221H TRUNK SECTION NUMBER,I5)
 19   CONTINUE
      JQ=1
      AZO(NPR*NPT*(I-1)+NPR*(J-1)+K)= P(11)
C
 20   CONTINUE
C
C    SOLVE FOR LOADED SHAPE PARAMETER DATA
C
      X(1)=.3*P(4)
      X(2)=X(1)-1.
      X(3)=2.5
      X(4)=4.*X(1)
      X(5)=X(4)-1.
      X(6)=1.5
      X(7)=0.
      DO 30 J=1,NPT
      IF(J.EQ.1) GO TO 215
      DO 214 M=1,7
 214  X(M)=SPT(M)
 215  CONTINUE
      P(6)=DPT*J
C
      JQ=0
      DO 30 K=1,NPR
      IF(K.EQ.1) GO TO 217
      DO 216 M=1,7
 216  X(M)=SPR(M)
 217  CONTINUE
      PR=(NPR-K)*DPR
      P(5)=P(6)*(1.-PR)
      P11=AZO(NPT*NPR*(I-1)+NPR*(J-1)+K)
      DO 30 L=1,NZ
      P(11)=P11*(NZ+1-L)/ANZ
      IF(L.GT.3) GO TO 888
C
      NY=0
      GO TO 104
 103  X(1)=.3*P(4)
      X(2)=X(1)+1.
      X(3)=2.5
      X(4)=2.*X(1)
      X(5)=X(4)+1.
      X(6)=1.5
      X(7)=.1*P(4)*(L-1)
      NY=1
      JQ=0
 104  CALL QNWT(X,7,7,ENDLS,P,.05,IP,JQ,R,RMS,AJ,BJ)
      IF(RMS.LE.0.05) GO TO 42
      IF(NY.EQ.0) GO TO 103
      WRITE(6,41) RMS,J,K,L,I
 41   FORMAT(/,2X,F8.3,3I5,5X,30HCONVERGENCE HAS FAILED - ENDLS,
     2 21H TRUNK SECTION NUMBER,I5)
 888  IF(L.LT.(NZ-1)) GO TO 30
```

```
      NN= L+NZ*(K-1)+NPR*NZ*(J-1)+NZ*NPR*NPT*(I-1)
      EYO(NN)= TBLU1(PZL(L),PZL(L-3),EYO(NN-3),2,3)
      EL1(NN)= TBLU1(PZL(L),PZL(L-3),EL1(NN-3),2,3)
      EL2(NN)= TBLU1(PZL(L),PZL(L-3),EL2(NN-3),2,3)
      EL3(NN)= TBLU1(PZL(L),PZL(L-3),EL3(NN-3),2,3)
      EE1(NN)= TBLU1(PZL(L),PZL(L-3),EE1(NN-3),2,3)
      EE3(NN)= TBLU1(PZL(L),PZL(L-3),EE3(NN-3),2,3)
      EVC(NN)= TBLU1(PZL(L),PZL(L-3),EVC(NN-3),2,3)
      EVS(NN)= TBLU1(PZL(L),PZL(L-3),EVS(NN-3),2,3)
      GO TO 30
 42   CONTINUE
C
      IF(L.NE.1) GO TO 219
      DO 218 M=1,7
 218  SPR(M)=X(M)
 219  CONTINUE
      IF((L+K).NE.2) GO TO 221
      DO 220 M=1,7
 220  SPT(M)=X(M)
 221  CONTINUE
      JQ=1
      NN= L+NZ*(K-1)+NPR*NZ*(J-1)+NZ*NPR*NPT*(I-1)
      EYO(NN)=YO
      EL1(NN)=L1
      EL2(NN)=L2
      EL3(NN)=X(7)
      EE1(NN)=E1
      EE3(NN)=E3
C
      Y2=P(11)-C
      Y2=AMIN1(Y2,C)
      YC=Y2/C
      A2=.5*A/C*(Y2*SQRT(C*C-Y2*Y2)+C*C*(PO2+ASIN(YC)))
      X2=.5*A*A/C/C*(C*C*(Y2+2.*C/3.)-Y2*Y2*Y2/3.)/A2
      EVC(NN)=A2*(P(10)+YO-X2)
C
      Y1=P(3)+P(11)-V
      Y1=AMIN1(Y1,V)
      YV=Y1/V
      A1=.5*Q/V*(Y1*SQRT(V*V-Y1*Y1)+V*V*(PO2+ASIN(YV)))
      X1=.5*Q*Q*(Y1+2.*V/3. - Y1*Y1*Y1/V/V/3.)/A1
      V1=A1*(P(10)+YO+L3+X1)-P(3)*(P(2)-L3-YO)*(P(10)+.5*(P(2)+L3+YO))
      EVS(NN)=V1+L3*P(11)*(P(10)+YO+.5*L3)+A2*(P(10)+YO-X2)+
     2 .5*P(2)*P(3)*(P(10)+2.*P(2)/3.)
 30   CONTINUE
      CALL SECOND(CPT)
      WRITE(6,5) CPT
      CALL ELWR(I,NPR,NPT,NZ,NEND,DPT,P(2),P(3),P(4),P(9),P(10),P(7),
     2 P(8))
 500  CONTINUE
 501  CONTINUE
C    ********
C    SIDE ELEMENT FREE SHAPE CALCULATIONS
C
      I1=NEND+1
      I2=NEND+NSIDE
```

```
      IF(NSIDE.EQ.0) GO TO 901
      DO 900 I=I1,I2
      P(2)=AA(I)
      P(3)=B(I)
      P(4)=LO(I)
      P(7)=EPI(I)
      P(8)=ETI(I)
      P(9)=NU(I)
      P(10)=RA(I)
C
      DO 36 IQ=1,NPHP
 36   P(NPA+IQ)=SPH(3+NPHS+NPHP*I+IQ)
C
      DO 40 J=1,NPT
      P(6)=DPT*J
      X(1)=.3*LO(I)
      X(2)=2.5
      X(3)=1.5
      JQ=0
      DO 40 K=1,NPR
      PR=(NPR-K)*DPR
      P(5)=P(6)*(1.-PR)
C
      NY=0
      GO TO 106
 105  X(1)=.4*P(4)
      X(2)=2.3
      X(3)=1.3
      NY=1
      JQ=0
 106  CALL QNWT(X,3,7,SIDEFS,P,.001,IP,JQ,R,RMS,AJ,BJ)
      IF(RMS.LE.0.001) GO TO 39
      IF(NY.EQ.0) GO TO 105
      WRITE(6,38) RMS,I,J,K,I
 38   FORMAT(/,5X,E15.4, 3I5,5X, 31HCONVERGENCE HAS FAILED - SIDEFS,
     2 21H TRUNK SECTION NUMBER,I5)
 39   CONTINUE
      JQ=1
      AZO(NPR*NPT*(I-1)+NPR*(J-1)+K)= P(11)
 40   CONTINUE
C
C    SOLVE FOR LOADED SHAPE PARAMETER DATA
C
      X(1)=.3*P(4)
      X(2)=2.5
      X(3)=1.5
      X(4)=0.
C
      DO 50 J=1,NPT
      P(6)=DPT*J
      JQ=0
      IF(J.EQ.1) GO TO 315
      DO 314 M=1,4
 314  X(M)=SPT(M)
 315  CONTINUE
      DO 50 K=I,NPR
```

```
      IF(K.EQ.1) GO TO 317
      DO 316 M=1,4
 316  X(M)=SPR(M)
 317  CONTINUE
      PR=(NPR-K)*DPR
      P(5)=P(6)*(1.-PR)
      DO 50 L=1,NZ
      P(11)=AZO(NPT*NPR*(I-1)+NPR*(J-1)+K)*(NZ+1-L)/ANZ
C
      NY=0
      GO TO 108
 107  X(1)=.4*P(4)
      X(2)=2.3
      X(3)=1.5
      X(4)=.1*P(4)*(L-1)
      NY=1
      JQ=0
 108  CALL QNWT(X,4,7,SIDELS,P,.001,IP,JQ,R,RMS,AJ,BJ)
      IF(RMS.LE.0.001)`GO TO 48
      IF(NY.EQ.0) GO TO 107
      WRITE(6,47) RMS,I,J,K,I
 47   FORMAT(/,5X,E15.4,3I5,5X,31HCONVERGENCE HAS FAILED - SIDELS,
     2 21H TRUNK SECTION NUMBER,I5)
 48   CONTINUE
      IF(L.NE.1) GO TO 319
      DO 318 M=1,4
 318  SPR(M)=X(M)
 319  CONTINUE
      IF((L+K).NE.2) GO TO 321
      DO 320 M=1,4
 320  SPT(M)=X(M)
 321  CONTINUE
      JQ=1
      NN= L+NZ*(K-1)+NPR*NZ*(J-1)+NZ*NPR*NPT*(I-1)
C
      S3=SIN(X(3))
      R2=X(1)*P(6)/P(5)
      YO=R2*S3
      EYO(NN)=YO
      EL1(NN)=X(1)*X(2)
      EL2(NN)=R2*X(3)
      EL3(NN)=X(4)
      EE1(NN)=E1
      EE3(NN)=E1
C
      ACV=.25*R2*R2*(2.*X(3)-SIN(2.*X(3)))
      EVC(NN)= ACV
      EVS(NN)=ACV + .25*X(1)*X(1)*(2.*X(2) + SIN(2.*X(2)))
     2 + (P(11)-X(1))*(AA(I)-YO) + X(1)*X(4) + .5*P(2)*P(3)
 50   CONTINUE
      CALL ELWR(I,NPR,NPT,NZ,NEND,DPT,P(2),P(3),P(4),P(9),P(10),P(7),
     2 P(8))
 900  CONTINUE
 901  CONTINUE
      RETURN
      END
```

Table 146: LISTING FOR SUBROUTINE ELFX

```
CELFX
      SUBROUTINE ELFX(TH,PHI,E)
C
C
C   VERSION 2                              REVISED MARCH 1979
C
C   WRITTEN BY - GS DULEBA
C
C   PURPOSE - TO COMPUTE AN INCOMPLETE ELLIPTIC INTEGRAL OF THE
C             SECOND KIND
C
C   OUTPUTS -
C       E     VALUE OF INTEGRAL
C
C   INPUTS  -
C       TH    INPUT MODULUS
C       PHI   INPUT PARAMETER
C
C
      IF(PHI.LT.0..OR.TH.LT.0.) GO TO 77
      GO TO 80
 77   E=0.
      RETURN
 80   CONTINUE
      AK=SIN(TH)
      SI=SIN(PHI)
      CO=COS(PHI)
      AN=1.
      AD=2.
      S1=.5*PHI-.25*SIN(2.*PHI)
      E=PHI-.5*AK*AK*S1
      I=1
 20   I=I+1
      IF(I.GT.20) GO TO 200
      AI=I
      N=2*I
      BN=N
      AN=AN*(2.*AI-3.)
      AD=2.*AD*AI
      S2=((BN-1.)*S1-CO*SI**(BN-1.))/BN
      S1=S2
      DE=AN/AD*S2*AK**BN
      E=E-DE
      IF((DE*DE).LT.1.E-06) GO TO 200
      GO TO 20
 200  RETURN
      END
```

Table 147: LISTING FOR SUBROUTINE ELKX

```
CELKX
      SUBROUTINE ELKX(TH,E)
C
C
C    VERSION 2                                    REVISED MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO COMPUTE A COMPLETE ELLIPTIC INTEGRAL OF THE SECOND
C              KIND
C
C    OUTPUTS -
C       E      VALUE F INTEGRAL
C
C    INPUTS  -
C       TH     INPUT MODULUS
C
C
      DATA PI2/1.570796/
      IF(TH.LT.0.) GO TO 77
      GO TO 80
 77   E=0.
      RETURN
 80   AK=SIN(TH)
      AN=1.
      AD=2.
      E=PI2*(1.-.25*AK*AK)
      I=1
 20   I=I+1
      IF(I.GT.20) GO TO 200
      AI=I
      N=2*I
      BN=N
      AN=AN*(2.*AI-1.)
      AD=AD*2.*AI
      DE=PI2*AN*AN/AD/AD/(BN-1.)*AK**BN
      E=E-DE
      IF((DE*DE).LT.1.E-06) GO TO 200
      GO TO 20
 200  RETURN
      END
```

Table 148: LISTING FOR SUBROUTINE ELWR

```
CELWR
      SUBROUTINE ELWR(I,NPR,NPT,NZ,NEND,DPT,A,B,LO,NU,RA,EPI,ETI)
C
C
C    VERSION 2                              REVISED MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO PRINT COMPUTED DATA ARRAYS FOR ACLS ELASTIC
C              TRUNK COMPONENT TS.
C
C    OUTPUTS - NONE
C
C    INPUTS  - SEE TS NOMENCLATURE FOR ARGUMENT LIST VARIABLES
C
C
      COMMON/STRCH/ EYO(2560),EL1(2560),EL3(2560),EVC(2560),
     2 EVS(2560),EE1(2560),EE3(2560),EL2(2560),AZO(512)
      COMMON/CIO/IREAD,IWRITE,IDIAG
      REAL LO,NU
      NI=NPR*NPT*(I-1)
      NPT2=NPT/2
      IF(I.NE.1) GO TO 199
      WRITE(IWRITE,100)
      WRITE(IWRITE,200)
      WRITE(IWRITE,300)
      WRITE(IWRITE,400)
 199  CONTINUE
      IF(I.GT.NEND) GO TO 5
      WRITE(IWRITE,101) I,A,B,LO,NU,RA,EPI,ETI
      GO TO 8
 5    WRITE(IWRITE,103) I,A,B,LO,NU,EPI,ETI
 8    CONTINUE
 101  FORMAT(///12H*** DATA SET,I2,23H *** TRUNK END ELEMENT ,
     2 4(4H****)/10X,2HA=,F6.2, 10X,2HB=,F6.2, 10X,3HLO=,F6.2,
     3 8X,3HNU=,F5.2,8X,3HRA=,F6.2,8X,4HEPI=,F6.3,8X,4HETI=,F6.3//
     4 53X,18H*** ZOFS ARRAY ***/)
 103  FORMAT(///,5X,12H*** DATA SET,I2,24H *** TRUNK SIDE ELEMENT ,
     2 4(4H****)/10X,2HA=,F6.2, 10X,2HB=,F6.2, 10X,3HLO=,F6.2,
     3 10X,3HNU=,F5.2,10X,4HEPI=,F6.3,10X,4HETI=,F6.3, //53X,18H*** ZO
     4FS ARRAY ***/)
C
C    OUTPUT ARRAYS
C
      DO 10 M=1,NPR
      K=NPR+1-M
      WRITE(IWRITE,105) (AZO(NI+NPR*(J-1)+K),J=1,NPT)
 10   CONTINUE
C
      WRITE(IWRITE,110)
      DO 20 J=1,NPT2
      PT1=DPT*(2*J-1)
      PT2=PT1+DPT
      WRITE(IWRITE,113) PT1,PT2
      DO 20 M=1,NPR
      K=NPR+1-M
```

```
      N1=NI*NZ+NZ*(K-1)+NPR*NZ*(2*J-2)
      N2=N1+NPR*NZ
      WRITE(IWRITE,115) (EYO(L+N1),L=1,NZ), (EYO(L+N2),L=1,NZ)
 20   CONTINUE
C
      WRITE(IWRITE,120)
      DO 30 J=1,NPT2
      PT1=DPT*(2*J-1)
      PT2=PT1+DPT
      WRITE(IWRITE,113) PT1,PT2
      DO 30 M=1,NPR
      K=NPR+1-M
      N1=NI*NZ+NZ*(K-1)+NPR*NZ*(2*J-2)
      N2=N1+NPR*NZ
      WRITE(IWRITE,115) (EL1(L+N1),L=1,NZ), (EL1(L+N2),L=1,NZ)
 30   CONTINUE
C
      WRITE(IWRITE,150)
      DO 50 J=1,NPT2
      PT1=DPT*(2*J-1)
      PT2=PT1+DPT
      WRITE(IWRITE,113) PT1,PT2
      DO 50 M=1,NPR
      K=NPR+1-M
      N1=NI*NZ+NZ*(K-1)+NPR*NZ*(2*J-2)
      N2=N1+NPR*NZ
      WRITE(IWRITE,115) (EL2(L+N1),L=1,NZ), (EL2(L+N2),L=1,NZ)
 50   CONTINUE
      WRITE(IWRITE,135)
      DO 40 J=1,NPT2
      PT1=DPT*(2*J-1)
      PT2=PT1+DPT
      WRITE(IWRITE,113) PT1,PT2
      DO 40 M=1,NPR
      K=NPR+1-M
      N1=NI*NZ+NZ*(K-1)+NPR*NZ*(2*J-2)
      N2=N1+NPR*NZ
      WRITE(IWRITE,116) (EL3(L+N1),L=1,NZ), (EL3(L+N2),L=1,NZ)
 40   CONTINUE
C
C
C
 100  FORMAT(39H1***** TRUNK ELEMENT SECTION PROPERTIES ,12(6H******))
 105  FORMAT(20X,8F10.2)
 110  FORMAT(//54X,16H*** YO ARRAY ***)
 113  FORMAT(//20X, 12HTRUNK PRES =, F5.2, 47X, 12HTRUNK PRES =, F5.2)
 115  FORMAT(2X,5F11.2,12X,5F11.2)
 116  FORMAT(2X,F11.0,4F11.2, 12X,F11.0,4F11.2)
 120  FORMAT(///54X,16H*** L1 ARRAY ***/)
 135  FORMAT(///54X,16H*** L3 ARRAY ***/)
 150  FORMAT(///54X,16H*** L2 ARRAY ***/)
 200  FORMAT(//,6X,67HFOR ALL ARRAYS- ROWS 1-8 CORRESPOND TO PR= 0, 1/8,
     2 2/8, . . . , 7/8)
 300  FORMAT(/,6X,75HFOR ZOFS ARRAY- COLUMNS 1-8 CORRESPOND TO PT= PTM*(
     2 1, 2, 3, . . . ,8)/9    )
 400  FORMAT(/,6X,75HFOR ALL OTHER ARRAYS- COLUMNS 1-5 CORRESPOND TO ZO/
```

Table 148: LISTING FOR SUBROUTINE ELWR (CONCLUDED)

```
2ZOFS= 1, .8, .6, .4, .2 )
 RETURN
 END
```

Table 149: LISTING FOR SUBROUTINE ENDFS

```
CENDFS
      SUBROUTINE ENDFS(Y,M,K,R,P)
C
C
C    VERSION 2                              REVISED MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO COMPUTE THE SHAPE PARAMETERS FOR AN ELASTIC TRUNK
C              (COMPONENT TS) END ELEMENT IN THE FREE OR UNLOADED
C              CONFIGURATION
C
C    METHOD - TRUNK MERIDIAN IS ASSUMED IN THE SHAPE OF TWO ELLIPTICAL
C             ARCS(INNER AND OUTER).  LOADS IN THE HOOP AND MERIDIAN
C             DIRECTIONS ARE CALCULATED FROM ASSUMED SHAPE USING
C             MEMBRANE THEORY. HOOP AND MERIDIAN STRAINS ARE FOUND
C             FROM LOAD/DEFLECTION CURVES AND MUST BE COMPATIBLE
C             WITH ASSUMED SHAPE FOR A VALID SOLUTION.
C
C    NOMENCLATURE
C       PR      PRESSURE RATIO (PT-PC)/(PT-PA)
C       Q       MAJOR AXIS FOR OUTER ELLIPSE
C       V       MINOR AXIS FOR OUTER ELLIPSE
C       PH1     SWEPT ANGLE FOR OUTER ELLIPSE
C       A       MAJOR AXIS FOR INNER ELLIPSE
C       C       MINOR AXIS FOR INNER ELLIPSE
C       G2      SWEPT ANGLE FOR INNER ELLIPSE
C
C
      COMMON/ELAST/L2,NPH,NTH,NTS,YO,L1,PI,PO2,SR,EP1,EP2
      REAL L1,L2,L10,L20,NP,NT,NET,NEP
      DIMENSION Y(5),R(5),P(70)
C
      PR=  P(5)/P(6)
      Q=Y(1)
      V=Y(2)
      PH1=Y(3)
      A=Y(4)
      G2=Y(5)
      C= PR*A*A*V/Q/Q
      SG=SIN(G2)
      CG=COS(G2)
      RE=A*C/SQRT(A*A*CG*CG+C*C*SG*SG)
C
      YO=RE*SG
      C2=COS(PH1)
      S2=SIN(PH1)
C
      RA=Q*V/SQRT(Q*Q*C2*C2+V*V*S2*S2)
      ZA=RA*S2
      R2Q=SQRT(Q*Q*(Q*Q-ZA*ZA)/V/V + ZA*ZA)
      PH2=ASIN(ZA/R2Q)
      IF(PH1.GT.PO2) PH2=PI-PH2
C    CALCULATE MERIDIAN LENGTH L2 FOR INNER ELLIPSE
C
C
```

```
      AYC=A*A*(A*A-YO*YO)
      IF(AYC.LT.0.) AYC=-AYC
      R2P=SQRT(AYC/C/C+YO*YO)
      PH3=ASIN(YO/R2P)
      IF(G2.GT.PO2) PH3=PI-PH3
      Z=0.
      AC=A*A-C*C
      YY=1.
      IF(AC.LT.0.) GO TO 80
      SQAC=YY*SQRT(AC)
      THE=ASIN(SQAC/A)
C
      IF(G2.GE.PO2) GO TO 76
      W=ASIN(RE*SG/A)
      CALL ELFX(THE,W,E)
      L2=A*E
      GO TO 78
 76   G3=PI-G2
      CALL ELKX(THE,E1)
      W=ASIN(RE*SIN(G3)/A)
      CALL ELFX(THE,W,E2)
      L2=A*(2.*E1-E2)
 78   CONTINUE
      GO TO 84
 80   CONTINUE
      ACC=YY*SQRT(-AC)/C
      THE=ASIN(ACC)
      IF(G2.GE.PO2) GO TO 82
      CALL ELKX(THE,E1)
      G3=PO2-G2
      W=ASIN(RE*SIN(G3)/C)
      CALL ELFX(THE,W,E2)
      L2=C*(E1-E2)
      GO TO 84
 82   G3=G2-PO2
      CALL ELKX(THE,E1)
      W=ASIN(RE*SIN(G3)/C)
      CALL ELFX(THE,W,E2)
      L2=C*(E1+E2)
 84   CONTINUE
C
C    CALCULATE MERIDIAN LENGTH L1 FOR OUTER ELLIPSE
      QV=Q*Q-V*V
      YY=1.
      IF(QV.LT.0.) GO TO 180
      SQV=YY*SQRT(QV)
      THE=ASIN(SQV/Q)
      IF(PH1.GT.(2.*PI)) GO TO 179
      IF(PH1.GT.(1.5*PI)) GO TO 177
      IF(PH1.GT.PI) GO TO 175
      IF(PH1.GE.PO2) GO TO 176
      W=ASIN(RA*S2/Q)
      CALL ELFX(THE,W,E)
      L1=Q*E
      GO TO 184
 175  G3=PH1-PI
```

```
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/Q)
      CALL ELFX(THE,W,E2)
      L1=Q*(2.*E1+E2)
      GO TO 184
 176  G3=PI-PH1
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/Q)
      CALL ELFX(THE,W,E2)
      L1=Q*(2.*E1-E2)
      GO TO 184
 177  CONTINUE
      G3=2.*PI-PH1
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/Q)
      CALL ELFX(THE,W,E2)
      L1=Q*(4.*E1-E2)
      GO TO 184
 179   G3=PH1-2.*PI
      CALL ELKX(THE,E1)
      L1=Q*4.*E1-100.*G3
      GO TO 184
 180  QVV=YY*SQRT(-QV)/V
      THE=ASIN(QVV)
      IF(PH1.GT.PI) GO TO 181
      IF(PH1.GE.PO2) GO TO 182
      CALL ELKX(THE,E1)
      G3=PO2-PH1
      W=ASIN(RA*SIN(G3)/V)
      CALL ELFX(THE,W,E2)
      L1=V*(E1-E2)
      GO TO 184
 181  G3=1.5*PI-PH1
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/V)
      CALL ELFX(THE,W,E2)
      L1=V*(3.*E1-E2)
      GO TO 184
 182   G3=PH1-PO2
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/V)
      CALL ELFX(THE,W,E2)
      L1=V*(E1+E2)
 184   CONTINUE
C
      B=P(10)+YO
      ET1=0.
      ET2=0.
      RRS1=0.
      RRS2=0.
      EP1=0.
      EP2=0.
      SR1=0.
      SR2=0.
C
C   DIVIDE OUTER ELLIPSE INTO J1 FINITE LENGTHS AND COMPUTE
```

```
C    LOADS AND STRAINS AT EACH POINT
      J1=20.*L1/(L1+L2)
      IF(J1.LT.1) J1=1
      IF(J1.GT.19) J1=19
      J2=20-J1
      DT1=PH2/J1
      DT2=PH3/J2
C
      RTH=P(4)/(1.+P(7))*SR + P(10)
C
      TH=PI-PH2-.5*DT1
      DO 65 J=1,J1
      TH=TH+DT1
      STH=SIN(TH)
      CTH=COS(TH)
C
      R2Q=Q*Q/SQRT(Q*Q*STH*STH + V*V*CTH*CTH)
      R1Q=R2Q*R2Q*R2Q*V*V/Q/Q/Q/Q
      RR=B+R2Q*STH
      R22=B/STH+R2Q
      NT=P(6)*R22*(1.-R2Q*(B+RR)/2./RR/R1Q)
      NP=P(6)*R2Q*(B+RR)/2./RR
      IF(SR.EQ.0.) RTH=RR
      NEP=RR/RTH*(NP-P(9)*NT)
      NET=(NT-P(9)*NP)*RR/RTH
      EP=R1Q*TBLU1(NEP,P(12),P(NPH+12),1,NPH)
      ET=R1Q*TBLU1(NET,P(NTS),P(NTH+NTS),1,NTH)
      SR1=SR1+R1Q
      Z=J
      EP1=EP1+EP
      ET1=ET1+ET
 65   RRS1=RRS1+RR
C
C
C    DIVIDE INNER ELLIPSE INTO J2 FINITE LENGTHS AND COMPUTE
C    LOADS AND STRAINS AT EACH POINT
      TH=-.5*DT2+PI
      DO 75 J=1,J2
      TH=TH+DT2
      STH=SIN(TH)
      CTH=COS(TH)
C
      R2P=A*A/SQRT(A*A*STH*STH + C*C*CTH*CTH)
      RR1=R2P*R2P*R2P*C*C/A/A/A/A
      RO=B+R2P*STH
      RO=AMAX1(RO,.1)
      NT=P(5)*(R2P+B/STH)*(1.-R2P*(B+RO)/2./RO/RR1)
      NP=P(5)*R2P*(B+RO)/2./RO
      IF(SR.EQ.0.) RTH=RO
      NEP=RO/RTH*(NP-P(9)*NT)
      NET=(NT-P(9)*NP)*RO/RTH
      EP=RR1*TBLU1(NEP,P(12),P(NPH+12),1,NPH)
      ET=RR1*TBLU1(NET,P(NTS),P(NTH+NTS),1,NTH)
      SR2=SR2+RR1
C
      ET2=ET2+ET
```

Table 149: LISTING FOR SUBROUTINE ENDFS (CONCLUDED)

```
      EP2=EP2+EP
 75   RRS2=RRS2+RO
C
C   COMPUTE AVERAGE STRAINS
      EP1=EP1/SR1
      EP2=EP2/SR2
      ET1=ET1/SR1
      ET2=ET2/SR2
C
C
      L1O=(1.+P(7))*L1/(1.+EP1)
      L2O=(1.+P(7))*L2/(1.+EP2)
C
C   COMPUTE RESIDUALS (=0. AT SOLUTION)
      R(1)=P(4) - L1O -L2O
      R(2)= P(2)- ZA - YO
      R(3)= P(3)- V+RA*C2 + C - RE*CG
C
C
      D2=P(2)*L2O/P(4)
      DO2= P(10) + .5*D2
      DO1= P(10) + D2 + .5*(P(2)-D2)
      IF(Z.EQ.0.) GO TO 110
      R41=ET1 - ((1.+P(8))*RRS1/DO1/Z - 1.)
      GO TO 115
 110  R41=0.
 115  IF(Z.EQ.20.) GO TO 120
      R42=ET2 - ((1.+P(8))*RRS2/DO2/(20.-Z) - 1.)
      GO TO 125
 120  R42=0.
 125  CONTINUE
      R(4)=R41
      R(5)=R42
      P(11)=C-RE*CG
      RETURN
      END
```

Table 150: LISTING FOR SUBROUTINE ENDLS

```
CENDLS
      SUBROUTINE ENDLS(Y,M,K,R,P)
C
C
C    VERSION 2                               REVISED MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO COMPUTE THE SHAPE PARAMETERS FOR AN ELASTIC TRUNK
C              (COMPONENT TS) END ELEMENT IN THE LOADED CONFIGURATION
C
C    METHOD - TRUNK MERIDIAN IS ASSUMED IN THE SHAPE OF TWO ELLIPTICAL
C             ARCS(INNER AND OUTER).  LOADS IN THE HOOP AND MERIDIAN
C             DIRECTIONS ARE CALCULATED FROM ASSUMED SHAPE USING
C             MEMBRANE THEORY. HOOP AND MERIDIAN STRAINS ARE FOUND
C             FROM LOAD/DEFLECTION CURVES AND MUST BE COMPATIBLE
C             WITH ASSUMED SHAPE FOR A VALID SOLUTION.
C
C    NOMENCLATURE
C       PR      PRESSURE RATIO (PT-PC)/(PT-PA)
C       Q       MAJOR AXIS FOR OUTER ELLIPSE
C       V       MINOR AXIS FOR OUTER ELLIPSE
C       PH1     SWEPT ANGLE FOR OUTER ELLIPSE
C       A       MAJOR AXIS FOR INNER ELLIPSE
C       C       MINOR AXIS FOR INNER ELLIPSE
C       G2      SWEPT ANGLE FOR INNER ELLIPSE
C       L3      MEMBRANE LENGTH IN GROUND CONTACT
C
C
      COMMON/ELAST/L2,NPH,NTH,NTS,YO,L1,PI,PO2,SR,EP1,EP3
      REAL L1,L2,L3,L10,L20,L30,NP,NT,NET,NEP
      DIMENSION ANT(2),ANP(2),Y(7),R(7),P(70)
C
      PR=  P(5)/P(6)
      Q=Y(1)
      V=Y(2)
      PH1=Y(3)
      A=Y(4)
      C=Y(5)
      G2=Y(6)
      L3=Y(7)
C
      SG=SIN(G2)
      CG=COS(G2)
      RE=A*C/SQRT(A*A*CG*CG+C*C*SG*SG)
C
      YO=RE*SG
      C2=COS(PH1)
      S2=SIN(PH1)
C
      RA=Q*V/SQRT(Q*Q*C2*C2+V*V*S2*S2)
      ZA=RA*S2
      R2Q=SQRT(Q*Q*(Q*Q-ZA*ZA)/V/V + ZA*ZA)
      PH2=ASIN(ZA/R2Q)
      IF(PH1.GT.PO2) PH2=PI-PH2
C
```

```
C     COMPUTE MERIDIAN LENGTH L2 FOR INNER ELLIPSE
         AYC=A*A*(A*A-YO*YO)
         IF(AYC.LT.0.) AYC=-AYC
         R2P=SQRT(AYC/C/C+YO*YO)
         PH3=ASIN(YO/R2P)
         IF(G2.GT.PO2) PH3=PI-PH3
         DTH=.05*(PH2+PH3)
         THO=PI-PH2
         TH=THO-.5*DTH
         Z=0.
         AC=A*A-C*C
         YY=1.
         IF(AC.LT.0.) GO TO 80
         SQAC=YY*SQRT(AC)
         THE=ASIN(SQAC/A)
C
C
         IF(G2.GE.PO2) GO TO 76
         W=ASIN(RE*SG/A)
         CALL ELFX(THE,W,E)
         L2=A*E
         GO TO 78
  76     G3=PI-G2
         CALL ELKX(THE,E1)
         W=ASIN(RE*SIN(G3)/A)
         CALL ELFX(THE,W,E2)
         L2=A*(2.*E1-E2)
  78     CONTINUE
         GO TO 84
  80     CONTINUE
         ACC=YY*SQRT(-AC)/C
         THE=ASIN(ACC)
         IF(G2.GE.PO2) GO TO 82
         CALL ELKX(THE,E1)
         G3=PO2-G2
         W=ASIN(RE*SIN(G3)/C)
         CALL ELFX(THE,W,E2)
         L2=C*(E1-E2)
         GO TO 84
  82     G3=G2-PO2
         CALL ELKX(THE,E1)
         W=ASIN(RE*SIN(G3)/C)
         CALL ELFX(THE,W,E2)
         L2=C*(E1+E2)
  84     CONTINUE
C
C     COMPUTE MERIDIAN LENGTH L1 FOR OUTER ELLIPSE
         QV=Q*Q-V*V
         YY=1.
         IF(QV.LT.0.) GO TO 180
         SQV=YY*SQRT(QV)
         THE=ASIN(SQV/Q)
         IF(PH1.GT.(2.*PI)) GO TO 179
         IF(PH1.GT.(1.5*PI)) GO TO 177
         IF(PH1.GT.PI) GO TO 175
         IF(PH1.GE.PO2) GO TO 176
```

```
      W=ASIN(RA*S2/Q)
      CALL ELFX(THE,W,E)
      L1=Q*E
      GO TO 184
 175  G3=PH1-PI
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/Q)
      CALL ELFX(THE,W,E2)
      L1=Q*(2.*E1+E2)
      GO TO 184
 176  G3=PI-PH1
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/Q)
      CALL ELFX(THE,W,E2)
      L1=Q*(2.*E1-E2)
      GO TO 184
 177  CONTINUE
      G3=2.*PI-PH1
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/Q)
      CALL ELFX(THE,W,E2)
      L1=Q*(4.*E1-E2)
      GO TO 184
 179   G3=PH1-2.*PI
      CALL ELKX(THE,E1)
      L1=Q*4.*E1-100.*G3
      GO TO 184
 180  QVV=YY*SQRT(-QV)/V
      THE=ASIN(QVV)
      IF(PH1.GT.PI) GO TO 181
      IF(PH1.GE.PO2) GO TO 182
      CALL ELKX(THE,E1)
      G3=PO2-PH1
      W=ASIN(RA*SIN(G3)/V)
      CALL ELFX(THE,W,E2)
      L1=V*(E1-E2)
      GO TO 184
 181  G3=1.5*PI-PH1
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/V)
      CALL ELFX(THE,W,E2)
      L1=V*(3.*E1-E2)
      GO TO 184
 182   G3=PH1-PO2
      CALL ELKX(THE,E1)
      W=ASIN(RA*SIN(G3)/V)
      CALL ELFX(THE,W,E2)
      L1=V*(E1+E2)
 184   CONTINUE
C
      B=P(10)+YO
      B1=B+L3
      ET1=0.
      ET2=0.
      RRS1=0.
      RRS2=0.
```

```
      EP1=0.
      EP2=0.
      SR1=0.
      SR2=0.
C
C   DIVIDE OUTER ELLIPSE INTO J1 FINITE LENGTHS AND COMPUTE
C   LOADS AND STRAINS AT EACH POINT
      J1=20.*L1/(L1+L2)
      IF(J1.LT.1) J1=1
      IF(J1.GT.19) J1=19
      J2=20-J1
      DT1=PH2/J1
      DT2=PH3/J2
C
      RTH=SR*P(4)/(1.+P(7)) + P(10)
C
      TH=PI-PH2-.5*DT1
C
      DO 65 J=1,J1
      TH=TH+DT1
      STH=SIN(TH)
      CTH=COS(TH)
C
      R2Q=Q*Q/SQRT(Q*Q*STH*STH + V*V*CTH*CTH)
      R1Q=R2Q*R2Q*R2Q*V*V/Q/Q/Q/Q
      RR=B1+R2Q*STH
      NT=P(6)*(R2Q+B1/STH)*(1.-R2Q*(B1+RR)/2./RR/R1Q)
      NP=P(6)*R2Q*(B1+RR)/2./RR
      IF(SR.EQ.0.) RTH=RR
      NEP=RR/RTH*(NP-P(9)*NT)
      NET=(NT-P(9)*NP)*RR/RTH
      EP=R1Q*TBLU1(NEP,P(12),P(NPH+12),1,NPH)
      ET=R1Q*TBLU1(NET,P(NTS),P(NTH+NTS),1,NTH)
      SR1=SR1+R1Q
      ANP(1)=EP/R1Q
      ANT(1)=ET/R1Q
      Z=J
      EP1=EP1+EP
      ET1=ET1+ET
 65   RRS1=RRS1+RR
C
C   DIVIDE INNER ELLIPSE INTO J2 FINITE LENGTHS AND COMPUTE
C   LOADS AND STRAINS AT EACH POINT
      TH=PI-.5*DT2
      DO 75 J=1,J2
      TH=TH+DT2
      STH=SIN(TH)
      CTH=COS(TH)
C
      R2P=A*A/SQRT(A*A*STH*STH + C*C*CTH*CTH)
      RR1=R2P*R2P*R2P*C*C/A/A/A/A
      RO=B+R2P*STH
      RO=AMAX1(.1,RO)
      NT=P(5)*(R2P+B/STH)*(1.-R2P*(B+RO)/2./RO/RR1)
      NP=P(5)*R2P*(B+RO)/2./RO
      IF(SR.EQ.0.) RTH=RO
```

Table 150: LISTING FOR SUBROUTINE ENDLS (CONCLUDED)

```
      NEP=RO/RTH*(NP-P(9)*NT)
      NET=(NT-P(9)*NP)*RO/RTH
      EP=RR1*TBLU1(NEP,P(12),P(NPH+12),1,NPH)
      ET=RR1*TBLU1(NET,P(NTS),P(NTH+NTS),1,NTH)
      SR2=SR2+RR1
C
      IF(J.GT.1) GO TO 70
      ANP(2)=EP/RR1
      ANT(2)=ET/RR1
 70   CONTINUE
      ET2=ET2+ET
      EP2=EP2+EP
 75   RRS2=RRS2+RO
C
C   COMPUTE AVERAGE STRAINS
      EP1=EP1/SR1
      EP2=EP2/SR2
      EP3= .5*(ANP(1)+ANP(2))
      ET3= .5*(ANT(1)+ANT(2))
      ET1=(ET1*DT1+L3*ET3)/(SR1*DT1+L3)
      ET2=ET2/SR2
C
C
      L10= (1.+P(7))*L1/(1.+EP1)
      L20= (1.+P(7))*L2/(1.+EP2)
      L30= (1.+P(7))*L3/(1.+EP3)
C
C   COMPUTE RESIDUALS (=0. AT SOLUTION)
      R(1)= P(4) - L10 - L20 - L30
      R(2)= P(2)-L3 - YO - ZA
      R(3)= P(3)- V+RA*C2 + C - RE*CG
C
C
C
      D2= L20*P(2)/P(4)
      DO2= P(10) + .5*D2
      DO1=P(10) + D2 + .5*P(2)/P(4)*(P(4)-L20)
      IF(Z.EQ.0.) GO TO 110
      R41=ET1 - ((1.+P(8))*RRS1/DO1/Z - 1.)
      GO TO 115
 110  R41=0.
 115  IF(Z.EQ.20.) GO TO 120
      R42=ET2 - ((1.+P(8))*RRS2/DO2/(20.-Z) - 1.)
      GO TO 125
 120  R42=0.
 125  CONTINUE
      R(4)=R41
      R(5)=R42
      R(6)=C-PR*A*A*V/Q/Q*(.25*L3+B)/(.75*L3+B)
      R(7)=P(11) - C + RE*CG
      RETURN
      END
```

Table 151: LISTING FOR SUBROUTINE ES

```
CES
      SUBROUTINE ES(TSR,TFN,TFP,TBT,TBP,TPO,TH,THD,ITH,FX,FZ,TY,
     1 FSP,FST,PPU,TPU,W2,T2,TCON,THR,AMN,GAMX,GAMZ,X0,Z0,
     2 PAM,TAM,P2,FAN,BLD,FX1)
C
C    VERSION 3.                    SEPT. 9  1977
C
C    PURPOSE   INPLANE, FIRST ORDER LAG ENGINE MODEL INCLUDING BLEED AIR
C              AND FAN AIR CALCULATIONS
C
C    METHOD   FIRST ORDER LAG FOLLOWED BY BODY AXIS TRANSFORMATION AND
C             CALCULATION OF PRESSURES AND TEMPERATURES  AT BLEED PORT
C             AND AT FAN STAGE (FUNCTIONS OF THRUST AND MACH NO.)
C
C   CALL SEQUENCE
C   *****  TABLES   *****
C     TSR          -ENGINE SPEED AS A FUNCTION OF MACH NO. AND
C                   INSTANTANEOUS THRUST (TWO DIMENSIONAL TABLE)
C     TFN          -TEMPERATURE RISE FAN-INLET TO OUTLET AS A
C                   FUNCTION OF CORRECTED ENG. SPEED(ONE DIM. TABLE)
C     TFP          -PRESSURE RATIO FAN OUTLET TO INLET AS A FUNCTION
C                   OF CORRECTED ENG. SPEED(ONE DIM. TABLE)
C     TBT          -TEMPERATURE RISE COMPRESSOR-INLET TO OUTLET AS A
C                   FUNCTION OF CORRECTED ENG. SPEED(ONE DIM. TABLE)
C     TBP          -PRESSURE RATIO COMPRESSOR OUTLET TO INLET AS A
C                   FUNCTION OF CORRECTED ENG. SPEED(ONE DIM. TABLE)
C     TPO          - CORRECTED BLEED FLOW RATE AS A FUNCTION OF
C                    ENGINE PORT PRESSURE DROP (ONE DIM. TABLE)
C
C   *****  OUTPUTS  *****
C     TH,THD,ITH     -THRUST,THRUST RATE, INT. CONTROL, LBS
C     FX,FZ        -X AND Z AXIS FORCES, LBS
C     TY           -Y AXIS TORQUE(PITCHING MOMENT), FT-LBS
C     FSP          -FAN STAGE DELIVERY PRESSURE (PSIA)
C     FST          -FAN STAGE DELIVERY TEMPERATURE (DEG RANKINE)
C     PPU          -BLEED PRESSURE UPSTREAM OF THE PORT (PSIA)
C     TPU          -BLEED TEMPERATURE UPSTREAM OF THE PORT (DEG RANKINE)
C     W2           -BLEED FLOW RATE DOWNSTREAM OF THE PORT (LB/MIN)NO 2
C     T2           -BLEED TEMPERATURE DOWNSTREAM OF THE PORT(DEG RANKINE)
C
C   *****  INPUTS   ******
C     TCON           -ENGINE TIME CONSTATN, SEC.
C     THR            -REQUESTED (INPUT) THRUST, LBS
C     AMN          -MACH NUMBER
C     GAMX,GAMZ    -X,Z DIRECTION COSINES
C     X0,Z0        -THRUST LOCATION COMPONENTS FROM C.G., FT
C     PAM          -AMBIENT PRESSURE (PSIA)
C     TAM          -AMBIENT TEMPERATURE (DEGREES RANKINE)
C     P2          - BLEED PRESSURE DNSTRM OF THE PORT(PSIA)NO 2
C     FAN          -INDICATOR FUNCTION FOR ENGINE FAN AIR CALCULATIONS
C                         0= TO BE INCLUDED
C                         1= TO BE EXCLUDED
C     BLD          -INDICATOR FUNCTION FOR ENGINE BLEED AIR CALCULATION
C                         0= TO BE INCLUDED
C                         1= TO BE EXCLUDED
C     FX1         -EXTERNAL FORCE X-AXIS(THRUST REDUCTION DUE TO
```

Table 151: LISTING FOR SUBROUTINE ES (CONTINUED)

```
C                    USE OF A THRUSTER ) LBS
C
C    WRITTEN BY   MAHINDER WAHI                                 MAY  1977
C
      DIMENSION  TSR(1),TFN(1),TFP(1),TBT(1),TBP(1),TPO(1)
C              INITIALIZATION
      IF(AMN.EQ. .99999) AMN=0.
      IF(PAM.EQ. .99999) PAM=14.7
      IF(TAM.EQ. .99999) TAM=459.
      IF(FX1.EQ. .99999) FX1=0.
      IF(TCON.NE.0.) GO TO 10
      TH= THR
      GO TO 20
   10 IF(ITH.NE.0)THD= (THR - TH)/TCON
      TH = TH + FX1
C  BODY AXIS TRANSOFRMATION
   20 FX= TH*GAMX
      FZ  = TH*GAMZ
      TY  = Z0*FX-X0*FZ
C
      IF(BLD.NE.0. .AND. FAN.NE.0.) GO TO 70
C  RAM RISE/COMPRESSOR INLET CONDITIONS
      PT = PAM*(1+ .2*AMN*AMN)**3.5
      TT = TAM*(1+ .2*AMN*AMN)
C  CORRECTED ENGINE SPEED RATIO
      N1 = TSR(3)+4
      N2 = TSR(2)+TSR(3)+4
      N3 = TSR(2)
      N4 = TSR(3)
      SPD= TBLU2(AMN,TH,TSR(N1),TSR(4),TSR(N2),1,1,-N3,-N4,N3,N4)
      ENC= SPD*SQRT(519./TT)
      IF(FAN.NE.0.) GO TO 60
C   FAN STAGE DELIVERY TEMPERATURE AND PRESSURE
      NX = TFN(2)
      DTF= TBLU1(ENC,TFN(4),TFN(NX+4),1,-NX)
      FST= TT*(1.+DTF)
      NX = TFP(2)
      FPR= TBLU1(ENC,TFP(4),TFP(NX+4),1,-NX)
      FSP= PT*FPR
   60 CONTINUE
      IF(BLD.NE.0.) GO TO 70
C  BLEED AIR PRESSURE AND TEMPERATURE CALCULATIONS
C  PRESSURE AND TEMPERATURE UPSTREAM OF BLEED PORT
      NX = TBT(2)
      DT = TBLU1(ENC,TBT(4),TBT(NX+4),1,-NX)
      TPU= TT*(1.+DT)
      NX = TBP(2)
      CPR = TBLU1(ENC,TBP(4),TBP(NX+4),1,-NX)
      PPU = PT*CPR
      PRAT=PPU/P2
      NX = TPO(2)
      WCR= TBLU1(PRAT,TPO(4),TPO(NX+4),1,-NX)
C  FLOWRATE AND TEMPERATURE DOWNSTREAM OF BLEED PORT
      W2 = WCR*PPU/SQRT(TPU)
      T2= TPU
   70 CONTINUE
```

Table 151: LISTING FOR SUBROUTINE ES (CONCLUDED)

```
RETURN
END
```

Table 152: LISTING FOR SUBROUTINE ETB2

```
CETB2
      SUBROUTINE ETB2(I,PR,PT,DPT,Z)
C
C
C    VERSION 2                                   REVISED MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO PROVIDE A TWO DIMENSIONAL TABLE LOOK-UP
C              CAPABILITY FOR ELASTIC TRUNK COMPONENT TS.
C
C    OUTPUTS -
C      Z       INTERPOLATED VARIABLE
C
C    INPUTS  -
C      I       TRUNK ELEMENT SET NUMBER
C      PR      PRESSURE RATIO
C      PT      TRUNK PRESSURE
C      DPT     SPACING OF TRUNK PRESSURE DATA POINTS
C
C
      COMMON/STRCH/DUM(20480),AZO(512)
      DATA NPR/8/,NPT/8/,NN/64/
C
      NI=NN*(I-1)
      PK=PR*NPR
      KP=PK
      KP=MAXO(0,MINO(6,KP))
      DKP=PK-KP
      ADKP=ABS(DKP)
      IF(ADKP.GT.1.) DKP=SIGN(1.,DKP)*(1.+.02*(ADKP-1.))
      K1=NPR-KP
      K2=K1-1
C
      PJ=PT/DPT
      PJ=AMAX1(0.,PJ)
      J1=PJ
      J1=MAXO(1,MINO(7,J1))
      J2=J1+1
C
      NK1=NI+NPR*(J1-1)
      N1=NK1+K1
      N2=NK1+K2
      G1=AZO(N1) + (AZO(N2)-AZO(N1))*DKP
      NK1=NI+NPR*(J2-1)
      N1=NK1+K1
      N2=NK1+K2
      G2=AZO(N1) + (AZO(N2)-AZO(N1))*DKP
      Z=G1 + (G2-G1)*(PJ-J1)
      RETURN
      END
```

Table 153: LISTING FOR SUBROUTINE ETB3

```
CETB3
      SUBROUTINE ETB3(I,PR,PT,DPT,ZR,E,F)
C
C
C   VERSION 2                              REVISED MARCH 1979
C
C   WRITTEN BY - GS DULEBA
C
C   PURPOSE - TO PROVIDE A THREE DIMENSIONAL TABLE LOOK-UP
C              CAPABILITY FOR ELASTIC TRUNK COMPONENT TS.
C
C   OUTPUTS -
C     F      INTERPOLATED VARIABLE
C
C   INPUTS  -
C     I      TRUNK ELEMENT SET NUMBER
C     PR     PRESSURE RATIO
C     PT     TRUNK PRESSURE
C     DPT    SPACING OF TRUNK PRESSURE DATA POINTS
C     ZR     RATIO OF LOADED MEMBRANE HEIGHT TO FREE HEIGHT
C     E      DATA ARRAY
C
C
      DIMENSION E(1)
      DATA NPR/8/,NPT/8/,NZ/5/,NN/320/,NJ/40/
C
      NI=NN*(I-1)
      ZL=ZR*NZ-1.
      ZL=AMAX1(ZL,-1.)
      LZ=ZL
      LZ=MAX0(0,MIN0(4,LZ))
      L1=NZ-LZ
      L2=L1-1
C
      PK=PR*NPR
      KP=PK
      KP=MAX0(0,MIN0(6,KP))
      DKP=PK-KP
      ADKP=ABS(DKP)
      IF(ADKP.GT.1.) DKP=SIGN(1.,DKP)*(1.+.02*(ADKP-1.))
      K1=NPR-KP
      K2=K1-1
C
      PJ=PT/DPT
      PJ=AMAX1(0.,PJ)
      J1=PJ
      J1=MAX0(1,MIN0(7,J1))
      J2=J1+1
C
      DZL=ZL-LZ
      NL1=NI+NJ*(J1-1)+NZ*(K1-1)+L1
      NL2=NL1-1
      NL2=MAX0(NL2,1)
      C1=E(NL1) + (E(NL2)-E(NL1))*DZL
C
      NL1=NI+NJ*(J1-1)+NZ*(K2-1)+L1
```

Table 153: LISTING FOR SUBROUTINE ETB3 (CONCLUDED)

```
      NL2=NL1-1
      NL2=MAX0(NL2,1)
      C2=E(NL1) + (E(NL2)-E(NL1))*DZL
C
      NL1=NI+NJ*(J2-1)+NZ*(K1-1)+L1
      NL2=NL1-1
      NL2=MAX0(NL2,1)
      C3=E(NL1) + (E(NL2)-E(NL1))*DZL
C
      NL1=NI+NJ*(J2-1)+NZ*(K2-1)+L1
      NL2=NL1-1
      NL2=MAX0(NL2,1)
      C4=E(NL1) + (E(NL2)-E(NL1))*DZL
C
      G1=C1 + (C2-C1)*DKP
      G2=C3 + (C4-C3)*DKP
      F=G1 + (G2-G1)*(PJ-J1)
      RETURN
      END
```

Table 154: LISTING FOR SUBROUTINE FD

```
CFD
      SUBROUTINE FD(U,UD,IU,V,VD,IV,P,PD,IP,R,RD,IR,ROL,ROLD,
     1 IROL,YAW,YAWD,IYAW,XD,YD,Z,ZD,IZ,PDOT,RDOT,UDOT,VDOT,TX,TZ,
     2 XXI,ZZI,XZI,PIT)
C  VERSION 2.                                          JULY  1977
C  PURPOSE   FOUR DEGREE OF FREEDOM RIGID BODY EQUATIONS OF MOTION
C  METHOD  EULER ANGLES
C  CALL SEQUENCE
C  *****  OUTPUTS  *****
C   LINEAR VELOCITIES  -- BODY AXES
C      U,UD,IU      - X AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C      V,VD,IV      - Y AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C   ANGULAR VELOCITIES -- BODY AXES
C      P,PD,IP      - X AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C      R,RD,IR      - Z AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C   EULER ANGLES  --  EARTH TO BODY - YAW,PITCH,ROLL
C     ROL,ROLD,IROL - ROLL ANGLE,RATE,INT CONTROL,DEG
C     YAW,YAWD,IYAW - YAW ANGLE,RATE,INT CONTROL,DEG
C   POSITION -- EARTH AXES
C      XD           - X AXIS LINEAR VELOCITY, FT/SEC
C      YD           - Y AXIS LINEAR VELOCITY, FT/SEC
C       Z,ZD,IZ      - Z AXIS POSITION(ALT),VELOCITY,INT CONTROL,FT
C   ANGULAR ACCELERATION  --  BODY AXES
C      PDOT         - X AXIS ANGULAR ACCELERATION, DEG/SEC2
C      RDOT         - Z AXIS ANGULAR ACCELERATION, DEG/SEC2
C  *****  INPUTS  *****
C   LINEAR ACCELERATION  --  BODY AXES
C      UDOT         - X AXIS LINEAR ACCELERATION, FT/SEC2
C      VDOT         - Y AXIS LINEAR ACCELERATION, FT/SEC2
C   MOMENTS
C      TX,TZ        - X,Z  AXIS TORQUES, FTLBS
C   MOMENTS OF INERTIA
C      XXI,ZZI      - X,Z  AXIS MOMENTS OF INERTIA, SLUG-FT2
C      XZI          - PRODUCT OF INERTIA, SLUG-FT2
C   EULER ANGLE  --  EARTH TO BODY - PITCH
C     PIT            - PITCH ANGLE,DEG
C    WRITTEN BY   M.K. WAHI                            MARCH 1977
C
      DATA RPD,DPR /.01745329,57.29578/
      CP=COS(PIT*RPD)
      SP=SIN(PIT*RPD)
      CR=COS(ROL*RPD)
      SR=SIN(ROL*RPD)
C ****************  LINEAR VELOCITY EQUATIONS  ***************
      IF (IU.NE.0)UD=UDOT
      IF(IV.NE.0)VD=VDOT
C ****************  ANGULAR VELOCITY EQUATIONS  *************
      IF(XZI.NE.0..AND.XZI.NE. .99999) GO TO 100
      IF(IP.NE.0)PD=(TX/XXI)*DPR
      IF(IR.NE.0)RD=(TZ/ZZI)*DPR
      GO TO 160
  100 IF(IP+IR.EQ.0)GO TO 160
      TEM=ZZI/XZI
      DIV=XXI*TEM-XZI
      IF(IP.NE.0)PD=((TX*TEM+TZ)/DIV)*DPR
      TEM=XXI/XZI
```

Table 154: LISTING FOR SUBROUTINE FD (CONCLUDED)

```
      IF(IR.NE.0)RD=((TX+TZ*TEM)/DIV)*DPR
C ****************  EULER ANGLE EQUATIONS  *************
  160 IF(CP.NE.0.)PSID=R*CR/CP
      IF(IYAW.NE.0)YAWD=PSID
      IF(IROL.NE.0)ROLD=P+PSID*SP
C ****************  POSITION EQUATIONS  **************
C -----      TEST IF X AND Y  (LATITUDE AND LONGITUDE) ARE BOTH FROZEN
      SPSR=SP*SR
      CY=COS(YAW*RPD)
      SY=SIN(YAW*RPD)
      XD=CY*CP*U+(-SY*CR+CY*SPSR)*V
      YD=SY*CP*U+(CY*CR+SY*SPSR)*V
      IF(IZ.NE.0)ZD=SP*U-CP*SR*V
C ****************  ANGULAR ACCELERATIONS(FOR OUTPUT PURPOSES ONLY) ****
      PDOT=PD
      RDOT=RD
      RETURN
      END
```

Table 155: LISTING FOR SUBROUTINE FG

```
CFG
      SUBROUTINE FG(FO,FODOT,IFO,FIN,G1,MX1,MN1,G2,MX2,MN2)
C
      REAL  MX1,MN1,MX2,MN2
C
C     VERSION 1.                    SEPT.1,1977
C
C     PURPOSE  -  SIMULATION OF A SIMPLE GENERAL PURPOSE FLIGHT
C                 AND GROUND CONTROLLER FOR AIRCRAFT
C
C     METHOD   -  SEE CODING
C
C     CALL SEQUENCE
C     *****  OUTPUTS  *****
C      FO       -CONTROLLER OUTPUT
C      FODOT    -OUTPUT DERIVATIVE
C      IFO      -INTEGRATOR CONTROL
C     *****  INPUTS  *****
C      FIN      -COMMAND SIGNAL
C      G1       -GAIN(SLOPE) FOR COMMAND SIGNAL INPUT
C      MX1      -UPPER LIMIT OF SATURATION ON COMMAND SIGNAL INPUT
C      MN1      -LOWER LIMIT OF SATURATION ON COMMAND SIGNAL INPUT
C      G2       -LOOP GAIN(SLOPE) FOR THE INTEGRATOR
C      MX2      -UPPER LIMIT OF SATURATION ON OUTPUT
C      MN2      -LOWER LIMIT OF SATURATION ON OUTPUT
C
C     WRITTEN BY   MAHINDER WAHI                               SEPT 1977
C
      X1= G1*FIN
      IF(X1.GT.MX1) X1= MX1+.01*(FIN-MX1)
      IF(X1.LT.MN1) X1= MN1+.01*(FIN-MN1)
      X2= X1-FO
      X3= G2*X2
      IF(X3.GT.MX2) X3= MX2+.01*(X2-MX2/G2)
      IF(X3.LT.MN2) X3= MN2+.01*(X2-MN2/G2)
      IF(IFO.NE.0) FODOT= X3
      RETURN
      END
```

Table 156: LISTING FOR SUBROUTINE FH

```
CFH
      SUBROUTINE FH(CF,CR,T2,WC,WCDOT,IWC,P1,P1DOT,IP1,T1,W1,
     1 P2,PRR,PRS,TC)
C
C  VERSION 2.                                      SEPT. 12, 1977
C               INCLUDES A TIME CONSTANT
C   PURPOSE -   FAN MODEL WITH TRASITION BETWEEN STALL AND RECOVERY
C
C   METHOD      IT IS ASSUMED THAT TRANSITION FROM NROMAL TO STALLED
C               OPERATION OCCURS WHEN THE PRESSURE RATIO EXCEEDS THE
C               STALL PRESSURE RATIO AND THAT TRANSITION FROM STALLED
C               TO NORMAL OCCURS WHEN PRESSURE RATIO FALLS BELOW A
C               REVERSE PRESSURE RATIO.
C
C   CALL SEQUENCE
C   *****  TABLES  *****
C     CF       -FAN FLOW RATE(FORWARD) AS A FUNCTION OF FAN
C               PRESSURE RATIO, ONE DIMENSIONAL TABEL
C     CR       -FAN FLOW RATE(REVERSE) AS A FUNCTION OF FAN
C               PRESSURE RATIO, ONE DIMENSIONAL TABLE
C   *****  OUTPUTS  *****
C     WC       -FAN FLOW RATE, LB/MIN
C     WCDOT    -FAN FLOW RATE DERIVATIVE, LB/MIN/SEC
C     IWC      -INTEGRATOR CONTROL FOR WC
C     P1       -INLET PRESSURE(PORT NO.1),PSIA
C     P1DOT    -INLET PRESSURE DERIVATIVE,PSIA/SEC
C     IP1      -INTEGRATOR CONTROL ON P1
C     T2       -OUTLET TEMPERATURE(PORT NO.2),DEGR
C   *****  INPUTS  *****
C     P2       -OUTLET PRESSURE(PORT NO.2), PSIA
C     W1       -INLET FLOWRATE(PORT NO.1),LB/MIN
C     T1       -INLET TEMPERATURE(PORT NO.1),DEGR
C     PRR      -PRESSURE RATIO BELOW WHICH TRANSITION FROM
C               STALLED TO NORMAL OPERATION OCCURS
C     PRS      -PRESSURE RATIO ABOVE WHICH TRANSITION FROM
C               NORMAL TO STALLED OPERATION OCCURS
C     TC       -FAN TIME CONSTANT
C
C   WRITTEN BY    MAHINDER WAHI AND ADAM LLOYD          SEPT. 1977
C
      DIMENSION  CF(1),CR(1)
      T2=T1
C   CALCULATE FAN PRESSURE RATIO
      P1CAL=W1*SQRT(T1)/WC
      IF(IP1.NE.0) P1DOT= (P1CAL-P1)/.01
      PR = P2/P1
      NX= CF(2)
C   CALCULATE FORWARD AND REVERSE FLOW RATE BY TABLE LOOK UP
      WCF= TBLU1(PR,CF(4),CF(NX+4),1,-NX)
      NX= CR(2)
      WCR= TBLU1(PR,CR(4),CR(NX+4),1,-NX)
C   LOGIC FOR DECIDING NORMAL(FORWARD) OR STALLED(REVERSE) OPERATION
      IF(WC.GT.WCF) CR(50)= 1
      IF(WC.LT.WCR) CR(50)=-1
      IF(PR.GT.PRS) CR(50)=-1
      IF(PR.LT.PRR) CR(50)= 1
```

Table 156: LISTING FOR SUBROUTINE FH (CONCLUDED)

```
      IF(CR(50)) 20,20,10
   10 WCCAL= WCF
      GO TO 30
   20 WCCAL= WCR
   30 IF(IWC.NE.0) WCDOT= (WCCAL-WC)/TC
      RETURN
      END
```

Table 157: LISTING FOR SUBROUTINE FL

```
CFL
      SUBROUTINE FL(PAM,TAM,PRM,TRM,ALT,AMN,DAY)
C
C    PURPOSE - TO CALCULATE AMBIENT DATA GIVEN ALTITUDE
C              AND MACH NUMBER.
C
C
C    METHOD  - USES CLIMATIC DATA FROM MIL-STD-210A
C                                      MIL-STD-210B
C                                      US STANDARD ATMOSPHERE
C              DEFAULTS TO US STANDARD ATMOSPHERE
C
C    WRITTEN BY -  ADAM LLOYD            LATEST REVISION   DEC 76
C    MODIFIED FOR ACLS BY - PAUL R. PERKINS                DEC.78
C
C
C    LIMITATIONS -   ALTITUDES OF 0-100000 FT
C
C
C    INPUT/OUTPUT LIST
C
C    PAM        AMBIENT PRESSURE                     PSIA     OUTPUT  VAR
C    TAM        AMBIENT TEMPERATURE                  DEGR     OUTPUT  VAR
C    PRM        RAM PRESSURE    (100P/C RECOVERY)    PSIA     OUTPUT  VAR
C    TRM        RAM TEMPERATURE (100P/C RECOVERY)    DEGR     OUTPUT  VAR
C    ALT        ALTITUDE                             FT       INPUT   VAR
C    AMN        MACH NUMBER                          ---      INPUT   VAR
C    DAY        CODE DESIGNATING DAY                 ---      INPUT PARAM
C                    =1   MIL-STD-210B OPERATIONAL (1P/C RISK) HOT DAY
C                    =2   MIL-STD-210A HOT DAY
C                    =3   MIL-STD-210A TROPICAL DAY
C                    =4   US STANDARD ATMOSHPERE (1962)
C                    =5   MIL-STD-210A POLAR DAY
C                    =6   MIL-STD-210A COLD DAY
C                    =7   MIL-STD-210B OPERATIONAL (1P/C RISK) COLD DAY
C
C    THE FOLLOWING ARE OUTPUT THROUGH COMMON BLOCK AMISS
C
C    PAMB       AMBIENT PRESSURE                     PSIA
C    TAMB       AMBIENT TEMPERATURE                  DEGR
C    PRAM       RAM PRESSURE    (100P/C RECOVERY)    PSIA
C    TRAM       RAM TEMPERATURE (100P/C RECOVERY)    DEGR
C    Z          ALTITUDE                             FT
C    AMNX       MACH NUMBER                          ---
      COMMON/CIO/IREAD,IWRITE,IDIAG
      COMMON/AMISS/PAMB,TAMB,PRAM,TRAM,Z,AMNX
      COMMON/ERMESS/IFATAL,IERR
C    ALL DATA ARRAYS INPUT AT INCREMENTS OF 5000 FT
      DIMENSION TA1(21),TA2(21),TA3(21),TA4(21),TA5(21),TA6(21),TA7(21),
     1 PA(21)
      DATA TA1 /
     1    580.,555.,534.,517.,501.,487.,481.,470.,451.,439.,430.,
     2    425.,425.,425.,425.,425.,426.,430.,433.,436.,443. /
      DATA TA2 /
     1    563.,543.,524.,505.,485.,466.,447.,430.,415.,417.,420.,
     2    421.,422.,422.,425.,428.,432.,436.,440.,444.,448. /
```

```
      DATA TA3 /
     1    550.,530.,511.,491.,472.,453.,433.,414.,395.,377.,360.,
     2    351.,362.,373.,384.,391.,397.,404.,411.,418.,424. /
      DATA TA4 /
     1    519.,501.,483.,465.,447.,430.,412.,394.,390.,390.,390.,
     2    390.,390.,390.,392.,395.,398.,400.,403.,406.,409. /
      DATA TA5 /
     1    444.,453.,450.,436.,422.,407.,393.,391.,390.,389.,388.,
     2    386.,385.,383.,382.,381.,380.,378.,378.,378.,378. /
      DATA TA6 /
     1    400.,445.,445.,431.,414.,396.,377.,375.,375.,361.,337.,
     2    335.,335.,347.,359.,365.,364.,362.,360.,358.,356. /
      DATA TA7 /
     1    382.,405.,411.,402.,390.,377.,366.,359.,360.,357.,346.,
     2    337.,337.,341.,341.,340.,339.,341.,342.,346.,349. /
      DATA PA  /
     1    14.69,12.23,10.11,8.30,6.76,5.46,4.375,3.465,2.73,2.15,1.692,
     2    1.332,1.049,0.826,0.651,0.514,0.406,0.322,0.255,0.203,0.1616 /
      NDAY=DAY
C    DEFAULT VALUE OF DAY=.99999, WHICH RESULTS IN SELECTION OF
C    STANDARD DAY (DAY=4)
      IF(DAY.LT.1.0.OR.DAY.GT.7.0)NDAY=4
C    CHECK IF ALTITUDE IS WITHIN ALLOWABLE RANGE
      IF(ALT.GE.0.0.AND.ALT.LE.100000.) GO TO 10
C    TEST FOR DIAGNOSTIC PRINT OUT
      IF(IERR.NE.1)GO TO 10
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,30HNON FATAL ERROR CALLED FROM FL/
     1 10X,41HALTITUDE OUTSIDE ALLOWABLE RANGE (0-100K))
C    CALCULATE AMBIENT PRESSURE
 10   X1=ALT/5000. + 1.
      X2=AMIN1(AMAX1(X1,1.),21.)
      I=X2
      I=MIN0(MAX0(I,1),20)
      PAMB=(X2-I)*(PA(I+1)-PA(I)) + PA(I)
C    CALCULATE AMBIENT TEMPERATURE
      GO TO (11,12,13,14,15,16,17) NDAY
C    MIL-STD-210B OPERATIONAL (1P/C RISK) HOT DAY
 11   TAMB=(X2-I)*(TA1(I+1)-TA1(I)) + TA1(I)
      GO TO 20
C    MIL-STD-210A HOT DAY
 12   TAMB=(X2-I)*(TA2(I+1)-TA2(I))+TA2(I)
      GO TO 20
C    MIL-STD-210A TROPICAL DAY
 13   TAMB=(X2-I)*(TA3(I+1)-TA3(I)) + TA3(I)
      GO TO 20
C    U.S. STANDARD ATMOSHPERE (1962)
 14   TAMB=(X2-I)*(TA4(I+1)-TA4(I)) + TA4(I)
      GO TO 20
C    MIL-STD-210A POLAR DAY
 15   TAMB=(X2-I)*(TA5(I+1)-TA5(I)) + TA5(I)
      GO TO 20
C    MIL-STD-210A COLD DAY
 16   TAMB=(X2-I)*(TA6(I+1)-TA6(I)) + TA6(I)
      GO TO 20
C    MIL-STD-210B OPERATIONAL (1P/C RISK) COLD DAY
```

Table 157: LISTING FOR SUBROUTINE FL (CONCLUDED)

```
 17    TAMB=(X2-I)*(TA7(I+1)-TA7(I)) + TA7(I)
C   CALCULATE RAM CONDITIONS. ASSUMES GAMMA=1.4
 20    PRAM=PAMB*(1.+.2*AMN*AMN)**3.5
       TRAM=TAMB*(1.+.2*AMN*AMN)
       PAM=PAMB
       TAM=TAMB
       TRM=TRAM
       PRM=PRAM
       AMNX=AMN
       Z=ALT
       RETURN
       END
```

Table 158: LISTING FOR SUBROUTINE FLOW

```
CFLOW
      SUBROUTINE FLOW
C SUBROUTINE TO FIND FLOW AND PRESSURE VALUES DURING DYNAMIC SIMULATION
C
      COMMON/COEFFS/CPA,CAF,CPC,CPT,CTC,CGAP,CTA,CVENT,CKK
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
       COMMON/COMPRS/AL0,AL1,AL2,AL3,AL4,G0,G1,G2,G3,G4,QP1
C
      DIMENSION PEN(60)
C
      TIRHO=2.0/RHO
C PLENUM TO TRUNK FLOW
       SIGN=1.0
       IF((PPLM-PTK).LT.0.0) SIGN=-1.0
      QPLTK=SIGN*CPT*APLTK*SQRT(ABS(TIRHO*(PPLM-PTK)))
C PLENUM TO CUSHION FLOW
       SIGN=1.0
       IF((PPLM-PCH).LT.0.0) SIGN=-1.0
      QPLCH=SIGN*CPC*APLCH*SQRT(ABS(TIRHO*(PPLM-PCH)))
C TRUNK TO CUSHION FLOW
       SIGN=1.0
       IF((PTK-PCH).LT.0.0) SIGN=-1.0
      QTKCH=SIGN*CTC*SQRT(ABS(TIRHO*(PTK-PCH)))*(ATKCH+0.66667*ATKCHC)
C TRUNK TO ATMOSPHERE FLOW
      SIGN=1.0
      IF(PTK.LT.0.0)SIGN=-1.
      QTKAT=SIGN*CTA*SQRT(TIRHO*ABS(PTK))*(ATKAT+0.66667*ATKATC)
C PRESSURE RELIEF VALVE FLOW
      QVENT=0.0
      IF(NPRV.LE.0) GO TO 10
      CALL VALVE
      SIGN=1.
      IF(PPLM.LE.0.0)SIGN=-1.
      QVENT=AVENT*CVENT*SQRT(TIRHO*ABS(PPLM))*SIGN*NPRV
C FAN INLET PRESSURE
10    CALL DYNFAN(QFANX,PFAN)
      PATFN=0.0
      IF(AATFN.GE.1.0)GO TO 20
      SIGN=1.0
      IF(QFANX.LT.0.0)SIGN=-1.0
      PATFN=-RHO/2.*SIGN*(QFANX/AATFN/CAF)**2
   20 SIGN=1.
      IF(PPLM.LT.0.0) SIGN=-1.0
C PLENUM TO ATMOSPHERE FLOW
      QPLAT=APLAT*CPA*SQRT(TIRHO*ABS(PPLM))*SIGN
C CUSHION TO ATMOSPHERE FLOW
      SIGN=1.0
      IF(PCH.LT.0.0) SIGN=-1.0
      QCHAT=AGAP*CGAP*SQRT(TIRHO*ABS(PCH))*SIGN
```

Table 158: LISTING FOR SUBROUTINE FLOW (CONCLUDED)

```
 RETURN
END
```

Table 159. LISTING FOR SUBROUTINE FM

```
CFM
      SUBROUTINE FM(STC,FAN,ORF,AII,PPRV,TRK,XXX,YYY,X1,X1D,IX1
     2,X2,X2D,IX2,X3,X3D,IX3,X4,X4D,IX4,X5
     3,X5D,IX5,X6,X6D,IX6,X7,X7D,IX7,X8,X8D
     4,IX8,X9,X9D,IX9,X10,X10D,IX10,X11,X11D
     5,IX11,X12,X12D,IX12,X13,X13D,IX13,CPU
     6,BM,BN,VLX,PPAT,TAM,AMS,VCD,VPL,VFN,AAT,APA
     7,APT,APC,TSI,FMC)
C
C
C    VERSION 1                               MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO INCORPORATE THE INELASTIC ACLS TRUNK MODEL
C              DEVELOPED BY FOSTER MILLER ASSOCIATES INTO THE
C              STANDARD EASY COMPONENT LIBRARY
C
C    METHOD - COMPONENT FM HAS REPLACED THE  FOSTER MILLER ACLS MAIN
C             PROGRAM.  NECESSARY FOSTER MILLER SUBROUTINES WHICH
C             SUPPORT THE MAIN PROGRAM WERE ADDED TO THE EASY LIBRARY
C
C    ARGUMENT LIST
C
C   ***** INPUT DATA ARRAYS *****
C
C      STC     -MISC DATA ARRAY;  PARAMETERS USED IN CALCULATION
C               OF HEAVE-PITCH-ROLL LOAD MAPS
C
C      FAN     -FAN DATA ARRAY;  POLYNOMIAL COEFFICIENTS, FAN AIR
C               INERTANCE, MAXIMUM STABLE PRESSURE
C
C      ORF     -MISC DATA ARRAY;  ORIFICE AREA, NUMBER OF ORIFICE
C               ROWS, SPACING, DIMENSIONS, X COORDINATE OF CG
C
C      AII     -AIRCRAFT DATA ARRAY;  MOMENTS AND PRODUCTS OF INERTIA
C               HORIZONTAL, VERTICAL, AND LATERAL CG DISTANCES
C               FROM CUSHION CENTER
C
C      PRV     -PRESSURE RELIEF VALVE DATA ARRAY;  NUMBER OF VALVES,
C               DIMENSIONS, STIFFNESS, MASS
C
C      TRK     -TRUNK DATA ARRAY;  ATTACH POINT DIMENSIONS, TRUNK
C               FREE HEIGHT, AND TRUNK POLYNOMIAL COEFFICIENTS
C
C      XXX     -MISC COEFFICIENT ARRAY;  VARIOUS DISCHARGE COEF-
C               FICIENTS, POLYTROPIC CONSTANT  (DEFAULT VALUES
C               FOR THIS TABLE ARE PROVIDED IN SUBROUTINE PARAMS)
C
C      YYY     -MISC DATA ARRAY;  GROUND EFFECT COEFFICIENT, PRV
C               DAMPING RATIO, COEFFICIENT OF FRICTION, TRUNK
C               DAMPING CONSTANT, BRAKING DECELERATION
C               (DEFAULT VALUES FOR THIS TABLE ARE PROVIDED IN
C               SUBROUTINE PARAMS)
C
C ***** OUTPUTS *****
```

```
C
C     X1,X1D,IX1         PLENUM PRESSURE,RATE,INT CONTROL; PSFG
C
C     X2,X2D,IX2         CUSHION PRESSURE,RATE,INT CONTROL; PSFG
C
C     X3,X3D,IX3         TRUNK PRESSURE,RATE,INT CONTROL; PSFG
C
C     X4,X4D,IX4         VEH SINK RATE,ACCEL,INT CONTROL; FT/SEC
C
C     X5,X5D,IX5         Y COORD OF CG,RATE,INT CONTROL; FT
C
C     X6,X6D,IX6         PITCH RATE,ACCEL,INT CONTROL; RAD/SEC
C
C     X7,X7D,IX7         ROLL RATE,ACCEL,INT CONTROL; RAD/SEC
C
C     X8,X8D,IX8         ROLL ANGLE,RATE,INT CONTROL; RAD
C
C     X9,X9D,IX9         PITCH ANGLE,RATE,INT CONTROL; RAD
C
C     X10,X10D,IX10      YAW ANGLE,RATE,INT CONTROL; RAD
C
C     X11,X11D,IX11      PRV STROKE,RATE,INT CONTROL; FT
C
C     X12,X12D,IX12      PRV VELOCITY,ACCEL,INT CONTROL; FT/SEC
C
C     X13,X13D,IX13      FAN AIR FLOW,RATE,INT CONTROL; CFS
C
C     CPU                CUMULATIVE CPU TIME; SEC
C
C ***** INPUTS *****
C
C     BM              NUMBER OF STRAIGHT TRUNK SEGMENTS PER QUARTER
C                     OF TRUNK PERIPHERY
C
C     BN              NUMBER OF CURVED TRUNK SEGMENTS PER QUARTER
C                     OF TRUNK PERIPHERY
C
C     VLX             AIRCRAFT FORWARD VELOCITY, FT/SEC
C
C     PPAT            AMBIENT PRESSURE, PSFG
C
C     TAM             AMBIENT TEMPERATURE, DEGF
C
C     AMS             AIRCRAFT WEIGHT, LBS
C
C     VCD             CUSHION DEAD VOLUME, CU FT
C
C     VPL             PLENUM VOLUME, CU FT
C
C     VFN             FAN VOLUME, CU FT
C
C     AAT             FAN INLET ORIFICE AREA, SQ FT
C
C     APA             PLENUM-TO-ATMOSPHERE ORIFICE AREA, SQ FT
C
C     APT             PLENUM-TO-TRUNK ORIFICE AREA, SQ FT
```

```
C
C     APC            PLENUM-TO-CUSHION ORIFICE AREA, SQ FT
C
C     TSI            STATIC/DYNAMIC OPTION PARAMETER
C                    = -1  DYNAMIC MODE ONLY. INITIAL CONDITIONS
C                          FOR STATES 1,2,3 AND 13 ARE ESTIMATED
C                          BY THE PROGRAM. REMAINING INIT CONDS
C                          MUST BY INPUT BY USER.
C
C                    = 0   DYNAMIC MODE ONLY. ALL INITIAL CONDS
C                          MUST BE INPUT BY USER.
C
C                    = 1   STATIC LOAD MAPS + DYNAMIC MODE. INIT
C                          CONDS FOR STATES 1,2,3 AND 13 ARE EST-
C                          IMATED BY THE PROGRAM. REMAINING MUST
C                          BE INPUT BY USER.
C
C                    = 2   STATIC LOAD MAPS + DYNAMIC MODE. ALL
C                          INIT CONDS MUST BE INPUT BY USER.
C
C                    = 3   STATIC LOAD MAPS + EQUILIBRIUM CALCS +
C                          DYNAMIC MODE. INIT CONDS FOR STATES 1,2,
C                          3 AND 13 ARE SET TO EQUILIBRIUM CALCS.
C                          USER MUST INPUT OTHERS.
C
C                    = 4   STATIC LOAD MAPS + EQUILIBRIUM CALCS +
C                          DYNAMIC MODE. INIT CONDS FOR STATES 1,2,
C                          3 AND 13 ARE ESTIMATED BY PROGRAM. USER
C                          MUST INPUT OTHERS.
C
C                    = 5   STATIC LOAD MAPS + EQUILIBRIUM CALCS +
C                            DYNAMIC MODE. USER MUST INPUT ALL INIT
C                            CONDITIONS.
C
C     FMC            COMPONENT MODE OPTION
C                     LE.0.  FOSTER MILLER MODE - DUPLICATES FOSTER
C                            MILLER/NASA ACLS PROGRAM.
C
C                     GT.0.  EASY MODE - ENABLES EASY ANALYSES WHICH
C                            REQUIRE LINEARIZATION (STEADY STATE, LINEAR
C                            ANALYSIS)
C
C
      COMMON/COEFFS/CPA,CAF,CPC,CPT,CTC,CGAP,CTA,CVENT,CKK
C
      COMMON/FORTQ/FCP,FTP,FORCT,FDF,FORCEY,TCPX,TTPX,TORQTX,TDFX,
     1TORQUEX,TCPZ,TTPZ,TORQTZ,TDFZ,TORFZ,TORQUEZ
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
C
C
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,XX1,XX2,HY
C
      COMMON/BTERM/B11,B12,B13,B21,B22,B23,B31,B32,B33
C
```

```
      COMMON/SHAPE/AGAPI(100),AGAPR(100),ATKI(100),ATKR(100),ATKCHI(100)
     1,ATKCHR(100),ATKATI(100),ATKATR(100),ACHI(100),ACHR(100)
     2,ATKCNI(100),ATKCNR(100),VCHI(100),VCHR(100),VTKI(100),VTKR(100)
     3,XCH(100),XTK(100),ZCH(100),ZTK(100),PERI(100)
C
C
      COMMON/VEHCL/MASS,AIX,AIZ,AIXY,AIYZ,AIZX,CC,GG,FF,AIFAN
C
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
C
      COMMON/LOADS/FORCNS(3,10),CCS(3,10),YCGS(3,10),PHIS(3,10),AGAPS(3,1C
     110),PCHS(3,10),QFANS(3,10),PFANS(3,10),PPLMS(3,10),THIS(3,10),
     1FFS(3,10),TORXS(3,10),TORZS(3,10),PTKS(3,10)
C
       COMMON/COMPRS/AL0,AL1,AL2,AL3,AL4,G0,G1,G2,G3,G4,QP1
C
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
C
      COMMON/DYNAMIC/TIM,FTIME,DTIME,IQ,NQ,DVCH,DVTK,VELX,DERY(13)
     1,DVCHP
C
      COMMON/ESTMD/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
C
      COMMON/LABL/LABEL(80)
C
      COMMON/STATIC/YSTRT,YSTOP,PSTRT,PSTOP,TSTRT,TSTOP
     1,PHIYC,THEYC,YCPHI,YCTHE
C
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
C
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
      COMMON/HCUR/AH0,AH1,AH2,AH3
C
      COMMON/SAVE/R1I,R2I,PHI1I,PHI2I,L1I,L2I,A1I,A2I,D2D2HBI,DXAMABI,
     1 BETAD2I,SINPHRI,D2I,SINPH2I,A1MA2,X1I,X2I,X12I,SI
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE
     2,SIE,XV,VV,QFANX
      COMMON/CPROV/PV(1)
      COMMON/CTIME/TIME
      COMMON/CSIMUL/DD(6),TINC,TMAX
      COMMON/COVRLY/INST
      COMMON/CNTRLS/I1,I2,IMODE,E(1)
      COMMON/FMERR/FMX
      COMMON/CORDER/NOX,NOV,NOP
      COMMON/CNAMEX/XS(1)
      COMMON/FMPCH/NPCH
      DIMENSION STC(1),AII(1),CRF(1),FAN(1),PPRV(1),TRK(1),XXX(1),YYY(1)
      REAL L,LS,LP,L1,L2,MASS
      DATA PI,RADIAN/3.141592653,0.0174532/
      DATA HKILL/0./
      DATA PCNX/1CHPCHFM       /
```

```
C
C     *******************************************************************
C
C   INITIAL PHASE OF PROGRAM SETS FOSTER MILLER VARIABLES
C   EQUAL TO USER INPUT DATA
C
      CALL SECOND(CPU)
      TIM=TIME
      VELX=VLX
      PAT=PPAT
      TEMPAT=TAM
      MASS=AMS
      VCHD=VCD
      VPLM=VFN+VPL
      VFAN=VFN
      AATFN=AAT
      APLAT=APA
      APLTK=APT
      APLCH=APC
      IF(TIME.NE.PV(27)) GO TO 20
      NFM=0
      NPCH=2
      IF(NOX.EQ.13) GO TO 2
      DO 3 I=1,NOX
 3    IF(PCNX.EQ.XS(I)) NPCH=I
 2    FMX=FMC
      IF(INST.NE.26.OR.IMODE.NE.7) FMX=1.
      SUM=FMX
      DO 10 J=4,13
      XJ=J
 10   SUM=SUM+STC(J)+ XJ*AII(J)
      DO 11 J=4,9
      XJ=J
 11   SUM=SUM+ XJ*PPRV(J)
      DO 12 J=4,15
      XJ=J
 12   SUM=SUM+XJ*FAN(J)
      IF(TSI.EQ.0.99999) TSI=0.
      ISTAT=TSI
      DO 13 J=4,11
      XJ=J
 13   SUM=TSI+SUM+ORF(J)*XJ+TRK(J)*XJ*XJ+XXX(J)+YYY(J)
      SUM=SUM+BM*BM+BN+AAT+APA+APT+APC
      IF(SUM.EQ.FAN(17)) GO TO 20
      FAN(17)= SUM
      M=BM
      N=BN
      G0=FAN(4)
      G1=FAN(5)
      G2=FAN(6)
      G3=FAN(7)
      G4=FAN(8)
      QP1=FAN(9)
      AL0=FAN(10)
      AL1=FAN(11)
      AL2=FAN(12)
```

```
      AL3=FAN(13)
      AL4=FAN(14)
      AIFAN=FAN(15)
      LS=ORF(9)
      D=ORF(10)
      XCG=ORF(11)
      NR=ORF(4)
      NH=ORF(5)
      AH=ORF(6)
      SH=ORF(7)
      LP=ORF(8)
      AIX=AII(4)
      AIZ=AII(5)
      AIXY=AII(6)
      AIYZ=AII(7)
      AIZX=AII(8)
      CC=AII(9)
      GG=AII(10)
      FF=AII(11)
      PHA=AII(12)
      HOC=AII(13)
      IF(PPRV(4).EQ.1.99999) PPRV(4)=0.
      NPRV=PPRV(4)
      DPRV=PPRV(5)
      PPLMB=PPRV(6)
      XA=PPRV(7)
      AKPRV=PPRV(8)
      AMPRV=PPRV(9)
      A=TRK(4)
      B=TRK(5)
      L=TRK(6)
      HYI=TRK(7)
      AHO=TRK(8)
      AH1=TRK(9)
      AH2=TRK(10)
      AH3=TRK(11)
      CALL PARAMS
      IF(XXX(2).EQ.1.99999) GO TO 5
      CKK=XXX(4)
      CPA=XXX(5)
      CAF=XXX(6)
      CPC=XXX(7)
      CPT=XXX(8)
      CTC=XXX(9)
      CTA=XXX(10)
      CGAP=XXX(11)
5     CONTINUE
      IF(YYY(2).EQ.1.99999) GO TO 6
      GEC=YYY(4)
      ZEPRV=YYY(5)
      U=YYY(6)
      DECCL=YYY(7)
      DAMPC=YYY(8)
      QP2=YYY(9)
      SLOPE=YYY(10)
      CVENT=YYY(11)
```

```
  6   CONTINUE
      YSTRT=STC(4)
      YSTOP=STC(5)
      PHIYC=STC(6)*RADIAN
      THEYC=STC(7)*RADIAN
      PSTRT=STC(8)*RADIAN
      PSTOP=STC(9)*RADIAN
      YCPHI=STC(10)
      TSTRT=STC(11)*RADIAN
      TSTOP=STC(12)*RADIAN
      YCTHE=STC(13)
C
C    SUBROUTINE FMWRIT WILL PRODUCE A LIST OF FOSTER MILLER
C    INPUT VARIABLES AND THEIR RESPECTIVE VALUES
C
      CALL FMWRIT
C
C
C
C
      ISHAPE=1
      LIMP=500
C
C ******************************************************************************
C DATA CONVERSION SECTION
C CONVERT DATA TO PROPER UNITS
C
      MASS=AMS/32.2
       PAT=PPAT*144.0
      AH=AH/144.0
      DPRV=DPRV/12.
      PPLMB=PPLMB*144.
      XA=XA/12.
      AKPRV=AKPRV*12.
      AMPRV=AMPRV/32.
C
C PARAMETER CALCULATION
C RHO IS AIR DENSITY
       RHO=1.241/(460.0+TEMPAT)
C CALCULATE PRESSURE RELIEF VALVE PARAMETERS
      APRV=3.141592653*DPRV*DPRV/4.0
      SPRV=3.141592653*DPRV
      ZPRV=2.*ZEPRV*SQRT(AKPRV*AMPRV)
C ******************************************************************************
C INITIAL ASSESSMENT OF AREAS , VOLUMES
C ISHAPE VALUE OF 0 MEANS INFEASIBLE TRUNK ,ERROR RETURN
C CALL GEOMETRY ROUTINES TO INITIALIZE TRUNK GEOMETRY
      HY=HYI
      D2I=0.
      CALL TRUNK(ISHAPE)
      IF(ISHAPE.EQ.0) WRITE(6,3999)
      IPLM=1
      IF(ISHAPE.EQ.0)GO TO 299
      CALL SEGMNT(0)
      CALL SHAPE1(0)
      IF(ISTAT)500,500,100
```

Table 159: LISTING FOR SUBROUTINE FM (CONTINUED)

```
C ****************************************************************************
C
C ****************************************************************************
C                              STATIC PART OF THE PROGRAM
C ****************************************************************************
C ****************************************************************************
C DETERMINE STATIC CHARACTERISTICS OF ACLS
C
100   ICLN=1
C
C SUBROUTINE STATIC IS STATIC ITERATOR
      CALL STATIC(ICASE)
C ICASE IS ERROR FLAG FOR ITERATION, ZERO CAUSES RETURN
      IF(ICASE.EQ.0) WRITE(6,9100)
9100   FORMAT(5X,*STATICS ERROR,PROGRAM TERMINATION*,/)
       IF(ICASE.EQ.0) GO TO 299
      IF(ISTAT.EQ.1.OR.ISTAT.EQ.2) GO TO 200
C
C NPRV IS NUMBER OF PRESSURE RELIEF VALVES
      IF(NPRV.EQ.0)GO TO 198
      IF(PPLM.LT.PPLMB) GO TO 198
      WRITE(6,9000) PPLM
9000  FORMAT(10X,*INFEASIBLE CONFIGURATION*,//,15X,*INCREASE PRESSURE ACTU
     1TUATION LIMIT OF*,/,15X,*PRESSURE RELIEF VALVE TO AT LEAST*,/,15X,
     1 F10.4,*PSF*,/)
      IPLM=0
      GO TO 299
  198 CONTINUE
C
C PARAMETER CALCULATION
C
C FAN HORSEPOWER
      HP=QFAN*PFAN/550.
C FAN STALL MARGIN
      SC=ABS(PFAN    -QP1)/QP1*100.
C WRITE FINAL EQUILIBRIUM CONDITIONS AND STATIC CHARACTERISTICS
C OBTAINED FROM SUBROUTINE STATIC
C
      WRITE(6,9201)
9201   FORMAT(1H1,///////)
      WRITE(6,9034)
9034   FORMAT(44X,31H STATIC EQUILIBRIUM CONDITIONS     ,/)
      WRITE(6,9037)YCG
9037    FORMAT(43X,27H HEIGHT OF CG              = ,F8.3,2X,3H FT )
      PHIC=PHIE*180./3.14159
      WRITE(6,9080)PHIC
 9080 FORMAT(43X,27H PITCH ANGLE              =,F8.3,2X,8H DEGREES)
      THETAEC=THETAE*180./3.14159
      WRITE(6,9082)THETAEC
 9082 FORMAT(43X,27H ROLL ANGLE               =,F8.3,2X,8H DEGREES,/)
      WRITE(6,9060) SI
9060   FORMAT(43X,27H CUSHION PERIMETER         =,F8.3,2X,10H FT        )
      WRITE(6,9061)VCH
9061  FORMAT(43X,27H CUSHION VOLUME           =,F8.3,2X,8H CU FT    )
      WRITE(6,9062)VTK
9062  FORMAT(43X,27H TRUNK VOLUME             =,F8.3,2X,8H CU FT    )
```

```
      WRITE(6,9038)AGAP
 9038 FORMAT(43X,27H AIR GAP AREA              =,F8.3,2X,8H SQ FT   )
      WRITE(6,9063)ACH
9063  FORMAT(43X,27H CUSHION AREA              =,F8.3,2X,8H SQ FT   )
      WRITE(6,9081)ATKCN
 9081 FORMAT(43X,27H GROUND CONTACT AREA       =,F8.3,2X,8H SQ FT  )
      WRITE(6,9064)ATKAT
9064  FORMAT(43X,27H ORIFICE AREA TRUNK-ATMOS =,F8.3,2X,8H SQ FT    )
       WRITE(6,9065)ATKCH
9065   FORMAT(43X,27H ORIFICE AREA TRUNK-CUSH  =,F8.3,2X,6H SQ FT ,/)
       WRITE(6,9039)PCH
9039    FORMAT(43X,27H CUSHION PRESSURE          =,F8.3,2X,9H PSFG     )
      WRITE(6,9040) PTK
9040    FORMAT(43X,27H TRUNK PRESSURE            =,F8.3,2X,9H PSFG     )
       WRITE(6,9041)PPLM
9041    FORMAT(43X,27H PLENUM PRESSURE           =,F8.3,2X,5H PSFG ,/)
       WRITE(6,9042)QFAN
9042  FORMAT(43X,27H TOTAL AIR FLOW            = ,F8.3,2X,10H CU FT/SEC )
      WRITE(6,9043)QCHAT
9043   FORMAT(43X,27H TOTAL CUSHION FLOW        =,F8.3,2X,10H CU FT/SEC)
      WRITE(6,9044)QPLCH
9044   FORMAT(43X,27H FLOW PLENUM TO CUSHION    =,F8.3,2X,10H CU FT/SEC)
      WRITE(6,9045)QPLTK
9045   FORMAT(43X,27H FLOW,PLENUM TO TRUNK      =,F8.3,2X,10H CU FT/SEC)
      WRITE(6,9046)QTKCH
9046   FORMAT(43X,27H FLOW,TRUNK TO CUSHION     =,F8.3,2X,10H CU FT/SEC)
      WRITE(6,9047)QTKAT
9047   FORMAT(43X,27H FLOW,TRUNK TO ATMOSPHERE =,F8.3,2X,10H CU FT/SEC)
      WRITE(6,9049) QPLAT
9049   FORMAT(43X,27H FLOW,PLENUM TO ATMOSPHERE=,F8.3,2X,10H CU FT/SEC ,
     1 /)
 9203 FORMAT(/)
      WRITE(6,9203)
      WRITE(6,9070)SC
 9070 FORMAT(43X,27H FAN STALL MARGINE         =,F8.3,2X,8H PERCENT)
      IF(SC.LT.5.0)WRITE(6,9071)
 9071 FORMAT(43X,*-WARNING-FAN CRITICALLY STABLE-*/)
      WRITE(6,9066)HP
9066  FORMAT(/,43X,27H THEORETICAL FAN POWER     =,F8.3,2X,4H HP
     1)
200   CONTINUE
      WRITE(6,9300)
9300   FORMAT(1H1,///////,52X,16H STATIC LOAD MAP,///,57X,6H HEAVE,//)
      PHIYC=PHIYC/RADIAN
      THEYC=THEYC/RADIAN
      WRITE(6,9301)PHIYC,THEYC
 9301 FORMAT(10X,15H PITCH ANGLE = ,F12.5,/,10X,15H ROLL ANGLE  = ,F12.5
     1,/)
      WRITE(6,9401)
9401  FORMAT(10X,*    LOAD          CG          TRUNK       CUSHION
     1    FAN          FAN          GAP          CONTACT     *,/
     2 10X,*                  HEIGHT       PRESSURE     PRESSURE     PRESSURE
     3RE       FLOW           AREA         AREA       *,/
     4 10X,*      LBS           FT          PSFG         PSFG         PSFG
     5       CU FT/SEC       SQ FT        SQ FT      *,/)
      DO 9400 IJ=1,10
```

```
      WRITE(6,9305)FORCNS(1,IJ),YCGS(1,IJ),PTKS(1,IJ),PCHS(1,IJ),PFANS(
     11,IJ),QFANS(1,IJ),AGAPS(1,IJ),CCS(1,IJ)
 9305 FORMAT(10X,F13.1,F13.4,3F13.1,F13.2,F13.4,F13.3)
9400  CONTINUE
      WRITE(6,9325)
9325  FORMAT(1H1,//////)
      WRITE(6,9310) YCPHI
9310  FORMAT(////,57X,6H PITCH,//,10X,16H CG ELEVATION = ,F12.5,/,10X,
     2 19H ROLL ANGLE    = 0.0,/)
      WRITE(6,9402)
9402  FORMAT(10X,*   MOMENT            PITCH          TRUNK        CUSHION
     1   FAN           FAN             GAP          CONTACT     *,/
     2 10X,*                   ANGLE        PRESSURE       PRESSURE      PRESSURE
     3RE         FLOW          AREA           AREA        *,/
     4 10X,*    FT LBS          DEG           PSFG           PSFG           PSFG
     5        CU FT/SEC       SQ FT          SQ FT     *,/)
      DO 9410 IJ=1,10
      WRITE(6,9305)TORZS(2,IJ),PHIS(2,IJ),PTKS(2,IJ),PCHS(2,IJ),PFANS(2,
     1IJ),QFANS(2,IJ),AGAPS(2,IJ),CCS(2,IJ)
9410   CONTINUE
      WRITE(6,9325)
      WRITE(6,9320) YCTHE
9320  FORMAT(////,58X,5H ROLL,//,10X,16H GC ELEVATION = ,F12.5,/,10X,
     2 19H PITCH ANGLE   = 0.0,/)
      WRITE(6,9403)
9403  FORMAT(10X,*   MOMENT             ROLL          TRUNK        CUSHION
     1   FAN           FAN             GAP          CONTACT     *,/
     2 10X,*                   ANGLE        PRESSURE       PRESSURE      PRESSURE
     3RE         FLOW          AREA           AREA        *,/
     4 10X,*    FT LBS          DEG           PSFG           PSFG           PSFG
     5        CU FT/SEC       SQ FT          SQ FT     *,/)
      DO 9420 IJ=1,10
      WRITE(6,9305)TORXS(3,IJ),THIS(3,IJ),PTKS(3,IJ),PCHS(3,IJ),PFANS(3,
     1IJ),QFANS(3,IJ),AGAPS(3,IJ),CCS(3,IJ)
9420  CONTINUE
 20   CONTINUE
      MASS=AMS/32.2
       PAT=PPAT*144.0
C
C
C ***********************************************************************
C ***********************************************************************
C************************************************************************
C                           DYNAMIC PART OF THE PROGRAM
C************************************************************************
C************************************************************************
C
C INITIALIZATION OF ACLS TO INPUT STATE VALUES
C
500   CONTINUE
C***********
C  ESTIMATION OF PRESSURE,FLOW INITIAL CONDITIONS
C
      IF(INST.NE.26) GO TO 777
      IF(IMODE.NE.7) GO TO 777
      IF(TIME.NE.PV(27)) GO TO 777
```

```
      DTIME=TINC
      IST=2 + MAX0(-1,MIN0(5,ISTAT))
      GO TO (72,75,75,72,73,72,75),IST
   72 PCH=0.
      PTK=0.8*QP1
      PPLM=0.8*QP1
      PFAN=PPLM
      CALL FMFAN
      QFANX=QFAN
      GO TO 74
C************
   73 DO 70 IJ=1,10
      IF(X5.GT.YCGS(1,IJ))GO TO 71
   70 CONTINUE
   71 IF(IJ.EQ.1)IJ=2
      IJ1=IJ-1
      YCGSIJ=YCGS(1,IJ)-YCGS(1,IJ1)
      IF(ABS(YCGSIJ).LE.0.0000001)GO TO 72
      YFACT=(X5-YCGS(1,IJ1))/(YCGS(1,IJ)-YCGS(1,IJ1))
      PCH=PCHS(1,IJ1)+(PCHS(1,IJ)-PCHS(1,IJ1))*YFACT
      PTK=PTKS(1,IJ1)+(PTKS(1,IJ)-PTKS(1,IJ1))*YFACT
      PPLM=PPLMS(1,IJ1)+(PPLMS(1,IJ)-PPLMS(1,IJ1))*YFACT
      QFANX=QFANS(1,IJ1)+(QFANS(1,IJ)-QFANS(1,IJ1))*YFACT
      PCH=AMAX1(PCH,0.)
      PTK=AMAX1(PTK,0.)
      PPLM=AMAX1(PPLM,0.)
      QFANX=AMAX1(QFANX,0.)
   74 CONTINUE
      XV=0.
      VV=0.
      SIE=0.
C
C    SET IC FOR EASY STATES TO FM VALUES
      X1=PPLM
      X2=PCH
      X3=PTK
      X10=SIE
      X11=XV
      X12=VV
      X13=QFANX
C
 75   CONTINUE
C************
C
C
C  EQUATE FM STATES WITH EASY STATES
C
      PPLM=X1
      PCH=X2
      PTK=X3
      SINKRT=X4
      YCG=X5
      DPHI=X6
      DTHETA=X7
      THETAE=X8
      PHIE=X9
```

```
      SIE=X10
      XV=X11
      VV=X12
      QFANX=X13
510   CONTINUE
       ICLN=0
C SET INDICATOR FOR PLENUM-TRUNK -CUSHION MODEL
      IPP=1
      PTESTP=ABS(PTK-PPLM)/PTK
      IF(PTESTP.LE.0.1.AND.INST.EQ.26)IPP=0
C OBTAIN INITIAL VALUE OF DVCHP AND INITIALIZE GEOMETRY
      PCHSS=(PCH+(PTK-PCH)*0.1)/PTK
      PRAT=PCHSS
      CALL HYCURV(PRAT,HX)
      HY=HYI*HX
      CALL TRUNK(ISHAPE)
      IF(ISHAPE.EQ.0) WRITE(6,3999)
      IF(ISHAPE.EQ.0)GO TO 299
      CALL SEGMNT(1)
      CALL COORDN
      CALL PROFILE
      CALL CLRNCE
      CALL SHAPE2
      VCHSS=VCH
      CALL HYCURV((PCH/PTK),HX)
      HY=HYI*HX
      CALL TRUNK(ISHAPE)
      IF(ISHAPE.EQ.0) WRITE(6,3999)
      IF(ISHAPE.EQ.0)GO TO 299
      CALL SEGMNT(1)
      CALL COORDN
      CALL PROFILE
      CALL CLRNCE
      CALL SHAPE2
      DVCHP=(VCH-VCHSS)/((PCH/PTK)-PCHSS)
      INUM=0
      DVTK=0.
      DVCH=SINKRT*ACH
      CALL STEQU
C
C    DEFINE EASY DERIVATIVES FROM FM DERIVATIVES
C
      IF(IX1.NE.0) X1D=DERY(1)
      IF(IX2.NE.0) X2D=DERY(2)
      IF(IX3.NE.0) X3D=DERY(3)
      IF(IX4.NE.0) X4D=DERY(4)
      IF(IX5.NE.0) X5D=DERY(5)
      IF(IX6.NE.0) X6D=DERY(6)
      IF(IX7.NE.0) X7D=DERY(7)
      IF(IX8.NE.0) X8D=DERY(8)
      IF(IX9.NE.0) X9D=DERY(9)
      IF(IX10.NE.0) X10D=DERY(10)
      IF(IX11.NE.0) X11D=DERY(11)
      IF(IX12.NE.0) X12D=DERY(12)
      IF(IX13.NE.0) X13D=DERY(13)
      CALL OUTFM(INUM)
```

```
      GO TO 999
C
C
 777  CONTINUE
C
C  EQUATE FM STATES WITH EASY STATES
C
      PPLM=X1
      PCH=X2
      PTK=X3
      SINKRT=X4
      YCG=X5
      DPHI=X6
      DTHETA=X7
      THETAE=X8
      PHIE=X9
      SIE=X10
      XV=X11
      VV=X12
      QFANX=X13
      IF(INST.EQ.26.AND.IMODE.EQ.7) GO TO 95
      DVCH=0.
      DVTK=0.
      DVCHP=0.
      CALL HYCURV((PCH/PTK),HX)
      HY=HYI*HX
      CALL TRUNK(ISHAPE)
      IF(ISHAPE.EQ.0) WRITE(6,3999)
      IF(ISHAPE.EQ.0)GO TO 299
      CALL SEGMNT(1)
      CALL COORDN
      CALL PROFILE
      CALL CLRNCE
      CALL SHAPE2
      NFM=-10
      GO TO 98
 95   CONTINUE
      NFM=NFM+1
 98   CALL STEQU
C   DEFINE EASY DERIVATIVES FROM FM DERIVATIVES
C
      IF(IX1.NE.0) X1D=DERY(1)
      IF(IX2.NE.0) X2D=DERY(2)
      IF(IX3.NE.0) X3D=DERY(3)
      IF(IX4.NE.0) X4D=DERY(4)
      IF(IX5.NE.0) X5D=DERY(5)
      IF(IX6.NE.C) X6D=DERY(6)
      IF(IX7.NE.0) X7D=DERY(7)
      IF(IX8.NE.0) X8D=DERY(8)
      IF(IX9.NE.0) X9D=DERY(9)
      IF(IX10.NE.0) X10D=DERY(10)
      IF(IX11.NE.0) X11D=DERY(11)
      IF(IX12.NE.0) X12D=DERY(12)
      IF(IX13.NE.0) X13D=DERY(13)
      IF(NFM.LT.4) GO TO 110
      INUM=1
```

Table 159: LISTING FOR SUBROUTINE FM (CONCLUDED)

```
      CALL OUTFM(INUM)
      NFM=0
110   CONTINUE
999   CONTINUE
3999  FORMAT(//,40X,31H***** FAILURE TO CONVERGE *****)
      GO TO 300
299   SK=1./HKILL
      SK1=SK*100.
300   CONTINUE
      RETURN
      END
```

Table 160: LISTING FOR SUBROUTINE FMFAN

```
CFMFAN
      SUBROUTINE FMFAN
C STATIC FAN MODEL SUBROUTINE
C INPUT PRESSURE AND OUTPUT FLOW
C
       COMMON/COMPRS/AL0,AL1,AL2,AL3,AL4,G0,G1,G2,G3,G4,QP1
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
C
C COMPUTE POWER SERIES TERMS
      PFAN2=PFAN*PFAN
      PFAN3=PFAN2*PFAN
      PFAN4=PFAN3*PFAN
C FAN CURVE FOR STATIC ITERATIONS
      QFAN=AL0+AL1*PFAN+AL2*PFAN2+AL3*PFAN3+AL4*PFAN4
      RETURN
      END
```

Table 161: LISTING FOR SUBROUTINE FMWRIT

```
CFMWRIT
      SUBROUTINE FMWRIT
C
C
C    VERSION 1                              MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO PRINT A LIST OF INPUT VARIABLES FOR THE FOSTER
C              MILLER ACLS TRUNK COMPONENT FM.  THE LIST INCLUDES
C              A DESCRIPTION OF THE VARIABLES AND THE VALUES
C              INPUT BY THE USER.
C
C
C
      COMMON/COEFFS/CPA,CAF,CPC,CPT,CTC,CGAP,CTA,CVENT,CKK
C
      COMMON/FORTQ/FCP,FTP,FORCT,FDF,FORCEY,TCPX,TTPX,TORQTX,TDFX,
     1TORQUEX,TCPZ,TTPZ,TORQTZ,TDFZ,TORFZ,TORQUEZ
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
C
C
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
C
      COMMON/BTERM/B11,B12,B13,B21,B22,B23,B31,B32,B33
C
      COMMON/SHAPE/AGAPI(100),AGAPR(100),ATKI(100),ATKR(100),ATKCHI(100)
     1,ATKCHR(100),ATKATI(100),ATKATR(100),ACHI(100),ACHR(100)
     2,ATKCNI(100),ATKCNR(100),VCHI(100),VCHR(100),VTKI(100),VTKR(100)
     3,XCH(100),XTK(100),ZCH(100),ZTK(100),PERI(100)
C
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
C
      COMMON/VEHCL/MASS,AIX,AIZ,AIXY,AIYZ,AIZX,CC,GG,FF,AIFAN
C
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
C
      COMMON/LOADS/FORCNS(3,10),CCS(3,10),YCGS(3,10),PHIS(3,10),AGAPS(3,1C
     110),PCHS(3,10),QFANS(3,10),PFANS(3,10),PPLMS(3,10),THIS(3,10),
     1FFS(3,10),TORXS(3,10),TORZS(3,10),PTKS(3,10)
C
        COMMON/COMPRS/AL0,AL1,AL2,AL3,AL4,G0,G1,G2,G3,G4,QP1
C
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
C
      COMMON/DYNAMIC/TIME,FTIME,DTIME,IQ,NQ,DVCH,DVTK,VELX,DERY(13)
     1,DVCHP
C
      COMMON/ESTMD/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
```

```
C
      COMMON/LABL/LABEL(80)
C
      COMMON/STATIC/YSTRT,YSTOP,PSTRT,PSTOP,TSTRT,TSTOP
     1,PHIYC,THEYC,YCPHI,YCTHE
C
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
C
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
      COMMON/HCUR/AH0,AH1,AH2,AH3
C
      COMMON/SAVE/R1I,R2I,PHI1I,PHI2I,L1I,L2I,A1I,A2I,D2D2HBI,DXAMABI,
     1 BETAD2I,SINPHRI,D2I,SINPH2I,A1MA2,X1I,X2I,X12I,SI
C     ******************************************************************
C
      WRITE(6,50)
      WRITE(6,70)
      WRITE(6,71)
      WRITE(6,72)
      WRITE(6,73)
      WRITE(6,74)
      WRITE(6,100)
      WRITE(6,101) MASS
      WRITE(6,102) AIX
      WRITE(6,103) AIZ
      WRITE(6,104) AIXY
      WRITE(6,105) AIYZ
      WRITE(6,106) AIZX
      WRITE(6,107) CC
      WRITE(6,108) GG
      WRITE(6,109) FF
      WRITE(6,110) PHA
      WRITE(6,111) HDC
      WRITE(6,112)
      WRITE(6,113) LS
      WRITE(6,114) D
      WRITE(6,115) A
      WRITE(6,116) B
      WRITE(6,117) L
      WRITE(6,118) HYI
      WRITE(6,119) NR
      WRITE(6,120) NH
      WRITE(6,121) AH
      WRITE(6,122) SH
      WRITE(6,123) LP
      WRITE(6,124)
      WRITE(6,125) AH0
      WRITE(6,126) AH1
      WRITE(6,127) AH2
      WRITE(6,128) AH3
      WRITE(6,129)
      WRITE(6,130) VFAN
      WRITE(6,131) AIFAN
      WRITE(6,132)
      WRITE(6,133) G0
```

```
      WRITE(6,134) G1
      WRITE(6,135) G2
      WRITE(6,136) G3
      WRITE(6,137) G4
      WRITE(6,138) AL0
      WRITE(6,139) AL1
      WRITE(6,140) AL2
      WRITE(6,141) AL3
      WRITE(6,142) AL4
      WRITE(6,143) QP1
      WRITE(6,144)
      WRITE(6,145) APLCH
      WRITE(6,146) APLTK
      WRITE(6,147) APLAT
      WRITE(6,148) AATFN
      WRITE(6,149) VPLM
      WRITE(6,150) VCHD
      WRITE(6,151)
      WRITE(6,152) NPRV
      WRITE(6,153) DPRV
      WRITE(6,154) PPLMB
      WRITE(6,155) XA
      WRITE(6,156) AKPRV
      WRITE(6,157) AMPRV
      WRITE(6,158)
      WRITE(6,160) PAT
      WRITE(6,161) TEMPAT
      WRITE(6,162)
      WRITE(6,163) M
      WRITE(6,164) N
50    FORMAT(1H1,///,48X,26HAIR CUSHION LANDING SYSTEM//,42X,40HEASY ADA
     2PTATION OF FOSTER-MILLER PROGRAM)
70    FORMAT(/,15X,20(5H*****)/,15X,20(5H*****))
71    FORMAT(/,33X,48HTHIS COMPONENT MAY BE USED IN TWO DIFFERENT OPER,
     2 13HATING MODES -,/,38X,35H1. FOSTER MILLER MODE (FMCFM.LE.0.),
     3 /,41X,44HIN THIS MODE, THE PROGRAM WILL DUPLICATE THE,/
     4 ,41X,42HFOSTER MILLER/NASA ACLS PROGRAM.  THE EASY,/
     5 ,41X,46HCOMMAND *SIMULATE* WILL INITIATE THE ANALYSIS.)
72    FORMAT(/,41X,46HIF THE DYNAMIC PORTION OF THE PROGRAM IS TO BE,/
     2 ,41X,43HEXECUTED, THE USER SHOULD SET *INT MODE=7*.,//
     3 ,41X,48HNO EASY ANALYTICAL COMMOND OTHER THAN *SIMULATE*,/
     4 ,41X,28HSHOULD BE USED IN THIS MODE.,/
     5 ,38X,27H2. EASY MODE  (FMCFM.GT.0.),/
     6 ,41X,48HIN THIS MODE, EASY ANALYTICAL TECHNIQUES SUCH AS )
73    FORMAT(41X,43H*STEADY STATE* AND *LINEAR ANALYSIS*, WHICH,/
     2 ,41X,40HREQUIRE MODEL LINEARIZATION MAY BE USED.,//
     3 ,41X,49HIF NON-LINEAR SIMULATION (*SIMULATE*) IS DESIRED,,/
     4 ,41X,33HTHE USER SHOULD SET *INT MODE=7*.,//
     5 ,38X,55HNOTE - RESULTS OF *LINEAR ANALYSIS* MAY BE ERRONEOUS IF,/
     6 ,45X,51HTHE SYSTEM IS NOT AT A STEADY STATE OPERATING POINT)
74    FORMAT(/,15X,20(5H*****)/,15X,20(5H*****)//)
100   FORMAT(51X,19HAIRCRAFT PARAMETERS)
101   FORMAT(40X,33HTOTAL WEIGHT OF AIRCRAFT        = ,F10.3, 5H  LBS)
102   FORMAT(40X,33HROLL  INERTIA                   = ,F10.3,13H  SLUG  S
     2Q FT)
103   FORMAT(40X,33HPITCH INERTIA                   = ,F10.3,13H  SLUG  S
```

Table 161: LISTING FOR SUBROUTINE FMWRIT (CONTINUED)

```
    2Q FT)
104  FORMAT(40X,33HPRODUCT OF INERTIA-IXY           = ,F10.3,13H  SLUG  S
    2Q FT)
105  FORMAT(40X,33HPRODUCT OF INERTIA-IYZ           = ,F10.3,13H  SLUG  S
    2Q FT)
106  FORMAT(40X,33HPRODUCT OF INERTIA-IZX           = ,F10.3,13H  SLUG  S
    2Q FT)
107  FORMAT(40X,33HCG HOR DIS FROM CUSH CNTR - CC = ,F10.3, 4H  FT)
108  FORMAT(40X,33HCG VER DIS FROM CUSH CNTR - GG = ,F10.3, 4H  FT)
109  FORMAT(40X,33HCG LAT DIS FROM CUSH CNTR - FF = ,F10.3, 4H  FT)
110  FORMAT(40X,33HPROJECTED HEAVE AREA            = ,F10.3, 7H  SQ FT)
111  FORMAT(40X,33HHEAVE DRAG COEFFICIENT          = ,F10.3)
112  FORMAT(//,53X,16HTRUNK PARAMETERS)
113  FORMAT(40X,33HSTRAIGHT SECTION LENGTH         = ,F10.3, 4H  FT)
114  FORMAT(40X,33HINNER ATTACHMENT DISTANCE       = ,F10.3, 4H  FT)
115  FORMAT(40X,33HHORIZ DIST BET ATTACH PNTS      = ,F10.3, 4H  FT)
116  FORMAT(40X,33HVERT  DIST BET ATTACH PNTS      = ,F10.3, 4H  FT)
117  FORMAT(40X,33HPERIMETER OF TRUNK CROSSECTION = ,F10.3,4H  FT)
118  FORMAT(40X,33HTRUNK FREE HEIGHT  HYI          = ,F10.3, 4H  FT/)
119  FORMAT(40X,33HNUMBER OF ORIFICE ROWS          = ,I6)
120  FORMAT(40X,33HNUMBER OF ORIFICES PER ROW      = , I6)
121  FORMAT(40X,33HAREA OF EACH ORIFICE            = , F10.3, 7H  SQ IN)
122  FORMAT(40X,33HSPACING BETWEEN ORIFICE ROWS    = , F10.3, 4H  FT)
123  FORMAT(40X,27HPERIPHERAL DISTANCE BETWEEN/
    2       40X,33HINNER ATTACHMT PNT + FIRST ROW = , F10.3, 4H  FT)
124  FORMAT(/,40X,31HTRUNK CHARACTERISTIC POLYNOMIAL/40X,19HZ=HY/HYI, X
    2=PCH/PTK/40X,29HZ=AH0+AH1*X+AH2*X*X+AH3*X*X*X/)
125  FORMAT(40X,33HAH0 TRUNK COEFFICIENT           = ,F10.3)
126  FORMAT(40X,33HAH1 TRUNK COEFFICIENT           = ,F10.3)
127  FORMAT(40X,33HAH2 TRUNK COEFFICIENT           = ,F10.3)
128  FORMAT(40X,33HAH3 TRUNK COEFFICIENT           = ,F10.3)
129  FORMAT(//,54X,14HFAN PARAMETERS)
130  FORMAT(40X,33HVOLUME OF FAN                   = ,F10.3, 7H  CU FT)
131  FORMAT(40X,33HFAN AIR INERTANCE               = ,F10.3,19H  LBS-SEC
    2*SEC/FT**5)
132  FORMAT(/,40X,29HFAN CHARACTERISTIC POLYNOMIAL/
    2         40X,36HP=G0+G1*A+G2*A*A+G3*A*A*A+G4*A*A*A*A/
    3         40X,41HQ=AL0+AL1*P+AL2*P*P+AL3*P*P*P+AL4*P*P*P*P/)
133  FORMAT(40X,33HG0 FAN COEFFICIENT              = ,E12.5 )
134  FORMAT(40X,33HG1 FAN COEFFICIENT              = ,E12.5 )
135  FORMAT(40X,33HG2 FAN COEFFICIENT              = ,E12.5 )
136  FORMAT(40X,33HG3 FAN COEFFICIENT              = ,E12.5 )
137  FORMAT(40X,33HG4 FAN COEFFICIENT              = ,E12.5 )
138  FORMAT(40X,33HAL0 FAN COEFFICIENT             = ,E12.5 )
139  FORMAT(40X,33HAL1 FAN COEFFICIENT             = , E12.5)
140  FORMAT(40X,33HAL2 FAN COEFFICIENT             = , E12.5)
141  FORMAT(40X,33HAL3 FAN COEFFICIENT             = , E12.5)
142  FORMAT(40X,33HAL4 FAN COEFFICIENT             = , E12.5)
143  FORMAT(40X,33HQP1 MAXIMUM STABLE PRESSURE     = , F10.3, 5H  PSF)
144  FORMAT(//,51X,21HAIR SUPPLY PARAMETERS)
145  FORMAT(40X,33HORIFICE AREA-PLENUM TO CUSH     = ,F10.3, 7H  SQ FT)
146  FORMAT(40X,33HORIFICE AREA-PLENUM TO TRUNK    = ,F10.3, 7H  SQ FT)
147  FORMAT(40X,33HAREA-PLENUM TO ATMOSPHERE       = ,F10.3, 7H  SQ FT)
148  FORMAT(40X,33HEFFECTIVE AREA-ATM TO FAN       = ,F10.3, 7H  SQ FT)
149  FORMAT(40X,33HPLENUM VOLUME                   = ,F10.3, 7H  CU FT)
150  FORMAT(40X,33HDEAD VOLUME OF CUSHION          = ,F10.3, 7H  CU FT)
```

Table 161: LISTING FOR SUBROUTINE FMWRIT (CONCLUDED)

```
151  FORMAT(//,52X,27HPRESSURE RELIEF VALVE (PRV))
152  FORMAT(40X,33HNUMBER OF PRV                      = , I6)
153  FORMAT(40X,33HDIAMTER OF PRV                     = ,F10.3,8H  INCHES)
154  FORMAT(40X,33HPRESSURE ACTUATION LIMIT           = ,F10.3,5H  PSI)
155  FORMAT(40X,33HSTROKE OF PRV                      = ,F10.3,8H  INCHES)
156  FORMAT(40X,33HSTIFFNESS OF PRV                   = ,F10.3,9H  LB/INCH)
157  FORMAT(40X,33HMASS OF PRV                        = ,F10.3,4H  LB)
158  FORMAT(//,50X,24HENVIRONMENTAL CONDITIONS)
160  FORMAT(40X,33HATMOSPHERIC PRESSURE               = ,F10.3,11H  LBS/SQ
    2IN)
161  FORMAT(40X,33HAMBIENT TEMPERATURE                = ,F10.3, 7H  DEG F)
162  FORMAT(//,51X,21HSIMULATION PARAMETERS)
163  FORMAT(40X,33HSTRAIGHT SEGMENTS/4                = , I6)
164  FORMAT(40X,33HCURVED SEGMENTS/4                  = , I6)
     RETURN
     END
```

Table 162: LISTING FOR SUBROUTINE FN

```
CFN
      SUBROUTINE FN(FANFLO,STALL,T2,W2,PIN,TIN,PR,P2,PAM,TAM,
     1PRM,TRM,NUI,NUF,COR,RPM)
C    PURPOSE -      COMPONENT FN DETERMINES THE OUTPUT FLOW
C                   RATE AND OUTPUT TEMPERATURE OF A FAN.
C
C    METHOD  -      THE USER MUST INPUT TABULAR VALUES OF
C                   FLOW AS FUNCTION OF BOTH PRESSURE RATIO
C                   AND FAN RPM. THIS IS AN INLET COMPONENT. THE INLET
C                   RAM EFFECT CAN BE INPUT BY THE USER ALONG
C                   WITH THE FAN EFFICIENCY. IF STALL DATA IS
C                   INPUT, AN ERROR MESSAGE IS PRINTED WHENEVER
C                   THE FAN IS OPERATING IN THE STALL REGION.
C
C   WRITTEN BY -   PAUL R. PERKINS                                DEC.78
C
C   INPUT/OUTPUT LIST
C     FANFLO        TABLE OF FAN OUTPUT=F(PR,RPM)      LB/SEC    INPUT TABLE
C     STALL         TABLE OF STALL POINTS=F(PR)        LB/SEC    INPUT TABLE
C     P2            OUTPUT PRESSURE                    PSIA      INPUT VAR
C     PAM           AMBIENT PRESSURE                   PSIA      INPUT VAR
C     TAM           AMBIENT TEMPERATURE                DEGR      INPUT VAR
C     PRM           RAM PRESSURE (100P/C RECOVERY)     PSIA      INPUT VAR
C     TRM           RAM TEMP     (100P/C RECOVERY)     DEGR      INPUT VAR
C     NUI           INLET RAM EFFICIENCY               ----      INPUT PARAM
C     NUF           FAN EFFICIENCY                     ----      INPUT PARAM
C     COR           THIS IS A LOGICAL VARIABLE WHICH             INPUT PARAM
C                   ELIMINATES THE FAN FLOW CORRECTIONS
C                   (I.E. FOR PIN/PO AND TIN/TO ) WHEN
C                   COR=0.0.
C     RPM           FAN SPEED                          RPM       INPUT VAR
C     PIN           FAN INPUT PRESSURE                 PSIA      OUTPUT VAR
C     TIN           FAN INLET TEMPERATURE              DEGR      OUTPUT VAR
C     T2            FAN OUTLET TEMPERATURE             DEGR      OUTPUT VAR
C     PR            PRESSURE RATIO  P2/PIN             ----      OUTPUT VAR
C     W2            FAN FLOW RATE                      LB/SEC    OUTPUT VAR
C
C
      DIMENSION FANFLO(80),STALL(40)
      COMMON/CIO/IREAD,IWRITE,IDIAG
      COMMON/ERMESS/IFATAL,IERR
      REAL NUI,NUF
      IF(NUI.EQ..99999) NUI=0.
      IF(NUF.EQ..99999) NUF=1.0
      IF(PRM.EQ..99999) PRM=PAM
      IF(TRM.EQ..99999) TRM=TAM
      IF(COR.EQ..99999) COR=0.0
C    CALCULATION OF THE INPUT PRESSURE AND TEMPERATURE
C    WITH RAM EFFECTS.
      PIN=(PRM-PAM)*NUI+PAM
      TIN=(TRM-TAM)*NUI+TAM
      PR=P2/PIN
C    CALCULATION OF THE GAS CONSTANTS AND THE RATIO OF
C    SPECIFIC HEATS BASED ON INLET TEMPERATURE.
      CP=SHCP(TIN,0.0)
      R=53.35
```

Table 162: LISTING FOR SUBROUTINE FN (CONCLUDED)

```
      GAMMA=1.+R/(778.*CP-R)
      G1=(GAMMA-1.)/GAMMA
C     CALCULATION OF THE CHANGE IN AIR TEMPERATURE PRODUCED
C     BY THE FAN AND THE RESULTING OUTPUT TEMPERATURE.
      DELT=(TIN/NUF)*(PR**G1-1.)
      T2=TIN+DELT
      NX=FANFLO(2)
      NY=FANFLO(3)
C     TABLE LOOK UP ROUTINE FOR DETERMINING THE IDEAL FAN FLOW
C     RATE GIVEN THE PRESSURE RATIO AND FAN RPM.
      WIDEAL=TBLU2(PR,RPM,FANFLO(4+NY),FANFLO(4),FANFLO(4+NX+NY),1,1,
     1NX,NY,NX,NY)
      RATIO=1.0
C     CORRECTION OF THE IDEAL FLOW RATE FOR PIN/PO AND TIN/TO
C     RATIOS. THIS CORRECTION IS NOT MADE IF COR=0.0.
      IF(COR.EQ.0.0) GO TO 177
      DELTA=PIN/14.696
      THETA=TIN/518.7
      RATIO=DELTA/SQRT(THETA)
C     CORRECTED FAN FLOW RATE.
  177 W2=WIDEAL*RATIO
C     LOGIC TO DETERMINE IF THE FAN IS OPERATING IN THE
C     STALL RANGE. IT USES THE TABLE LOOK UP DATA FOR
C     FAN STALL POINTS -VS- PRESSURE RATIO. IF STALL DATA
C     HAS NOT BEEN INPUT, THIS SECTION IS SKIPPED.
      IF(STALL(2).EQ.1.99999) GO TO 277
      NX=STALL(2)
      WSTALL=TBLU1(PR,STALL(4),STALL(4+NX),1,NX)
      IF(WSTALL.LE.WIDEAL) GO TO 277
      IF(IERR.NE.1) GO TO 277
      WRITE(IWRITE,270)
  270 FORMAT(10X,42HTHE FAN IS OPERATING IN THE STALL REGION. )
  277 RETURN
      END
```

Table 163: LISTING FOR SUBROUTINE FNFLOW

```
CFNFLOWS
      SUBROUTINE FNFLOW(P1,P2,T,CA,AK,FN,W)
C
C    PURPOSE - TO CALCULATE FLOW (W) AND/OR CHESTER SMITH COMPRESSIBLE
C              FLOW FUNCTION (FN) GIVEN UPSTREAM AND DOWNSTREAM PRESSURES
C              (P1 AND P2), TEMPERATURE (T), EFFECTIVE AREA (CA) AND LOSS
C              FACTOR (AK).
C              * NOTE * IF P2 IS GREATER THAN P1, OUTPUTS FN AND W WILL
C              BE NEGATIVE.  FN IS BASED ON  P/PS.
C              FOR ORIFICE CALCULATIONS, INPUT AK=1.
C
C    METHOD  - USES TABULAR DATA FOR FN=F(PRESSURE RATIO)
C
C
C    WRITTEN BY  -   ADAM LLOYD          LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - K FACTOR MUST NOT EQUAL ZERO
C
C
C    INPUT/OUTPUT LIST
C
C    P1          UPSTREAM PRESSURE                         PSIA        INPUT
C    P2          DOWNSTREAM PRESSURE                       PSIA        INPUT
C    T           TEMPERATURE                               DEGR        INPUT
C    CA          EFFECTIVE AREA                            IN2         INPUT
C    AK          K FACTOR (MUST NOT EQUAL ZERO)            ---         INPUT
C    FN          COMPRESSIBLE FLOW FACTOR                  ---         OUTPUT
C    W           FLOW RATE                                 LB/MIN      OUTPUT
      DIMENSION ANF(34)
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      DATA ANF/
     1    0.0    ,.1453 ,.2044 ,.2491 ,.2862 ,.3183 ,.3469 ,.3729 ,
     2    .3966 ,.4186 ,.4390 ,.4582 ,.4762 ,.5394 ,.5916 ,.6361 ,
     3    .6745 ,.7081 ,.7378 ,.7642 ,.8188 ,.8609 ,.8939 ,.9199 ,
     4    .9404 ,.9566 ,.9693 ,.9791 ,.9866 ,.9920 ,.9959 ,.9984 ,
     5    .9997 ,1.0   /
      IF(P2.GT.P1)GO TO 100
      PS=(P1*(AK-1.)+P2)/AK
      IF(PS.GE.0.)PS=AMAX1(PS,0.00001)
      IF(PS.LT.0.)PS=AMIN1(PS,-.00001)
      PR=P1/PS
      IF(PR.LT.0.)GO TO 10
      X1=1.+ 200.*(PR-1.)
      IF(PR.GT.1.06) X1=13. + 50.*(PR-1.06)
      IF(PR.GT.1.20) X1=20. + 20.*(PR-1.20)
      IF(PR.GT.1.90) GO TO 10
      I=X1
      FN=(X1-I)*(ANF(I+1)-ANF(I))+ANF(I)
      GO TO 20
C    CHOKED FLOW
 10   FN=1.
 20   W=31.9*FN*CA*P1/SQRT(T)
      IF(P1.LT.0.) W=-W
      GO TO 200
```

Table 163: LISTING FOR SUBROUTINE FNFLOW (CONCLUDED)

```
C    REVERSE FLOW - FN AND W ARE NEGATIVE
 100  PS=(P2*(AK-1.)+P1)/AK
      IF(PS.GE.0.)PS=AMAX1(PS,0.00001)
      IF(PS.LT.0.)PS=AMIN1(PS,-.00001)
      PR=P2/PS
      IF(P1.LT.0.)GO TO 150
      IF(PR.LT.0.)GO TO 110
      X1=1.+200.*(PR-1.)
      IF(PR.GT.1.06) X1=13. + 50.*(PR-1.06)
      IF(PR.GT.1.20) X1=20. + 20.*(PR-1.20)
      IF(PR.GT.1.90) GO TO 110
      I=X1
      FN=-(X1-I)*(ANF(I+1)-ANF(I)) - ANF(I)
      GO TO 120
C    CHOKED REVERSE FLOW
 110  FN=-1.
 120  W=31.9*FN*CA*P2/SQRT(T)
      GO TO 200
C    REVERSE FLOW - NEGATIVE P1
 150  FN=-1.+.1*P1
      W=31.9*FN*CA*P2/SQRT(T)
      IF(P2.LT.0.) W=-W
 200  RETURN
      END
```

Table 164: LISTING FOR SUBROUTINE FORCE

```
CFORCE
      SUBROUTINE FORCE
C FORCES AND TORQUES ASSOCIATED WITH A PARTICULAR ACLS ORIENTATION
C ARE CALCULATED
C
      COMMON/ESTMD/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
      COMMON/DYNAMIC/TIME,FTIME,DTIME,IQ,NQ,DVCH,DVTK,VELX,DERY(13)
     1,DVCHP
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/VEHCL/MASS,AIX,AIZ,AIXY,AIYZ,AIZX,CC,GG,FF,AIFAN
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      COMMON/SHAPE/AGAPI(100),AGAPR(100),ATKI(100),ATKR(100),ATKCHI(100)
     1,ATKCHR(100),ATKATI(100),ATKATR(100),ACHI(100),ACHR(100)
     2,ATKCNI(100),ATKCNR(100),VCHI(100),VCHR(100),VTKI(100),VTKR(100)
     3,XCH(100),XTK(100),ZCH(100),ZTK(100),PERI(100)
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/FORTQ/FCP,FTP,FORCT,FDF,FORCEY,TCPX,TTPX,TORQTX,TDFX,
     1TORQUEX,TCPZ,TTPZ,TORQTZ,TDFZ,TORFZ,TORQUEZ
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
C CALCULATE TRANSCENDENTALS ONLY ONCE
      CSSCS=COS(PHIE)*SIN(THETAE)*SIN(SIE)-COS(SIE)*SIN(PHIE)
      CPCT=COS(PHIE)*COS(THETAE)
C CLEAR TOTAL FORCES AND TORQUES TO ZERO
      FORCT=0.0
      TTPX=0.0
      TTPZ=0.0
      TCPX=0.0
      TCPZ=0.0
      TORFZ=0.0
      TORQTX=0.0
      TORQTZ=0.0
C***********************************************************************
C   FORCES AND TORQUES INDEPENDENT OF SEGMENTS INDIVIDUALLY
C***********************************************************************
C HEAVE FORCES CUSHION AND TRUNK
      FCP=PCH*ACH
      FTP=PTK*ATKCN
C COMPUTE VELOCITY FOR DRAG FORCE
      V=VELX*CSSCS+SINKRT*CPCT
      SIGN=1.
      IF(V.GT.0.0)SIGN=-1.0
C HEAVE DRAG FORCE
      FDF=0.5*HDC*PHA*RHO*V*V*SIGN
C DRAG TORQUE
      TDFZ=FDF*CENFX
      TDFX=-FDF*CENFZ
```

Table 164: LISTING FOR SUBROUTINE FORCE (CONCLUDED)

```
C*****************************************************************************
C  FORCES AND TORQUES DEPENDENT ON SEGMENTS INDIVIDUALLY
C*****************************************************************************
C SUM INDIVIDIDUAL SEGMENTS TO FIND TOTALS
      DO 10 I=1,NSTOP
C CUSHION PRESSURE TORQUES
      TCPZ=TCPZ+(XCH(I)-CC)*PCH*(ACHI(I)-ACHR(I))
      TCPX=TCPX-(ZCH(I)-FF)*PCH*(ACHI(I)-ACHR(I))
C TORQUES DUE TO CONTACT FORCE
      TTPZ=TTPZ+(XTK(I)-CC)*(PTK*(ATKCNI(I)+ATKCNR(I)))
      TTPX=TTPX-(ZTK(I)-FF)*(PTK*(ATKCNI(I)+ATKCNR(I)))
      IF((ATKCNI(I).GT.0.0).OR.(ATKCNR(I).GT.0.)) GO TO 11
      GO TO 10
C DAMPING FORCE AND TORQUES
   11 VELT=SINKRT*CPCT+DPHI*(XTK(I)-CC)-DTHETA*(ZTK(I)-FF)
      FORD=-VELT*DAMPC*PERI(I)
      FORCT=FORCT+FORD
      TORQTZ=TORQTZ+(XTK(I)-CC)*FORD
      TORQTX=TORQTX-(ZTK(I)-FF)*FORD
      IF(VELX.EQ.0.0) GO TO 10
C FRICTION TORQUE
      TORFZ=TORFZ-(GG+YGH(I))*PTK*(ATKCNI(I)+ATKCNR(I))*U
   10 CONTINUE
C*****************************************************************************
C  SUMMATION OF FORCE AND TORQUE COMPONENTS
C*****************************************************************************
C TOTAL HEAVE FORCE
      FORCEY=(FCP+FTP+FORCT+FDF)*CPCT
C TOTAL TORQUE X AXIS
      TORQUEX=TCPX+TTPX+TORQTX+TDFX
C TOTAL TORQUE Z AXIS
      TORQUEZ=TCPZ+TTPZ+TORQTZ+TDFZ+TORFZ
      RETURN
      END
```

Table 165: LISTING FOR SUBROUTINE FR

```
CFR
      SUBROUTINE FR(PRTAB,ET,T2,W2,P1,P1DOT,IP1,WCO,WKC,ETC,
     1 T1,W1,P2,EN,UA,TAM)
C
C    PURPOSE - ANALYSIS OF COMPRESSOR OR FAN, WITH CAPABILITY TO
C              MODEL FLOW IN SECOND QUADRANT (REVERSE FLOW) AND
C              FOURTH QUADRANT (PRESSURE RATIOS LESS THAN ONE).
C
C    METHOD  - USES INPUT TABLES DEFINING STEADY STATE CHARACTERISTICS
C              OF COMPRESSOR. UNLIKE CM AND CN, INDEPENDENT VARIABLES
C              ON MAPS ARE CORRECTED SPEED AND CORRECTED FLOW. PRESSURE
C              RATIO AND EFFICIENCY ARE DEPENDENT VARIABLES.
C              NOTE THAT IN THE FOURTH QUADRANT (TURBINE REGION), A
C              POSITIVE VALUE OF EFFICIENCY CORRESPONDS TO A TEMPERATURE
C              DROP, AND NEGATIVE TO TEMPERATURE RISE.
C              HEAT TRANSFER TO AMBIENT CALCULATED FROM INPUT
C              VALUE OF UA
C
C    LIMITATIONS - TEMPERATURE CALCULATIONS BASED ON T1 BEING THE INLET
C                  TEMPERATURE FOR BOTH POSITIVE AND REVERSE FLOW.
C                  MAX ALLOWABLE SIZE OF TABULAR ARRAYS 20X10 PER TABLE
C
C   WRITTEN BY    ADAM LLOYD AS COMPONENT *CR* IN
C                 ECS LIBRARY      MARCH  1977
C
C    MODIFIED BY  - MAHINDER WAHI             AUGUST   1977
C
C
C    INPUT/OUTPUT LIST
C
C    PR       PRESS RATIO=F(WCO,N/SQRT(T))        ---         INPUT  TABLE
C    ET       EFFICIENCY =F(WCO,N/SQRT(T))        ---         INPUT  TABLE
C    T2       OUTLET TEMPERATURE(PORT NO 2)       DEGR        OUTPUT  VAR
C    W2       OUTLET FLOW                         LB/MIN      OUTPUT  VAR
C    P1       INLET PRESSURE(PORT NO 1)           PSIA        OUTPUT STATE
C    P1DOT    INLET PRESSURE DERIVATIVE           PSIA/SEC    OUTPUT DERIV
C    IP1      INTEGRATOR CONTROL                  ---         PROGRAM VAR
C    WCO      COMPRESSOR CORRECTED FLOW           ---         OUTPUT  VAR
C    WKC     COMPRESSOR WORK  INPUT               FT/LBF/SEC  OUTPUT  VAR
C    ETC      COMPRESSOR EFFICIENCY               ---         OUTPUT  VAR
C   T1       INLET TEMPERATURE(PORT NO 1)        DEGR        INPUT   VAR
C    W1       INLET FLOW RATE                     LB/MIN      INPUT   VAR
C    P2       OUTLET PRESSURE(PORT NO 2)          PSIA        INPUT   VAR
C    EN       COMPRESSOR SPEED                    RPM/1000.   INPUT   VAR
C    UA       OVERALL CONDUCTANCE                 BTU/HRDEGR  INPUT  PARAM
C    TAM      EFFECTIVE LOCAL AMBIENT TEMP        DEGR        INPUT  PARAM
C
      DIMENSION PRTAB(1),ET(1)
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
C    CALCULATE GAS CONSTANTS AND RATIO OF SPECIFIC HEATS, BASED ON
C    INLET TEMPERATURE
      CP=SHCP(T1,0.)
      R=(53.3)
      GAMMA=1.+R/(778.*CP-R)
      G1=(GAMMA-1.)/GAMMA
```

Table 165: LISTING FOR SUBROUTINE FR (CONCLUDED)

```
C     CALCULATE INTERNAL TEMPERATURE TI BASED ON INPUT VALUE OF UA AND  W1
      WBAR=AMAX1(ABS(W1),.01)
      TI=TAM+(T1-TAM)/EXP(UA/(CP*WBAR))
C     WARNING DIAGNOSTIC IF TEMPERATURE CHANGE EXCEEDS 300 DEGR
      IF(ABS(T1-TI).LE.300.)GO TO 10
      IF(T1.GT.TI)TI=T1-300.
      IF(TI.GT.T1)TI=T1+300.
C     TEST FOR DIAGNOSTIC PRINTOUT
      IF(IERR.NE.1)GO TO 10
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,44HNON FATAL ERROR CALLED FROM FAN COMPONENT FR/
     1 10X,65HTEMPERATURE CHANGE FROM HEAT TRANSFER TO AMBIENT EXCEEDS 3
     200 DEGR/10X,23HCHECK INPUT VALUE OF UA)
 10   CONTINUE
C     CALCULATE COMPRESSOR PERFORMANCE FROM INPUT TABLES
      ENC=1000.*EN/SQRT(TI)
      WCO=W1*SQRT(TI)/P1
      N1=PRTAB(3)+4
      N2=PRTAB(2)+PRTAB(3)+4
      N3=PRTAB(2)
      N4=PRTAB(3)
      PR=TBLU2(WCO,ENC,PRTAB(N1),PRTAB(4),PRTAB(N2),1,1,-N3,-N4,N3,N4)
      N1=ET(3)+4
      N2=ET(2)+ET(3)+4
      N3=ET(2)
      N4=ET(3)
      ETC=TBLU2(WCO,ENC,ET(N1),ET(4),ET(N2),1,1,-N3,-N4,N3,N4)
      IF(ETC.LT.0.)GO TO 15
      ETC=AMAX1(ETC,.01)
      GO TO 20
 15   ETC=AMIN1(ETC,-.01)
 20   CONTINUE
      W2=W1
      DELT=TI*(PR**G1-1.)/ETC
      T2=TI+DELT
      TM=(TI+T2)/2.
      CPM=SHCP(TM,0.)
      WKC=W2*CPM*DELT*12.967
C     CONSTANT =778/60 AND CONVERTS WORK TO FT-LBF/SEC
      PR=AMAX1(PR,.01)
      P1CAL=P2/PR
      IF(IP1.NE.0)P1DOT=(P1CAL-P1)/.01
      RETURN
      END
```

Table 166: LISTING FOR SUBROUTINE FS

```
CFS
      SUBROUTINE FS(T2,W2,T3,W3,P1,P1DOT,IP1,T1,W1,P2,P3,AK2,D2,
     1 AK3,D3,DHY,AHT,TAM,HO,VOL,FC)
C
C   PURPOSE - ANALYSIS OF FLOW SPLIT, WITH ONE INLET PORT AND TWO
C             OUTLET PORTS
C
C
C   METHOD  - SIMILAR TO DUCT MODEL DE
C
C
C   LIMITATIONS - AS FOR DE
C
C
C  WRITTEN BY ADAM LLOYD AS COMPONENT *SP* IN
C             ECS LIBRARY      NOV. 1975
C
C   MODIFIED BY  - MAHINDER WAHI                          AUGUST  1977
C
C
C   INPUT/OUTPUT LIST
C
C   T2          OUTLET TEMPERATURE         )PORT      DEGR      OUTPUT VAR
C   W2          OUTLET FLOW                ) NO 2  LB/MIN    OUTPUT VAR
C   T3          OUTLET TEMPERATURE         )PORT      DEGR      OUTPUT VAR
C   W3          OUTLET FLOW                ) NO 3  LB/MIN    OUTPUT VAR
C   P1          INLET  PRESSURE            )PORT   PSIA      OUTPUT STATE
C   P1DOT       INLET PRESSURE DERIVATIVE)NO 1     PSIA/SEC  OUTPUT DERIV
C   IP1         INTEGRATOR CONTROL                 ---       PROGRAM VAR
C   T1          INLET TEMPERATURE       )PORT NO 1 DEGR      INPUT  VAR
C   W1          INLET FLOW                         LB/MIN    INPUT  VAR
C   P2          OUTLET PRESSURE (PORT NO 2)        PSIA      INPUT  VAR
C   P3          OUTLET PRESSURE (PORT NO 3)        PSIA      INPUT  VAR
C   AK2         K FACTOR (PORT NO 2)               ---       INPUT  PARAM
C   D2          DIAMETER (PORT NO 2)               IN        INPUT  PARAM
C   AK3         K FACTOR (PORT NO 3)               ---       INPUT  PARAM
C   D3          DIAMETER (PORT NO 3)               IN        INPUT  PARAM
C   DHY         HYDRAULIC DIAMETER ) TO CALCULATE  IN        INPUT  PARAM
C   AHT         HEAT TRANSFER AREA ) UA            FT2       INPUT  PARAM
C   TAM         EFFECTIVE LOCAL AMBIENT TEMP       DEGR      INPUT  PARAM
C   HO          EXTERNAL HEAT TRANSFER COEFFICIENT BTU/FT2   INPUT  PARAM
C                                                  HR DEGR
C   VOL         INTERNAL VOLUME                    FT3       INPUT  PARAM
C   FC          FREQUENCY CONTROL ON P1.(FC.GE.1.) ---       INPUT  PARAM
C               A VALUE OF FC GREATER THAN 1.
C               DECREASES FREQUENCY RESPONSE OF P1
C               CORRESPONDINGLY
C
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
C   CALCULATE GAS CONSTANT AND RATIO OF SPECIFIC HEATS, BASED ON
C   INLET TEMPERATURE
      CP=SHCP(T1,0.)
      R=53.3
      GAMMA=1.+R/(778.*CP-R)
      G1=1./(GAMMA-1.)
```

Table 166: LISTING FOR SUBROUTINE FS (CONCLUDED)

```
      G2=(GAMMA-1.)/2.
C   CALCULATE OUTLET FLOWS BASED ON INLET TEMPERATURE
      CA2=.785398*D2*D2
      CALL FNFLOW(P1,P2,T1,CA2,AK2,FN,W2)
      CA3=.785398*D3*D3
      CALL FNFLOW(P1,P3,T1,CA3,AK3,FN,W3)
C   CALCULATE OVERALL CONDUCTANCE (UA) BASED ON AVERAGE FLOW
      WBAR=(ABS(W1)+ABS(W2)+ABS(W3))/3.
      WBAR=AMAX1(WBAR,.01)
C   CALCULATE EFFECTIVE LENGTH AL FOR  HEAT TRANS COEFF CALCULATION
      AL=183.35*AHT/(DHY*DHY)
C   CONSTANT 183.35=144./(PI/4.)        AL IS IN FEET.
      HINT=HI(1,T1,T1,WBAR,0.,DHY,AL,0.)
C   THE INPUT VALUE OF THE EXTERNAL HEAT TRANSFER COEFFICIENT IS BASED
C   ON THE INTERNAL WETTED AREA. HENCE UA IS GIVEN BY
      UA=AHT*HINT*HO/(60.*(HINT+HO))
C   UA IS IN BTU/MIN DEGR
C   CALCULATE OUTLET TEMPERATURES T2 AND T3
      T2=TAM+(T1-TAM)/EXP(UA/(CP*WBAR))
      T3=T2
C   WARNING DIAGNOSTIC IF TEMPERATURE CHANGE EXCEEDS 300 DEGR
      IF(ABS(T1-T2).LE.300.)GO TO 10
      IF(T1.GT.T2) T2=T1-300.
      IF(T2.GT.T1) T2=T1+300.
      T3=T2
C   TEST FOR DIAGNOSTIC PRINT OUT
      IF(IERR.NE.1)GO TO 10
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,46HNON FATAL ERROR CALLED FROM SPLIT COMPONENT FS/
     1 10X,35HTEMPERATURE CHANGE EXCEEDS 300 DEGR)
  10  TBAR=(T1+T2+T3)/3.
      PBAR=(P1+P2+P3)/3.
      CABAR=(CA2+CA3)/2.
      AM=AMACH(PBAR,TBAR,CABAR,WBAR,0.)
      IF(IP1.NE.0)P1DOT=R*TBAR*(W1-W2-W3)*(1.+G2*AM*AM)**G1/(8640.*VOL*
     1 FC)
      RETURN
      END
```

Table 167: LISTING FOR SUBROUTINE FSFLOW

```
CFSFLOW
      SUBROUTINE FSFLOW(P1,P2,T,CA,AK,FN,SFN,W)
C
C   PURPOSE - TO CALCULATE FLOW (W) AND/OR CHESTER SMITH COMPRESSIBLE
C             FLOW FUNCTION (FN) GIVEN UPSTREAM AND DOWNSTREAM PRESSURES
C             (P1 AND P2), TEMPERATURE (T), EFFECTIVE AREA (CA) AND LOSS
C             FACTOR (AK).
C             * NOTE * IF P2 IS GREATER THAN P1, OUTPUTS FN AND W WILL
C             BE NEGATIVE.  FN IS BASED ON  P/PS.
C             FOR ORIFICE CALCULATIONS, INPUT AK=1.
C
C   METHOD  - USES TABULAR DATA FOR FN=F(PRESSURE RATIO)
C
C
C   WRITTEN BY  -   ADAM LLOYD          LATEST REVISION     FEB 78
C
C
C
C
C   LIMITATIONS - K FACTOR MUST NOT EQUAL ZERO
C
C
C   INPUT/OUTPUT LIST
C
C   P1          UPSTREAM PRESSURE                        PSIA       INPUT
C   P2          DOWNSTREAM PRESSURE                      PSIA       INPUT
C   T           TEMPERATURE                              DEGR       INPUT
C   CA          EFFECTIVE AREA                           IN2        INPUT
C   AK          K FACTOR (MUST NOT EQUAL ZERO)           ---        INPUT
C   FN          COMPRESSIBLE FLOW FACTOR                 ---        OUTPUT
C   SFN         SLOPE FN WRT P1/P2                       ---        OUTPUT
C   W           FLOW RATE                                LB/MIN     OUTPUT
      DIMENSION ANF(34),ASF(34)
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      DATA ANF/
     1    0.0   ,.1453 ,.2044 ,.2491 ,.2862 ,.3183 ,.3469 ,.3729 ,
     2    .3966 ,.4186 ,.4390 ,.4582 ,.4762 ,.5394 ,.5916 ,.6361 ,
     3    .6745 ,.7081 ,.7378 ,.7642 ,.8188 ,.8609 ,.8939 ,.9199 ,
     4    .9404 ,.9566 ,.9693 ,.9791 ,.9866 ,.9920 ,.9959 ,.9984 ,
     5    .9997 ,1.0   /
C
      DATA ASF/
     1    37.68 ,20.44 ,10.38 ,8.18  ,6.92  ,6.07  ,5.46  ,4.97  ,
     2    4.57  ,4.24  ,3.96  ,3.72  ,3.38  ,3.16  ,2.61  ,2.225 ,
     3    1.92  ,1.68  ,1.485 ,1.32  ,1.092 ,.842  ,.66   ,.52   ,
     4    .41   ,.324  ,.254  ,.196  ,.15   ,.108  ,.078  ,.05   ,
     5    .026  ,.006/
      IF(P2.GT.P1)GO TO 100
      PS=(P1*(AK-1.)+P2)/AK
      IF(PS.GE.O.)PS=AMAX1(PS,0.00001)
      IF(PS.LT.O.)PS=AMIN1(PS,-.00001)
      PR=P1/PS
      IF(PR.LT.O.)GO TO 10
      X1=1.+ 200.*(PR-1.)
      IF(PR.GT.1.06) X1=13. + 50.*(PR-1.06)
```

```
      IF(PR.GT.1.20) X1=20. + 20.*(PR-1.20)
      IF(PR.GT.1.90) GO TO 10
      I=X1
      FN= (X1-I)*(ANF(I+1)-ANF(I))+ANF(I)
      SFN=(X1-I)*(ASF(I+1)-ASF(I))+ASF(I)
      GO TO 20
C     CHOKED FLOW
 10   FN=1.
      SFN=0.
 20   W=31.9*FN*CA*P1/SQRT(T)
      IF(P1.LT.0.) W=-W
      GO TO 200
C     REVERSE FLOW - FN AND W ARE NEGATIVE
 100  PS=(P2*(AK-1.)+P1)/AK
      IF(PS.GE.0.)PS=AMAX1(PS,0.00001)
      IF(PS.LT.0.)PS=AMIN1(PS,-.00001)
      PR=P2/PS
      IF(P1.LT.0.)GO TO 150
      IF(PR.LT.0.)GO TO 110
      X1=1.+200.*(PR-1.)
      IF(PR.GT.1.06) X1=13. + 50.*(PR-1.06)
      IF(PR.GT.1.20) X1=20. + 20.*(PR-1.20)
      IF(PR.GT.1.90) GO TO 110
      I=X1
      FN= -(X1-I)*(ANF(I+1)-ANF(I)) - ANF(I)
      SFN=-(X1-I)*(ASF(I+1)-ASF(I))-ASF(I)
      GO TO 120
C     CHOKED REVERSE FLOW
 110  FN=-1.
      SFN=0.
 120  W=31.9*FN*CA*P2/SQRT(T)
      GO TO 200
C     REVERSE FLOW - NEGATIVE P1
 150  FN=-1.+.1*P1
      W=31.9*FN*CA*P2/SQRT(T)
      IF(P2.LT.0.) W=-W
 200  RETURN
      END
```

Table 168: LISTING FOR SUBROUTINE FT

```
CFT
      SUBROUTINE FT(WC,TOT,T3,W3,P1,P1DOT,IP1,T1,W1,T2,P2,P3,VOL,FC)
C
C    VERSION 2.                                              SEPT 16 1977
C
C    PURPOSE   SIMULATE A HUB OR TIP DRIVEN AXIAL TURBO FAN AS
C              USED ON THE JINDIVIK ACLS VEHICLE
C
C    METHOD  - USES INPUT TABLES DEFINING STEADY STATE CHARACTERISTICS
C              OF TURBINE AND FAN.DRIVE/BLEED AIR TURBINE INLET
C              PRESSURE IS A STATE.
C
C    CALL SEQUENCE
C    ******  TABLE  ******
C      WC      -TABLE OF CORRECTED TURBINE FLOW AS A FUNCTION OF
C              -DRIVE(BLEED AIR) TO CUSHION/TRUNK PRESSURE RATIO,
C              -ONE DIMENSIONAL TABLE
C      TOT     -TABLE OF TOTAL FLOW FROM TURBOFAN AS A FUNCTION OF
C              -CUSHION/TRUNK PRESSURE(PSFG) AND DRIVE PRESSURE(PSIA)
C              -TWO DIMENSIONAL TABLE
C    ******  OUTPUTS  ******
C      T3      -TEMPERATURE OF FAN AIR EXIT, DEG RANKINE
C      W3      -TOTAL FLOW FROM TURBOFAN TO CUSHION/TRUNK, LB/MIN(PORT NO 3
C      P1      -DRIVE/BLEED AIR PRESSURE, PSIA(PORT NO 1)
C      P1DOT   -DERIVATIVE OF P1, PSIA/SEC
C      IP1     -INTEGRATOR CONTROL FOR P1
C    ******  INPUTS  ******
C      T1      -DRIVE/BLEED AIR TEMPERATURE, DEG RANKINE
C      W1      -DRIVE/BLEED AIR FLOW RATE, LB/MIN(PORT NO 1)
C      T2      -AMBIENT AIR TERMPERATURE, DEG RANKINE
C      P2      -AMBIENT AIR PRESSURE, PSIA(PORT NO 2)
C      P3      -PRESSURE OF FAN AIR EXIT, PSIG(PORT NO 3)
C      VOL     -INTERNAL VOLUME,CU.FT.
C      FC      -FREQUENCY CONTROL ON P1 (FC.GE.1.)
C            -A VALUE OF FC GREATER THAN 1. DECREASES
C              -FREQUENCY RESPONSE OF P1 CORRESPONDINGLY
C
C    WRITTEN BY  MAHINDER WAHI                                JUNE 1977
C
      DIMENSION  WC(1),TOT(1)
      DATA R/53.32/
C    CALCULATE TURBINE FLOW RATE FROM INPUT TABLE
      PRAT = P1/P3
      PRAT= AMAX1(1.,PRAT)
      NX= WC(2)
      WCOR= TBLU1(PRAT,WC(4),WC(NX+4),1,-NX)
      W1CAL= 60*WCOR*1.55*P1/SQRT(T1)
      IF(IP1.NE.0) P1DOT=R*T1*(W1-W1CAL)/(8640.*FC*VOL)
C    CALCULATE FAN PERFORMANCE FROM INPUT TABLES
      PSF = (P3-P2)*144.
      N1= TOT(3)+4
      N2= TOT(2)+TOT(3)+4
      N3= TOT(2)
      N4= TOT(3)
      W3= 60*TBLU2(PSF,P1,TOT(N1),TOT(4),TOT(N2),1,1,-N3,-N4,N3,N4)
      W3=AMAX1(W1CAL,W3)
```

Table 168: LISTING FOR SUBROUTINE FT (CONCLUDED)

```
W2= W3-W1CAL
T3= (W1*T1+W2*T2)/W3
T3=AMAX1(400.,T3)
RETURN
END
```

Table 169: LISTING FOR SUBROUTINE FU

```
CFU
      SUBROUTINE FU(FTA,FO,FIN,AN)
C
C   PURPOSE - TO CALCULATE OUTPUT FO AS AN ARBITRARY FUNCTION OF
C             INPUT FIN USING TABULAR INPUT FTA GIVING FO=F(FIN)
C
C   METHOD  - SELF EXPLANATORY
C
C
C   LIMITATIONS - NONE
C
C
C   WRITTEN BY - ADAM LLOYD             LATEST REVISION     NOV  75
C
C
C   INPUT/OUTPUT LIST
C
C   FTA         TABULAR INPUT FO=F(FIN)       ANY           INPUT  TABLE
C   FO          OUTPUT                        ANY           OUTPUT VAR
C   FIN         INPUT                         ANY           INPUT  VAR
C   AN          DEGREE OF INTERPOLATION       ---           INPUT  PARAM
C               A NEGATIVE VALUE OF AN WILL
C               PREVENT EXTRAPOLATION BEYOND
C               TABLE LIMITS
C
      DIMENSION FTA(1)
      NA=FTA(2)*AN/ABS(AN)
      NB=FTA(2)+4
      N=ABS(AN)
      FO=TBLU1(FIN,FTA(4),FTA(NB),N,NA)
      RETURN
      END
```

Table 170: LISTING FOR SUBROUTINE FV

```
CFV
      SUBROUTINE FV(FTA,FO,FNA,FNB,AN,BN)
C
C    PURPOSE - TO CALCULATE OUTPUT FO AS AN ARBITRARY FUNCTION OF INPUT
C              VARIABLES FNA AND FNB. INPUT TABLE FTA IS USED GIVING
C              FO=F(FNA,FNB)
C
C    METHOD  - TWO DIMENSIONAL TABLE LOOKUP
C
C
C    LIMITATIONS - MAX ALLOWABLE SIZE OF TABULAR ARRAY IS 12X12.
C
C
C    WRITTEN BY  - GEORGE DULEBA                       LATEST REVISION   MAY 76
C
C
C    INPUT/OUTPUT LIST
C
C    FTA        TABULAR INPUT                          ---          INPUT TABL
C    FO         OUTPUT                                 ANY          OUTPUT VA
C    FNA        INPUT A                                ANY          INPUT  VA
C    FNB        INPUT B                                ANY          INPUT
C    AN         DEGREE OF INTERPOLATION FOR FNA        ---          INPUT  PA
C                    A NEGATIVE VALUE INDICATES THAT THE NEAREST END
C                    POINT IS TO BE USED UPON EXTRAPOLATION.
C    BN         DEGREE OF INTERPOLATION FOR FNB        ---          INPUT  PA
C                    A NEGATIVE VALUE INDICATES THAT THE NEAREST END
C                    POINT IS TO BE USED UPON EXTRAPOLATION.
C
      DIMENSION FTA(1)
      N1=FTA(3)+4
      N2=FTA(2)+FTA(3)+4
      N3=FTA(2)
      N4=FTA(3)
      N5= FTA(2)*ABS(AN)/AN
      N6= FTA(3)*ABS(BN)/BN
      NAN=ABS(AN)
      NBN=ABS(BN)
      FO=TBLU2(FNA,FNB,FTA(N1),FTA(4),FTA(N2),NAN,NBN,N5,N6,N3,N4)
      RETURN
      END
```

Table 171: LISTING FOR SUBROUTINE GW

```
CGW
      SUBROUTINE GW(UG,UGD,IUG,VG,VGD,IVG,VX,VXD,IVX,WG,WGD,IWG,
     1 WX,WXD,IWX,PG,PGD,IPG,QX,QXD,IQX,QG,RX,RXD,IRX,RG,VS,XNU,
     2 XNV,XNW,XNP,SLH,SLV,VO,SIH,SIV,B)
      DATA PI,ROOT3,ISW/3.1415927,1.7320508,0/
      DATA DPR/57.29578/
C    VERSION 1.                                         MAY  27 1977
C    PURPOSE   SIMULATE RANDOM WIND GUST COMPONENTS
C    METHOD    PASS WHITE NOISE THROUGH DRYDEN TRANSFORMS TO
C              SIMULATE U,V,W,P,Q,R, GUST VELOCITIES IN
C              ACCORDANCE WITH SECTION 3.7, MIL-F-8785B
C
C    CALL SEQUENCE
C    ******  OUTPUTS  ******
C    LINEAR VELOCITIES -- BODY AXES
C      UG,UGD,IUG      -X AXIS WIND VELOCITY STATE VARIABLES, FT/SEC
C      VG,VGD,IVG      -Y AXIS WIND VELOCITY   +       +        +
C      VX,VXD,IVX      -Y AXIS INTERMEDIATE    +       +        +
C      WG,WGD,IWG      -Z AXIS WIND VELOCITY   +       +        +
C      WX,WXD,IWX      -Z AXIS INTERMEDIATE    +       +        +
C    ANGULAR VELOCITIES -- BODY AXES
C      PG,PGD,IPG      -X AXIS ANGULAR RATE    +       +     DEG/SEC
C      QX,QXD,IQX      -Y AXIS ANGULAR RATE    +       +        +
C      RX,RXD,IRX      -Z AXIS ANGULAR RATE    +       +        +
C      QG,RG           -Y AND Z AXIS ANGULAR RATE OUTPUTS, DEG/SEC
C    VELOCITY
C      VS              -STEADY STATE (TRIM) AIRSPEED, FT/SEC
C    ******  INPUTS  ******
C      XNU,XNV,XNW     -RANDOM NOISE INPUTS FOR U,V,W GUST VELOCITIES
C      XNP             -RANDOM NOISE INPUT FOR P ANGULAR RATE GUST
C      SLH,SLV         -HORIZONTAL AND VERTICAL SCALES, FT
C      VO              -STEADY STATE AIRSPEED INPUT, FT/SEC
C      SIH,SIV         -HORIZONTAL AND VERTICAL RMS GUST INTENSITY, FT/SEC
C      B               -WING SPAN,  FT
C
C    WRITTEN BY  A.W.WARREN AS COMPONENT *WM* IN
C                FLIGHT CONTROLS LIBRARY      SEPT 1976
C
C     MODIFIED BY MAHINDER WAHI                              MAY 1977
C            SET DEFAULTS
C
      IF(ISW.EQ.1) GO TO 10
      IF(SLH.EQ. .99999) SLH=1750.
      IF(SLV.EQ. .99999) SLV=1750.
      IF(SIH.EQ. .99999) SIH=0.
      IF(SIV.EQ. .99999) SIV=0.
      ISW=1
C            COMPUTE GAINS AND COEFFICIENT TERMS
C
   10 VS=VO
      SLHP= SLH/VO
      SLVP= SLV/VO
      GAINU= SIH*SQRT((SLHP+SLHP)/PI)
      GAINV= 0.707107*GAINU*XNV
      GAINU= GAINU*XNU
      GAINW= SIV*SQRT(SLVP/PI)*XNW
```

Table 171: LISTING FOR SUBROUTINE GW (CONCLUDED)

```
      COEFH= 4.*B/(PI*VO)
      COEFV= 0.75*COEFH
      GAINP= SIV*SQRT(0.8*(0.25*PI*SLV/B)**0.333333/(SLV*VO))*XNP
C
C           COMPUTE STATE DERIVATIVES
      IF(IUG.NE.0) UGD= (GAINU-UG)/SLHP
      IF(IVX.NE.0) VXD= (GAINV-VG)/SLHP**2
      IF(IVG.NE.0) VGD= VX+(GAINV*ROOT3-VG-VG)/SLHP
      IF(IWX.NE.0) WXD= (GAINW-WG)/SLVP**2
      IF(IWG.NE.0) WGD= WX+(GAINW*ROOT3-WG-WG)/SLVP
      IF(IPG.NE.0) PGD= (GAINP*DPR-PG)/COEFH
      QG= QX + DPR*WG/(VO*COEFH)
      IF(IQX.NE.0)QXD= -QG/COEFH
      RG= RX - DPR*VG/(VO*COEFV)
      IF(IRX.NE.0) RXD= -RG/COEFV
C
      RETURN
      END
```

Table 172: LISTING FOR SUBROUTINE HI

```
CHI
      FUNCTION HI(IFL,T,TW,W,SH,D,AL,CODE)
C
C    PURPOSE - TO CALCULATE HEAT TRANSFER COEFFICIENTS FOR FLOW IN DUCTS
C              AND FOR FLOW ACROSS CYLINDERS (SUCH AS SENSORS)
C
C    METHOD  - USES CONVENTIONAL CORRELATIONS FOR LAMINAR AND TURBULENT
C              FLOW. FOR TRANSITION FLOW IN DUCTS AN EQUATION OF THE FORM
C                   NU=C1*RE**C2*PR**.333
C              IS ASSUMED. C1 AND C2 ARE EVALUATED FROM VALUES OF NU AT
C              THE LIMITS OF THE LAMINAR AND TURBULENT REGIMES
C
C    LIMITATIONS -  FOR AIR FLOW ACROSS CYLINDERS THE VALID RANGE OF
C                   REYNOLDS NUMBER IS  1000.LE.RE.LE.50000
C                   FOR LIQUIDS THE VALID RANGE IS 50.LE.RE.LE.10000
C
C
C    WRITTEN BY  -   ADAM LLOYD          LATEST REVISION     NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    HI          CONVECTIVE HEAT TRANSFER COEFF.    BTU/FT2 HRDEGR OUTPUT
C    IFL         INTEGER DESIGNATING FLUID(SEE PROP)---            INPUT
C    T           FLUID TEMPERATURE                  DEGR           INPUT
C    TW          WALL TEMPERATURE (USED FOR LIQUIDS)DEGR           INPUT
C    W           FLOW RATE                          LB/MIN         INPUT
C    SH          SPECIFIC HUMIDITY OF VAPOR         LB/LB          INPUT
C    D           DIAMETER (DUCT)                    IN             INPUT
C    AL          LENGTH (DUCT LENGTH IN FT FOR LAM  FT OR IN       INPUT
C                FLOW OR CYLINDER DIAMETER (INCHES)
C    CODE        =0. FLOW IN DUCTS                  --             INPUT
C                =1. FLOW ACROSS CYLINDERS
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      REAL NU,NU2
      AK=PROP(IFL,3,T)
      IF(CODE.GT.0.5)GO TO 100
C    FLOW IN DUCTS
      RE=RENVX(IFL,W,T,D)
      IF(RE.GT.10000.)GO TO 20
      IF(RE.GT.2100.)GO TO 10
C    LAMINAR FLOW IN DUCTS
      IF(IFL.NE.1)GO TO 5
C    LAMINAR FLOW OF AIR IN DUCTS
      PR=PRND(1,T,SH)
      HI=6.048*(RE*PR*D/AL)**0.4*AK/D
C    CONSTANT 6.048 =1.5*12**.6*(PI/4)**.4
      GO TO 200
C    LAMINAR FLOW OF LIQUIDS IN DUCTS
 5    AMUB=PROP(IFL,2,T)
      AMUW=PROP(IFL,2,TW)
      PR=PRND(IFL,T,SH)
      HI=9.757*AK*(RE*PR*D/AL)**.333*(AMUB/AMUW)**.14/D
C    CONSTANT 9.757=1.86*(12)**.667
      GO TO 200
```

```
C    TRANSITION FLOW IN DUCTS
 10    IF(IFL.NE.1)GO TO 15
C    TRANSITION FLOW OF AIR IN DUCTS
C    CALCULATE CONSTANTS C1 AND C2. THIS REQUIRES CALCULATION OF NUSSELT
C    NUMBER AT UPPER LIMIT OF LAMINAR FLOW AND LOWER LIMIT OF TURBULENT
C    FLOW.
       PR=PRND(IFL,T,SH)
       NU=10.74776728*(PR*D/AL)**0.4
C    NU IS NUSSELT NUMBER AT RE=2100
       NU2=36.45254342*PR**0.4
       C2=.640759583*ALOG(NU2/NU)
       C1=NU/(2100.**C2*PR**.333)
       HI=12.*AK*C1*RE**C2*PR**.333/D
       GO TO 200
C    TRANSITION FLOW OF LIQUIDS IN DUCTS
 15    PR=PRND(IFL,T,SH)
       AMUB=PROP(IFL,2,T)
       AMUW=PROP(IFL,2,TW)
       VISR=(AMUB/AMUW)**.14
       NU=10.38591142*(PR*D/AL)**.333*VISR
C    NU IS NUSSELT NUMBER AT RE=2100
       NU2=36.45254342*(PR)**.333*VISR
       C2=.640759583*ALOG(NU2/NU)
       C1=NU/(2100.**C2*PR**.333)
       HI=12.*AK*C1*RE**C2*PR**.333*VISR/D
       GO TO 200
C    TURBULENT FLOW IN DUCTS
 20    IF(IFL.NE.1)GO TO 25
C    TURBULENT FLOW OF AIR IN DUCTS
       PR=PRND(1,T,SH)
       HI=.276*RE**0.8*PR**0.4*AK/D
C    CONSTANT .276=.023*12.
       GO TO 200
C    TURBULENT FLOW OF LIQUIDS IN DUCTS
 25    PR=PRND(IFL,T,SH)
       AMUB=PROP(IFL,2,T)
       AMUW=PROP(IFL,2,TW)
       HI=.276*RE**0.8*PR**0.333*(AMUB/AMUW)**14*AK/D
       GO TO 200
C    FLOW ACROSS CYLINDERS
C    SINGLE CORRELATION FOR WHOLE REYNOLDS NUMBER RANGE
 100   IF(IFL.NE.1)GO TO 120
C    FLOW OF AIR ACROSS CYLINDERS. REYNOLDS NUMBER BASED ON AL WHICH IS
C    CYLINDER DIAMETER IN INCHES
C    RE=RHO*V*AL/MU
C      =916.73*W*AL/(D*D*MU)
       AMU=PROP(1,2,T)
       RE=ABS(916.73*W*AL/(D*D*AMU))
       HI=2.88*AK*RE**0.6/AL
C    CONSTANT 2.88=.24*12.
C      CHECK IF VALID REYNOLDS NUMBER
       IF(1000.LE.RE.AND.RE.LE.50000.) GO TO 200
C    TEST FOR DIAGNOSTIC PRINT OUT
       IF(IERR.NE.1)GO TO 200
       WRITE(IWRITE,9999)
 9999 FORMAT(10X,30HNON FATAL ERROR CALLED FROM HI/
```

Table 172: LISTING FOR SUBROUTINE HI (CONCLUDED)

```
     1 10X,67HREYNOLDS NUMBER FOR FLOW OF AIR ACROSS CYLINDER OUTSIDE VA
     2LID RANGE)
      GO TO 200
C   FLOW OF LIQUID ACROSS CYLINDERS
 120  AMU=PROP(IFL,2,T)
      RE=ABS(916.73*W*AL/(D*D*AMU))
      PR=PRND(IFL,T,SH)
      HI=7.2*AK*RE**0.5*PR**0.31/AL
C   CONSTANT 7.2=.6*12.
C   CHECK IF VALID REYNOLDS NUMBER
      IF(50.LE.RE.AND.RE.LE.10000.)GO TO 200
C   TEST FOR DIAGNOSTIC PRINT OUT
      IF(IERR.NE.1)GO TO 200
      WRITE(IWRITE,9998)
 9998 FORMAT(10X,30HNON FATAL ERROR CALLED FROM HI/
     1 10X,70HREYNOLDS NUMBER FOR FLOW OF LIQUID ACROSS CYLINDER OUTSIDE
     2 VALID RANGE)
 200  RETURN
      END
```

Table 173: LISTING FOR SUBROUTINE HYCURV

```
CHYCURV
      SUBROUTINE HYCURV(X,Z)
C SUBROUTINE TO CALCULATE POSITIONS OF
C SIDE TRUNK LOBES
C THE POSITION  (EXPRESSED BY HY) DEPENDS ON PRESSURES.
C I.E. HY/HYI=F(PCH/PTK)
      COMMON/HCUR/AH0,AH1,AH2,AH3
C FORCE INPUT PRESSURE RATIO BETWEEN 0.0 AND 1.0
      X=AMIN1(1.0,AMAX1(0.0,X))
      Z=AH0+AH1*X+AH2*X*X+AH3*X*X*X
      IF(Z.LT.0.1)Z=0.1
      IF (Z.GT.1.0) Z=1.0
      RETURN
      END
```

Table 174: LISTING FOR SUBROUTINE IC

```
CIC
      SUBROUTINE IC(NSET,NPTS,ITYPE,A,B,LO,DPR)
C
C   VERSION 3.          FEB.20 1978
C   PURPOSE - SOLVE FOR TRUNK ELEMENT PARAMETERIC DATA FOR FREE
C             AND LOADED ACLS TRUNK SHAPES AT INITIAL CONDITIONS
C
C             ITYPE=1   DATA GENERATED FOR MEMBRANE TRUNK
C             ITYPE=0   DATA GENERATED FOR FROZEN TRUNK
C
C   METHOD  - SEE AFFDL-TR-71-50, THEORY OF AN ACLS
C             FOR AIRCRAFT BY KENNERLY H. DIGGES, JUNE 1971
C
C   LIMITATIONS - CALLED ONLY BY THE TRUNK MODEL COMPONENT TK
C   **CALL SEQUENCE**
C   ***OUTPUTS***
C   ARRAYS   ACV,AS,L1,L3,L3P,Y0,AND Z0 FOR VARIOUS
C          TRUNK TO CUSHION PRESSURE RATIOS.(SEE TRUNK NOMEN.)
C   ***INPUTS***
C   NSET,NPTS,ITYPE,A,B,LO,AND DPR  (SEE TRUNK NOMEN.)
C
C   WRITTEN BY  - J.R.KILNER AND M.K.WAHI
C
      COMMON/SECT/AZ0(10,6),AY0(10,10,6),AL1(10,10,6),AL3P(10,10,6),
     * AL3(10,10,6),AACV(10,10,6),AAS(10,10,6)
      COMMON/CIO/IREAD,IWRITE,IDIAG
      DIMENSION ITYPE(6)
      DIMENSION A(6),B(6),LO(6),X(5),R(5),P(6),AJ(5,5),Z(5,7),IP(6)
      DIMENSION D1(10),D2(10),R1(10,10,6),R2(10,10,6)
      REAL LO
      EXTERNAL ICFS
      EXTERNAL ICLS
C
      DPR=1.0/NPTS
      ANPT=NPTS
      P(1)=0.
C
      DO 10 I=1,NSET
C
      P(2)=A(I)
      P(3)=B(I)
      P(4)=LO(I)
      X(1)=0.25*LO(I)
      X(2)=0.25*LO(I)
      X(3)=2.0
      X(4)=2.0
      IP(1)=-50
      JQ=0
C
C  SOLVE FOR FREE SHAPE PARAMETER DATA
C
      DO 20 J=1,NPTS
      P(5)=DPR*(J-1)
C
      CALL QNWT(X,4,5,ICFS,P,.0001,IP,JQ,R,RMS,AJ,Z)
C
```

```
      JQ=1
      AZO(J,I)=X(2)*(1.-COS(X(4)))
C
      IF(RMS.LE. .0001) GO TO 15
      WRITE(IWRITE,14) RMS,NSET
   14 FORMAT(//, 5X, 20HQNWT FAILURE IN ICFS,
     *      /5X,*RMS=*,E12.4,5X,*NSET=*,I2//)
      STOP
C
   15 CONTINUE
      IF(ITYPE(I).EQ.0)GOTO 90
   20 CONTINUE
C
C  SOLVE FOR LOADED TRUNK MEMBRANE PARAMETRIC DATA
C
      X(1)=.25*LO(I)
      X(2)=.25*LO(I)
      X(3)=2.
      X(4)=2.
      X(5)=0.
      JQ=0
      DO 30 J=1,NPTS
      P(5)=DPR*(J-1)
      DO 30 K=1,NPTS
      P(6)=AZO(J,I)*(1.-(K-1)/ANPT)
C
      CALL QNWT(X,5,5,ICLS,P,.0001,IP,0,R,RMS,AJ,Z)
C
      JQ=1
      AYO(J,K,I)=X(2)*SIN(X(4))
      AL1(J,K,I)=X(1)*X(3)
      AL3(J,K,I)=X(5)
      AACV(J,K,I)=.25*X(2)*X(2)*(2.*X(4)-SIN(2.*X(4)))
      AAS(J,K,I)=.25*X(1)*X(1)*(2.*X(3)+SIN(2.*X(3)))
     *           +AACV(J,K,I)+(P(6)-X(1))*(A(I)-AYO(J,K,I))+X(1)*X(5)
      R1(J,K,I)=X(1)
      R2(J,K,I)=X(2)
C
      IF(RMS.LE. .0001) GO TO 25
      WRITE(IWRITE,24) RMS,NSET
   24 FORMAT(//, 5X, 20HQNWT FAILURE IN ICLS,
     *        /5X,*RMS=*,E12.4,5X,*NSET=*,I2//)
      STOP
C
   25 CONTINUE
   30 CONTINUE
C
      GOTO 33
C
C  SOLVE FOR LOADED TRUNK FROZEN PARAMETRIC DATA
C
   90 C1=X(2)*SIN(X(4))
      C2=AZO(1,I)-X(2)
C
      DO 92 K=1,NPTS
C
```

```
      ZO=AZO(1,I)*(1.-(K-1)/ANPT)
      ZCR=(ZO-C2)/X(2)
      IF(ZCR.GT.1.)ZCR=1.
      THE=ACOS(ZCR)
      AL1(1,K,I)=X(1)*(X(3)-THE)
      AL3(1,K,I)=2.*X(2)*SIN(THE)
      AL3P(1,K,I)=2.*THE*X(2)
      AYO(1,K,I)=C1-.5*AL3(1,K,I)
      AACV(1,K,I)=.5*((C1-AL3(1,K,I))*C2+(X(4)-THE)*X(2)*X(2)
     *             -.5*(ZO-C2)*AL3(1,K,I))
      AAS(1,K,I)=.5*((X(3)+X(4)-2.*THE)*X(2)*X(2)+C1*C2
     *        +(A(I)-C1)*(C2-B(I))+(ZO-C2)*AL3(1,K,I))
      R1(1,K,I)=X(1)
      R2(1,K,I)=X(2)
   92 CONTINUE
C
      DO 94 J=2,NPTS
      AZO(J,I)=AZO(1,I)
      DO 94 K=1,NPTS
      AL1(J,K,I)=AL1(1,K,I)
      AL3(J,K,I)=AL3(1,K,I)
      AL3P(J,K,I)=AL3P(1,K,I)
      AYO(J,K,I)=AYO(1,K,I)
      AACV(J,K,I)=AACV(1,K,I)
      AAS(J,K,I)=AAS(1,K,I)
      R1(J,K,I)=R1(1,K,I)
      R2(J,K,I)=R2(1,K,I)
   94 CONTINUE
C
C
C  OUTPUT ARRAYS
C
 33   IF(I.EQ.1)WRITE(IWRITE,6000)
      IF(ITYPE(I).EQ.1)WRITE(IWRITE,6002)I
      IF(ITYPE(I).EQ.0)WRITE(IWRITE,6004)I
      WRITE(IWRITE,6006)NPTS,DPR,A(I),B(I),LO(I)
      WRITE(IWRITE,6008)AZO(1,I),AYO(1,1,I),AL1(1,1,I),AACV(1,1,I),
     *                  AAS(1,1,I)
6000  FORMAT(39H1***** TRUNK ELEMENT SECTION PROPERTIES ,12(6H******))
6002  FORMAT(////14H *** DATA SET ,I1,30H   *** MEMBRANE TRUNK ELEMENT *,
     * 4(4H****)/)
6004  FORMAT(////14H *** DATA SET ,I1,30H   *** FROZEN TRUNK ELEMENT ***,
     * 4(4H****)/)
6006  FORMAT(5X,*NPTS=*,I2,2X,*DPR=*,F4.3,8X,*A=*,F6.2,*    B=*,F6.2,
     * *    LO=*,F6.2/)
6008  FORMAT(5X, 50HELEMENT PROPERTIES FREE OF GROUND EFFECTS AND PR=0,
     * 5X,*ZO=*,F6.2,*  YO=*,F6.2,*  L1=*,F6.2,*  ACV=*,F7.1,*  AS=*,
     * F7.1//)
C
      WRITE(IWRITE,6010)
      DO 40 J=1,NPTS
      DO 42 K=1,NPTS
42    D1(K)=AZO(J,I)*(NPTS-K+1)/ANPT
      WRITE(IWRITE,6020)(D1(K),K=1,NPTS)
40    CONTINUE
6010  FORMAT(5X,16H*** ZO ARRAY ***//7X,*1ST COLUMN IS OUT-OF-GROUND*,
```

```
     *         *-EFFECTS ZO FOR INCREASING VALUES OF PR*/)
6020  FORMAT(5X,10F8.2)
6022  FORMAT(5X,10F8.1)
C
      WRITE(IWRITE,6030)
      DO 50 J=1,NPTS
      WRITE(IWRITE,6020)(AYO(J,K,I),K=1,NPTS)
50    CONTINUE
C
      WRITE(IWRITE,6032)
      DO 52 J=1,NPTS
      WRITE(IWRITE,6020)(AL1(J,K,I),K=1,NPTS)
52    CONTINUE
C
      WRITE(IWRITE,6034)
      DO 54 J=1,NPTS
      WRITE(IWRITE,6020)(AL3(J,K,I),K=1,NPTS)
54    CONTINUE
C
      IF(ITYPE(I).EQ.1)GOTO 55
      WRITE(IWRITE,6036)
      DO 56 J=1,NPTS
      WRITE(IWRITE,6020)(AL3(J,K,I),K=1,NPTS)
56    CONTINUE
C
55    WRITE(IWRITE,6038)
      DO 58 J=1,NPTS
      WRITE(IWRITE,6022)(AACV(J,K,I),K=1,NPTS)
58    CONTINUE
C
      WRITE(IWRITE,6040)
      DO 60 J=1,NPTS
      WRITE(IWRITE,6022)(AAS(J,K,I),K=1,NPTS)
60    CONTINUE
C
      WRITE(IWRITE,6042)
      DO 62 J=1,NPTS
      WRITE(IWRITE,6020)(R1(J,K,I),K=1,NPTS)
   62 CONTINUE
C
      WRITE(IWRITE,6044)
      DO 64 J=1,NPTS
      WRITE(IWRITE,6020)(R2(J,K,I),K=1,NPTS)
   64 CONTINUE
6030  FORMAT(//5X,*THE FOLLOWING ARRAYS CORRESPOND ELEMENT-TO-ELEMENT*
     * * WITH THE ZO ARRAY*//5X,16H*** YO ARRAY ***/)
6032  FORMAT(//5X,16H*** L1 ARRAY ***/)
6034  FORMAT(//5X,16H*** L3 ARRAY ***/)
6036  FORMAT(//5X,17H*** L3P ARRAY ***/)
6038  FORMAT(//5X,17H*** ACV ARRAY ***/)
6040  FORMAT(//5X,16H*** AS ARRAY ***/)
6042  FORMAT(//5X,16H*** R1 ARRAY ***/)
6044  FORMAT(//5X,16H*** R2 ARRAY ***/)
C
   10 CONTINUE
C
```

Table 174: LISTING FOR SUBROUTINE IC (CONCLUDED)

```
RETURN
END
```

Table 175: LISTING FOR SUBROUTINE ICB

```
CICB
      SUBROUTINE ICB(NSET,NPTS,A,B,LO,GA,GB,DMU,
     *           AZO,AYO,AL1,AL3,AAS,BYO,BL1,BL3,BAS)
C
C    PURPOSE - SOLVE FOR AIR BAG ELEMENT PARAMETERIC DATA FOR FREE
C              AND LOADED AIR BAG ELEMENT SHAPES AT INITIAL CONDITIONS
C
C    METHOD  -SOLVE MEMBRANE GEOMETRY AND FORCE BALANCE EQUATIONS
C              FOR EVEN INCREMENTS OF ZO AND MUT AND STORES
C              PARAMETER VALUES IN TABLE LOOK UP ARRAYS.
C
C    LIMITATIONS - CALLED ONLY BY THE AIR BAG MODEL COMPONENT AB
C    **CALL SEQUENCE**
C    ***OUTPUTS***
C    ARRAYS   YO,L1,L3,AS  FOR VARIOUS ZO AND MUT
C
C    ***INPUTS***
C    NSET,NPTS,A,B,LO,GA,GB,AND DMU  (SEE AIR BAG NOMEN.)
C
C    WRITTEN BY  - J.R.KILNER              APRIL 10,1978
C
      DIMENSION AZO(6),AYO(10,10,6),AL1(10,10,6),AL3(10,10,6),
     * AAS(10,10,6),BYO(10,10,6),BL1(10,10,6),BL3(10,10,6),
     * BAS(10,10,6)
      COMMON/CIO/IREAD,IWRITE,IDIAG
      DIMENSION A(6),B(6),LO(6),X(5),R(5),P(16),AJ(5,5),Z(5,7),IP(6)
      DIMENSION D1(10),D2(10),GA(6),GB(6)
      REAL LO
      EXTERNAL ICFSB
      EXTERNAL ICLSB
C
      DMU=1.0/NPTS
      ANPT=NPTS
      P(1)=0.
      IP(1)=100
C
      DO 10 I=1,NSET
C
      P(2)=A(I)
      P(3)=B(I)
      P(4)=LO(I)
      P(7)=.01745*GB(I)
      P(8)=.01745*GA(I)
      P(9)=SIN(P(7))
      P(10)=COS(P(7))
      P(11)=SIN(P(8))
      P(12)=COS(P(8))
C
      X(1)=0.25*LO(I)
      X(2)=2.0
      X(3)=2.0
      JQ=0
C
C   SOLVE FOR FREE SHAPE PARAMETER DATA
C
C
```

```
      CALL QNWT(X,3,5,ICFSB,P,.0001,IP,JQ,R,RMS,AJ,Z)
C
      AZO(I)=X(1)*(1.-COS(X(3)))
C
      IF(RMS.LE. .0001) GO TO 15
      WRITE(IWRITE,14) RMS,NSET
   14 FORMAT(//, 5X, 21HQNWT FAILURE IN ICFSB,
     *       /5X,*RMS=*,E12.4,5X,*NSET=*,I2//)
      STOP
C
   15 CONTINUE
C
C  SOLVE FOR LOADED ELEMENT MEMBRANE PARAMETRIC DATA
C
      DMUX=DMU
      DO 30 M=1,2
      IF(M.EQ.2)DMUX=-DMU
      DO 30 J=1,NPTS
      X(1)=.25*LO(I)
      X(2)=.25*LO(I)
      X(3)=.5*LO(I)
      X(4)=.5*LO(I)
      X(5)=0.
      JQ=0
      P(5)=DMUX*(J-1)
      DO 30 K=1,NPTS
      P(6)=AZO(I)*(1.-(K-1)/ANPT)
      IX=0
C
27    CALL QNWT(X,5,5,ICLSB,P,.0001,IP,0,R,RMS,AJ,Z)
C
      IF(RMS.LE. .0001) GO TO 25
      IF(IX.EQ.1)GOTO 26
      IX=1
      X(1)=.25*LO(I)
      X(2)=.25*LO(I)
      X(3)=.5*LO(I)
      X(4)=.5*LO(I)
      X(5)=0.
      JQ=0
      GOTO 27
   26 WRITE(IWRITE,24) RMS,NSET
   24 FORMAT(//, 5X, 21HQNWT FAILURE IN ICLSB,
     *          /5X,*RMS=*,E12.4,5X,*NSET=*,I2//)
      STOP
C
   25 CONTINUE
      JQ=1
      DYO=X(2)*SIN(P(14))-P(16)*P(12)
      DL1=X(3)
      DL3=X(5)
      AX=X(1)*X(1)*(.5*P(13)+.25*SIN(2.*P(13))-SIN(P(13)))
     *  +X(2)*X(2)*(.5*P(14)+.25*SIN(2.*P(14))-SIN(P(14)))
      AXX=.5*(X(1)*(1.-COS(P(13)))+P(6)+P(3))*P(15)*P(10)
     *    +.5*(X(2)*(1.-COS(P(14)))+P(6)      )*P(16)*P(12)
      DAS=AX+AXX+P(6)*P(2)+.5*P(2)*P(3)
```

```
C
      IF(M.EQ.2)GOTO 33
C   STORE POSITIVE MU IN A ARRAYS
      AYO(J,K,I)=DYO
      AL1(J,K,I)=DL1
      AL3(J,K,I)=DL3
      AAS(J,K,I)=DAS
      GOTO 34
C   STORE NEGATIVE MU IN B ARRAYS
33    BYO(J,K,I)=DYO
      BL1(J,K,I)=DL1
      BL3(J,K,I)=DL3
      BAS(J,K,I)=DAS
34    CONTINUE
C
   30 CONTINUE
C
C  OUTPUT ARRAYS
C
      IF(I.EQ.1)WRITE(IWRITE,6000)
      WRITE(IWRITE,6002)I
      WRITE(IWRITE,6006)NPTS,DMU,A(I),B(I),LO(I),GA(I),GB(I)
      WRITE(IWRITE,6008)AZO(I),AYO(1,1,I),AL1(1,1,I),AAS(1,1,I)
6000  FORMAT(41H1***** AIR BAG ELEMENT SECTION PROPERTIES ,14(5H*****))
6002  FORMAT(////14H *** DATA SET ,I1,24H  *** MEMBRANE ELEMENT *,
     * 4(4H****)/)
6006  FORMAT(5X,*NPTS=*,I2,2X,*DMU=*,F4.3,8X,*A=*,F6.2,*   B=*,F6.2,
     * *    LO=*,F6.2,*   GA=*,F4.1,*   GB=*,F4.1/)
6008  FORMAT(5X,*FREE SHAPE ELEMENT PROPERTIES *,
     * 5X,*ZO=*,F6.2,*   YO=*,F6.2,*   L1=*,F6.2,*  AS=*,F7.1//)
C
      WRITE(IWRITE,6010)
      DO 42 K=1,NPTS
42    D1(K)=AZO(I)*(NPTS-K+1)/ANPT
      WRITE(IWRITE,6020)(D1(K),K=1,NPTS)
6010  FORMAT(5X,16H*** ZO ARRAY ***/)
6020  FORMAT(5X,10F8.2)
6022  FORMAT(5X,10F8.1)
C
      WRITE(IWRITE,6030)
      DO 50 J=1,NPTS
      WRITE(IWRITE,6020)(AYO(J,K,I),K=1,NPTS)
50    CONTINUE
      WRITE(IWRITE,6042)
      DO 51 J=1,NPTS
      WRITE(IWRITE,6020)(BYO(J,K,I),K=1,NPTS)
51    CONTINUE
C
      WRITE(IWRITE,6032)
      DO 52 J=1,NPTS
      WRITE(IWRITE,6020)(AL1(J,K,I),K=1,NPTS)
52    CONTINUE
      WRITE(IWRITE,6042)
      DO 53 J=1,NPTS
      WRITE(IWRITE,6020)(BL1(J,K,I),K=1,NPTS)
53    CONTINUE
```

Table 175: LISTING FOR SUBROUTINE ICB (CONCLUDED)

```
C
      WRITE(IWRITE,6034)
      DO 54 J=1,NPTS
      WRITE(IWRITE,6020)(AL3(J,K,I),K=1,NPTS)
54    CONTINUE
      WRITE(IWRITE,6042)
      DO 55 J=1,NPTS
      WRITE(IWRITE,6020)(BL3(J,K,I),K=1,NPTS)
55    CONTINUE
C
      WRITE(IWRITE,6040)
      DO 60 J=1,NPTS
      WRITE(IWRITE,6022)(AAS(J,K,I),K=1,NPTS)
60    CONTINUE
      WRITE(IWRITE,6042)
      DO 61 J=1,NPTS
      WRITE(IWRITE,6022)(BAS(J,K,I),K=1,NPTS)
61    CONTINUE
C
6030  FORMAT(//5X,*THE FOLLOWING ARRAYS CORRESPOND COLUMN-TO-ELEMENT*
     * * WITH THE ZO ARRAY*/5X,*ROWS CORRESPOND TO*
     * * INCREASING MAGNITUDES OF MU BEGINNING AT MU=0*/
     * 5X,*DATA GENERATED FOR POSITIVE MU STORED IN 1ST ARRAY*/
     * 5X,*DATA GENERATED FOR NEGATIVE MU STORED IN 2ND ARRAY*/
     * /5X,16H*** YO ARRAY ***/)
6032  FORMAT(//5X,16H*** L1 ARRAY ***/)
6034  FORMAT(//5X,16H*** L3 ARRAY ***/)
6040  FORMAT(//5X,16H*** AS ARRAY ***/)
6042  FORMAT(/)
C
   10 CONTINUE
C
      RETURN
      END
```

Table 176: LISTING FOR SUBROUTINE ICFS

```
CICFS
      SUBROUTINE ICFS(X,M,K,R,P)
C
C   VERSION 3.          FEB.1978
C
C   PURPOSE - DEFINE INITIAL CONDITION FREE SHAPE DIGGES MODEL
C           - FOR ACLS TRUNKS
C
C   MEHTOD  - SEE AFFDL-TR-71-50, THEORY OF AN ACLS FOR AIRCRAFT
C           - BY  KENNERLY H. DIGGES, JUNE 1971.
C
C   LIMITATIONS - USED ONLY BY ROUTINE QNWT IN SUBROUTINE IC
C
C   ***OUTPUTS***
C   X      THE SOLUTION ESTIMATE FROM THE LAST ITERATION
C   R      THE RESIDUAL VECTOR
C   ***INPUTS***
C   X      INITIAL ESTIMATE VECTOR
C   P(1)   PRINT CONTROL OPTION
C   P(1)   PRINT CONTROL OPTION
C          P=0   NO PRINT OUT
C          P=K   FOR PRINT OUT AT EVERY K-TH ITERATION
C   P(2)...P(N)   STORAGE AVAILABLE FOR PASSING DATA
C                 TO SUBROUTINE FUN(ICFS) BY QNWT.
C   M,K    DIMENSION INDICATORS FOR X AND AJ MATRICES
C   WRITTEN BY  - J.R.KILNER AND M.K.WAHI                OCT. 21 1977
C
      DIMENSION X(1),R(1),P(6)
C
      R(1)=X(1)-X(2)*(1.-P(5))
      R(2)=P(4)-X(1)*X(3)-X(2)*X(4)
      R(3)=P(2)-X(1)*SIN(X(3))-X(2)*SIN(X(4))
      R(4)=P(3)-X(1)*(1.-COS(X(3)))+X(2)*(1.-COS(X(4)))
C
      RETURN
      END
```

Table 177: LISTING FOR SUBROUTINE ICFSB

```
CICFSB
      SUBROUTINE ICFSB(X,M,K,R,P)
C
C    PURPOSE - EVALUATE GEOMETRY EQUATIONS FOR FREE SHAPE AIR BAG
C            - ELEMENT
C
C    METHOD  - EQUATIONS DESCRIBE INELASTIC MEMBRANE SUBJECT TO
C            - UNIFORM INTERNAL PRESSURE
C
C    LIMITATIONS - USED ONLY BY ROUTINE QNWT IN SUBROUTINE ICB
C
C    ***OUTPUTS***
C    X     THE SOLUTION ESTIMATE FROM THE LAST ITERATION
C    R     THE RESIDUAL VECTOR
C    ***INPUTS***
C    X     INITIAL ESTIMATE VECTOR
C    P(1)  PRINT CONTROL OPTION
C          P=0   NO PRINT OUT
C          P=K   FOR PRINT OUT AT EVERY K-TH ITERATION
C    P(2)...P(N)   STORAGE AVAILABLE FOR PASSING DATA
C                  TO SUBROUTINE FUN(ICFS) BY QNWT.
C    M,K   DIMENSION INDICATORS FOR X AND AJ MATRICES
C
C    WRITTEN BY  J.R.KILNER             APRIL 10,1978
C
      DIMENSION X(1),R(1),P(1)
C
      R(1)=P(4)-X(1)*(X(2)+X(3))
      R(2)=P(2)-X(1)*(SIN(X(2))+SIN(X(3)))
      R(3)=P(3)-X(1)*(COS(X(3))-COS(X(2)))
C
      RETURN
      END
```

Table 178: LISTING FOR SUBROUTINE ICLS

```
CICLS
      SUBROUTINE ICLS(X,M,K,R,P)
C    PURPOSE - DEFINE INITIAL CONDITION LOADED SHAPES DIGGES MODEL
C            - FOR ACLS TRUNKS
C
C    VERSION 3.            FEB.20 1978
C
C    MEHTOD  - SEE AFFDL-TR-71-50, THEORY OF AN ACLS FOR AIRCRAFT
C            - BY KENNERLY H. DIGGES, JUNE 1971.
C
C    LIMITATIONS - USED ONLY BY ROUTINE QNWT IN SUBROUTINE IC
C
C     FOR INPUTS AND OUTPUTS  SEE SUBROUTINE ICFS
C    WRITTEN BY  - J.R.KILNER AND M.K.WAHI              OCT. 21 1977
C
      DIMENSION X(1),R(1),P(6)
C
      R(1)=X(1)-X(2)*(1.-P(5))
      R(2)=P(4)-X(1)*X(3)-X(2)*X(4)-X(5)
      R(3)=P(2)-X(1)*SIN(X(3))-X(2)*SIN(X(4))-X(5)
      R(4)=P(3)-X(1)*(1.-COS(X(3)))+X(2)*(1.-COS(X(4)))
      R(5)=P(6)-X(2)*(1.-COS(X(4)))
C
      RETURN
      END
```

Table 179: LISTING FOR SUBROUTINE ICLSB

```
CICLSB
      SUBROUTINE ICLSB(X,M,K,R,P)
C
C    PURPOSE - EVALUATE ELEMENT GEOMETRY AND FORCE BALANCE EQUATIONS
C            - FOR LOADED SHAPE AIR BAG ELEMENT
C
C    METHOD  - EQUATIONS DESCRIBE INELASTIC MEMBRANE SUBJECT
C            - TO UNIFORM INTERNAL PRESSURE AND GROUND REACTION
C
C    LIMITATIONS - USED ONLY BY ROUTINE QNWT IN SUBROUTINE ICB
C
C     FOR INPUTS AND OUTPUTS  SEE SUBROUTINE ICFS
C    WRITTEN BY  - J.R.KILNER                      APRIL 10,1978
C
      DIMENSION X(1),R(1),P(1)
C
      DO 10 I=1,2
      P(I+14)=X(I+2)-X(I)*(3.1416+P(I+6))
      IF(P(I+14).LT.0.)P(I+14)=0.
      P(I+12)=3.1416+P(I+6)
10    IF(P(I+14).LE.0.)P(I+12)=X(I+2)/X(I)
C
      R(1)=X(1)-X(2)-P(5)*X(5)
      R(2)=P(4)-X(3)-X(4)-X(5)
      R(3)=P(2)-X(1)*SIN(P(13))-X(2)*SIN(P(14))-X(5)
     *    +P(15)*P(10)+P(16)*P(12)
      R(4)=P(3)-X(1)*(1.-COS(P(13)))+X(2)*(1.-COS(P(14)))
     *    +P(15)*P(9)-P(16)*P(11)
      R(5)=P(6)-X(2)*(1.-COS(P(14)))+P(16)*P(11)
C
      RETURN
      END
```

Table 180: LISTING FOR SUBROUTINE IT

```
CIT
      SUBROUTINE IT(FO,FODOT,IFO,FIN,GKI,GKL,AMA,AMI)
C  VERSION 2.                          REVISED: OCT 8 1976
C
C   PURPOSE - SIMULATION OF AN INTEGRATOR WITH SATURATION
C
C
C   METHOD  - SEE CODING
C
C
C   LIMITATIONS - EXCESSIVELY HIGH VALUES OF GKL MAY RESULT IN POOR
C                 STEADY STATE CONVERGENCE
C
C
C   WRITTEN BY  - ADAM LLOYD                     LATEST REVISION - NOV 75
C
C
C   INPUT/OUTPUT LIST
C
C   FO          INTEGRATOR OUTPUT                ANY            OUTPUT STATE
C   FODOT       OUTPUT DERIVATIVE                ANY            OUTPUT DERIV
C   IFO         INTEGRATOR CONTROL               ---            PROGRAM VAR
C   FIN         FUNCTION INPUT                   ANY            INPUT   VAR
C   GKI         INTEGRATOR GAIN                  ANY            INPUT   PARAM
C   GKL         DERIVATIVE LIMITER GAIN          ANY            INPUT   PARAM
C   AMA         UPPER LIMIT OF OUTPUT            ANY            INPUT   PARAM
C               WHERE DERIV. LIMITER STARTS
C   AMI         LOWER LIMIT OF OUTPUT            ANY            INPUT   PARAM
C               WHERE DERIV. LIMITER STARTS
      EPS=FIN
C  ------------ PROVIDE DEFAULTS THAT ELLIMINATE SATURATION
      IF(AMA.EQ..99999)AMA=1.E36
      IF(AMI.EQ..99999)AMI=-1.E36
      IF(FO.GT.AMA)EPS = FIN - GKL*(FO-AMA)
      IF(FO.LT.AMI)EPS = FIN - GKL*(FO-AMI)
      IF(IFO.NE.0)FODOT=GKI*EPS
      RETURN
      END
```

Table 181: LISTING FOR SUBROUTINE KINK

```
CKINK
      SUBROUTINE KINK(VO,CSI,EC,DNC,PKW,C,X)
C
C    PURPOSE - DETERMINE ARRESTING CABLE KINK WAVE ANGLE
C
C    MEHTOD  - SEE ARRESTING SYSTEM DOCUMENTATION,VOL I, FINAL REPORT
C
C    LIMITATIONS - CALLED ONLY BY THE ARRESTING SYSTEM COMPONENT AS
C
C    *** CALL SEQUENCE ***
C    *** INPUTS ***
C
C    VO          - VEHICLE VELOCITY , FT/SEC
C    CSI         - INITIAL CABLE STRESS, LBS/SQ IN
C    EC          - CABLE MODULUS OF ELASTICITY, LBS/SQ IN
C    DNC         - CABLE WEIGHT DENSITY, LBS/CU IN
C
C    *** OUTPUT ***
C
C    PKW         - KINK WAVE ANGLE, RAD
C    C           - CABLE SONIC VELOCITY, FT/SEC
C    CSTR        - STRAIN IN WAKE OF TRANSVERSE WAVE, IN/IN
C
C       WRITTEN BY  -  J.R.KILNER                  MAY 1978
C
      DIMENSION X(1),P(3),IP(2),R(1),AJ(1,1),BB(1,3)
      COMMON/CIO/IREAD,IWRITE,IDIAG
      EXTERNAL RES
C
      C=SQRT(386.*EC/DNC)/12.
      VC=VO/C
      P(2)=VC*VC
      P(3)=CSI/EC
      X(1)=P(3)
      P(1)=0.
      IP(1)=50
C
      CALL QNWT(X,1,1,RES,P,.0001,IP,0,R,RMS,AJ,BB)
C
      SINPKW=VC/SQRT(X(1)*(1.+X(1)))
      PKW=ASIN(SINPKW)
C
      RETURN
      END
```

Table 182: LISTING FOR SUBROUTINE LA

```
CLA
      SUBROUTINE LA(FO,FODOT,IFO,FIN,GAI,TC)
C
C    PURPOSE - TO SIMULATE FIRST ORDER LAG        FO        GAI
C                                                ---- = ----------
C                                                FIN     (1.+TC*S)
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  -   ADAM LLOYD          LATEST REVISION      NOV 75
C
C
C    LIMITATIONS -   TIME CONSTANT TC SHOULD NOT EQUAL ZERO
C
C
C    INPUT/OUTPUT LIST
C
C    FO          TRANSFER FUNCTION OUTPUT              ANY       OUTPUT STATE
C    FODOT       TRANSFER FUNCTION OUTPUT DERIV.       ANY       OUTPUT STATE
C    IFO         INTEGERATOR CONTROL                   ---       PROGRAM VAR
C    FIN         TRANSFER FUNCTION INPUT               ANY       INPUT  VAR
C    GAI         TRANSFER FUNCTION GAIN                ---       INPUT  PARAM
C    TC          TIME CONSTANT                         SECS      INPUT  PARAM
      COMMON/CIO/IREAD,IWRITE,IDIAG
      IF(IFO.NE.0) FODOT=(GAI*FIN-FO)/TC
      RETURN
      END
```

Table 183: LISTING FOR SUBROUTINE LE

```
CLE
      SUBROUTINE LE(X1,X1D,IX1,FO,FIN,GAI,ZO,PO)
C  VERSION 2.                          REVISED: SEPT 17 1976
C  PURPOSE:  PROVIDE FIRST ORDER LEAD-LAG TRANSFER FUNCTION
C                  FO         GAI( S + ZO )
C                 ----   =   ---------------
C                  FIN          S  +  PO
C  CALL SEQUENCE:
C       X1,X1D,IX1   -  STATE,RATE,INT CONTROL
C       FO           -  OUTPUT
C       FIN          -  INPUT
C       GAI          -  HIGH FREQUENCY GAIN
C       ZO           -  ZERO LOCATION
C       PO           -  POLE LOCATION
C  DESIGNED BY:  J.D. BURROUGHS              MAY 1976
      GFIN=GAI*FIN
      FO=X1+GFIN
      IF(IX1.NE.0)X1D=GFIN*ZO-FO*PO
      RETURN
      END
```

Table 184: LISTING FOR SUBROUTINE LG

```
CLG
      SUBROUTINE LG(FO,FOD,IFO,FIN,ZO,PO)
C  VERSION 2.                    REVISED: SEPT 17 1976
C  PURPOSE:  PROVIDE FIRST ORDER LAG TRANSFER FUNCTION
C                   FO              ZO
C                  -----   =   ----------
C                   FIN          S   +  PO
C  CALL SEQUENCE:
C       FO,FOD,IFO - OUTPUT STATE,RATE,INT CONTROL
C       FIN        - INPUT
C       ZO         - NUMERATOR COEFFICIENT
C       PO         - DENOMINATOR COEFFICIENT
C  DESIGNED BY: J.D. BURROUGHS                  MAY 1976
      IF(IFO.NE.0)FOD=ZO*FIN-PO*FO
      RETURN
      END
```

Table 185: LISTING FOR SUBROUTINE LL

```
CLL
      SUBROUTINE LL(X1,X1DOT,IX1,FO,FIN,TC1,TC2,GAI)
C
C    PURPOSE - TO SIMULATE LEAD LAG TRANSFER FUNCTION
C
C                            FO      GAI*(1.+TC1*S)
C                           ----- = ---------------
C                            FIN       (1.+TC2*S)
C
C
C    METHOD  - SELF EXPLANATORY
C
C
C    LIMITATIONS -   NONE
C
C
C    WRITTEN BY  -   ADAM LLOYD          LATEST REVISION  NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    X1          STATE VARIABLE                       ANY        OUTPUT STATE
C    X1DOT       STATE VARIABLE DERIVATIVE            ANY        OUTPUT STATE
C    IX1         INTEGRATOR CONTROL                   ---        PROGRAM VAR
C    FO          TRANSFER FUNCTION OUTPUT             ANY        OUTPUT VAR
C    FIN         TRANSFER FUNCTION INPUT              ANY        INPUT  VAR
C    TC1         TIME CONSTANT (NUMERATOR)            SECS       INPUT  PARAM
C    TC2         TIME CONSTANT (DENOMINATOR)          SECS       INPUT  PARAM
C    GAI         TRANSFER FUNCTION GAIN               ---        INPUT  PARAM
      COMMON/CIO/IREAD,IWRITE,IDIAG
      FO=(X1+FIN*TC1*GAI)/TC2
      IF(IX1.NE.0)X1DOT= GAI*FIN-FO
      RETURN
      END
```

Table 186: LISTING FOR SUBROUTINE MA

```
CMA
      SUBROUTINE MA(FO,FIN,C1,C2)
C
C    PURPOSE - TO SIMULATE THE EQUATION    OUTPUT=C1*INPUT + C2
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  -  ADAM LLOYD           LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - NONE
C
C
C    INPUT/OUTPUT LIST
C
C    FO         OUTPUT VARIABLE           ANY            OUTPUT VAR
C    FIN        INPUT  VARIABLE           ANY            INPUT  VAR
C    C1         CONSTANT MULTIPLIER       ---            INPUT  PARAM
C    C2         CONSTANT ADDITION         ---            INPUT  PARAM
      FO=C1*FIN + C2
      RETURN
      END
```

Table 187: LISTING FOR SUBROUTINE MB

```
CMB
      SUBROUTINE MB(FO,FNA,FNB,C1,C2,C3,C4,C5)
C
C   PURPOSE - TO SIMULATE THE EQUATION   Y=C1*XA+C2*XB+C3*XA*XB+C4*XA/XB+C
C
C
C   WRITTEN BY - GEORGE DULEBA                    LATEST REVISION   MAY 76
C
C
C   LIMITATIONS - IF FNB=0 DURING DIVISION, FNB IS SET TO E-20.
C                 DIAGNOSTIC MESSAGE IS GIVEN.
C
C
C   INPUT/OUTPUT LIST
C
C   FO         OUTPUT VARIABLE                          ANY          OUTPUT
C   FNA        INPUT VARIABLE A                         ANY          INPUT
C   FNB        INPUT VARIABLE B                         ANY          INPUT
C   C1         MULTIPLIER 1                             ANY          INPUT
C   C2         MULTIPLIER 2                             ANY          INPUT
C   C3         MULTIPLIER 3                             ANY          INPUT
C   C4         MULTIPLIER 4                             ANY          INPUT
C   C5         ADDITIVE VARIABLE                        ANY          INPUT
C
C
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      FO= C1*FNA + C2*FNB + C3*FNA*FNB + C5
      IF(C4.EQ.0.99999) GO TO 30
      IF(FNB.EQ.0.) GO TO 10
      FO= FO + C4*FNA/FNB
      RETURN
 10   WRITE(IWRITE,20)
 20   FORMAT(/,30X, 53HWARNING- DIVISOR IN MB EQUALS 0., HAS BEEN SET=1.
     2E-20)
      FO= FO + C4*FNA*1.E+20
 30   RETURN
      END
```

Table 188: LISTING FOR SUBROUTINE MC

```
CMC
      SUBROUTINE MC(FO,FIN,FIO,FIP,C1,C2,C3,C4)
C
C   PURPOSE - TO SIMULATE THE EQUATION  FO=C1*FIN+C2*FIO+C3*FIP+C4
C
C
C   METHOD  - SEE CODING
C
C
C   WRITTEN BY  -  ADAM LLOYD          LATEST REVISION   NOV 75
C
C
C   LIMITATIONS - NONE
C
C
C   INPUT/OUTPUT LIST
C
C   FO          OUTPUT VARIABLE              ANY           OUTPUT VAR
C   FIN         INPUT  VARIABLE              ANY           INPUT  VAR
C   FIO         INPUT  VARIABLE              ANY           INPUT  VAR
C   FIP         INPUT  VARIABLE              ANY           INPUT  VAR
C   C1          CONSTANT MULTIPLIER          ---           INPUT  PARAM
C   C2          CONSTANT MULTIPLIER          ---           INPUT  PARAM
C   C3          CONSTANT MULTIPLIER          ---           INPUT  PARAM
C   C4          CONSTANT ADDITION            ---           INPUT  PARAM
      FO=C1*FIN+C2*FIO+C3*FIP+C4
      RETURN
      END
```

Table 189: LISTING FOR SUBROUTINE MG

```
CMG
      SUBROUTINE MG(T3,W3,P,PDOT,IP,T1,W1,T2,W2,P3,AK,D3,DHY,
     1 AHT,TAM,HO,VOL,FC)
C
C    PURPOSE - ANALYSIS OF FLOW MERGE, WITH TWO INLET PORTS AND ONE
C              OUTLET PORT
C
C    METHOD  - SIMILAR TO DUCT MODEL DE
C
C
C    LIMITATIONS - AS FOR DE
C
C
C    WRITTEN BY  ADAM LLOYD AS COMPONENT *ME* IN
C          ECS LIBRARY        NOV. 1975
C
C    MODIFIED BY  - MAHINDER WAHI                         AUGUST  1977
C
C
C    INPUT/OUTPUT LIST
C
C    T3          OUTLET TEMPERATURE          )PORT    DEGR       OUTPUT VAR
C    W3          OUTLET FLOW                 )NO 3    LB/MIN     OUTPUT VAR
C    P           INTERNAL PRESSURE                    PSIA       OUTPUT STATE
C    PDOT        INTERNAL PRESSURE DERIVATIVE         PSIA/SEC   OUTPUT DERIV
C    IP          INTEGRATOR CONTROL                   ---        PROGRAM VAR
C    T1          INLET TEMPERATURE           )PORT    DEGR       INPUT  VAR
C    W1          INLET FLOW                  ) NO 1   LB/MIN     INPUT  VAR
C    T2          INLET TEMPERATURE           )PORT    DEGR       INPUT  VAR
C    W2          INLET FLOW                  ) NO 2   LB/MIN     INPUT  VAR
C    P3          OUTLET PRESSURE(PORT NO 3)           PSIA       INPUT  VAR
C    AK          K FACTOR ) FOR PRESSURE DROP         ---        INPUT  PARAM
C    D3          DIAMETER ) CALCULATION               IN         INPUT  PARAM
C    DHY         HYDRAULIC DIAMETER ) TO CALCULATE    IN         INPUT  PARAM
C    AHT         HEAT TRANSFER AREA ) UA              FT2        INPUT  PARAM
C    TAM         EFFECTIVE LOCAL AMBIENT TEMP         DEGR       INPUT  PARAM
C    HO          EXTERNAL HEAT TRANSFER COEFFICIENT BTU/FT2      INPUT  PARAM
C                (BASED ON INTERNAL WETTED AREA)      HR DEGR
C    VOL         INTERNAL VOLUME                      FT3        INPUT  PARAM
C    FC          FREQUENCY CONTROL ON P1.(FC.GE.1.) ---          INPUT  PARAM
C                A VALUE OF FC GREATER THAN 1.
C                DECREASES FREQUENCY RESPONSE OF P1
C                CORRESPONDINGLY
C
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
C    CALCULATE GAS CONSTANT AND RATIO OF SPECIFIC HEATS, BASED ON AVERAGE
C    INLET TEMPERATURE
      TINB=(T1*ABS(W1)+T2*ABS(W2))/(ABS(W1)+ABS(W2))
      TINB=AMAX1(AMIN1(TINB,1600.),300.)
      CP=SHCP(TINB,0.)
      R=53.3
      GAMMA=1.+R/(778.*CP-R)
      G1=1./(GAMMA-1.)
      G2=(GAMMA-1.)/2.
C    CALCULATE OUTLET FLOW BASED ON AVERAGE INLET TEMPERATURE
```

Table 189: LISTING FOR SUBROUTINE MG (CONCLUDED)

```
      CA=.785398*D3*D3
      CALL FNFLOW(P,P3,TINB,CA,AK,FN,W3)
C    CALCULATE OVERALL CONDUCTANCE (UA) BASED ON AVERAGE FLOW
      WBAR=(ABS(W1)+ABS(W2)+ABS(W3))/3.
      WBAR=AMAX1(WBAR,.01)
C    CALCULATE EFFECTIVE LENGTH AL FOR  HEAT TRANS COEFF CALCULATION
      AL=183.35*AHT/(DHY*DHY)
C    CONSTANT 183.35=144./(PI/4.)        AL IS IN FEET.
      HINT=HI(1,TINB,TINB,WBAR,0.,DHY,AL,0.)
C    THE INPUT VALUE OF THE EXTERNAL HEAT TRANSFER COEFFICIENT IS BASED
C    ON THE INTERNAL WETTED AREA. HENCE UA IS GIVEN BY
      UA=AHT*HINT*HO/(60.*(HINT+HO))
C    UA IS IN BTU/MIN DEGR
C    CALCULATE EXIT TEMPERATURE T3
      T3=TAM+(TINB-TAM)/EXP(UA/(CP*WBAR))
C    WARNING DIAGNOSTIC IF TEMPERATURE CHANGE EXCEEDS 300 DEGR
      IF(ABS(TINB-T3).LE.300.)GO TO 10
      IF(TINB.GT.T3)T3=TINB-300.
      IF(T3.GT.TINB)T3=TINB+300.
C    TEST FOR DIAGNOSTIC PRINT OUT
      IF(IERR.NE.1)GO TO 10
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,46HNON FATAL ERROR CALLED FROM MERGE COMPONENT MG/
     1 10X,35HTEMPERATURE CHANGE EXCEEDS 300 DEGR)
 10   TBAR=(T1+T2+T3)/3.
      PBAR=(2.0*P+P3)/3.
      AM=AMACH(PBAR,TBAR,CA,WBAR,0.)
      IF(IP.NE.0)PDOT=R*TBAR*(W1+W2-W3)*(1.+G2*AM*AM)**G1/(8640.*VOL*FC)
      RETURN
      END
```

Table 190: LISTING FOR SUBROUTINE OC

```
COC
      SUBROUTINE OC(G,S,AK,FK,X,XDOT,INT,Y,YOP,U,UOP,N,NY,NU)
C   VERSION 1.                                REVISED: MAY 8 1975
C   PURPOSE:  OPTIMAL CONTROLLER COMPONENT
C   CALL SEQUENCE:  G     - OUTPUT MATRIX
C                   S     - INPUT MATRIX
C                   AK    - STABILITY MATRIX
C                   FK    - D.C. GAIN MATRIX
C                   X     - STATE VECTOR
C                   XDOT  - RATE VECTOR
C                   INT   - INTEGRATOR CONTROL VECTOR
C                   Y     - INPUT VECTOR
C                  YOP     - INPUT OPERATING POINT OFFSET VECTOR
C                   U     - OUTPUT VECTOR
C                  UOP     - OUTPUT OPERATING POINT OFFSET VECTOR
C                   N     - OPTIMAL CONTROLLER ORDER
C                   NY    - NUMBER OF INPUTS
C                   NU    - NUMBER OF OUTPUTS
C    DESIGNED BY: J.D.BURROUGHS                          APRIL 1975
      DIMENSION G(NU,N),S(N,NY),AK(N,N),FK(NU,NY),X(N),XDOT(N)
     1 ,INT(N),Y(NY),U(NU),YOP(NY),UOP(NU)
C --->        CALCULATE O.C. RATES        XDOT = S * Y + AK * X
C --->              TEST FOR ZERO ORDER CONTROLLER
      IF(N.LE.0)GO TO 200
      DO 160 I=1,N
C --->        BYPASS CALCULATION FOR FROZEN STATES
      IF(INT(I).EQ.0)GO TO 160
      SUM=0.
      DO 100 J=1,NY
100   SUM=SUM+S(I,J)*(Y(J)-YOP(J))
      DO 120 J=1,N
120   SUM=SUM+AK(I,J)*X(J)
      XDOT(I)=SUM
160   CONTINUE
C --->        CALCULATE OUTPUTS         U = FK * Y + G * X
200   DO 260 I=1,NU
      SUM=UOP(I)
      DO 220 J=1,NY
220   SUM=SUM+FK(I,J)*(Y(J)-YOP(J))
C --->        TEST FOR ZERO ORDER CONTROLLER
      IF(N.LE.0)GO TO 250
      DO 240 J=1,N
240   SUM=SUM+G(I,J)*X(J)
250   U(I)=SUM
260   CONTINUE
      RETURN
      END
```

Table 191: LISTING FOR SUBROUTINE OL

```
COL
      SUBROUTINE OL(FX,FZ,TY,UD,WD,AMAS,XCP,CXO,CXA,CXU,CXDE,
     1 CXTR,CXSP,CXGE,KCXB,CZO,CZA,CZAD,CZQ,CZU,CZDE,CZTR,CZSP,
     2 CZGE,KCZB,CZDS,CMO,CMA,CMAD,CMQ,CMU,CMDE,CMTR,CMSP,CMGE,
     3 KCMB,CMDS,CMB,KGE,AMASS,C,XAC,DIM,CAS,SAS,SWI,FXIN,FZIN,
     4 TYIN,ELEV,STAB,SPO,AL,ALP,U,UP,WP,VBAR,QBAR,Q,QW,EU,EW)
      REAL KCXB,KCZB,KCMB,KGE
C
C    COMMENT:WIND TUNNEL DATA MAY ALSO BE USED,IN
C            WHICH CASE CXA,CZA,AND CMA ARE SET EQUAL
C            TO ZERO.CXO,CZO,AND CMO VALUES ARE INTER
C            POLATED FROM TABLES.
C
C    VERSION 2.                                    AUG.18 1977
C    PURPOSE   COMPUTE LONGITUDINAL FORCES AND MOMENTS
C    METHOD    USE LINEAR DERIVATIVES TO COMPUTE LIFT AND DRAG FORCES.
C              SOLVE IMPLICIT EQUATIONS FOR BODY AXIS FORCES AND ACCELE-
C              RATIONS.COMPUTE PITCHING MOMENT INCLUDING C.P. TORQUE.
C    CALL SEQUENCE
C    ******  OUTPUTS  ******
C      FX,FZ            -X AND Z BODY AXIS FORCE SUM, LBS
C      TY               -Y BODY AXIS (PITCHING) MOMENT, FT-LBS
C      UD,WD            -X AND Z BODY AXIS ACCELERATIONS, FT/SEC**2
C      AMAS             -RIGID BODY MASS, SLUGS
C      XCP              -X AXIS DISTANCE C.P. - C.G. , FT
C    ******  INPUTS  ******
C                   AERO-DERIVATIVES -- UNITS FOR DIMENSIONAL CASE
C    DRAG FORCE COEFFICIENTS
C      CXO              -BIAS COEFFICIENT FOR TRIM, LBS
C      CXA              -ALPHA COEFFICIENT (NONDIM.)
C                       -Z AXIS VELOCITY COEFFICIENT (DIM.), LB-SEC/FT
C      CXU,CXDE         -X AXIS VELOCITY, ELEVATOR COEFFICIENT,
C                       - LB-SEC/FT, LB/DEG
C      CXTR             - TAKEOFF OR RECOVERY TRUNK COEFFICIENT
C      CXSP             - FLIGHT PLUS GROUND SPOILER COEFFICIENT, LB/DEG
C      CXGE             - GROUND EFFECT FACTOR ON CXO
C      KCXB             - LARGE SIDE SLIP ANGLE FACTOR FOR CXO
C    LIFT FORCE COEFFICIENTS
C      CZO              -BIAS COEFFICIENT FOR TRIM, LBS
C      CZA,CZAD         -ALPHA AND ALPHA DOT COEFFICIENTS (NONDIM.)
C                       -Z AXIS VELOCITY AND ACCEL. COEFFICIENTS (DIM.),
C                       - LB-SEC/FT, LB-SEC**2/FT
C      CZQ,CZU,CZDE     - Q, X AXIS VELOCITY, AND ELEVATOR COEFFICIENTS,
C                       - LB-SEC/DEG, LB-SEC/FT, LB/DEG
C      CZTR             - TAKEOFF OR RECOVERY TRUNK COEFFICIENT
C      CZSP             - FLIGHT PLUS GROUND SPOILER COEFFICIENT, LB/DEG
C      CZGE             - GROUND EFFECT FACTOR ON CZO
C      KCZB             - LARGE SIDE SLIP ANGLE FACTOR FOR CZO
C      CZDS             - STABILIZER COEFFICIENT
C
C    PITCHING MOMENT COEFFICIENTS
C      CMO              -BIAS COEFFICIENT FOR TRIM, FT-LBS
C      CMA,CMAD         -ALPHA AND ALPHA DOT COEFFICIENTS (NONDIM.)
C                       -Z AXIS VELOCITY AND ACCEL. COEFFICIENTS (DIM.)
C                       - LB-SEC , LB-SEC**2
C      CMQ,CMU,CMDE     - Q, X AXIS VELOCITY, AND ELEVATOR COEFFICIENTS
```

```
C                          - FT-LB-SEC/DEG, LB-SEC, FT-LB/DEG
C     CMTR                 -TAKEOFF OR RECOVERY TRUNK COEFFICIENT
C     CMSP                 -FLIGHT PLUS GROUND SPOILER COEFFICIENT, LB/DEG
C     CMGE                 -GROUND EFFECT FACTOR ON CMO
C     KCMB                 -LARGE SIDE SLIP ANGLE FACTOR FOR PITCHING MOMENT
C     CMDS                 -STABILIZER COEFFICIENT
C     CMB                  -LARGE SIDE SLIP ANGLE COEFFICIENT
C     KGE                  -GROUND EFFECT HEIGHT FACTOR
C
C   CONSTANTS
C     AMASS                -RIGID BODY MASS, SLUGS
C     C                    -MEAN AERODYNAMIC CHORD, FT
C     XAC                  -X AXIS DISTANCE C.P.-C.G., FT
C     DIM                  -INDICATOR FUNCTION FOR AERO-COEFFICIENTS
C                                0=   BODY AXIS, DIM.
C                                1=   BODY AXIS, NONDIM.
C                                2=   STABILITY AXIS, DIM.
C                                3=   STABILITY AXIS, NONDIM.
C     CAS,SAS              -DIRECTION COSINES FOR STABILITY AXES OR
C                          -BODY AXES DEPENDING ON DIM
C     SWI                  -INDICATOR FUNCTION FOR DEGREES OF FREEDOM
C                                1=  SINGLE DOF; U=X BODY AXIS VELOCITY
C                                2=  TWO DOF; W=Z BODY AXIS VELOCITY AND
C                                    Q=Y BODY AXIS ANGULAR RATE
C                                3=  THREE DOF; U,W,Q;FULL MODEL
C
C   EXTERNAL FORCES AND MOMENTS
C     FXIN,FZIN            -X AND Z BODY AXIS FORCES, LBS
C     TYIN                 -Y BODY AXIS (PITCHING) MOMENT, FT-LBS
C   AERO-VARIABLES
C     ELEV                 -ELEVATOR DEFLECTION, DEG
C     STAB                 -STABILIZER DEFLECTION, DEG
C     SPO                  -SPOILER DEFLECTION, DEG
C     AL, ALP              -ANGLE OF ATTACK IN BODY AND STABILITY AXES, DEG
C     U                    -X BODY AXIS VELOCITY, FT/SEC
C     UP,WP                -X AND Z PERTURBATION VELOCITIES (NONDIM.)
C                          -X AND Z STABILITY AXIS VELOCITIES (DIM.), FT/SEC
C     VBAR                 -TRUE AIRSPEED, FT/SEC
C     QBAR                 -DYNAMIC PRESSURE TIMES REFERENCE AREA, LBS
C     Q                    -Y BODY AXIS ANGULAR RATE, DEG/SEC
C     QW                   -Y BODY AXIS ANGULAR RATE GUST, DEG/SEC
C     EU,EW                -X AND Z BODY AXIS ACCEL. TERMS FOR EXPLICIT
C                          -UDOT AND WDOT SOLUTIONS, FT/SEC**2
C
C   WRITTEN BY A.W. WARREN  AS COMPONENT *LO* IN
C             FLT.CONTROLS LIBRARY      SEPT 1976
C
C   MODIFIED BY  MAHINDER WAHI                                  MAY 1977
C
C                          INITIALIZATION
      DATA IND, RPD /0, .01745329/
      IF(IND.EQ.1) GO TO 10
      IF(CXO .EQ. .99999) CXO =0.
      IF(CXA .EQ. .99999) CXA =0.
      IF(CXU .EQ. .99999) CXU =0.
      IF(CXDE.EQ. .99999) CXDE=0.
```

```
      IF(CXTR.EQ. .99999) CXTR=0.
      IF(CXSP.EQ. .99999) CXSP=0.
      IF(CXGE.EQ. .99999) CXGE=0.
      IF(KCXB.EQ. .99999) KCXB=1.
C
      IF(CZO .EQ. .99999) CZO =0.
      IF(CZA .EQ. .99999) CZA =0.
      IF(CZAD.EQ. .99999) CZAD=0.
      IF(CZQ .EQ. .99999) CZQ =0.
      IF(CZU .EQ. .99999) CZU =0.
      IF(CZDE.EQ. .99999) CZDE=0.
      IF(CZTR.EQ. .99999) CZTR=0.
      IF(CZSP.EQ. .99999) CZSP=0.
      IF(CZGE.EQ. .99999) CZGE=0.
      IF(CZDS.EQ. .99999) CZDS=0.
      IF(KCZB.EQ. .99999) KCZB=1.
C
      IF(CMO .EQ. .99999) CMO =0.
      IF(CMA .EQ. .99999) CMA =0.
      IF(CMAD.EQ. .99999) CMAD=0.
      IF(CMQ .EQ. .99999) CMQ =0.
      IF(CMU .EQ. .99999) CMU =0.
      IF(CMDE.EQ. .99999) CMDE=0.
      IF(CMTR.EQ. .99999) CMTR=0.
      IF(CMSP.EQ. .99999) CMSP=0.
      IF(CMGE.EQ. .99999) CMGE=0.
      IF(CMDS.EQ. .99999) CMDS=0.
      IF(CMB .EQ. .99999) CMB =0.
      IF(KCMB.EQ. .99999) KCMB=1.
C
      IF(KGE .EQ. .99999) KGE =0.
      IF(XAC .EQ. .99999) XAC =0.
      IF(ELEV.EQ. .99999) ELEV=0.
      IF(STAB.EQ. .99999) STAB=0.
      IF(SPO .EQ. .99999) SPO =0.
      IND=1
   10 XCP= XAC
      AMAS = AMASS
      IF(SWI.EQ.1.) GO TO 80
      IF(SWI.EQ.2.) GO TO 90
      IF(SWI.EQ.3.) GO TO 100
   80 WP=EW=Q=QW=CMQ=CZQ=CXA=CZA=CZAD=CMA=CMAD=0.
      GO TO 100
   90 CXU=CZU=CMU=UP=EU=0.
  100 CONTINUE
      IF(DIM.EQ.1. .OR. DIM.EQ.3.) GO TO 20
C
C                       DIMENSIONAL FORCE AND MOMENT SOLUTION
C
      WWDOT= QW*VBAR*RPD
C
C                 EXPLICIT SOLUTION FOR AERO-FORCES USING ASOL,BSOL
      ASOL= 1.- CZAD/AMASS
      BSOL= EW- SAS*EU- WWDOT
      FX=(CXO + CXA*WP + CXU*UP + CXDE*ELEV + CXTR
     1                            + CXSP*SPO  + CXGE*KGE)*KCXB
```

```
      FZ= ( CZO +CZA*WP +CZAD*BSOL +CZQ*Q +CZU*UP +CZDE*ELEV
     1          +CZTR +CZSP*SPO + CZGE*KGE +CZDS*STAB)*KCZB
      FZAERO= FZ/ASOL
C             COMPUTE BODY AXIS FORCE SUMS AND ACCELERATIONS
C
      IF(DIM.EQ.0.) GO TO 30
C             STABILITY AXIS SUMS AND TRANSFORMATION
C
      FXS= FX + FZIN*SAS + FXIN*CAS
      FZS= (FZ + FZIN*CAS - FXIN*SAS)/ASOL
      FZAERO= FZAERO*CAS + FX*SAS
      FX=FXS*CAS - FZS*SAS
      FZ=FZS*CAS + FXS*SAS
      GO TO 40
C                          BODY AXIS SUMS
   30 FX= FX + FXIN
      FZ= (FZ + FZIN)/ASOL
   40 UD= FX/AMASS + EU
      WD= FZ/AMASS + EW
C                          PITCHING MOMENT COMPUTATIONS
C
      WDP= WD - SAS*UD - WWDOT
      TY= ( CMO +CMA*WP +CMAD*WDP +CMQ*Q +CMU*UP +CMDE*ELEV
     1          +CMTR +CMSP*SPO +CMGE*KGE + CMDS*STAB +CMB)*KCMB
   70 TY= TY + TYIN - XAC*FZAERO
      RETURN
C
C              NONDIMENSIONAL FORCE AND MOMENT SOLUTION
C
   20 CDIM = C/(VBAR+VBAR)
      AL1= AL*RPD
      ALP1= ALP*RPD
      Q1= Q*RPD
      ELEV1= ELEV*RPD
      SPO1= SPO*RPD
      STAB1= STAB*RPD
C          EXPLICIT SOLUTION FOR AERO-FORCES USING ASOL,BSOL
      DIV= 1./(AMASS*U)
      ASOL= 1.- QBAR*CDIM*CZAD*DIV
      FX= QBAR*( CXO +CXA*ALP1 +CXU*UP +CXDE*ELEV1 +CXTR
     1             +CXSP*SPO1 +CXGE*KGE)*KCXB
      FXS= FX +FXIN*CAS +FZIN*SAS
      BSOL= FXS*(SAS-AL1)*DIV + (EW-AL1*EU)/U - QW*RPD
      FZ= QBAR*( CZO +CZA*ALP1 +CDIM*(CZAD*BSOL +CZQ*Q1)
     1          +CZU*UP +CZDE*ELEV1 + CZTR +CZSP*SPO1
     2                + CZGE*KGE + CZDS*STAB1)*KCZB
      FZAERO= FZ/ASOL
      IF(DIM.EQ.1.) GO TO 50
C                          STABILITY AXIS SUMS AND TRANSFORMATION
C
      FZAERO= FZAERO*CAS + FX*SAS
      FZS= (FZ +FZIN*CAS -FXIN*SAS)/ASOL
 6000 FORMAT(1H0,F12.5,F12.5)
      FX= FXS*CAS - FZS*SAS
      FZ= FZS*CAS + FXS*SAS
      GO TO 60
```

Table 191: LISTING FOR SUBROUTINE OL (CONCLUDED)

```
C                          BODY AXIS SUMS AND ACCELERATIONS
C
   50 FX= FXS
      FZ= (FZ + FZIN)/ASOL
      FZS= FZ
   60 UD= FX/AMASS + EU
      WD= FZ/AMASS + EW
C                          PITCHING MOMENT COMPUTATION
C
      ALDOT= FZS*DIV + BSOL
      TY= QBAR*C*(CMO +CMA*ALP1 + CDIM*(CMAD*ALDOT +CMQ*Q1)
     1           +CMU*UP +CMDE*ELEV1 +CMTR +CMSP*SPO1
     2           +CMGE*KGE +CMDS*STAB1 +CMB)*KCMB
      GO TO 70
      END
```

Table 192: LISTING FOR SUBROUTINE OO

```
COO
      SUBROUTINE OO(QFN,QPT,QPC,QPA,QV,QTC,QTA,QCA,PAF,PF,FCY
     2,FPT,FCT,FPC,FFD,TX,TTX,TQX,TCX,TDX,TZ,TTZ,TQZ,TDZ,TCZ,TFZ,HHY
     3,RR1,RR2,LL1,LL2,AGP,AACH,XTN,XTA,XTC,ATA,ATC,VVTK
     4,VVCH,DUM)
C
C
C     VERSION 1                          MARCH 1979
C
C     WRITTEN BY - GS DULEBA
C
C     PURPOSE - TO CONVERT FOSTER MILLER OUTPUT VARIABLES INTO
C                  EASY COMPONENT OUTPUT VARIABLES
C
C ***** OUTPUTS *****
C
C     QTC       TRUNK-TO-CUSHION FLOW, CFS
C     QTA       TRUNK-TO-ATMOSPHERE FLOW, CFS
C     QCA       CUSHION-TO-ATMOSPHERE FLOW, CFS
C     QFN       TOTAL FAN FLOW, CFS
C     QPA       PLENUM-TO-ATMOSPHERE FLOW, CFS
C     QPC       PLENUM-TO-CUSHION FLOW, CFS
C     QPT       PLENUM-TO-TRUNK FLOW, CFS
C     QV        PRESSURE RELIEF VALVE FLOW, CFS
C     PAF       FAN INLET PRESSURE, PSFG
C     PF        FAN PRESSURE RISE, PSF
C     FCY       TOTAL VERTICAL FORCE, LBS
C     FPT       TRUNK CONTACT FORCE, LBS
C     FCT       TRUNK DAMPING FORCE, LBS
C     FPC       CUSHION FORCE, LBS
C     FFD       AERO DRAG FORCE IN HEAVE, LBS
C     TX        TOTAL TORQUE X AXIS, FT-LBS
C     TTX       TRUNK PRESSURE TORQUE X AXIS, FT-LBS
C     TQX       TRUNK DAMPING TORQUE X AXIS, FT-LBS
C     TCX       CUSHION PRESSURE TORQUE X AXIS, FT-LBS
C     TDX       AERO DRAG TORQUE X AXIS, FT-LBS
C     TZ        TOTAL TORQUE Z AXIS, FT-LBS
C     TTZ       TRUNK PRESSURE TORQUE Z AXIS, FT-LBS
C     TQZ       TRUNK DAMPING TORQUE Z AXIS, FT-LBS
C     TDZ       AERO DRAG TORQUE Z AXIS, FT-LBS
C     TCZ       CUSHION PRESSURE TORQUE Z AXIS, FT-LBS
C     TFZ       GROUND FRICTION TORQUE Z AXIS, FT-LBS
C     HHY       TRUNK HEIGHT, FT
C     RR1       OUTER TRUNK RADIUS OF CURV, FT
C     RR2       INNER TRUNK RADIUS OF CURV, FT
C     LL1       TRUNK LENGTH OUTER-HORIZ ATTACH POINT, FT
C     LL2       TRUNK LENGTH INNER-HORIZ ATTACH POINT, FT
C     AGP       AREA OF CUSHION GAP, SQ FT
C     AACH      AREA OF CUSHION, SQ FT
C     XTN       AREA OF TRUNK-GROUND CONTACT, SQ FT
C     XTA       TRUNK-ATMOS AREA IN CONTACT AREA, SQ FT
C     XTC       TRUNK-CUSHION AREA IN CONTACT AREA, SQ FT
C     ATA       TRUNK TO ATMOS AREA, SQ FT
C     ATC       TRUNK TO CUSHION AREA, SQ FT
C     VVTK      TRUNK VOLUME, CU FT
C     VVCH      CUSHION VOLUME, CU FT
```

```
C
C ***** INPUTS *****
C   DUM      DUMMY VARIABLE
C
C
      COMMON/COEFFS/CPA,CAF,CPC,CPT,CTC,CGAP,CTA,CVENT,CKK
C
      COMMON/FORTQ/FCP,FTP,FORCT,FDF,FORCEY,TCPX,TTPX,TORQTX,TDFX,
     1TORQUEX,TCPZ,TTPZ,TORQTZ,TDFZ,TORFZ,TORQUEZ
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
C
C
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
C
      COMMON/BTERM/B11,B12,B13,B21,B22,B23,B31,B32,B33
C
      COMMON/SHAPE/AGAPI(100),AGAPR(100),ATKI(100),ATKR(100),ATKCHI(100)
     1,ATKCHR(100),ATKATI(100),ATKATR(100),ACHI(100),ACHR(100)
     2,ATKCNI(100),ATKCNR(100),VCHI(100),VCHR(100),VTKI(100),VTKR(100)
     3,XCH(100),XTK(100),ZCH(100),ZTK(100),PERI(100)
C
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
C
      COMMON/VEHCL/MASS,AIX,AIZ,AIXY,AIYZ,AIZX,CC,GG,FF,AIFAN
C
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
C
      COMMON/LOADS/FORCNS(3,10),CCS(3,10),YCGS(3,10),PHIS(3,10),AGAPS(3,1C
     110),PCHS(3,10),QFANS(3,10),PFANS(3,10),PPLMS(3,10),THIS(3,10),
     1FFS(3,10),TORXS(3,10),TORZS(3,10),PTKS(3,10)
C
       COMMON/COMPRS/AL0,AL1,AL2,AL3,AL4,G0,G1,G2,G3,G4,QP1
C
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
C
      COMMON/DYNAMIC/TIME,FTIME,DTIME,IQ,NQ,DVCH,DVTK,VELX,DERY(13)
     1,DVCHP
C
      COMMON/ESTMO/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
C
      COMMON/LABL/LABEL(80)
C
      COMMON/STATIC/YSTRT,YSTOP,PSTRT,PSTOP,TSTRT,TSTOP
     1,PHIYC,THEYC,YCPHI,YCTHE
C
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
C
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
```

```
      COMMON/HCUR/AH0,AH1,AH2,AH3
C
      COMMON/SAVE/R1I,R2I,PHI1I,PHI2I,L1I,L2I,A1I,A2I,D2D2HBI,DXAMABI,
     1 BETAD2I,SINPHRI,D2I,SINPH2I,A1MA2,X1I,X2I,X12I,SI
      REAL L1,LL1,L2,LL2
C     *****************************************************************
C
      QTC=QTKCH
      QTA=QTKAT
      QCA=QCHAT
      QFN=QFAN
      QPA=QPLAT
      QPC=QPLCH
      QPT=QPLTK
      QV=QVENT
      PAF=PATFN
      PF=PFAN
      FCT=FORCT
      FPC=FCP
      TCX=TCPX
      TDX=TDFX
      TFZ=TORFZ
      TQX=TORQTX
      TTX=TTPX
      FCY=FORCEY
      TX=TORQUEX
      TZ=TORQUEZ
      FPT=FTP
      FFD=FDF
      TCZ=TCPZ
      TDZ=TDFZ
      TQZ=TORQTZ
      TTZ=TTPZ
      HHY=HY
      RR1=R1
      RR2=R2
      LL1=L1
      LL2=L2
      ATA=ATKAT
      ATC=ATKCH
      AACH=ACH
      XTA=ATKATC
      VVCH=VCH
      VVTK=VTK
      XTN=ATKCN
      AGP=AGAP
      XTC=ATKCHC
      RETURN
      END
```

Table 193: LISTING FOR SUBROUTINE OUTFM

```
COUTFM
      SUBROUTINE OUTFM(INUM)
C
C
C   VERSION 1                         MARCH 1979
C
C   WRITTEN BY - GS DULEBA
C
C   PURPOSE - TO REPLACE FOSTER MILLER SUBROUTINE -OUTPUT-.
C             OUTPUT CONTAINED PLOTTING LOGIC WHICH IS INCOM-
C             PATIBLE WITH THE EASY PROGRAM.
C
C
C INTEGRATION GEOMETRY CALCULATION AND OUTPUT
C
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
      COMMON/DYNAMIC/TIME,FTIME,DTIME,IQ,NQ,DVCH,DVTK,VELX,DERY(13)
     1,DVCHP
      COMMON/ESTMD/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
       COMMON/COMPRS/AL0,AL1,AL2,AL3,AL4,G0,G1,G2,G3,G4,QP1
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
      COMMON/SHAPE/AGAPI(100),AGAPR(100),ATKI(100),ATKR(100),ATK3HI(100)
     1,ATKCHR(100),ATKATI(100),ATKATR(100),ACHI(100),ACHR(100)
     2,ATKCNI(100),ATKCNR(100),VCHI(100),VCHR(100),VTKI(100),VTKR(100)
     3,XCH(100),XTK(100),ZCH(100),ZTK(100),PERI(100)
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      DATA PI,RADIAN/3.141592653,57.295779/
C
      IF(INUM)100,1,100
1     J=1
      IF(MM.LE.0)MM=1
      I=MM-1
      K=50
      VCHS=VCH
      VTKS=VTK
      IXT=0
      JXT=1
100    CONTINUE
C UPDATE AREAS, VOLUMES AND DVCHP.
C CALCULATE DVCH
      CALL TRUNK(ISHAPE)
      IF(ISHAPE.EQ.0) GO TO 555
      CALL SEGMNT(1)
       CALL COORDN
      CALL PROFILE
```

```
       CALL CLRNCE
      CALL SHAPE2
      DVCH=(VCH-VCHS)/DTIME
C UPDATE DVCHP
      CALL CDVCHP
      PRAT=AMAX1(0.0,AMIN1(1.0,(PCH/PTK)))
C COMPUTE NEW HY FROM PCH/PTK FOR NEXT STEP
      CALL HYCURV(PRAT,HX)
      HY=HX*HYI
      CALL TRUNK(ISHAPE)
      IF(ISHAPE.EQ.0) GO TO 555
      CALL SEGMNT(1)
       CALL COORDN
      CALL PROFILE
       CALL CLRNCE
      CALL SHAPE2
C CALCULATE DVTK
      DVTK=(VTK-VTKS)/DTIME
C COMPUTE HORIZONTAL VELOCITY
      VELX=VELX-DECCL*DTIME
      IF(VELX.LE.0.0)VELX=0.
      I=I+1
C
   13 VCHS=VCH
      VTKS=VTK
C
C IF NO PRV SET IPRV=0 AND SKIP
      IF(NPRV.LE.0) GO TO 75
C DETERMINE IF PRESSURE RELIEF VALVE OPENS
      PRFOR=PPLM*APRV
      IF(XV.LE.0.0001)GO TO 70
      XXX=XA-0.0001
      IF(XV.GE.XXX)GO TO 62
      GO TO 65
C CLOSED PRV
   70 COFOR=PPLMB*APRV
      IF(PRFOR.GT.COFOR)GO TO 65
75    IPRV=0
      GO TO 66
C FULL OPEN PRV
   62 COFOR=PPLMB*APRV+AKPRV*XA
      IF(PRFOR.LT.COFOR)GO TO 65
      IPRV=0
      XV=XA
      GO TO 66
C NORMAL RELIEF OPERATION
   65 IPRV=1
   66 CONTINUE
      XCG=XCG+VELX*DTIME
   10 RETURN
  555 WRITE(6,556)
  556 FORMAT(//,40X,27H*** FAILURE TO CONVERGE ***)
      KILL=1/ISHAPE
      END
```

Table 194: LISTING FOR SUBROUTINE PARAMS

```
CPARAMS
      SUBROUTINE PARAMS
C INITIALIZE PARAMETERS FOR FMA ACLS PROGRAM
C
      COMMON/ESTMD/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/VEHCL/MASS,AIX,AIZ,AIXY,AIYZ,AIZX,CC,GG,FF,AIFAN
      COMMON/COEFFS/CPA,CAF,CPC,CPT,CTC,CGAP,CTA,CVENT,CKK
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
C
C FLOW COEFFICIENT
      CKK=1.4
      CPA=0.6
      CAF=1.0
      CPC=0.60
      CPT=0.90
      CTC=0.76
      CTA=0.76
      CGAP=1.0
      CVENT=0.70
C
C OTHER ESTIMATED PARAMETERS
      GEC=0.2
      ZEPRV=0.15
      U=0.5
      DECCL=0.
      CENFX=-CC
      CENFZ=-FF
      DAMPC=3.2
      QP2=5.0
      SLOPE=10.0
C
      NSTOP=4*(N+M)
C CLEAR DELTA ARRAY TO ZERO
      DO 10 I=1,NSTOP
      DELTA(I)=0.0
   10 CONTINUE
C ************************
      RETURN
      END
```

Table 195: LISTING FOR SUBROUTINE PERF

```
CPERF
      SUBROUTINE PERF(ZGAP,L1,L3,L3P,LP,LH,RA,YO,YOFS,D,AP,PT,PC,PA,
     *                BET,AHA1,AHA2,AHC2,AHC1)
      REAL L1,L3,L3P,LP,LH,LHA1,LHC1,L3A,L3C
C
C   VERSION 1.           FEB.1978
C
C   PURPOSE - TO CALCULATE ORIFICE AREAS FOR FLOW TO
C             ATMOSPHERE AND TO CUSHION.
C
C   MATHOD  - SEE TRUNK DOCUMENTATION,VOLUME I,
C             FINAL REPORT.
C
C   LIMITATIONS - CALLED ONLY BY TRUNK COMPONENT TK
C
C   WRITTEN BY J.R.KILNER
C
C  TEST FOR GROUND CONTACT
      DA=D*AP
      IF(ZGAP.GE.0.)GOTO 50
C
C  LOADED TRUNK SHAPE
C
      LHA1=L1-LP
      LHC1=LP+LH-L1-L3P
      PX=L3/(2.*PT-PC-PA)
      L3A=(PT-PA)*PX
      L3C=(PT-PC)*PX
      EA1=EA2=EC2=EC1=1.
C
C  TEST FOR SIDE ELEMENT
C
      IF(BET.EQ.0.)GOTO 20
C
      RN=RA+YO
      RD=RA+YOFS
      EA1=(RN+L3+.5*LHA1)/RD
      EA2=(RN+L3-.5*L3A)/RD
      EC2=(RN+.5*L3C)/RD
      EC1=(RN-.5*LHC1)/RD
C
20    IF(LHA1.LE.0.)GOTO 22
      IF(LHC1.LE.0.)GOTO 24
C  CASE 1
      AHA1=AHA1+LHA1*DA*EA1
      AHA2=AHA2+L3A*DA*EA2
      AHC2=AHC2+L3C*DA*EC2
      AHC1=AHC1+LHC1*DA*EC1
      RETURN
C
24    IF(-LHC1.GE.L3C)GOTO 26
C  CASE 2
      AHA1=AHA1+LHA1*DA*EA1
      AHA2=AHA2+L3A*DA*EA2
      AHC2=AHC2+(L3C+LHC1)*DA*EC2
      RETURN
```

```
C
26     IF(-LHC1.GE.L3)GOTO 28
C  CASE 3
       AHA1=AHA1+LHA1*DA*EA1
       AHA2=AHA2+(L3+LHC1)*DA*EA2
       RETURN
C
C  CASE 4
28     AHA1=AHA1+LH*DA*EA1
       RETURN
C
22     IF(-LHA1.GE.L3A)GOTO 30
       IF(LHC1.LE.0.)GOTO 32
C  CASE 5
       AHA2=AHA2+(L3A+LHA1)*DA*EA2
       AHC2=AHC2+L3C*DA*EC2
       AHC1=AHC1+LHC1*DA*EC1
       RETURN
C
32     IF(-LHC1.GE.L3C)GOTO 34
C  CASE 6
       AHA2=AHA2+(L3A+LHA1)*DA*EA2
       AHC2=AHC2+(L3C+LHC1)*DA*EC2
       RETURN
C
C  CASE 7
34     AHA2=AHA2+(L3+LHA1+LHC1)*DA*EA2
       RETURN
C
30     IF(-LHA1.GE.L3)GOTO 36
       IF(LHC1.LE.0.)GOTO 38
C  CASE 8
       AHC2=AHC2+(L3+LHA1)*DA*EC2
       AHC1=AHC1+LHC1*DA*EC1
       RETURN
C
C  CASE 9
38     AHC2=AHC2+(L3+LHA1+LHC1)*DA*EC2
       RETURN
C
C  CASE 10
36     AHC1=AHC1+LH*DA*EC1
       RETURN
C
C  FREE TRUNK SHAPE
C
50     LHA1=L1-LP
       LHC1=LP+LH-L1
       EA1=EC1=1.
C
C  TEST FOR SIDE ELEMENT
C
       IF(BET.EQ.0.)GOTO 52
C
       RD=RA+YOFS
       EA1=(RD+.5*LHA1)/RD
```

```
      EC1=(RD-.5*LHC1)/RD
C
52    IF(LHA1.LE.0.)GOTO 54
      IF(LHC1.LE.0.)GOTO 56
C  CASE 11
      AHA1=AHA1+LHA1*DA*EA1
      AHC1=AHC1+LHC1*DA*EC1
      RETURN
C
C  CASE 12
56    AHA1=AHA1+LH*DA*EA1
      RETURN
C
C  CASE 13
54    AHC1=AHC1+LH*DA*EC1
      RETURN
C
      END
```

Table 196: LISTING FOR SUBROUTINE PERFB

```
CPERFB
      SUBROUTINE PERFB(ZGAP,L1,L3,LP,LH,D,AP,AH1,AH2)
      REAL L1,L3,LP,LH,LHA,LHB
C
C   VERSION 1.          APRIL 10,1978
C
C   PURPOSE - TO CALCULATE ORIFICE AREAS FOR AIR BAG
C           - OUTFLOW THRU SURFACE PERFORATIONS
C
C   METHOD  - SEE AIR BAG DOCUMENTATION,VOLUME I,
C             FINAL REPORT.
C
C   LIMITATIONS - CALLED ONLY BY AIR BAG COMPONENT AB
C
C   WRITTEN BY J.R.KILNER
C
C  TEST FOR GROUND CONTACT
      DA=D*AP
      IF(ZGAP.GE.0.)GOTO 50
C
C  LOADED SHAPE
C
      LHB=L1-LP
      LHA=LP+LH-L1-L3
C
      IF(LHB.LE.0.)GOTO 22
      IF(LHA.LE.0.)GOTO 24
C  CASE 1
      AH1=AH1+(LHA+LHB)*DA
      AH2=AH2+L3*DA
      RETURN
C
24    IF(-LHA.GE.L3)GOTO 26
C  CASE 2
      AH1=AH1+LHB*DA
      AH2=AH2+(L3+LHA)*DA
      RETURN
C  CASE 3
26    AH1=AH1+LH*DA
      RETURN
C
22    IF(-LHB.GE.L3)GOTO 30
      IF(LHA.GE.0.)GOTO 32
C  CASE 4
      AH2=AH2+LH*DA
      RETURN
C
C  CASE 5
32    AH1=AH1+LHA*DA
      AH2=AH2+(L3+LHB)*DA
      RETURN
C
C  CASE 6
30    AH1=AH1+LH*DA
      RETURN
C
```

Table 196: LISTING FOR SUBROUTINE PERFB (CONCLUDED)

```
C
C  FREE SHAPE
C
50     AH1=AH1+LH*DA
       RETURN
C
       END
```

Table 197: LISTING FOR SUBROUTINE PRND

```
CPRND
      FUNCTION PRND(IFL,T,SH)
C
C   PURPOSE - TO CALCULATE PRANDTL NUMBER
C
C
C   METHOD   - PR=SPECIFIC HEAT X VISCOSITY/THERMAL CONDUCTIVITY
C            - FLUID PROPERTIES OBTAINED FROM FUNCTIONS PROP AND SHCP
C
C
C   WRITTEN BY  -   ADAM LLOYD          LATEST REVISION     NOV 75
C
C
C   LIMITATIONS - DEPENDENT ON FLUID. SEE PROP AND SHCP
C
C
C   INPUT/OUTPUT LIST
C
C   PRND   PRANDTL NUMBER                                      OUTPUT
C   IFL    INTEGER DESIGNATING FLUID                ---        INPUT
C          (AS DEFINED IN FUNCTION PROP)
C   T      TEMPERATURE                              DEGR       INPUT
C   SH    SPECIFIC HUMIDITY OF WATER VAPOR          LB/LB      INPUT
C          (USED FOR IFL=1 ONLY)
C
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      IF(IFL.GT.1)GO TO 20
      CP=SHCP(T,SH)
      GO TO 30
 20   CP=PROP(IFL,1,T)
 30   XMU=PROP(IFL,2,T)
      CON=PROP(IFL,3,T)
      PRND=CP*XMU/CON
      RETURN
      END
```

Table 198: LISTING FOR SUBROUTINE PROFILE

```
CPROFILE
      SUBROUTINE PROFILE
C USER SPECIFIED GROUND PROFILE.
C ELEVATION YG(I) IS EXPRESSED AS A FUNCTION OF X AND Z COORDINATES
C OF GROUND POINT I, I.E. XG(I) AND ZG(I)
C
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
      DO 10 I=1,NSTOP
C SET FOR FLAT TERRAIN
      YG(I)=0.
   10 CONTINUE
      RETURN
      END
```

Table 199: LISTING FOR SUBROUTINE PROP

```
CPROP
      FUNCTION PROP(I,IP,T)
C
C     PURPOSE - TO CALCULATE FLUID PROPERTIES (SPECIFIC HEAT, ABSOLUTE
C             - VISCOSITY, THERMAL CONDUCTIVITY, DENSITY)
C
C     METHOD  - USES POLYNOMIAL APPROXIMATIONS  PROP=F(TEMP)
C               LOGARITHMIC FIT FOR VISCOSITY OF LIQUIDS
C
C
C     WRITTEN BY  -    ADAM LLOYD          LATEST REVISION     NOV 75
C
C
C     LIMITATIONS -    VALID TEMPERATURE RANGES
C
C                      FLUID                                  TMIN      TMAX
C                                                                 (DEGR)
C                      DRY AIR                                300       1600
C                      WATER AT SAT PRESSURE                  500       1000
C                      60/40 EGW                              400        800
C                      HEAT TRANSPORT FLUID FC-75             400        650
C                      FUEL JP-4 (MIL-F-5624)                 400        600
C                      HEAT TRANSPORT FLUID DC-331            400        700
C                      HYDRAULIC FLUID (MIL-H-83282)          400        900
C                      HYDRAULIC FLUID (MIL-H-5606)           400        600
C
C
C     INPUT/OUTPUT LIST
C
C     PROP        FLUID PROPERTY                                   OUTPUT
C     I           INTEGER DESIGNATING FLUID            ---         INPUT
C                 =1   DRY AIR
C                 =2   WATER AT SATURATION PRESSURE
C                 =3   60/40 ETHYLENE GLYCOL/WATER
C                 =4   HEAT TRANSPORT FLUID FC-75
C                 =5   FUEL JP-4 (MIL-F-5624)
C                 =6   HEAT TRANSPORT FLUID DC-331
C                 =7   HYDRAULIC FLUID (MIL-H-83282)
C                 =8   HYDRAULIC FLUID (MIL-H-5606)
C
C     IP      INTEGER DESIGNATING PROPERTY TO BE CALCULATED       INPUT
C                 =1   SPECIFIC HEAT                   BTU/LB DEGR
C                 =2   ABSOLUTE VISCOSITY              LB/FT HR
C                 =3   THERMAL CONDUCTIVITY            BTU/FT HR DEGR
C                 =4   DENSITY                         LB/FT3
C                      **** DENSITY CALCULATION NOT VALID FOR I=1 ****
C
C     T       TEMPERATURE AT WHICH PROPERTY IS REQD  DEGR          INPUT
C
      DIMENSION CP(5,8),VIS(5,8),THK(5,8),RO(5,8)
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      DATA CP/
     1 .24788     ,-42.046E-06, 5.7679E-08,-14.931E-12,0.        ,
     2 4.5302     ,-2.0410E-02, 4.3086E-05,-4.0179E-08, 1.4219E-11,
     3-9.2843E-01, 7.5662E-03,-1.3232E-05, 1.1255E-08,-3.7296E-12,
```

```
     4 3.5504E-02, 9.0754E-04,-1.9672E-06, 2.5322E-09,-1.2045E-12,
     5 1.1300E-01, 1.0223E-03,-1.4367E-06, 2.2667E-09,-1.3333E-12,
     6 3.2800E-01, 1.8000E-04, 0.        , 0.        , 0.        ,
     7 4.2498E-01,-6.6351E-04, 2.5118E-06,-2.4196E-09, 8.6285E-13,
     8-1.4701E-01, 2.9948E-03,-7.1652E-06, 9.1111E-09,-4.2424E-12/
      DATA VIS/
     1 5.5029E-03, 8.7157E-05,-2.9464E-08, 6.2500E-12, 0.        ,
     2 2.3649E+01,-9.1079E-02, 1.2969E-04,-8.4487E-08, 2.0822E-11,
     3 1.1786E+02,-6.4304E-01, 1.3668E-03,-1.3199E-06, 4.8195E-10,
     4 4.7377E+01,-2.5971E-01, 5.7398E-04,-5.9191E-07, 2.3372E-10,
     5 7.7977E+01,-5.2305E-01, 1.3745E-03,-1.6488E-06, 7.5126E-10,
     6 4.0799E+01,-1.9543E-01, 3.9815E-04,-3.8320E-07, 1.4228E-10,
     7 7.0421E+01,-2.9404E-01, 4.8815E-04,-3.7464E-07, 1.1020E-10,
     8 8.9064E+01,-4.4104E-01, 8.5289E-04,-7.4108E-07, 2.4158E-10/
      DATA THK/
     1 1.3500E-03, 2.7780E-05,-4.2857E-09, 1.0416E-12, 0.        ,
     2-1.4950E+00, 8.6283E-03,-1.5038E-05, 1.2059E-08,-3.8228E-12,
     3 5.7303E-01,-2.2979E-03, 5.5721E-06,-5.7990E-09, 2.0979E-12,
     4 4.7389E-02,-1.1812E-05,-2.5552E-08, 2.6998E-11,-1.0308E-14,
     5 7.6450E-02, 1.1592E-04,-4.4917E-07, 6.3333E-10,-3.3333E-13,
     6-1.5520E-01, 1.7181E-03,-4.4217E-06, 4.8222E-09,-2.0000E-12,
     7 1.2821E-01,-2.0731E-04, 3.3615E-07,-2.7287E-10, 8.2338E-14,
     8 1.2497E-01,-1.4185E-04, 1.4045E-07,-1.5556E-10, 6.0606E-14/
      DATA RO/
     1 8.0800E-02, 0.        , 0.        , 0.        , 0.        ,
     2 5.2289E+00, 3.2185E-01,-6.4108E-04, 5.4250E-07,-1.8182E-10,
     3 7.6413E+01,-3.5580E-02, 9.2685E-05,-1.4564E-07, 6.5734E-11,
     4 2.7751E+02,-1.0204E+00, 2.6867E-03,-3.3947E-06, 1.5762E-09,
     5 6.4000E+01,-3.0000E-02, 0.        , 0.        , 0.        ,
     6 7.6479E+01,-3.5024E-02, 3.3333E-06,-2.9190E-18, 1.2947E-21,
     7 5.6567E+01, 3.3975E-02,-1.4570E-04, 1.5879E-07,-6.2118E-11,
     8 2.5754E+01, 2.7183E-01,-7.9985E-04, 9.5556E-07,-4.2424E-10/
      T=AMAX1(T,350.)
      GO TO (10,20,30,40) IP
C     CALCULATION OF SPECIFIC HEAT
 10   PROP=(((CP(5,I)*T+CP(4,I))*T+CP(3,I))*T+CP(2,I))*T+CP(1,I)
      GO TO 50
C     CALCULATION OF VISCOSITY
 20   PROP=(((VIS(5,I)*T+VIS(4,I))*T+VIS(3,I))*T+VIS(2,I))*T+VIS(1,I)
      IF(I.GT.1.5) PROP=EXP(PROP)
      GO TO 50
C     CALCULATION OF THERMAL CONDUCTIVITY
 30   PROP=(((THK(5,I)*T+THK(4,I))*T+THK(3,I))*T+THK(2,I))*T+THK(1,I)
      GO TO 50
C     CALCULATION OF DENSITY    ****  INVALID FOR I=1   ****
 40   PROP=(((RO(5,I)*T+RO(4,I))*T+RO(3,I))*T+RO(2,I))*T+RO(1,I)
C     TEST FOR DIAGNOSTIC PRINTOUT
 50   IF(IERR.NE.1)GO TO 200
C     TEST IF TEMPERATURES ARE WITHIN RANGE
      GO TO(110,120,130,140,150,160,170,180) I
 110  IF(T.GT.300.AND.T.LT.1600) GO TO 200
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,32HNON FATAL ERROR CALLED FROM PROP/
     1 10X,41HTEMPERATURE OF AIR NOT WITHIN VALID RANGE)
      GO TO 200
 120  IF(T.GT.500.AND.T.LT.1000.)GO TO 200
```

```
      WRITE(IWRITE,9998)
 9998 FORMAT(10X,32HNON FATAL ERROR CALLED FROM PROP/
     1 10X,43HTEMPERATURE OF WATER NOT WITHIN VALID RANGE)
      GO TO 200
 130  IF(T.GT.400.AND.T.LT.800.)GO TO 200
      WRITE(IWRITE,9997)
 9997 FORMAT(10X,32HNON FATAL ERROR CALLED FROM PROP/
     1 10X,47HTEMPERATURE OF 60/40 EGW NOT WITHIN VALID RANGE)
      GO TO 200
 140  IF(T.GT.400.AND.T.LT.650.)GO TO 200
      WRITE(IWRITE,9996)
 9996 FORMAT(10X,32HNON FATAL ERROR CALLED FROM PROP/
     1 10X,43HTEMPERATURE OF FC-75 NOT WITHIN VALID RANGE)
      GO TO 200
 150  IF(T.GT.400.AND.T.LT.600.)GO TO 200
      WRITE(IWRITE,9995)
 9995 FORMAT(10X,32HNON FATAL ERROR CALLED FROM PROP/
     1 10X,42HTEMPERATURE OF JP-4 NOT WITHIN VALID RANGE)
      GO TO 200
 160  IF(T.GT.400.AND.T.LT.700.)GO TO 200
      WRITE(IWRITE,9994)
 9994 FORMAT(10X,32HNON FATAL ERROR CALLED FROM PROP/
     1 10X,44HTEMPERATURE OF DC-331 NOT WIHTIN VALID RANGE)
      GO TO 200
 170  IF(T.GT.400.AND.T.LT.900.)GO TO 200
      WRITE(IWRITE,9993)
 9993 FORMAT(10X,32HNON FATAL ERROR CALLED FROM PROP/
     1 10X,61HTEMPERATURE OF HYD FLUID (MIL-H-83282) NOT WITHIN VALID RA
     2NGE)
      GO TO 200
 180  IF(T.GT.400.AND.T.LT.700) GO TO 200
      WRITE(IWRITE,9992)
 9992 FORMAT(10X,32HNON FATAL ERROR CALLED FROM PROP/
     1 10X,60HTEMPERATURE OF HYD FLUID (MIL-H-5606) NOT WITHIN VALID RAN
     2GE)
 200  RETURN
      END
```

Table 200: LISTING FOR SUBROUTINE PT

```
CPT
      SUBROUTINE PT(FX,FXDOT,IFX,FZ,TY,ED,TM,ST,SR,C1,C2,SIG,
     1 GA,TC,TH,XA)
C   VERSION 1.                      AUGUST 19, 1977
C
C   PURPOSE  PITCH CONTROL THRUSTER
C
C--METHOD   VECTORED THRUST IS A FUNCTION OF SIG IN A SATURATION FUNCTION,
C           WHERE SIG IS AN INPUT FROM AIRCRAFT CONTROL SYSTEM COMPONENT.
C           IF THE ENGINE DEPENDENCE INDICATOR ED IS 1, VECTORED THRUST
C           CAPABILITY IS DEPENDENT ON ENGINE THRUST  ED=0 SETS AVAILABLE
C           VECTORED THRUST TO A CONSTANT.
C--CALL SEQUENCE
C  *****  OUTPUTS  *****
C    FX         ENGINE THRUST REDUCTION
C    FXDOT,IFX  THRUST REDUCTION RATE,INT CONTROL
C    FZ         VECTORED THRUST-VERTICAL FORCE
C    TY         PITCH MOMENT DUE TO THRUSTER
C  *****  INPUTS  *****
C    ED         ENGINE DEPENDENCE INDICATOR(ED=1.0,YES ED=0.0,NO)
C    TM         THRUSTER MAXIMUM FORCE FOR ENGINE INDEPENDENT SYSTEM
C    ST         SLOPE FOR VECTORED THRUST AS FUNCTION OF ENGINE THRUST
C    SR         SLOPE OF ENGINE THRUST REDUCTION AS FUNCTION OF VECTORED TH
C    C1         SATURATION FUNCTION SLOPE
C    C2         SATURATION SLOPE
C    SIG        AIRCRAFT CONTROL SYSTEM SIGNAL TO THRUSTER
C    GA         FIRST ORDER LAG GAIN
C    TC         FIRST ORDER LAG TIME CONSTANT
C    TH         ENGINE THRUST
C    XA         THRUSTER PITCH MOMENT ARM
C
C   WRITTEN BY    MAHINDER WAHI   (BASED ON COMPONENT *YC*)
C
C--SWITCH FOR ENGINE DEPENDENCE
      IF(ED.GT.0.5) GO TO 1
      TVA=TM
      GO TO 2
C--AVAILABLE VECTORED THRUST
    1 TVA=ST*TH
C--SATURATION INTERCEPT
    2 C3=TVA/C1
      C6=-C3
      C4=C1
      C5=C2
C   SATURATION FUNCTION, FZ(SIG)
      CALL SA(FZ,SIG,C1,C2,C3,C4,C5,C6)
      IF(ED.LT.0.5) GO TO 3
C--ENGINE THRUST REDUCTION FUNCTION
      FR=-SR*ABS(FZ)
C--1ST ORDER LAG ON ENGINE RESPONSE
      IF(IFX.NE.0) FXDOT = (FR*GA-FX)/TC
    3 CONTINUE
C--VECTORED THRUST MOMENTS
      TY = -FZ*XA
      RETURN
      END
```

Table 201: LISTING FOR SUBROUTINE RA

```
CRA
      SUBROUTINE RA(XNU,XNV,XNW,XNP)
      COMMON /CSIMUL/I1,I2,I3,N1,N2,N3,TINC,TMAX
C    VERSION 2.                                        REVISED OCT 22 1976
C    PURPOSE:  GENERATE RANDOM VARIABLES FOR WIND MODEL
C    METHOD:   DISCRETE RANDOM VARIABLES WITH MEAN ZERO AND
C              VARIANCE =2*TINC APPROXIMATE UNIT VARIANCE WHITE NOISE
C
C    ******  OUTPUTS  ******
C      XNU,XNV,XNW    -NOISE SAMPLES FOR U,V,W GUST VELOCITIES
C      XNP            -NOISE SAMPLE FOR P ANGULAR RATE GUST
C    DESIGNED BY A.W.WARREN                            SEPT. 1976
C
      DATA DUM/.99999/
      SIG= SQRT(TINC+TINC)
      CALL RN(XNU,DUM,SIG, 0.)
      CALL RN(XNV,DUM,SIG,0.)
      CALL RN(XNW,DUM,SIG,0.)
      CALL RN(XNP,DUM,SIG,0.)
C
      RETURN
      END
```

Table 202: LISTING FOR SUBROUTINE RENVX

```
CRENVX
      FUNCTION RENVX(IFL,W,T,D)
C
C    PURPOSE - TO CALCULATE REYNOLDS NUMBER OF A FLUID IN AN ELEMENT OF
C              CIRCULAR CROSS-SECTION
C
C    METHOD  - RE=(4.*W/PI*MU*D)
C
C
C    LIMITATIONS -
C
C
C    WRITTEN BY       ADAM LLOYD           LATEST REVISION      NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    RENVX        REYNOLDS NUMBER                         ---          OUTPUT
C    IFL          INTEGER DESIGNATING FLUID(SEE PROP)---               INPUT
C    W            FLOW RATE                               LB/MIN       INPUT
C    T            TEMPERATURE                             DEGR         INPUT
C    D            DIAMETER                                IN           INPUT
C    NOTE - VISCOSITY IS ONLY FLUID PROPERTY. HENCE INPUT OF SPECIFIC
C           HUMIDITY NOT REQUIRED
      COMMON/CIO/IREAD,IWRITE,IDIAG
      XMU= PROP(IFL,2,T)
      RENVX=ABS(916.732*W/(XMU*D))
C    CONSTANT = 4*12*60/PI
      RETURN
      END
```

Table 203: LISTING FOR SUBROUTINE RES

```
CRES
      SUBROUTINE RES(X,M,K,R,P)
C
C    PURPOSE - EVALUATE CABLE STRAIN EQUATION
C
C    LIMITATIONS - CALLED ONLY BY ROUTINE QNWT IN ROUTINE KINK
C
C      WRITTEN BY  -  J.R.KILNER
C
      DIMENSION X(1),R(1),P(1)
C
      A=X(1)*(1.+X(1))
      B=SQRT(A)-X(1)+P(3)
      B=B*B
C
      R(1)=P(2)+B-A
C
      RETURN
      END
```

Table 204: LISTING FOR SUBROUTINE RG

```
CRG
      SUBROUTINE RG(P2,P2D,IP2,Q2,Q2D,IQ2,R2,R2D,IR2,PX,PXD,IPX,QX,QXD,
     1 IQX,RX,RXD,IRX,P1,Q1,R1,SL,DMP,WN)
C  VERSION: 1.                        REVISED:  JUNE 1 1976
C  PURPOSE:  MODEL DYNAMICS AND SATURATION OF THREE RATE GYROS
C  METHOD:  THE SAME 2ND ORDER DYNAMICS AND SATURATION LIMIT IS
C           APPLIED TO EACH RATE GYRO.  SATURATION IS APPLIED  .
C           BY INCREASING FEEDBACK SIGNAL AROUND INTEGRATORS BY A
C           FACTOR OF 100.
C  CALL SEQUENCE:
C  ********************  OUTPUTS  *********************
C   P2,Q2,R2     -  OUTPUT RATES, RAD/SEC.
C   PX,QX,RX     -  INTERMEDIATE STATES
C  ********************  INPUTS  ********************
C   P1,Q1,R1     -  INPUT (SENSED) RATES, RAD/SEC.
C   SL           -  SATURATION LIMIT, RAD/SEC  (SAME FOR ALL GYROS)
C   DMP          -  DAMPING COEFFICIENT (SAME FOR ALL GYROS)
C   WN           -  NATURAL FREQUENCY, RAD/SEC.  (SAME FOR ALL GYROS)
C  DESIGNED BY: J.D. BURROUGHS                       JUNE 1976
C  =============  ROLL RATE  ===============
      FB=P2
C ------- SATURATION FEEDBACK CALCULATION ========
      IF(ABS(P2).GT.SL)FB=100*(P2-SIGN(SL,P2))+SIGN(SL,P2)
      IF(IPX.NE.0)PXD=WN*(P1-FB)
      IF(IP2.NE.0)P2D=WN*(PX-2*DMP*FB)
C  =============  PITCH RATE  ==============
      FB=Q2
C ------- SATURATION FEEDBACK CALCULATION ========
      IF(ABS(Q2).GT.SL)FB=100*(Q2-SIGN(SL,Q2))+SIGN(SL,Q2)
      IF(IQX.NE.0)QXD=WN*(Q1-FB)
      IF(IQ2.NE.0)Q2D=WN*(QX-2*DMP*FB)
C  =============  YAW RATE  ==============
      FB=R2
C ------- SATURATION FEEDBACK CALCULATION ========
      IF(ABS(R2).GT.SL)FB=100*(R2-SIGN(SL,R2))+SIGN(SL,R2)
      IF(IRX.NE.0)RXD=WN*(R1-FB)
      IF(IR2.NE.0)R2D=WN*(RX-2*DMP*FB)
      RETURN
      END
```

Table 205: LISTING FOR SUBROUTINE RN

```
CRN
      SUBROUTINE RN(U,AX,SIG,AMN)
C  VERSION 1.                          REVISED: OCT 7 1976
C  PURPOSE - GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C  CALL SEQUENCE
C               U   - THE RANDOM NUMBER OUTPUT
C               AX  - A START PARAMETER WHICH CONTROLS THE BEGINNING POINT
C                     OF THE OUTPUT SEQUENCE.  AX SHOULD BE ANY ODD NUMBER
C                     GREATER THAN ONE. THE DEFAULT VALUE OF AX IS 43146971.
C                     AX IS UPDATED FOR NEW CALLS TO THE SUBROUTINE
C               SIG- THE DESIRED STANDARD DEVIATION OF THE SEQUENCE
C               AMN- THE DESIRED MEAN OF THE SEQUENCE
C  DESIGNED BY      ROGER W. CALL                   SEPT.  1976
      DATA IY/25396781/
      IF(AX.EQ..99999)AX=43146971.
      IX=AX
      SUM=0.
      DO 1 I=1,12
      IX=IX*IY
      SUM=SUM+IX/281474976710655.
    1 AX=IX
      U=(SUM-6.0)*SIG+AMN
      RETURN
      END
```

Table 206: LISTING FOR SUBROUTINE ROTATE

```
CROTATE
      SUBROUTINE ROTATE
C BMATRIX TRANSFORMS A VECTOR FROM VEHICLE FRAME TO INERTIAL FRAME
C
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,OFANX
      COMMON/BTERM/B11,B12,B13,B21,B22,B23,B31,B32,B33
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
C CALCULATE TRANSCENDENTALS
      CSIE=COS(SIE)
      CPHIE=COS(PHIE)
      CTHETAE=COS(THETAE)
      SSIE=SIN(SIE)
      SPHIE=SIN(PHIE)
      STHETAE=SIN(THETAE)
C
C COMPUTE TRANSLATION MATRIX ELEMENTS
      B11=CSIE*CPHIE+STHETAE*SPHIE*SSIE
      B12=SPHIE*CTHETAE
      B13=-SSIE*CPHIE+STHETAE*SPHIE*CSIE
      B21=-SPHIE*CSIE+SSIE*CPHIE*STHETAE
      B22=CPHIE*CTHETAE
      B23=SPHIE*SSIE+CSIE*CPHIE*STHETAE
      B31=SSIE*CTHETAE
      B32=-STHETAE
      B33=CSIE*CTHETAE
      RETURN
      END
```

Table 207: LISTING FOR SUBROUTINE RT

```
CRT
      SUBROUTINE RT(FX,FXDOT,IFX,FZ,TX,ED,TM,ST,SR,C1,C2,SIG,
     1 GA,TC,TH,YA)
C   VERSION 1.                      AUGUST 19, 1977
C
C   PURPOSE  ROLL CONTROL THRUSTER
C
C--METHOD   VECTORED THRUST IS A FUNCTION OF SIG IN A SATURATION FUNCTION,
C           WHERE SIG IS AN INPUT FROM AIRCRAFT CONTROL SYSTEM COMPONENT.
C           IF THE ENGINE DEPENDENCE INDICATOR ED IS 1, VECTORED THRUST
C           CAPABILITY IS DEPENDENT ON ENGINE THRUST  ED=0 SETS AVAILABLE
C           VECTORED THRUST TO A CONSTANT.
C--CALL SEQUENCE
C  *****  OUTPUTS  *****
C    FX         ENGINE THRUST REDUCTION
C    FXDOT,IFX THRUST REDUCTION RATE,INT CONTROL
C    FZ         VECTORED THRUST-VERTICAL FORCE
C    TX         ROLL MOMENT DUE TO THRUSTER NOT ON X-AXIS
C  *****  INPUTS  *****
C    ED         ENGINE DEPENDENCE INDICATOR(ED=1.0,YES ED=0.0,NO)
C    TM         THRUSTER MAXIMUM FORCE FOR ENGINE INDEPENDENT SYSTEM
C    ST         SLOPE FOR VECTORED THRUST AS FUNCTION OF ENGINE THRUST
C    SR         SLOPE OF ENGINE THRUST REDUCTION AS FUNCTION OF VECTORED TH
C    C1         SATURATION FUNCTION SLOPE
C    C2         SATURATION SLOPE
C    SIG        AIRCRAFT CONTROL SYSTEM SIGNAL TO THRUSTER
C    GA         FIRST ORDER LAG GAIN
C    TC         FIRST ORDER LAG TIME CONSTANT
C    TH         ENGINE THRUST
C    YA         THRUSTER ROLL MOMENT ARM
C
C   WRITTEN BY    MAHINDER WAHI  (BASED ON COMPONENT *YC*)
C
C--SWITCH FOR ENGINE DEPENDENCE
      IF(ED.GT.0.5) GO TO 1
      TVA=TM
      GO TO 2
C--AVAILABLE VECTORED THRUST
    1 TVA=ST*TH
C--SATURATION INTERCEPT
    2 C3=TVA/C1
      C6=-C3
      C4=C1
      C5=C2
C   SATURATION FUNCTION, FZ(SIG)
      CALL SA(FZ,SIG,C1,C2,C3,C4,C5,C6)
      IF(ED.LT.0.5) GO TO 3
C--ENGINE THRUST REDUCTION FUNCTION
      FR=-SR*ABS(FZ)
C--1ST ORDER LAG ON ENGINE RESPONSE
      IF(IFX.NE.0) FXDOT = (FR*GA-FX)/TC
    3 CONTINUE
C--VECTORED THRUST MOMENTS
      TX = FZ*YA
      RETURN
      END
```

Table 208: LISTING FOR SUBROUTINE SA

```
CSA
      SUBROUTINE SA(FO,FIN,C1,C2,C3,C4,C5,C6)
C
C    PURPOSE - TO SIMULATE SATURATION
C
C
C    METHOD - SEE CODING. C3 AND C6 ARE VALUES OF THE INPUT AT WHICH
C             SATURATION OCCURS.  C3 IS GREATER THAN C6. THE ROUTINE
C             CAN SIMULATE A CHANGE OF SLOPE AT THE ORIGIN (C1.NE.C4)
C             PROVIDED C6 IS LESS THAN ZERO. SIMILARLY THE SLOPES
C             IN THE SATURATION REGION (C2 AND C5)CAN DIFFER.
C             THE SLOPES CAN BE POSITIVE OR NEGATIVE
C
C
C    WRITTEN BY  -  ADAM LLOYD                 LATEST REVISION - NOV 75
C
C    LIMITATIONS - USE OF ZERO SLOPES (C2=0 OR C5=0) IN THE SATURATION
C                  REGION SHOULD BE AVOIDED. IT IS DESIRABLE THAT THE
C                  SLOPE RATIOS C1/C2 AND C4/C5 SHOULD NOT EXCEED 100.
C                  EXCESSIVE SLOPE RATIOS MAY RESULT IN VERY SLOW
C                  CONVERGENCE
C
C
C    INPUT/OUTPUT LIST
C
C    FO         OUTPUT VARIABLE                   ANY          OUTPUT VAR
C    FIN        INPUT  VARIABLE                   ANY          INPUT  VAR
C    C1         SLOPE                ) FIRST      ANY          INPUT  PARAM
C    C2         SATURATION SLOPE     ) SLOPE      ANY          INPUT  PARAM
C    C3         SATURATION INTERCEPT)             ANY          INPUT  PARAM
C    C4         SLOPE                ) SECOND     ANY          INPUT  PARAM
C    C5         SATURATION SLOPE     ) SLOPE      ANY          INPUT  PARAM
C    C6         SATURATION INTERCEPT)             ANY          INPUT  PARAM
C
      IF(FIN.GT.C3)GO TO 10
      IF(FIN.LT.C6)GO TO 20
      IF(FIN.LT.0.)GO TO 30
      FO=C1*FIN
      GO TO 100
C    POSITIVE SATURATION
 10   FO=C1*C3+C2*(FIN-C3)
      GO TO 100
C    NEGATIVE SATURATION
 20   FO=C4*C6+C5*(FIN-C6)
      GO TO 100
C    NEGATIVE UNSATURATED
 30   FO=C4*FIN
 100  RETURN
      END
```

Table 209: LISTING FOR SUBROUTINE SB

```
CSB
      SUBROUTINE SB(FO,FIN,C1,C2,C3,C4,C5,C6,C7,C8)
C
C    PURPOSE - TO SIMULATE SATURATION WITH A DEAD BAND
C
C
C    METHOD - SEE CODING.
C
C
C    WRITTEN BY - GEORGE DULEBA               LATEST REVISION - FEB 77
C
C    LIMITATIONS - C1 MUST BE POSITIVE
C                  C3 MUST BE GREATER THAN C1
C                  C5 MUST BE POSITIVE
C                  C7 MUST BE GREATER THAN 0.
C
C                  C2 MUST BE NEGATIVE
C                  C4 MUST BE LESS THAN C2
C                  C6 MUST BE NEGATIVE
C                  C8 MUST BE GREATER THAN 0.
C
C
C    INPUT/OUTPUT LIST
C
C    FO          OUTPUT VARIABLE                   ANY       OUTPUT VAR
C    FIN         INPUT VARIABLE                    ANY       INPUT  VAR
C    C1          POSITIVE DEAD BAND ON FIN         ANY       INPUT  VAR
C    C2          NEGATIVE DEAD BAND ON FIN         ANY       INPUT  VAR
C    C3          POSITIVE SATURATION INTERCEPT     ANY       INPUT  VAR
C    C4          NEGATIVE SATURATION INTERCEPT     ANY       INPUT  VAR
C    C5          POSITIVE SATURATION LIMIT ON FO   ANY       INPUT  VAR
C    C6          NEGATIVE SATURATION LIMIT ON FO   ANY       INPUT  VAR
C    C7          SATURATION SLOPE                  ANY       INPUT  VAR
C    C8          SATURATION SLOPE                  ANY       INPUT  VAR
C
C
      IF(FIN.GT.C3) GO TO 50
      IF(FIN.LT.C4) GO TO 60
      SLZERO= .001*C6/C2
      IF(C5.LT.-C6) SLZERO= .001*C5/C1
      YP= SLZERO*C1
      YN= SLZERO*C2
      SLNEG=(C6-YN)/(C4-C2)
      SLPLUS=(C5-YP)/(C3-C1)
      FO =SLZERO*FIN
      IF(FIN.LT.0.) GO TO 40
      IF(FIN.GT.C1) FO=YP+SLPLUS*(FIN-C1)
      GO TO 100
   40 IF(FIN.LT.C2) FO=YN+SLNEG*(FIN-C2)
      GO TO 100
   50 FO=C5+C7*(FIN-C3)
      GO TO 100
   60 FO=C6+C8*(FIN-C4)
  100 RETURN
      END
```

Table 210: LISTING FOR SUBROUTINE SEGMNT

```
CSEGMNT
      SUBROUTINE SEGMNT(ICALL)
C DIVISION OF THE TRUNK INTO SEGMENTS
C
      REAL L,L1,L2,LS,LP,MASS
C
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
      COMMON/SAVE/R1I,R2I,PHI1I,PHI2I,L1I,L2I,A1I,A2I,D2D2HBI,DXAMABI,
     1 BETAD2I,SINPHRI,D2I,SINPH2I,A1MA2,X1I,X2I,X12I,SI
      DATA PI/3.141592653/
C
C IF FIRST CALL,COMPUTE PARTIAL TERMS AND NUMBER SEGMENTS
      IF(ICALL) 20,30,20
30    RLSH=0.5*LS
C BETA IS CURVED SEGMENT ARC ANGLE
      BETA=PI/2./FLOAT(N)
C DELX IS STRAIGHT SEGMENT LENGTH
      DELX=LS/FLOAT(2*M)
      BETA2=1.33333*SIN(BETA/2.)/BETA
C  NUMBERING OF SEGMENTS ACCORDING TO THEIR POSITION IN THE TRUNK
      DO 11 I=1,NSTOP
      IF(I.LE.N)ISEG(I)=1
      IF(I.GT.N.AND.I.LE.N+M)ISEG(I)=2
      IF(I.GT.N+M.AND.I.LE.N+2*M)ISEG(I)=3
      IF(I.GT.N+2*M.AND.I.LE.2*(N+M))ISEG(I)=4
      IF(I.GT.2*(N+M).AND.I.LE.3*N+2*M)ISEG(I)=5
      IF(I.GT.3*N+2*M.AND.I.LE.3*(N+M))ISEG(I)=6
      IF(I.GT.3*(N+M).AND.I.LE.3*N+4*M)ISEG(I)=7
      IF(I.GT.3*N+4*M.AND.I.LE.4*(N+M))ISEG(I)=8
   11 CONTINUE
C
C
C*************************************************************************
C  EVALUATING PROPERTIES OF SEGMENTS
C*************************************************************************
C  ITYP=1 FOR CURVED SEGMENT,=0 FOR STRAIGHT SEGMENT
C  XCX AND ZCX ARE X AND Z COORDINATES RESP. OF THE SEGMENT CENTER
C  XCHI AND ZCHI ARE X AND Z COORDINATES RESP. OF THE CUSHION
C  PRESSURE CENTER FOR A SEGMENT,WHEN IT IS OUT OF GROUND CONTACT
C DELTA IS SEGMENT CENTER ANGLE RELATIVE TO CG
C
20     CONTINUE
      D2=0.5*D+R2*SIN(PHI2)
      DO 10 I=1,NSTOP
      KGO=ISEG(I)
      GO TO (1,2,3,4,5,6,7,8), KGO
C CURVED SEGMENT
C IF NOT INITIAL CALL SKIP CALCULATIONS
1     IF(ICALL)10,100,10
100   ITYP(I)=1
      DELTA(I)=(FLOAT(I-1)+0.5)*BETA
```

```
      COSDEL=COS(DELTA(I))
      XCX(I)=-(RLSH+D2I*COSDEL)
      ZCX(I)=D2I*SIN(DELTA(I))
      XCHI(I)=-(RLSH+D2I*BETA2*COSDEL)
      ZCHI(I)=ZCX(I)*BETA2
      GO TO 10
C STRAIGHT SEGMENT
    2 ITYP(I)=0
      XCX(I)=-RLSH+(FLOAT(I-1-N)+0.5)*DELX
      ZCX(I)=D2
      XCHI(I)=XCX(I)
      ZCHI(I)=ZCX(I)*0.5
      GO TO 10
C STRAIGHT SEGMENT
    3 ITYP(I)=0
      XCX(I)=(FLOAT(I-N-M-1)+0.5)*DELX
      ZCX(I)=D2
      XCHI(I)=XCX(I)
      ZCHI(I)=ZCX(I)*0.5
      GO TO 10
C CURVED SEGMENT
C IF NOT INITIAL CALL SKIP CALCULATIONS
4     IF(ICALL)10,400,10
400     ITYP(I)=1
      DELTA(I)=(FLOAT(I-N-2*M-1)+0.5)*BETA
      SINDEL=SIN(DELTA(I))
      XCX(I)=RLSH+D2I*SINDEL
      ZCX(I)=D2I*COS(DELTA(I))
      XCHI(I)=RLSH+D2I*BETA2*SINDEL
      ZCHI(I)=ZCX(I)*BETA2
      GO TO 10
C CURVED SEGMENT
C IF NOT INITIAL CALL SKIP CALCULATIONS
5     IF(ICALL) 10,500,10
500   ITYP(I)=1
      DELTA(I)=(FLOAT(I-2*N-2*M-1)+0.5)*BETA
      COSDEL=COS(DELTA(I))
      XCX(I)=RLSH+D2I*COSDEL
      ZCX(I)=-D2I*SIN(DELTA(I))
      XCHI(I)=RLSH+D2I*COSDEL*BETA2
      ZCHI(I)=ZCX(I)*BETA2
      GO TO 10
C STRAIGHT SEGMENT
    6 ITYP(I)=0
      XCX(I)=RLSH-(FLOAT(I-3*N-2*M-1)+0.5)*DELX
      ZCX(I)=-D2
      XCHI(I)=XCX(I)
      ZCHI(I)=ZCX(I)*0.5
      GO TO 10
C STRAIGHT SEGMENT
    7 ITYP(I)=0
      XCX(I)=-(FLOAT(I-3*N-3*M-1)+0.5)*DELX
      ZCX(I)=-D2
      XCHI(I)=XCX(I)
      ZCHI(I)=ZCX(I)*0.5
      GO TO 10
```

```
C CURVED SEGMENT
C IF NOT INITIAL CALL SKIP CALCULATIONS
8       IF(ICALL) 10,800,10
800    ITYP(I)=1
       DELTA(I)=(FLOAT(I-3*N-4*M-1)+0.5)*BETA
       SINDEL=SIN(DELTA(I))
       XCX(I)=-(RLSH+D2I*SINDEL)
       ZCX(I)=-D2I*COS(DELTA(I))
       XCHI(I)=-(RLSH+D2I*SINDEL*BETA2)
       ZCHI(I)=ZCX(I)*BETA2
   10 CONTINUE
       RETURN
       END
```

Table 211: LISTING FOR SUBROUTINE SG

```
CSG
      SUBROUTINE SG(U,UD,IU,V,VD,IV,W,WD,IW,P,PD,IP,Q,QD,IQ,
     1 R,RD,IR,ROL,ROLD,IROL,PIT,PITD,IPIT,YAW,YAWD,IYAW,
     2 X,XD,IX,Y,YD,IY,Z,ZD,IZ,PDOT,QDOT,RDOT,
     3 ROLDOT,PITDOT,YAWDOT,XDOT,YDOT,
     4 UDOT,VDOT,WDOT,TX,TY,TZ,XXI,YYI,ZZI,XZI,XYI,YZI)
C   VERSION 5.                     MAY 11 1978
C  PURPOSE   SIX DEGREE OF FREEDOM NONSYMETRIC RIGID BODY EQUATIONS OF MOT
C  METHOD   EULER ANGLES
C  CALL SEQUENCE
C  *****  OUTPUTS  *****
C   LINEAR VELOCITIES  --  BODY AXES
C      U,UD,IU      - X AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C      V,VD,IV      - Y AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C      W,WD,IW      - Z AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C   ANGULAR VELOCITIES --  BODY AXES
C      P,PD,IP      - X AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C      Q,QD,IQ      - Y AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C      R,RD,IR      - Z AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C   EULER ANGLES  --  EARTH TO BODY - YAW,PITCH,ROLL
C      ROL,ROLD,IROL - ROLL ANGLE,RATE,INT CONTROL,DEG
C      PIT,PITD,IPIT - PITCH ANGLE,RATE,INT CONTROL, DEG
C      YAW,YAWD,IYAW - YAW ANGLE,RATE,INT CONTROL, DEG
C   POSITIONS  --  EARTH AXES
C      X,XD,IX      - X AXIS POSITION,VELOCITY,INT CONTROL,FT
C      Y,YD,IY      - Y AXIS POSITION,VELOCITY,INT CONTROL,FT
C      Z,ZD,IZ      - -Z AXIS POSITION (ALT),VELOCITY,INT CONTOL, FT
C   ANGULAR ACCELERATIONS  --  BODY AXES
C      PDOT         - X AXIS ANGULAR ACCELERATION, DEG/SEC2
C      QDOT         - Y AXIS ANGULAR ACCELERATION, DEG/SEC2
C      RDOT         - Z AXIS ANGULAR ACCELERATION, DEG/SEC2
C   EULER ANGULAR RATES  --  EARTH TO BODY AXES
C      ROLDOT       - ROLL ANGLE RATE,DEG/SEC2
C      PITDOT       - PITCH ANGLE RATE,DEG/SEC2
C      YAWDOT       - YAW ANGLE RATE,DEG/SEC2
C   LINEAR VELOCITIES  --  EARTH AXES
C      XDOT         - X AXIS LINEAR VELOCITY,FT/SEC
C      YDOT         - Y AXIS LINEAR VELOCITY,FT/SEC
C ***************    INPUTS   *******************
C   LINEAR ACCELERATIONS  --  BODY AXES
C      UDOT         - X AXIS LINEAR ACCELERATION, FT/SEC2
C      VDOT         - Y AXIS LINEAR ACCELERATION, FT/SEC2
C      WDOT         - Z AXIS LINEAR ACCELERATION, FT/SEC2
C   MOMENTS
C      TX,TY,TZ     - X,Y,Z  AXIS TORQUES, FTLBS
C   MOMENTS OF INERTIA
C      XXI,YYI,ZZI - X,Y,Z AXIS MOMENTS OF INERTIA, SLUG-FT2
C      XZI,XYI,YZI - PRODUCT OF INERTIA, SLUG-FT2
C
C   WRITTEN BY  J.D. BURROUGHS          MAY 1976
C
C     MODIFIED BY   M.K. WAHI         NOV 1977
C
      DATA RPD,DPR /.01745329,57.29578/
      CP=COS(PIT*RPD)
      SP=SIN(PIT*RPD)
```

```
      CR=COS(ROL*RPD)
      SR=SIN(ROL*RPD)
      P1= P*RPD
      Q1= Q*RPD
      R1= R*RPD
C ****************  LINEAR VELOCITY EQUATIONS  *************
      IF(IU.NE.0)UD=UDOT
      IF(IV.NE.0)VD= VDOT
      IF(IW.NE.0)WD= WDOT
C ****************  ANGULAR VELOCITY EQUATIONS  *************
      IF(XZI.EQ. .99999) XZI=0.
      IF(XYI.EQ. .99999) XYI=0.
      IF(YZI.EQ. .99999) YZI=0.
      TXE=TX+YZI*(Q1**2-R1**2)+XZI*P1*Q1-XYI*R1*P1
     1   +(YYI-ZZI)*Q1*R1
      TYE=TY+XZI*(R1**2-P1**2)+XYI*Q1*R1-YZI*P1*Q1
     1   +(ZZI-XXI)*R1*P1
      TZE=TZ+XYI*(P1**2-Q1**2)+YZI*R1*P1-XZI*Q1*R1
     1   +(XXI-YYI)*P1*Q1
      DETI=XXI*(YYI*ZZI-YZI**2)-XYI*(YZI*XZI+ZZI*XYI)
     1   -XZI*(XYI*YZI+YYI*XZI)
      PD=0.
      QD=0.
      RD=0.
      IF(IP.NE.0) PD=DPR*(TXE*(YYI*ZZI-YZI**2)+TYE*(XYI*ZZI
     1               +YZI*XZI)+TZE*(XYI*YZI+YYI*XZI))/DETI
      IF(IQ.NE.0) QD=DPR*(TXE*(XYI*ZZI+YZI*XZI)+TYE*(XXI*ZZI
     1               -XZI**2)+TZE*(XXI*YZI+XYI*XZI))/DETI
      IF(IR.NE.0) RD=DPR*(TXE*(XYI*YZI+YYI*XZI)+TYE*(XXI*YZI
     1               +XYI*XZI)+TZE*(XXI*YYI-XYI**2))/DETI
C ****************  EULER ANGLE EQUATIONS  *************
      IF(IPIT.NE.0)PITD=Q*CR-R*SR
      IF(CP.NE.0.)PSID=(Q*SR+R*CR)/CP
      IF(IYAW.NE.0)YAWD=PSID
      IF(IROL.NE.0)ROLD=P+PSID*SP
C ***************  POSITION EQUATIONS  ***************
C -----      TEST IF X AND Y (LATITUDE AND LONGITUDE) ARE BOTH FROZEN
      SPSR=SP*SR
      SPCR=SP*CR
      CY=COS(YAW*RPD)
      SY=SIN(YAW*RPD)
      IF(IX.NE.0)XD=CY*CP*U+(-SY*CR+CY*SPSR)*V+(SY*SR+CY*SPCR)*W
      IF(IY.NE.0)YD=SY*CP*U+(CY*CR+SY*SPSR)*V+(-CY*SR+SY*SPCR)*W
      IF(IZ.NE.0)ZD=SP*U-CP*SR*V-CP*CR*W
C ****************  ANGULAR ACCELERATIONS  (FOR OUTPUT PURPOSES ONLY) ****
      PDOT=PD
      QDOT=QD
      RDOT=RD
C ***** EULER ANGLE RATES (FOR OUTPUT PURPOSES ONLY) *****
      ROLDOT=ROLD
      PITDOT=PITD
      YAWDOT=YAWD
C ***** LINEAR VELOCITIES IN EARTH AXES (OUTPUT ONLY) *****
      XDOT=XD
      YDOT=YD
      RETURN
```

Table 211: LISTING FOR SUBROUTINE SG (CONCLUDED)

```
END
```

```
CSHAPE1
      SUBROUTINE SHAPE1(ICALL)
C INITIAL ASSESSMENT OF AREAS,VOLUMES ASSUMING
C NO GROUND CONTACT
C
      REAL L,L1,L2,LS,LP,MASS
      REAL L1I,L2I
C
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      COMMON/SHAPE/AGAPI(100),AGAPR(100),ATKI(100),ATKR(100),ATKCHI(100)
     1,ATKCHR(100),ATKATI(100),ATKATR(100),ACHI(100),ACHR(100)
     2,ATKCNI(100),ATKCNR(100),VCHI(100),VCHR(100),VTKI(100),VTKR(100)
     3,XCH(100),XTK(100),ZCH(100),ZTK(100),PERI(100)
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/SAVE/R1I,R2I,PHI1I,PHI2I,L1I,L2I,A1I,A2I,D2D2HBI,DXAMABI,
     1 BETAD2I,SINPHRI,D2I,SINPH2I,A1MA2,X1I,X2I,X12I,SI
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
C COMPUTE GEOMETRY TERMS
      SINPH2=SIN(PHI2)
      SINPHR=SINPH2*R2
      D2=D/2.+SINPHR
      DD2=DELX*D2
      BDD2=BETA*D2*D2*0.5
      X=B*(A-SINPHR)/(B+HY-R1)
C COMPUTE AREAS OF TRUNK SECTORS
      A1=PHI2/2.0*R2**2
      A2=(R2-HY)/2.0*SINPHR
      A3=PHI1/2.0*R1**2
      A4=X*B/2.0
      A5=(A-SINPHR-X)/2.0*(HY-R1)
      X1=SINPHR-4.0*(SIN(PHI2/2.0))**2*R2/(3.0*PHI2)
      X2=0.66667*SINPHR
      X3=SINPHR+4.0*(SIN(PHI1/2.0))**2*R1/(3.0*PHI1)
      X4=A-0.333333*X
      X5=SINPHR+0.333333*(A-SINPHR-X)
      AA=A1+A3+A5-A2-A4
      AX=A1*X1-A2*X2+A3*X3-A4*X4+A5*X5
      IF(ICALL.GT.0) GO TO 20
C SAVE TRUNK GEOMETRY TERMS FOR END TRUNK CALCULATIONS
      S=2.0*LS+6.28318*D2
      R1I=R1
      R2I=R2
      PHI1I=PHI1
      PHI2I=PHI2
      L1I=L1
      L2I=L2
      A1I=A1
      A2I=A2
```

```
      SINPH2I=SINPH2
      SINPHRI=SINPHR
      X1I=X1
      X2I=X2
       A1MA2I=A1-A2
      D2I=D2
      SI=S
      BETAD2I=BETA*D2
      X12=(X1*A1-X2*A2)/A1MA2I
      DXAMABI=(D*0.5+X12)*A1MA2I*BETA
      D2D2HBI=D2*D2*0.5*BETA
      DD2I=DELX*D2
      BDD2I=BETA*D2*D2*0.5
20    CONTINUE
C COMPUTE TRUNK SEGMENT AREA,VOLUME,CUSHION AREA
      DO 10 I=1,NSTOP
      IF(ITYP(I).EQ.1)GO TO 11
C STRAIGHT PART OF TRUNK
13    ATKI(I)=AA
      VTKI(I)=DELX*ATKI(I)
      ACHI(I)=DD2
      GO TO 10
C CURVED PART OF TRUNK
11    IF(ICALL.GT.0) GO TO 10
      ATKI(I)=AA
      XE=AX/ATKI(I)
      VTKI(I)=BETA*(D/2.+XE)*ATKI(I)
      ACHI(I)=BDD2
10      CONTINUE
      RETURN
      END
```

Table 213: LISTING FOR SUBROUTINE SHAPE2

```
CSHAPE2
      SUBROUTINE SHAPE2
C CALCULATION OF AREAS AND VOLUMES ASSOCIATED WITH ACLS,KNOWING ITS
C ORIENTATION
C
      REAL L,L1,L2,LS,LP,MASS
      REAL L1I,L2I
C
      COMMON/ESTMD/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      COMMON/SHAPE/AGAPI(100),AGAPR(100),ATKI(100),ATKR(100),ATKCHI(100)
     1,ATKCHR(100),ATKATI(100),ATKATR(100),ACHI(100),ACHR(100)
     2,ATKCNI(100),ATKCNR(100),VCHI(100),VCHR(100),VTKI(100),VTKR(100)
     3,XCH(100),XTK(100),ZCH(100),ZTK(100),PERI(100)
      COMMON/SPACE/N,M,BETA,DELX,XCX(100),ZCX(100),XG(100),YG(100),
     1ZG(100),SL4(100),YGH(100),ISEG(100),ITYP(100),DELTA(100),XCHI(100)
     2,ZCHI(100),XCG
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/SAVE/R1I,R2I,PHI1I,PHI2I,L1I,L2I,A1I,A2I,D2D2HBI,DXAMABI,
     1 BETAD2I,SINPHRI,D2I,SINPH2I,A1MA2,X1I,X2I,X12I,SI
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
C
      DIMENSION ATKRB(100),VTKRA(100),VTKRB(100)
C
      DATA PI/3.141592653/
C
C CALL SHAPE1 TO GET TRUNK SIDE SHAPES
      CALL SHAPE1(1)
C COMPUTE PARTIAL TERMS
       SINPH2=SIN(PHI2)
      SINPHR=SINPH2*R2
      D2=D*0.5+SINPHR
      D2ISQ=D2I*D2I
      A1MA2=A1-A2
      BETA2=1.33333*SIN(BETA/2.)/BETA
      BETAD2=BETA*D2
      RLSH=LS*0.50
      ADS=AH*DELX/SI
      ABDSI=AH*BETA*D2I/SI
      RNN=FLOAT(NR*NH)
C***********************************************************************=
C PART 1                    I VALUE OF VCH AND AGAP
C***********************************************************************=
      DO 17 I=1,NSTOP
C TEST FOR TRUNK SEGMENT, WHETHER CURVED OR STRAIGHT
      IF(ITYP(I).EQ.1)GO TO 11
C
C STRAIGHT PART OF TRUNK
C
C CALCULATE CUSHION SEGMENT INITIAL VOLUME
13     VCHI(I)=(YGH(I)*D2-A1MA2)*DELX
```

```
C CALCULATE SEGMENT GAP AREA
      AGAPI(I)=(YGH(I)-HY)*DELX
      GO TO 10
C
C CURVED PART OF TRUNK
C
11    VCHI(I)=YGH(I)*D2D2HBI-DXAMABI
      AGAPI(I)=(YGH(I)-HYI)*BETAD2I
C**********************************************************************
C PART 2                    R VALUE CALCULATIONS
C**********************************************************************
C
C TEST FOR GROUND CONTACT AT EACH SEGMENT
10    CONTINUE
C FORCE VOLUME AREAS .GE.0
      VCHI(I)=AMAX1(0.0,VCHI(I))
      AGAPI(I)=AMAX1(0.0,AGAPI(I))
C TEST SEGMENT FOR CONTACT
      IF(ITYP(I).EQ.1.AND.YGH(I).LE.HYI) GO TO 14
      IF(ITYP(I).EQ.0.AND.YGH(I).LE.HY) GO TO 23
C
C NO GROUND CONTACT
C SET CONTACT AND REMOVE TERMS TO ZERO
      ATKR(I)=0.0
      ACHR(I)=0.0
      VTKR(I)=0.0
      VCHR(I)=0.0
      AGAPR(I)=0.0
      ATKCNI(I)=0.0
      ATKCNR(I)=0.0
      ATKCHR(I)=0.0
      ATKATR(I)=0.0
      PERI(I)=0.0
C SET DISTANCES X,Z TO FREE TRUNK VALUES
      XCH(I)=XCHI(I)
      ZCH(I)=ZCHI(I)
      ZTK(I)=ZCH(I)
      XTK(I)=XCH(I)
C COMPUTE TRUNK-CUSHION-ATMOSPHERE BLEED AREAS
      IF(ITYP(I)) 16,16,18
16    CONTINUE
C NO CONTACT STRAIGHT SECTIONS
      ATKCHI(I)=FLOAT(IFIX((L2-LP)/SH+1.0)*NH)*ADS
      ATKATI(I)=RNN*ADS-ATKCHI(I)
      GO TO 17
18    CONTINUE
C NO CONTACT CURVED SECTIONS
      ATKCHI(I)=FLOAT(IFIX((L2I-LP)/SH+1.0)*NH)*ABDSI
       ATKATI(I)=RNN*ABDSI-ATKCHI(I)
       GO TO 17
C**********************************************************************
C
C TRUNK GROUND CONTACT
C
C CURVED PART OF TRUNK
C CALCULATE DEFORMATION ANGLES FOR SEGMENT
```

```
14    PHI3=ACOS((R2I-(HYI-YGH(I)))/R2I)
      PHI4=ACOS((R1I-(HYI-YGH(I)))/R1I)
      SINPH3=SIN(PHI3)
      SINPH4=SIN(PHI4)
C COMPUTE PARTIAL TERMS
      DRSP=(D2I-R2I*SINPH3)
      DRSP2=DRSP*DRSP
      COSDEL=COS(DELTA(I))
      SINDEL=SIN(DELTA(I))
      BEDRSN=BETA2*DRSP*SINDEL
      BEDRCS=BETA2*DRSP*COSDEL
C COMPUTE REMOVAL SECTORS
      A6=R2I*R2I*PHI3*0.5
      A7=(R2I-HYI+YGH(I))*0.5*R2I*SINPH3
      A6MA7=A6-A7
      A8=R1I*R1I*PHI4*0.5
      A9=(R1I-HYI+YGH(I))*0.5*R1I*SINPH4
      A10=A9
      A11=A8
C COMPUTE SECTOR CENTROIDS
      X6=SINPHRI-1.333333*(SIN(PHI3*0.5)**2)*R2I/PHI3
      X7=SINPHRI-0.333333*R2I*SINPH3
      X8=SINPHRI+1.3333333*(SIN(PHI4*0.5)**2)*R1I/PHI4
      X9=SINPHRI+0.3333333*R1I*SINPH4
      X10=X9
      X11=X8
      PII2=PI*0.5
      IF(PHI4.LT.PII2) GO TO 50
C IF PHI4 GREATER THAN 90 DEGREES, SET TO 90 DEGREES
      PHI4=PII2
      SINPH4=SIN(PHI4)
      A10=(R1I-HYI+YGH(I))*R1I
      X10=SINPHRI+0.5*R1I
      A11=R1I*R1I*PHI4*0.5
      X11=SINPHR+1.333333*(SIN(PHI4*0.5)**2)*R1I/PHI4
   50 CONTINUE
C COMPUTE TRUNK AREA CHANGE
      ATKR(I)=A6MA7+A8-A9
      ATKRB(I)=A6MA7+A11-A10
      XER=(A6*X6-A7*X7+A8*X8-A9*X9)/ATKR(I)
      XERB=(A6*X6-A7*X7+A11*X11-A10*X10)/ATKRB(I)
C COMPUTE TRUNK VOLUME CHANGE
      VTKRA(I)=BETA*(D*0.5+XER)*ATKR(I)
      VTKRB(I)=ATKRB(I)*BETA*(D*0.5+XERB)
      VTKR(I)=2.*VTKRA(I)-VTKRB(I)
      XCP=(A6*X6-A7*X7)/A6MA7
C COMPUTE TRUNK EXIT AREAS
      ATKCHI(I)=FLOAT(IFIX((L2I-LP-R2I*PHI3)/SH+1.0)*NH)*ABDSI
      ATKATI(I)=FLOAT(IFIX((L1I-L+LP+FLOAT(NR-1)*SH-R1I*PHI4)/SH+1.0)*NH
     1)*ABDSI
      ATKCHR(I)=FLOAT(IFIX((L2I-LP)/SH+1.0)*NH)*ABDSI-ATKCHI(I)
      ATKATR(I)=RNN*ABDSI-ATKCHI(I)-ATKATI(I)-ATKCHR(I)
C COMPUTE CONTACT PERIMETER
      PERI(I)=BETA*(DRSP+D2I+R1I*SINPH4)
C COMPUTE TRUNK CONTACT AREA
      ATKCNI(I)=BETA*0.5*(D2ISQ-DRSP2)
```

```
      ATKCNR(I)=BETA*0.5*((D2I+R1I*SINPH4)**2-D2ISQ)
      ACHR(I)=ATKCNI(I)
C COMPUTE CUSHION VOLUME CHANGE
      VCHR(I)=-BETA*A6MA7*(D*0.5+XCR)
C COMPUTE GAP AREA CHANGE
      AGAPR(I)=AGAPI(I)
C DISTANCE OF SEGMENT PRESSURE CENTERS FROM CUSHION CENTER
   29 RR=D2I+R1I*SINPH4
      RR1=D2I-R2I*SINPH3
      XX2=1.333333*SIN(BETA*0.5)/BETA*(RR**3-RR1**3)/(RR*RR-RR1*RR1)
      KGO=ISEG(I)
      GO TO (61,23,23,64,65,23,23,68),KGO
61    XCH(I)=-RLSH-BEDRCS
      ZCH(I)=BEDRSN
      XTK(I)=-RLSH-XX2*COSDEL
      ZTK(I)=XX2*SINDEL
      GO TO 17
64    XCH(I)=RLSH+BEDRSN
      ZCH(I)=BEDRCS
      XTK(I)=RLSH+XX2*SINDEL
      ZTK(I)=XX2*COSDEL
      GO TO 17
65    XCH(I)=RLSH+BEDRCS
      ZCH(I)=-BEDRSN
      XTK(I)=RLSH+XX2*COSDEL
      ZTK(I)=-XX2*SINDEL
      GO TO 17
68    XCH(I)=-RLSH-BEDRSN
      ZCH(I)=-BEDRCS
      XTK(I)=-RLSH-XX2*SINDEL
      ZTK(I)=-XX2*COSDEL
      GO TO 17
C*****************************************************************************
C
C TRUNK GROUND CONTACT
C STRAIGHT PART OF TRUNK
23    CONTINUE
C COMPUTE DEFORMATION ANGLES
      RHY=((R2-(HY-YGH(I)))/R2)
      PHI3=ACOS(AMAX1(-1.0,AMIN1(1.0,RHY)))
      RHY=((R1-(HY-YGH(I)))/R1)
      PHI4=ACOS(AMAX1(-1.0,AMIN1(1.0,RHY)))
C DO TRANSCENDENTALS ONLY ONCE
      SINPH3=SIN(PHI3)
      SINPH4=SIN(PHI4)
C COMPUTE PARTIAL TERMS
      DRSP=(D2-R2*SINPH3)
      DRSP2=DRSP*DRSP
      COSDEL=COS(DELTA(I))
      SINDEL=SIN(DELTA(I))
      BEDRSN=BETA2*DRSP*SINDEL
      BEDRCS=BETA2*DRSP*COSDEL
C COMPUTE REMOVAL SECTORS
      A6=R2*R2*PHI3*0.5
      A7=(R2-HY+YGH(I))*0.5*R2*SINPH3
      A6MA7=A6-A7
```

```
      A8=R1*R1*PHI4*0.5
      A9=(R1-HY+YGH(I))*0.5*R1*SINPH4
      A10=A9
      A11=A8
C COMPUTE SECTOR CENTROIDS
      X6=SINPHR-1.333333*(SIN(PHI3*0.5)**2)*R2/PHI3
      X7=SINPHR-0.333333*R2*SINPH3
      X8=SINPHR+1.3333333*(SIN(PHI4*0.5)**2)*R1/PHI4
      X9=SINPHR+0.3333333*R1*SINPH4
      X10=X9
      X11=X8
      PII2=PI*0.5
      IF(PHI4.LT.PII2) GO TO 70
C IF PHI4 IS GREATER THAN 90 DEGREES, SET TO 90 DEGREES
      PHI4=PII2
      SINPH4=SIN(PHI4)
      A10=(R1-HY+YGH(I))*R1
      X10=SINPHR+0.5*R1
      A11=R1*R1*PHI4*0.5
      X11=SINPHR+1.3333333*(SIN(PHI4*0.5)**2)*R1/PHI4
70    CONTINUE
C COMPUTE TRUNK AREA CHANGE
      ATKR(I)=A6MA7+A8-A9
      ATKRB(I)=A6MA7+A11-A10
      VTKRA(I)=ATKR(I)*DELX
      VTKRB(I)=ATKRB(I)*DELX
      VTKR(I)=2.*VTKRA(I)-VTKRB(I)
C COMPUTE CUSHION VOLUME CHANGE
      VCHR(I)=-DELX*A6MA7
C COMPUTE TRUNK EXIT AREAS
      ATKATI(I)=FLOAT(IFIX((L1-L+LP+FLOAT(NR-1)*SH-R1*PHI4)/SH+1.0)*NH)
     1*ADS
      ATKCHI(I)=FLOAT(IFIX((L2-LP-R2*PHI3)/SH+1.)*NH)*ADS
      ATKCHR(I)=FLOAT(IFIX((L2-LP)/SH+1.0)*NH)*ADS-ATKCHI(I)
      ATKATR(I)=RNN*ADS-ATKCHI(I)-ATKATI(I)-ATKCHR(I)
C COMPUTE TRUNK CONTACT AREA
      ATKCNI(I)=R2*SINPH3*DELX
      ATKCNR(I)=R1*SINPH4*DELX
C COMPUTE TRUNK CONTACT PERIMETER
C COMPUTE TRUNK VOLUME CHANGE
      PERI(I)=2.*DELX
C COMPUTE GAP AREA CHANGE
      ACHR(I)=ATKCNI(I)
       AGAPR(I)=AGAPI(I)
      KGO=ISEG(I)
C COMPUTE SEGMENT CONTACT CENTER OF PRESSURE FOR CUSHION AND TRUNK
      GO TO (17,62,62,17,17,66,66,17),KGO
   62 XCH(I)=XCX(I)
      ZCH(I)=0.5*(D2-R2*SINPH3)
      XTK(I)=XCX(I)
      ZTK(I)=D2+0.5*(R1*SINPH4-R2*SINPH3)
      GO TO 17
   66 XCH(I)=XCX(I)
      ZCH(I)=-0.5*(D2-R2*SINPH3)
      XTK(I)=XCX(I)
      ZTK(I)=-(D2+0.5*(R1*SINPH4-R2*SINPH3))
```

Table 213: LISTING FOR SUBROUTINE SHAPE2 (CONCLUDED)

```
   17 CONTINUE
C*****************************************************************************
C PART 3                    SUMMATION OF SEGMENT AREAS VOLUMES
C*****************************************************************************
C
C SET TOTAL AREA AND VOLUMES TO ZERO
      ATKCN=0.0
       VTK=0.0
      ACH=0.0
      ATKCH=0.0
      ATKAT=0.0
      VCH=0.0
      AGAP=0.0
      ATKATC=0.
      ATKCHC=0.
C LOOP ON SEGMENTS TO FIND TOTALS OF AREAS AND VOLUMES.
      DO 30 I=1,NSTOP
      VTK=VTK+    (VTKI(I)-VTKR(I))
      ACH=ACH+    (ACHI(I)-ACHR(I))
      ATKCH=ATKCH+ATKCHI(I)
      ATKAT=ATKAT+ATKATI(I)
      VCH=VCH+    (VCHI(I)-VCHR(I))
      ATKCN=ATKCN+ATKCNI(I)+ATKCNR(I)
      AGAP=AGAP+    (AGAPI(I)-AGAPR(I))
      ATKATC=ATKATC+ATKATR(I)
      ATKCHC=ATKCHC+ATKCHR(I)
30    CONTINUE
      AGAP=AMAX1(AGAP,0.0)
      VTK=AMAX1(0.0,VTK)
      VCH=AMAX1(0.0,(VCH+VCHO))
      VCH=AMAX1(0.0,VCH)
      ATKCH=AMAX1(0.000,ATKCH)
      ATKAT=AMAX1(0.000,ATKAT)
      ACH=AMAX1(0.0,ACH)
      ATKATC=AMAX1(0.0,ATKATC)
      ATKCHC=AMAX1(0.0,ATKCHC)
      RETURN
      END
```

Table 214: LISTING FOR SUBROUTINE SHCP

```
CSHCP
      FUNCTION SHCP(T,SH)
C
C    PURPOSE - TO CALCULATE SPECIFIC HEAT OF MOIST AIR (AIR + VAPOR)
C
C
C    METHOD  - SPECIFIC HEAT OF DRY AIR FROM KEENAN AND KAYE, USING
C            - POLYNOMIAL  APPROXIMATIONS
C            - SPECIFIC HEAT OF VAPOR FROM KEENAN AND KEYES, TAKEN AT
C            - 0.2 PS1 PARTIAL PRESSURE
C
C
C    WRITTEN BY  -   ADAM LLOYD          LATEST REVISION     NOV 75
C
C
C    LIMITATIONS -   TEMPERATURE 300-1600 DEGR
C
C    INPUT/OUTPUT LIST
C
C    SHCP        SPECIFIC HEAT OF MOIST AIR              BTU/LB DEGR   OUTPUT
C    T           TEMPERATURE                             DEGR          INPUT
C    SH          SPECIFIC HUMIDITY                       LB/LB         INPUT
C
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      X=T/1000.
      X=AMAX1(AMIN1(X,1.6),0.4)
      CP=((-.01493056*X+.05767857)*X-.04204563)*X+.2478786
      SHCP= (CP + .46*SH)/(1.+SH)
C    TEST IF AIR TEMPERATURES ARE WITHIN VALID RANGE
      IF(T.GE.300.AND.T.LE.1600.)GO TO 100
C    TEST FOR DIAGNOSTIC PRINT OUT
      IF(IERR.NE.1)GO TO 100
      WRITE(IWRITE,9999)
 9999 FORMAT(10X,32HNON FATAL ERROR CALLED FROM SHCP/
     1 10X,34HTEMPERATURE NOT WITHIN VALID RANGE)
  100 RETURN
      END
```

Table 215: LISTING FOR SUBROUTINE SIDEFS

```
CSIDEFS
      SUBROUTINE SIDEFS(Y,M,K,R,P)
C
C
C     VERSION 2                               REVISED MARCH 1979
C
C     WRITTEN BY - GS DULEBA
C
C     PURPOSE - TO COMPUTE THE SHAPE PARAMETERS FOR AN ELASTIC
C               TRUNK SIDE ELEMENT IN THE FREE CONFIGURATION.
C
C     METHOD - MERIDIAN SHAPE IS ASSUMED TO BE TWO CONNECTED
C              CIRCULAR ARCS(INNER AND OUTER). MERIDIAN LOAD
C              IS COMPUTED USING MEMBRANE THEORY AND
C              ASSUMED SHAPE.  MERIDIAN STRAIN IS CALCULATED
C              FROM LOAD/DEFLECTION CURVE AND MUST BE COMPATIBLE
C              WITH ASSUMED SHAPE TO BE A VALID SOLUTION.
C
C     NOMENCLATURE
C       R1     RADIUS OF OUTER ARC
C       PH1    SWEPT ANGLE OF OUTER ARC
C       R2     RADIUS OF INNER ARC
C       PH2    SWEPT ANGLE OF OUTER ARC
C       NP     MERIDIAN LOAD
C       EP     MERIDIAN STRAIN
C
C
      COMMON/ELAST/L2,NPH,DUM(7),EP,D
      REAL L1,L2,L10,L20,NP,NT,NEP
      DIMENSION Y(1),R(1),P(1)
      DATA PI2/6.283185/, THC/6./
C
      R1=Y(1)
      PH1=Y(2)
      PH2=Y(3)
      R2=R1*P(6)/P(5)
C
      S1=SIN(PH1)
      C1=COS(PH1)
      S2=SIN(PH2)
      C2=COS(PH2)
C
      XL1=0.
      IF(PH1.GT.THC) XL1=.5/(THC-PI2)*(PH1-PI2)*(PH1-PI2)
      L1=R1*(PH1+1000.*XL1)
      L2=R2*PH2
C
C     CALCULATE MERIDIAN LOAD AND STRAIN
      NP=P(6)*R1
      NT=NP*P(9)
      NEP=NP-P(9)*NT
      EP=TBLU1(NEP,P(12),P(NPH+12),1,NPH)
C
C     COMPUTE DEFLATED LENGTHS FOR L1 AND L2
      L10=(1.+P(7))*L1/(1.+EP)
      L20=(1.+P(7))*L2/(1.+EP)
```

Table 215: LISTING FOR SUBROUTINE SIDEFS (CONCLUDED)

```
C
C     COMPUTE RESIDUALS (=0. FOR SOLUTION)
      R(1)= P(4)-L10-L20
      R(2)= P(2)-R1*S1-R2*S2
      R(3)= P(3)-R1*(1.-C1)+R2*(1.-C2)
      P(11)=R2*(1.-C2)
C
      RETURN
      END
```

Table 216: LISTING FOR SUBROUTINE SIDELS

```
CSIDELS
      SUBROUTINE SIDELS(Y,M,K,R,P)
C
C
C    VERSION 2                          REVISED MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO COMPUTE THE SHAPE PARAMETERS FOR AN ELASTIC
C              TRUNK SIDE ELEMENT IN THE LOADED CONFIGURATION.
C
C    METHOD - MERIDIAN SHAPE IS ASSUMED TO BE TWO CONNECTED
C             CIRCULAR ARCS(INNER AND OUTER). MERIDIAN LOAD
C             IS COMPUTED USING MEMBRANE THEORY AND
C             ASSUMED SHAPE.  MERIDIAN STRAIN IS CALCULATED
C             FROM LOAD/DEFLECTION CURVE AND MUST BE COMPATIBLE
C             WITH ASSUMED SHAPE TO BE A VALID SOLUTION.
C
C    NOMENCLATURE
C      R1      RADIUS OF OUTER ARC
C      PH1     SWEPT ANGLE OF OUTER ARC
C      R2      RADIUS OF INNER ARC
C      PH2     SWEPT ANGLE OF OUTER ARC
C      L3      MERIDIAN LENGTH IN CONTACT WITH GROUND
C      NP      MERIDIAN LOAD
C      EP      MERIDIAN STRAIN
C
C
       COMMON/ELAST/L2,NPH,DUM(7),EP,D
       REAL L1,L2,L3,L10,L20,L30,NP,NT,NEP
       DIMENSION Y(1),R(1),P(1)
       DATA PI2/6.283185/, THC/6./
C
       R1=Y(1)
       PH1=Y(2)
       PH2=Y(3)
       R2=R1*P(6)/P(5)
       L3=Y(4)
C
       S1=SIN(PH1)
       C1=COS(PH1)
       S2=SIN(PH2)
       C2=COS(PH2)
C
       XL1=0.
       IF(PH1.GT.THC) XL1=.5/(THC-PI2)*(PH1-PI2)*(PH1-PI2)
       L1=R1*(PH1+1000.*XL1)
       L2=R2*PH2
C
C    COMPUTE MERIDIAN LOAD AND STRAIN
      NP=P(6)*R1
      NT=P(9)*NP
      NEP=NP-P(9)*NT
      EP=TBLU1(NEP,P(12),P(12+NPH),1,NPH)
C
C    COMPUTE DEFLATED LENGTHS FOR L1,L2 AND L3
```

Table 216: LISTING FOR SUBROUTINE SIDELS (CONCLUDED)

```
      L10=(1.+P(7))*L1/(1.+EP)
      L20=(1.+P(7))*L2/(1.+EP)
      L30=(1.+P(7))*L3/(1.+EP)
C
C    COMPUTE RESIDUALS (=0. FOR SOLUTION)
      R(1)= P(4)-L10-L20-L30
      R(2)= P(2)-R1*S1-R2*S2-L3
      R(3)= P(3)-R1*(1.-C1)+R2*(1.-C2)
      R(4)= P(11)-R2*(1.-C2)
C
      RETURN
      END
```

Table 217: LISTING FOR SUBROUTINE STATIC

```
CSTATIC
      SUBROUTINE STATIC(ICASE)
C STATIC CHARACTERISTICS SUBROUTINE
C
      REAL L,L1,L2,LS,LP,MASS
C
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      COMMON/COMPRS/AL0,AL1,AL2,AL3,AL4,G0,G1,G2,G3,G4,QP1
      COMMON/LOADS/FORCNS(3,10),CCS(3,10),YCGS(3,10),PHIS(3,10),AGAPS(3,
     110),PCHS(3,10),QFANS(3,10),PFANS(3,10),PPLMS(3,10),THIS(3,10),
     1FFS(3,10),TORXS(3,10),TORZS(3,10),PTKS(3,10)
      COMMON/STATIC/YSTRT,YSTOP,PSTRT,PHSTP,TSTRT,TSTOP
     1,PHIYC,THEYC,YCPHI,YCTHE
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      COMMON/VEHCL/MASS,AIX,AIZ,AIXY,AIYZ,AIZX,CC,GG,FF,AIFAN
      COMMON/DYNAMIC/TIME,FTIME,DTIME,IQ,NQ,DVCH,DVTK,VELX,DERY(13)
     1,DVCHP
      COMMON/FORTQ/FCP,FTP,FORCT,FDF,FORCEY,TCPX,TTPX,TORQTX,TDFX,
     1TORQUEX,TCPZ,TTPZ,TORQTZ,TDFZ,TORFZ,TORQUEZ
      COMMON/COEFFS/CPA,CAF,CPC,CPT,CTC,CGAP,CTA,CVENT,CKK
      COMMON/FLUID/QFAN,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
C
      DIMENSION ZCC(3,3,3),ZWT(3,3,3),ZFF(3,3,3)
      DIMENSION AGAPP(10)
      DIMENSION PFSAV(10)
      DIMENSION HYTEST(3),HYPRES(3),HYERROR(3)
C
C THIS SUBROUTINE CONSISTS OF FOUR NESTED ITERATION LOOPS
C
C....ITERATION 1
C      VALUES OF YCG,PHIE AND THETAE ARE ITERATED.
C....ITERATION 2
C      VALUE OF HY IS ITERATED SO THAT THE ASSUMED HY MATCHES THE VALUE
C      REQUIRED BY FUNCTIONAL RELATIONSHIP HY/HYI=F(PCH/PTK)
C....ITERATION 3
C     VALUE OF PFAN IS ITERATED SO THAT THE VALUE OF AGAP REQUIRED
C     BY PRESSURE FLOW RELATIONS MATCH AGAP GENERATED BY YCG ,PHIE,
C     THETAE,AND HY
C....ITERATION 4
C      VALUE OF PCH IS ITERATED SO THAT THE ASSUMED VALUE
C      MATCHES THE VALUE REQUIRED BY THE PRESSURE RELATIONSHIP
C ALL FOUR  ITERATIONS INVOLVE CHOOSING INITIAL VALUES,
C TESTING THE GENERATED VARIABLE AND SELECTING NEXT ITERATIVE VALUES
C THE USER MAY HAVE TO CHANGE NITE OR XTOL
C IF SO INDICATED BY ERROR MESSAGE
C  THE SUBROUTINE USES THESE ITERATIONS IN TWO WAYS
C   (1) GENERATE STATIC LOAD MAP
C      IN THIS PART THE VALUES OF YCG, PHIE AND THETAE IN ITERATIONS
C      ARE USER SPECIFIED
```

Table 217: LISTING FOR SUBROUTINE STATIC (CONTINUED)

```
C    (2) OBTAIN STATIC EQUILIBRIUM CONDITIONS
C      IN THIS PART THE VALUES OF YCG, PHIE AND THETAE ARE ITERATED,
C      ACCORDING TO A CONVERGING ALGORITHM, SO THAT THE CONFIGURATION
C      GENERATES ENOUGH FORCE AT CORRECT DISTANCE TO BALANCE THE
C      WEIGHT OF THE SYSTEM
C***********************************************************************
C PARAMETERS FOR ITERATION
      DATA RADIAN/57.2957795/
      TIRHO=2.0/RHO
      GMASS=32.2*MASS
C********
C INITIALIZE VARIABLES
      XCG=0.0
      SINKRT=0.
      DTHETA=0.
      DPHI=0.
      SIE=0.
      VELX=0.
C********
C TOLERENCES FOR ITERATION
      PTOL=0.5
      HTOL=HYI*0.005
      ATOL=0.003
      XTOL=0.05
      NITE=10
C********
C SAVE INPUT VALUES OF CC,FF
      CCI=CC
      FFI=FF
      GGI=GG
C********
C INITIALIZE FLAGS
      IPAS=0
      IODIN=0
      ICON=1
      ICREST=1
C*********
C SET TESTING TOLERANCE INDEX BASED ON GEOMETRY
      ICGF=1
      IF(ABS(CC).GE.0.001.AND.ABS(FF).LE.0.001)ICGF=2
      IF(ABS(CC).LE.0.001.AND.ABS(FF).GE.0.001)ICGF=3
      IF(ABS(CC).LE.0.001.AND.ABS(FF).LE.0.001)ICGF=4
C***********************************************************************
C INITIAL VALUE,ITERATION 1
C SET BOUNDARIES FOR STATIC LOAD MAP PART
C
      NYCG=10
      NPHI=1
      NTHET=1
C SET YCG ITERATION BOUNDARIES
      YCGSTOP=(HYI+GG)*0.9
      YCGSTRT=(HYI+GG)*1.1
C SET PITCH ANGLE ITERATION BOUNDARIES
      PHISTRT=PHIYC
      PHISTOP=0.0
C SET ROLL ANGLE ITERATION BOUNDARIES
```

```
      THSTRT=THEYC
      THSTOP=0.0
C IF INPUT BOUNDARIES ZERO USE DEFAULT VALUES
      IF((YSTRT.EQ.0.0).AND.(YSTOP.EQ.0.0)) GO TO 13
      YCGSTRT=YSTRT
      YCGSTOP=YSTOP
C**********************************************************************
C**********************************************************************
C ITERATION 1 BEGINS. YCG,PHIE AND THETAE ITERATED
C**********************************************************************
C**********************************************************************
13    CONTINUE
      PHIDELT=0.0
      THDELT=0.0
      YCGDELT=(YCGSTOP-YCGSTRT)/9.0
10    CONTINUE
C COMPUTE TEST POINT VARIABLE DELTAS
      IF(IPAS.LT.4) GO TO 12
      YCGDELT=(YCGSTOP-YCGSTRT)*0.5
      PHIDELT=(PHISTOP-PHISTRT)*0.5
      THDELT=(THSTOP-THSTRT)*0.5
12    CONTINUE
      IPAS=IPAS+1
      IDID=1
C THETAE LOOP
      THETAE=THSTRT
       IIT=0
       DO 400 ITH=1,NTHET
C PHIE LOOP
      PHIE=PHISTRT
      DO 200 IPHI=1,NPHI
C YCG LOOP
      YCG=YCGSTRT
      DO 300 IYCG=1,NYCG
      IIT=IIT+1
      CC=0.
       FF=0.
   11 IF(ICREST.EQ.2)CC=ZCC(IS,JS,KS)
      IF(ICREST.EQ.2)FF=ZFF(IS,JS,KS)
      IDEX=0
       IFLAG=0
C
C *********************************************************************
C *********************************************************************
C ITERATION 2 BEGINS.  HY ITERATED
C *********************************************************************
C *********************************************************************
      HYSTRT=0.9*HYI
      HYSTOP=HYI
16    CONTINUE
      IDON=1
      HYSTRT=AMAX1(HYSTRT,(HYI*0.5))
      HYSTOP=AMIN1(HYI,HYSTOP)
      HYDELT=(HYSTOP-HYSTRT)*C.5
      HY=HYSTRT
      DO 15 IHY=1,3
```

```
      HYTEST(IHY)=HY
C CALL GEOMETRY SUBROUTINES
      CALL TRUNK(ISHAPE)
      IF(ISHAPE.EQ.0)ICASE=0
      IF(ISHAPE.EQ.0)RETURN
      CALL SEGMNT(1)
      CALL COORDN
      CALL PROFILE
      CALL CLRNCE
      CALL SHAPE2
C TEST IF EXHAUST AREAS CLOSED
      PCH=QP1
      PTK=QP1
      IF(ATKCH.LT.0.000001)GO TO 601
      IF(ATKAT.LT.0.000001)GO TO 601
      IF(AGAP.LT.0.000001)GO TO 601
      IPN=1
      IPREST=1
      IRST=0
      ICASE=1
C***********************************************************************
C INITIAL VALUE, ITERATION 3
C SET FAN PRESSURE ITERATION BOUNDARIES
      PSTART=QP1/2.
      PSTOP=QP1
C
C***********************************************************************
C***********************************************************************
C ITERATION 3 BEGINS,PFAN ITERATED
C***********************************************************************
C***********************************************************************
24     PINC1=(PSTOP-PSTART)/80.0
25     PINC=(PSTOP-PSTART)/9.0
      PFAN=PSTART
      DO 100 I=1,10
      IF(IPREST.EQ.2) PFAN=PFSAV(ICASE)
C
C CALL FAN MODEL TO FIND QFAN
      CALL FMFAN
C
C FAN INLET PRESSURE
      IF(AATFN.GE.1.0)PATFN=0.0
      IF(AATFN.GE.1.0)GO TO 259
      PATFN=-RHO/2.0*(QFAN/(AATFN*CAF))**2
  259 CONTINUE
C PLENUM PRESSURE
      PPLM=PFAN+PATFN
       IF(PPLM.LT.0.0) GO TO 35
      SIGN=1.0
C PLENUM TO ATMOSPHERE FLOW
      QPLAT=APLAT*CPA*SQRT(TIRHO*ABS(PPLM))*SIGN
C FAN TO PLENUM FLOW
       QFNPL=QFAN-QPLAT
C***********************************************************************
C INITIAL VALUE, ITERATION 4
      PCH=PPLM/2.
```

Table 217: LISTING FOR SUBROUTINE STATIC (CONTINUED)

```
C
C*****************************************************************************
C*****************************************************************************
C ITERATION 4 BEGINS,PCH ITERATED
C*****************************************************************************
C*****************************************************************************
       DO 50 KK=1,20
C ITERATION OF CUSHION PRESSURE
       SIGN=1.0
       IF(PCH.GT.PPLM)SIGN=-1.0
C PLENUM TO CUSHION FLOW
      QPLCH=APLCH*CPC*SQRT(TIRHO*ABS(PPLM-PCH))*SIGN
C PLENUM TO TRUNK FLOW
       QPLTK=QFNPL-QPLCH
       SIGN=1.0
       IF (QPLTK.LT.0.0) SIGN=-1.0
C TRUNK PRESSURE
      PTK=PPLM-RHO/2.0*(QPLTK/(APLTK*CPT))**2  *SIGN
      SIGN=1.0
       IF(PTK.LT.0.0) SIGN=-1.0
C TRUNK TO ATMOSPHERE FLOW
      QTKAT=SIGN*CTA*SQRT(TIRHO*ABS(PTK))*(ATKAT+0.66667*ATKATC)
C TRUNK TO CUSHION FLOW
      QTKCH=QPLTK-QTKAT
      SIGN=1.0
      IF(QTKCH.LT.0.0) SIGN=-1.0
C CUSHION PRESSURE
      PCHI=PTK-RHO/2.0*(QTKCH/((ATKCH+0.66667*ATKCHC)*CTC))**2*SIGN
C*****************************************************************************
C TESTING VARIABLE, ITERATION 4
       IF(ABS(PCH-PCHI).LE.PTOL)GO TO 75
C
C*****************************************************************************
C SELECTING NEW VALUE, ITERATION 4
       PCH=(PCHI+PCH)/2.0
C
C ITERATION CONTINUED
50       CONTINUE
C*****************************************************************************
C*****************************************************************************
C ITERATION 4 ENDS
C*****************************************************************************
C*****************************************************************************
C INFEASIBLE CONFIGURATION. CHOOSE NEXT VALUE OF PFAN
35      PSTART=PSTART+PINC1
        IRST=IRST+1
        IRTT=IRST-300
       IF(IRTT.GT.0)IRST=0
       IF(IRTT)41,41,601
   41  PDRF=PSTOP-1.5*PINC1
       IF(PDRF.LT.PSTART) GO TO 24
       GO TO 25
   75  IF(PTK.LT.PCH.AND.I.LE.2)GO TO 35
       IF(PTK.LT.PCH)PSTOP=PFAN-PINC
        IF(PTK.LT.PCH)GO TO 25
        IF(PCH.LT.0.0) GO TO 35
```

Table 217: LISTING FOR SUBROUTINE STATIC (CONTINUED)

```
C CUSHION TO ATMOSPHERE FLOW
      QCHAT=QPLCH+QTKCH
      IF(IPREST.EQ.2)GO TO 600
C COMPUTE ITERATED GAP AREA
      AGAPP(I)=QCHAT/(CGAP*SQRT(TIRHO*PCH))
      PFSAV(I)=PFAN
      PFAN=PFAN+PINC
  100 CONTINUE
      IRST=0
C**********************************************************************
C TESTING VARIABLE, ITERATION 3
      DIF=ABS(AGAPP(1)-AGAP)
      IDIF=1
      DO 500 J=2,10
      XTEST=ABS(AGAPP(J)-AGAP)
      IF(XTEST.LT.DIF)IDIF=J
      IF(XTEST.LT.DIF)DIF=XTEST
      IF(XTEST.LT.ATOL)IPREST=2
  500 CONTINUE
C
      ICASE=IDIF
C**********************************************************************
C SELECTING NEW VALUE, ITERATION 3
      PSTART=PFSAV(IDIF)-PINC
      PSTOP=PFSAV(IDIF)+PINC
C
      IPN=IPN+1
      IF(IPN.GE.10) GO TO 601
      GO TO 24
  601 IDON=0
C**********************************************************************
C**********************************************************************
C ITERATION 3 ENDS
C**********************************************************************
C**********************************************************************
  600 CONTINUE
      PRAT=AMAX1(0.0,AMIN1(1.0,PCH/PTK))
      CALL HYCURV(PRAT,HX)
       HYPRES(IHY)=HYI*HX
      HYERROR(IHY)=HY-HYPRES(IHY)
      IF(IFLAG.EQ.1) GO TO 90
15    HY=HY+HYDELT
       IDEX=IDEX+1
      IF (IDEX.LT.15) GO TO 603
C IMPOSSIBLE CONDITIONS , SET ERRORS TO 1000000.
      IF(IPAS.LE.3)GO TO 604
  602 ZWT(IYCG,IPHI,ITH)=1.0E+06
      ZCC(IYCG,IPHI,ITH)=1.0E+06
      ZFF(IYCG,IPHI,ITH)=1.0E+06
  604 CONTINUE
      IDID=0
      GO TO 300
 603  CONTINUE
C TEST HY VALUE
      IHX=1
       HX=ABS(HYERROR(1))
```

```
      DO 80 IHY=2,3
      IF(ABS(HYERROR(IHY)).GE.HX) GO TO 80
      IHX=IHY
      HX=ABS(HYERROR(IHX))
80     CONTINUE
C***********************************************************************
C TESTING VARIABLE, ITERATION 2
      IF(HX.LT.HTOL.AND.IDON.NE.0)IFLAG=1
      IF(IFLAG.EQ.1) GO TO 86
C***********************************************************************
C SELECTING NEW VALUES, ITERATION 2
      IHXS=IHX
88    IF(IHX-2)81,81,82
81    IHX=1
      IF((HYERROR(1).GT.0.0).AND.(HYERROR(2).LT.0.0)) GO TO 83
      IF((HYERROR(1).LT.0.0).AND.(HYERROR(2).GT.0.0)) GO TO 83
       IF(IHXS.EQ.1) GO TO 84
82     IHX=2
      IF((HYERROR(2).GT.0.0).AND.(HYERROR(3).LT.0.0)) GO TO 83
      IF((HYERROR(2).LT.0.0).AND.(HYERROR(3).GT.0.0)) GO TO 83
      IF(IHXS.EQ.3) GO TO 85
83    HT=ABS(HYERROR(IHX)/(HYERROR(IHX+1)-HYERROR(IHX)))
      HYSTRT=HYTEST(IHX)+HT*(HYTEST(IHX+1)-HYTEST(IHX))
      HT=AMIN1(HT,(1.0-HT))
      HT=AMAX1(HT,0.25)
      HYDELT=HT*(HYTEST(IHX+1)-HYTEST(IHX))
      HYSTRT=HYSTRT-HYDELT
      HYSTOP=HYSTRT+2.0*HYDELT
       GO TO 16
84     HYSTRT=HYSTRT-HYDELT
      HYSTOP=HYSTRT+2.0*HYDELT
      GO TO 16
85    HYSTRT=HYSTRT+HYDELT
      HYSTOP=HYSTRT+2.0*HYDELT
      GO TO 16
86    HYSTRT=HYTEST(IHX)
      HYSTOP=HYSTRT+2.0*HYDELT
      GO TO 16
90     CONTINUE
      IF((ICREST.EQ.2).AND.(IPAS.GE.4)) GO TO 2001
C **********************************************************************
C **********************************************************************
C ITERATION 2 ENDS
C **********************************************************************
C **********************************************************************
C CALCULATE FORCES AND TORQUES
      CALL FORCE
      CC=(TCPZ+TTPZ)/(FCP+FTP)
      FF=-(TCPX+TTPX)/(FCP+FTP)
C VERTICAL LOAD CAPABILITY
      FORCN=(FCP+FTP)*COS(PHIE)*COS(THETAE)
C STORE VALUES OF PERFORMANCE VARIABLES FOR LOAD MAP
      IF(IPAS.GE.4) GO TO 298
      FORCNS(IPAS,IIT)=FORCN
      CCS(IPAS,IIT)=ATKCN
      YCGS(IPAS,IIT)=YCG+CC*SIN(PHIE)*COS(THETAE)-FF*SIN(THETAE)
```

```
      PHIS(IPAS,IIT)=PHIE*RADIAN
      AGAPS(IPAS,IIT)=AGAP
      PCHS(IPAS,IIT)=PCH
      PTKS(IPAS,IIT)=PTK
      PPLMS(IPAS,IIT)=PPLM
      PFANS(IPAS,IIT)=PFAN
      QFANS(IPAS,IIT)=QFAN
      THIS(IPAS,IIT)=THETAE*RADIAN
      FFS(IPAS,IIT)=FF
      TORXS(IPAS,IIT)=TCPX+TTPX
      TORZS(IPAS,IIT)=TCPZ+TTPZ
298   CONTINUE
C IF ITERATION FOR EQUILIBRIUM SET FORCE MATRICIES
      IF(IPAS.LE.3) GO TO 299
      ZCC(IYCG,IPHI,ITH)=CC
      ZWT(IYCG,IPHI,ITH)=FORCN
      ZFF(IYCG,IPHI,ITH)=FF
299   CONTINUE
  300 YCG=YCG+YCGDELT
C END OF YCG ITERATION LOOP
C
  200 PHIE=PHIE+PHIDELT
C END OF PHIE ITERATION LOOP
C
 400  THETAE=THETAE+THDELT
C END OF THETAE ITERATION LOOP
      IF((ISTAT.EQ.2).AND.(IPAS.GE.3)) GO TO 2001
      IF(IPAS.GE.4) GO TO 6000
       GO TO (3000,4000,5000),IPAS
C ***********************************************************************
C DONE WITH HEAVE PART OF LOAD MAP, SET UP PITCH PART
 3000 YCGEQUI=0.
      IF(YCPHI.GT.0.0.AND.YCTHE.GT.0.0)GO TO 3010
C
C DETERMINE EQUILIBRIUM YCG BY INTERPOLATION
      IMD=0
      DO 3500 I=2,10
      IF((FORCNS(1,I).LE.GMASS).AND.(FORCNS(1,I-1).GE.GMASS)) IMD=-I
      IF((FORCNS(1,I).GE.GMASS).AND.(FORCNS(1,I-1).LE.GMASS)) IMD=I
3500   CONTINUE
      IF(IMD.EQ.0) GO TO 3700
      IF(IMD.GT.0) GO TO 3600
      IMD=IABS(IMD)
C INCREASING FORCE
      YCGEQUI=YCGSTRT+FLOAT(IMD-1)*YCGDELT-(GMASS-FORCNS(1,IMD))/(FORCNS(1
     1(1,IMD-1)-FORCNS(1,IMD))*YCGDELT
      YCGDELT=0.0
      GO TO 3010
C DECREASING FORCE
3600  YCGEQUI=YCGSTRT+YCGDELT*FLOAT (IMD-1)-(GMASS-FORCNS(1,IMD-1))/(
     1 FORCNS(1,IMD)-FORCNS(1,IMD-1))*YCGDELT
      YCGDELT=0.0
      GO TO 3010
3700   WRITE(6,3701)
3701   FORMAT(/,5X,37H ***** ERROR,MODIFY LOAD OR YCG RANGE
      ICASE=0
```

```
      RETURN
3010  NYCG=1
       NPHI=10
      YCGSTOP=0.0
      PHISTRT=0.0
      PHISTOP=0.10
C IF INPUT BOUNDARIES ZERO USE DEFAULT VALUES
      IF((PSTRT.EQ.0.0).AND.(PHSTP.EQ.0.0)) GO TO 3001
      PHISTRT=PSTRT
      PHISTOP=PHSTP
3001   PHIDELT=(PHISTOP-PHISTRT)/9.0
      YCGSTRT=YCGEQUI
      IF(YCPHI.GT.0.0)YCGSTRT=YCPHI
      YCPHI=YCGSTRT
      GO TO 10
C ************************************************************************
C DONE WITH PITCH PART, SET UP ROLL PART
4000   NPHI=1
      NTHET=10
      THSTRT=0.0
      PHISTOP=0.0
       THSTOP=0.1
C IF INPUT BOUNDARIES ZERO USE DEFAULT VALUES
       IF((TSTRT.EQ.0.0).AND.(TSTOP.EQ.0.0)) GO TO 4001
      THSTRT=TSTRT
      THSTOP=TSTOP
4001   THDELT=(THSTOP-THSTRT)/9.0
      PHIDELT=0.0
      YCGSTRT=YCGEQUI
      IF(YCTHE.GT.0.0)YCGSTRT=YCTHE
      YCTHE=YCGSTRT
      GO TO 10
C ************************************************************************
C DONE WITH ROLL PART
C********
C INTIAL VALUES ITERATION 1
C SET BOUNDARIES FOR EQUILIBRIUM CONDITION PART
5000  NYCG=3
      NPHI=3
      NTHET=3
      PHISTRT=-0.025
      PHISTOP=0.025
      THSTRT=-0.0125
      THSTOP=0.0125
      YCGSTOP=1.02*YCGSTRT
      YCGSTRT=YCGSTRT*0.99
      PHIDELT=(PHISTOP-PHISTRT)/2.0
      YCGDELT=(YCGSTOP-YCGSTRT)*0.5
      THDELT=(THSTOP-THSTRT)*0.5
      GO TO 10
C ************************************************************************
C
6000  CONTINUE
      IS=1
      JS=1
      KS=1
```

```
C********
C TESTING VARIABLE ,ITERATION 1
C
C DIFFERENT QUADRATIC INDICES ARE FORMED DEPENDING ON ZERO OR
C NONZERO VALUES OF FFI AND CCI
      GMASS=MASS*32.2
      GO TO(61,62,63,64)ICGF
C ZWT(2,2,2) IS CENTRAL POINT OF FORCE MATRIX
61    XTEST=((ZWT(2,2,2)-GMASS)/GMASS)**2
     1+((ZCC(2,2,2)-CCI)/CCI)**2+((ZFF(2,2,2)-FFI)/FFI)**2
      GO TO 65
62    XTEST=((ZWT(2,2,2)-GMASS)/GMASS)**2
     1+((ZCC(2,2,2)-CCI)/CCI)**2
      GO TO 65
63    XTEST=((ZWT(2,2,2)-GMASS)/GMASS)**2
     1                            +((ZFF(2,2,2)-FFI)/FFI)**2
      GO TO 65
64    XTEST=((ZWT(2,2,2)-GMASS)/GMASS)**2
      XTOL=0.001
   65 CONTINUE
C TEST ENTIRE MATRIX AGAINST INDEX FOR MINIMUM ERROR STATE
      DO 1000 IQQ=1,3
      DO 1000 JQQ=1,3
      DO 1000 KQQ=1,3
      I=IQQ
      J=JQQ
      K=KQQ
      IF(ABS(CCI).LE.0.001)J=2
      IF(ABS(FFI).LE.0.001)K=2
      GO TO (91,92,93,94)ICGF
C USE PERFORMANCE INDEX SPECIFIED ABOVE
91    XDIS=((ZWT(I,J,K)-GMASS)/GMASS)**2
     1+((ZCC(I,J,K)-CCI)/CCI)**2+((ZFF(I,J,K)-FFI)/FFI)**2
      GO TO 95
92    XDIS=((ZWT(I,J,K)-GMASS)/GMASS)**2
     1+((ZCC(I,J,K)-CCI)/CCI)**2
      GO TO 95
93    XDIS=((ZWT(I,J,K)-GMASS)/GMASS)**2
     1                            +((ZFF(I,J,K)-FFI)/FFI)**2
      GO TO 95
94    XDIS=((ZWT(I,J,K)-GMASS)/GMASS)**2
   95 CONTINUE
      IF(XDIS.GT.XTEST)GO TO 1000
C SET INDEX FOR BEST POINT
      IS=I
      JS=J
      KS=K
      XTEST=XDIS
 1000 CONTINUE
C IF POINT OK, SET FOR FINAL PASS
      IF (XTEST.LT.XTOL) GO TO 2000
C
      IF(XTEST.GT.1.0E+08) GO TO 1201
C**********************************************************************=
C SELECTING NEXT VALUES, ITERATION 1
C SIDE POINT IS THE BEST, MOVE THE CUBE SIDEWAYS, SUCH THAT THE
```

```
C BEST POINT BECOMES THE CENTER POINT
C
      YCG=YCGSTRT+FLOAT(IS-1)*YCGDELT
      PHIE=PHISTRT+FLOAT(JS-1)*PHIDELT
      THETAE=THSTRT+FLOAT(KS-1)*THDELT
      IF(IS.EQ.2.AND.JS.EQ.2.AND.KS.EQ.2)GO TO 1200
      YCGSTRT=YCG-YCGDELT
      YCGSTOP=YCG+YCGDELT
      PHISTRT=PHIE-PHIDELT
      PHISTOP=PHIE+PHIDELT
      THSTRT=THETAE-THDELT
      THSTOP=THETAE+THDELT
      IODIN=IODIN+1
      IF(ICON.EQ.0.AND.IODIN.GE.NITE)GO TO 2003
C
C TEST IF ITERATION UNABLE TO SOLVE
      IF(IODIN.GE.NITE)GO TO 2002
      GO TO 10
C
C MID POINT IS THE BEST, REDUCE EACH SIDE OF THE CUBE BY FACTOR 2
 1200 YCGSTRT=YCG-YCGDELT/2.
      YCGSTOP=YCG+YCGDELT/2.
      PHISTOP=PHIE+PHIDELT/2.
      PHISTRT=PHIE-PHIDELT/2.
      THSTRT=THETAE-THDELT/2.
      THSTOP=THETAE+THDELT/2.
      IODIN=IODIN+1
      IF(IDID.EQ.1) ICON=0
      IF(IODIN.GE.NITE)GO TO 2003
      GO TO 10
C
C NO POINT IS FEASIBLE, LIFT UP THE CUBE, SUCH THAT VALUES OF
C YCG ARE HIGHER
 1201 YCGSTRT=YCGSTRT+3.*YCGDELT
      YCGSTOP=YCGSTOP+3.*YCGDELT
      IODIN=IODIN+1
      IF(IODIN.GE.NITE)GO TO 2002
      GO TO 10
C
 2000 PHIE=FLOAT(JS-1)*PHIDELT+PHISTRT
      THETAE=FLOAT(KS-1)*THDELT+THSTRT
      YCG=FLOAT(IS-1)*YCGDELT+YCGSTRT+ZCC(IS,JS,KS)*SIN(PHIE)*COS(THETAE
     1)-ZFF(IS,JS,KS)*SIN(THETAE)
      ICREST=2
      GO TO 11
C **  ERROR  RETURN AFTER MESSAGE
 2002 WRITE(6,1202)
 1202 FORMAT(10X,*INFEASIBLE CONFIGURATION OR XTOL TOO SMALL*//)
      ICASE=1
      RETURN
 2003 WRITE(6,1203)XTEST
 1203 FORMAT(10X,*INSUFFICIENT NUMBER OF ITERATIONS*/15X,*(A) INCREASE V
     1ALUE OF NITE, OR*/15X,*(B) INCREASE VALUE OF XTOL TO AT LEAST*,E10
     2.3/15X,*SOLUTION (A) INCREASES BOTH ACCURACY AND COST*)
      ICASE=0
      RETURN
```

Table 217: LISTING FOR SUBROUTINE STATIC (CONCLUDED)

```
C***********************************************************************
C***********************************************************************
C ITERATION 1 ENDS
C **********************************************************************
C **********************************************************************
C RESET CC,FF VALUES TO INPUT AND RETURN
 2001 CC=CCI
      FF=FFI
      ICASE=1
      RETURN
      END
```

Table 218: LISTING FOR SUBROUTINE STEQU

```
CSTEQU
      SUBROUTINE STEQU
C DYNAMIC FAN VERSION FOR FMA4
C STATE EQUATIONS FOR THE DYNAMIC SYSTEM
C
      REAL L,L1,L2,LS,LP,MASS
C
      COMMON/COEFFS/CPA,CAF,CPC,CPT,CTC,CGAP,CTA,CVENT,CKK
      COMMON/DYNAMIC/TIME,FTIME,DTIME,IQ,NQ,DVCH,DVTK,VELX,DERY(13)
     1,DVCHP
      COMMON/ESTMD/GEC,DAMPC,U,DECCL,HDC,PHA,CENFX,CENFZ,ZEPRV,ZPRV
     1,QP2,SLOPE
      COMMON/FLUID/QFNX,QPLAT,QPLCH,QPLTK,QTKAT,QTKCH,QCHAT,PATFN,PFAN,
     1 PAT,TEMPAT,RHO,QVENT
      COMMON/FORTQ/FCP,FTP,FORCT,FDF,FORCEY,TCPX,TTPX,TORQTX,TDFX,
     1TORQUEX,TCPZ,TTPZ,TORQTZ,TDFZ,TORFZ,TORQUEZ
      COMMON/VEHCL/MASS,AIX,AIZ,AIXY,AIYZ,AIZX,CC,GG,FF,AIFAN
      COMMON/AREAV/S,AATFN,APLAT,APLCH,APLTK,ATKAT,ATKCH,AGAP,ATK,ACH,
     1 ATKCN,APRV,VCH,VTK,VPLM,VCHD,VFAN,ATKATC,ATKCHC
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      COMMON/COVRLY/INST
      COMMON/CX/XX(1)
      COMMON/FMPCH/NPCH
      COMMON/CNTRLS/I1,I2,MODE,E(1)
      COMMON/FMERR/FMX
C
C FOLLOWING SUBROUTINES ARE CALLED TO UPDATE VALUES OF
C FORCES,TORQUES AND FLOWS, GIVEN THE NEW VALUES OF THE
C STATE VARIABLES
C
      CALL FLOW
      CALL FORCE
C
C
C
C*********************************************************************
C*********************************************************************
C                          STATE EQUATIONS
C*********************************************************************
C*********************************************************************
C
C***THE STATE VARIABLES***
C 1)PPLM..PLENUM PRESSURE (GAGE)
C 2)PCH..CUSHION PRESSURE (GAGE)
C 3)PTK..TRUNK PRESSURE (GAGE)
C 4)SINKRT..VERTICAL SINK RATE, POSITIVE UPWARDS
C 5)YCG..CG ELEVATION
C 6)DPHI..PITCH RATE,VEHICLE FRAME
C 7)DTHETA..ROLL RATE, VEHICLE FRAME
C 8)THETAE..EULERIAN ROLL ANGLE
C 9)PHIE..EULERIAN PITCH ANGLE
```

Table 218: LISTING FOR SUBROUTINE STEQU (CONTINUED)

```
C 10)SIE..EULERIAN YAW ANGLE (APPROX. ZERO)
C 11)XV..DISPLN OF PRESSURE RELIEF VALVE
C 12)VV..VELOCITY OF PRESSURE RELIEF VALVE
C 13)QFANX..FAN AIR INERTANCE FLOW
C
C VARIATIONS IN THE EQUATIONS ARE MADE TO ACCOMODATE SPECIAL CONDITIONS
C
      IF(IPP)11,10,11
C COMBINED TRUNK-PLENUM DYNAMICS , IPP=0
10    DERY(1)=CKK*(PPLM+PAT)/(VPLM+VTK)*(QFANX-QPLCH-QPLAT-QTKCH-QTKAT
     1-DVTK-QVENT)
      DERY(3)=DERY(1)
      GO TO 12
C SEPARATE TRUNK-PLENUM DYNAMICS , IPP=1
11    DERY(1)=(CKK*(PPLM+PAT)/VPLM)*(QFANX-QPLCH-QPLTK-QPLAT-QVENT)
      DERY(3)=(CKK*(PTK+PAT)/VTK)*(QPLTK-QTKCH-QTKAT-DVTK)
C CUSHION FLOW ABOVE GROUND EFFECT TRANSITION ZONE
   12 QCHFT=QPLCH+QTKCH-DVCH
C CALCULATE GROUND EFFECT TRANSITION ZONE
      TBOUND=GG+HYI*(1.+GEC)
       BBOUND=GG+HYI
C DETERMINE IF ACLS IN TRANSITION ZONE
      IF(FMX.GT.0.) GO TO 16
      IF(YCG.GT.TBOUND)GO TO 13
       IF(YCG.GT.BBOUND) GO TO 14
      GO TO 16
C
C ABOVE TRANSITION ZONE
   13 QCHAT=QCHFT
      IFLAG=0
      IQ=0
      NQ=0
      GO TO 15
C
C IN TRANSITION ZONE
14    IFLAG=1
      NQ=100
      IQ=IFIX(ABS((TBOUND-YCG)/(TBOUND-BBOUND)*FLOAT(NQ)))
      IF(IQ.GE.NQ)IQ=NQ
      GO TO 17
C
C IN GROUND EFFECT ZONE
16    NQ=1
      IQ=1
      IFLAG=2
C COMPUTE CUSHION TO ATMOSPHERE FLOW
17     QCHAT=FLOAT(NQ-IQ)/FLOAT(NQ)*QCHFT+FLOAT(IQ)/FLOAT(NQ)*QCHAT
C CUSHION PRESSURE DERIVATIVE
   15 DERY(2)=(QPLCH+QTKCH-QCHAT-DVCH+DVCHP*PCH*DERY(3)/(PTK*PTK))/
     1(VCH/(CKK*(PCH+PAT))+DVCHP/PTK)
C YCG ELEVATION DERIVATIVES
      DERY(4)=FORCEY/MASS-32.2
      DERY(5)=SINKRT
C ANGULAR POSITION DERIVATIVES
      DERY(6)=(AIX*(TORQUEZ-DTHETA*(AIXY*DTHETA+AIYZ*DPHI))-AIZX*(TORQUEX
     1X+DPHI*(AIXY*DTHETA+AIYZ*DPHI)))/(AIZ*AIX-AIZX*AIZX)
```

```
      DERY(7)=(AIZ*(TORQUEX+DPHI*(AIXY*DTHETA+AIYZ*DPHI))-AIZX*(TORQUEZ
     1-DTHETA*(AIXY*DTHETA+AIYZ*DPHI)))/(AIZ*AIX-AIZX*AIZX)
      IF(INST.NE.31.OR.ATKCN.NE.0.) GO TO 20
      DERY(6)=-PHIE
      DERY(7)=-THETAE
 20   CONTINUE
      DERY(8)=DTHETA*COS(PHIE)
      DERY(9)=DPHI+DTHETA*SIN(PHIE)*TAN(THETAE)
      DERY(10)=DTHETA*SIN(PHIE)/COS(THETAE)
C TEST IF PRV OPEN
      IF(IPRV)61,60,61
C COMPUTE PRV VALVE VELOCITY,LOCATION
61     DERY(11)=VV
      DERY(12)=(PPLM*APRV-AKPRV*XV-PPLMB*APRV-ZPRV*VV)/AMPRV
      IF(FMX.GT.0.) DERY(12)=(DERY(12)*AMPRV - 100.*AKPRV*
     2(AMAX1(0.,XV-XA)+AMIN1(0.,XV)))/AMPRV
      GO TO 62
C PRV FULL OPEN,AT STOPS,OR NO PRV
60    DERY(11)=0.0
      DERY(12)=0.0
   62 CONTINUE
C DYNAMICS FOR FAN,FAN FLOW RATE OF CHANGE
      DERY(13)=(PFAN+PATFN-PPLM)/AIFAN
C CUSHION PRESSURE IS ZERO ABOVE TRANSITION ZONE
      IF(FMX.GT.0.) GO TO 75
      IF(IFLAG.EQ.0) XX(NPCH)=0.0
 75   CONTINUE
      RETURN
      END
```

Table 219: LISTING FOR SUBROUTINE SV

```
CSV
      SUBROUTINE SV(UW,VW,WW,PW,QW,RW,UWS,VWS,WWS,UG,VG,WG,PG,QG,RG)
C    VERSION 1.                                                MAY 26 1977
C    PURPOSE   SUM TWO SETS OF 3 AXIS VELOCITIES AND ANGULAR RATES
C    METHOD    ADD STEADY OR SHEAR WIND COMPONENTS TO THE RANDOM
C              GUST COMPONENTS AND GUST ANGULAR RATES
C    CALL SEQUENCE
C    ******  OUTPUTS  ******
C    LINEAR VELOCITIES -- BODY AXES
C      UW,VW,WW       -SUM OF X,Y,Z AXIS WIND VELOCITIES, FT/SEC
C    ANGULAR VELOCITIES -- BODY AXES
C      PW,QW,RW       -SUM OF X,Y,Z AXIS ANGULAR VELOCITIES, DEG/SEC
C    ******  INPUTS  ******
C     .LINEAR VELOCITIES -- BODY AXES
C      UWS,VWS,WWS    -X,Y,Z AXIS STEADY/SHEAR WIND COMPONENTS, FT/SEC
C      UG,VG,WG       -X,Y,Z AXIS GUST WIND COMPONENTS, FT/SEC
C    ANGULAR VELOCITIES -- BODY AXES
C      PG,QG,RG       -X,Y,Z AXIS GUST ANGULAR COMPONENTS, DEG/SEC
C    WRITTEN BY MAHINDER WAHI                                  MAY  1977
C          ***  SUM LINEAR VELOCITIES  ***
      UW= UWS + UG
      VW= VWS + VG
      WW= WWS + WG
C          ***  SUM ANGULAR VELOCITIES  ***
      PW= PG
      QW= QG
      RW= RG
      RETURN
      END
```

Table 220: LISTING FOR SUBROUTINE SW

```
CSW
      SUBROUTINE SW(VO1,VA1,VB1,SW1,TC1,TC2)
C
C    PURPOSE - TO PROVIDE SWITCH CONTROL FOR ONE VARIABLE
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  - ADAM LLOYD            LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - NOT MORE THAN TWO SWITCHINGS AT TIMES TC1 AND TC2
C
C
C    INPUT/OUTPUT LIST
C
C    VO1         OUTPUT VARIABLE NO 1                 ANY       OUTPUT VAR
C    VA1         INPUT VARIABLE NO A1                 ANY       INPUT  VAR
C    VB1         INPUT VARIABLE NO B1                 ANY       INPUT  VAR
C    SW1         SWITCH CONTROL INITIAL VALUE         ---       INPUT  PARAM
C                    =1.   VO=VB
C                    =0.   VO=VA
C    TC1         TIME FOR FIRST SWITCH                SECS      INPUT  PARAM
C    TC2         TIME FOR SECOND SWITCH               SECS      INPUT  PARAM
C                    (TC2.GT.TC1)
      COMMON/CTIME/TIME
      COMMON/CIO/IREAD,IWRITE,IDIAG
      SX=SW1
      IF(TIME.GT.TC1.AND.TIME.LT.TC2)SX=ABS(SW1-1.)
      VO1=VA1
      IF(SX.GT.0.5)VO1=VB1
      RETURN
      END
```

Table 221: LISTING FOR SUBROUTINE SX

```
CSX
      SUBROUTINE SX(VO1,VO2,VA1,VA2,VB1,VB2,SW1,TC1,TC2)
C
C    PURPOSE - TO PROVIDE A SWITCH COMPONENT FOR TWO VARIABLES
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  - ADAM LLOYD             LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - NOT MORE THAN TWO SWITCHINGS AT TIMES TC1 AND TC2
C
C
C    INPUT/OUTPUT LIST
C
C    VO1         OUTPUT VARIABLE NO 1                ANY       OUTPUT VAR
C    VO2         OUTPUT VARIABLE NO 2                ANY       OUTPUT VAR
C    VA1         INPUT VARIABLE NO A1                ANY       INPUT  VAR
C    VA2         INPUT VARIABLE NO A2                ANY       INPUT  VAR
C    VB1         INPUT VARIABLE NO B1                ANY       INPUT  VAR
C    VB2         INPUT VARIABLE NO B2                ANY       INPUT  VAR
C    SW1         SWITCH CONTROL INITIAL VALUE        ---       INPUT  PARAM
C                    =1.   VO=VB
C                    =0.   VO=VA
C    TC1         TIME FOR FIRST SWITCH               SECS      INPUT  PARAM
C    TC2         TIME FOR SECOND SWITCH              SECS      INPUT  PARAM
C                    (TC2.GT.TC1)
      COMMON/CTIME/TIME
      COMMON/CIO/IREAD,IWRITE,IDIAG
      SW=SW1
      IF(TIME.GT.TC1.AND.TIME.LT.TC2)SW=ABS(SW1-1.)
      VO1=VA1
      VO2=VA2
      IF(SW.GT.0.5)VO1=VB1
      IF(SW.GT.0.5)VO2=VB2
      RETURN
      END
```

Table 222: LISTING FOR SUBROUTINE SY

```
CSY
      SUBROUTINE SY(VO1,VO2,VO3,VA1,VA2,VA3,VB1,VB2,VB3,SW1,TC1,TC2)
C
C    PURPOSE - TO PROVIDE A SWITCH COMPONENT FOR THREE VARIABLES
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  - ADAM LLOYD            LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - NOT MORE THAN TWO SWITCHINGS AT TIMES TC1 AND TC2
C
C
C    INPUT/OUTPUT LIST
C
C    VO1         OUTPUT VARIABLE NO 1             ANY       OUTPUT VAR
C    VO2         OUTPUT VARIABLE NO 2             ANY       OUTPUT VAR
C    VO3         OUTPUT VARIABLE NO 3             ANY       OUTPUT VAR
C    VA1         INPUT VARIABLE NO A1             ANY       INPUT  VAR
C    VA2         INPUT VARIABLE NO A2             ANY       INPUT  VAR
C    VA3         INPUT VARIABLE NO A3             ANY       INPUT  VAR
C    VB1         INPUT VARIABLE NO B1             ANY       INPUT  VAR
C    VB2         INPUT VARIABLE NO B2             ANY       INPUT  VAR
C    VB3         INPUT VARIABLE NO B3             ANY       INPUT  VAR
C    SW1         SWITCH CONTROL INITIAL VALUE     ---       INPUT  PARAM
C                    =1.   VO=VB
C                    =0.   VO=VA
C    TC1         TIME FOR FIRST SWITCH            SECS      INPUT  PARAM
C    TC2         TIME FOR SECOND SWITCH           SECS      INPUT  PARAM
C                    (TC2.GT.TC1)
      COMMON/CTIME/TIME
      COMMON/CIO/IREAD,IWRITE,IDIAG
      SW=SW1
      VO1=VA1
      VO2=VA2
      VO3=VA3
      IF(TIME.GT.TC1.AND.TIME.LT.TC2)SW=ABS(SW1-1.)
      IF(SW.GT.0.5)VO1=VB1
      IF(SW.GT.0.5)VO2=VB2
      IF(SW.GT.0.5)VO3=VB3
      RETURN
      END
```

Table 223: LISTING FOR SUBROUTINE SZ

```
CSZ
      SUBROUTINE SZ(VO1,VO2,VO3,VO4,VA1,VA2,VA3,VA4,VB1,VB2,VB3,VB4,
     1 SW1,TC1,TC2)
C
C    PURPOSE - TO PROVIDE A SWITCH COMPONENT FOR FOUR VARIABLES
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  - ADAM LLOYD             LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - NOT MORE THAN TWO SWITCHINGS AT TIMES TC1 AND TC2
C
C
C    INPUT/OUTPUT LIST
C
C    VO1         OUTPUT VARIABLE NO 1                ANY        OUTPUT VAR
C    VO2         OUTPUT VARIABLE NO 2                ANY        OUTPUT VAR
C    VO3         OUTPUT VARIABLE NO 3                ANY        OUTPUT VAR
C    VO4         OUTPUT VARIABLE NO 4                ANY        OUTPUT VAR
C    VA1         INPUT VARIABLE NO A1                ANY        INPUT  VAR
C    VA2         INPUT VARIABLE NO A2                ANY        INPUT  VAR
C    VA3         INPUT VARIABLE NO A3                ANY        INPUT  VAR
C    VA4         INPUT VARIABLE NO A4                ANY        INPUT  VAR
C    VB1         INPUT VARIABLE NO B1                ANY        INPUT  VAR
C    VB2         INPUT VARIABLE NO B2                ANY        INPUT  VAR
C    VB3         INPUT VARIABLE NO B3                ANY        INPUT  VAR
C    VB4         INPUT VARIABLE NO B4                ANY        INPUT  VAR
C    SW1         SWITCH CONTROL INITIAL VALUE        ---        INPUT  PARAM
C                    =1.   VO=VB
C                    =0.   VO=VA
C    TC1         TIME FOR FIRST SWITCH               SECS       INPUT  PARAM
C    TC2         TIME FOR SECOND SWITCH              SECS       INPUT  PARAM
C                    (TC2.GT.TC1)
      COMMON/CTIME/TIME
      COMMON/CIO/IREAD,IWRITE,IDIAG
      SW=SW1
      IF(TIME.GT.TC1.AND.TIME.LT.TC2)SW=ABS(SW1-1.)
      VO1=VA1
      VO2=VA2
      VO3=VA3
      VO4=VA4
      IF(SW.GT.0.5)VO1=VB1
      IF(SW.GT.0.5)VO2=VB2
      IF(SW.GT.0.5)VO3=VB3
      IF(SW.GT.0.5)VO4=VB4
      RETURN
      END
```

Table 224: LISTING FOR SUBROUTINE S2

```
CS2
      SUBROUTINE S2(FX3,FY3,FZ3,AL3,AM3,AN3,FX1,FY1,FZ1,AL1,AM1,AN1,
     1 FX2,FY2,FZ2,AL2,AM2,AN2)
C  VERSION 1.                            REVISED: MAY 21 1976
C  PURPOSE:  SUM TWO SETS OF 3 AXIS FORCES AND MOMENTS
C  CALL SEQUENCE:
C  =============  OUTPUTS  ==============
C  FX3,FY3,FZ3  -  SUM OF FORCES
C  AL3,AM3,AN3  -  SUM OF MOMENTS
C  =============  INPUTS  ===============
C  FX1,FY1,FZ1  -  FORCES INPUT  PORT 1
C  AL1,AM1,AN1  -  MOMENTS INPUT  PORT 1
C  FX2,FY2,FZ2  -  FORCES INPUT  PORT2
C  AL2,AM2,AN2  -  MOMENTS INPUT  PORT 2
C  DESIGNED BY:  J.D. BURROUGHS               MAY 1976
C
C ================  SUM FORCES  =============
      FX3=FX1+FX2
      FY3=FY1+FY2
      FZ3=FZ1+FZ2
C ================  SUM MOMENTS  ============
      AL3=AL1+AL2
      AM3=AM1+AM2
      AN3=AN1+AN2
      RETURN
      END
```

Table 225: LISTING FOR SUBROUTINE S3

```
CS3
      SUBROUTINE S3(FX4,FY4,FZ4,AL4,AM4,AN4,FX1,FY1,FZ1,AL1,AM1,AN1,
     1 FX2,FY2,FZ2,AL2,AM2,AN2,FX3,FY3,FZ3,AL3,AM3,AN3)
C  VERSION 1.                             REVISED: MAY 21 1976
C  PURPOSE:  SUM THREE SETS OF 3 AXIS FORCES AND MOMENTS
C  CALL SEQUENCE:
C  =============  OUTPUTS  ==============
C  FX4,FY4,FZ4  -  SUM OF FORCES
C  AL4,AM4,AN4  -  SUM OF MOMENTS
C  =============  INPUTS  ===============
C  FX1,FY1,FZ1  -  FORCES INPUT  PORT 1
C  AL1,AM1,AN1  -  MOMENTS INPUT  PORT 1
C  FX2,FY2,FZ2  -  FORCES INPUT  PORT2
C  AL2,AM2,AN2  -  MOMENTS INPUT  PORT 2
C  FZ3,FY3,FZ3  -  FORCES INPUT  PORT3
C  AL3,AM3,AN3  -  MOMENTS INPUT  PORT 3
C  DESIGNED BY:  J.D. BURROUGHS               MAY 1976
C
C ================  SUM FORCES  =============
      FX4=FX1+FX2+FX3
      FY4=FY1+FY2+FY3
      FZ4=FZ1+FZ2+FZ3
C ================  SUM MOMENTS  =============
      AL4=AL1+AL2+AL3
      AM4=AM1+AM2+AM3
      AN4=AN1+AN2+AN3
      RETURN
      END
```

Table 226: LISTING FOR SUBROUTINE TA

```
CTA
      SUBROUTINE TA(A2T,B2T,C2T,D2T,A2,B2,C2,D2)
C
C    PURPOSE - TO PROVIDE CAPABILITY TO INPUT VARIABLES AS FUNCTIONS
C              OF TIME
C
C    METHOD  - TABLE LOOK-UP USES 1 DEGREE INTERPOLATION
C              AND DOES NOT PERMIT EXTRAPOLATION
C
C
C    WRITTEN BY  -  ADAM LLOYD                  LATEST REVISION     NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    A2T         TABULAR INPUT OF VARIABLE A2       ANY        INPUT  TABLE
C    B2T         TABULAR INPUT OF VARIABLE B2       ANY        INPUT  TABLE
C    C2T         TABULAR INPUT OF VARIABLE C2       ANY        INPUT  TABLE
C    D2T         TABULAR INPUT OF VARIABLE D2       ANY        INPUT  TABLE
C                ABOVE TABLES ARE ALL ONE DIMENSIONAL
C                WITH TIME AS INDEPENDENT VARIABLE
C    A2          VARIABLE A2                        ANY        OUTPUT VAR
C    B2          VARIABLE B2                        ANY        OUTPUT VAR
C    C2          VARIABLE C2                        ANY        OUTPUT VAR
C    D2          VARIABLE D2                        ANY        OUTPUT VAR
      DIMENSION A2T(1),B2T(1),C2T(1),D2T(1)
      COMMON/CTIME/TIME
      NA=A2T(2)
      NB=B2T(2)
      NC=C2T(2)
      ND=D2T(2)
      A2=TBLU1(TIME,A2T(4),A2T(NA+4),1,-NA)
      B2=TBLU1(TIME,B2T(4),B2T(NB+4),1,-NB)
      C2=TBLU1(TIME,C2T(4),C2T(NC+4),1,-NC)
      D2=TBLU1(TIME,D2T(4),D2T(ND+4),1,-ND)
      RETURN
      END
```

Table 227: LISTING FOR SUBROUTINE TB

```
CTB
      SUBROUTINE TB(A2T,B2T,A2,B2)
C
C    PURPOSE - TO PROVIDE CAPABILITY TO INPUT VARIABLES AS FUNCTIONS
C              OF TIME
C
C    METHOD  - TABLE LOOK-UP USES 1 DEGREE INTERPOLATION
C              AND DOES NOT PERMIT EXTRAPOLATION
C
C
C    WRITTEN BY  -  ADAM LLOYD                    LATEST REVISION     NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    A2T         TABULAR INPUT OF VARIABLE A2       ANY        INPUT  TABLE
C    B2T         TABULAR INPUT OF VARIABLE B2       ANY        INPUT  TABLE
C                ABOVE TABLES ARE ALL ONE DIMENSIONAL
C                WITH TIME AS INDEPENDENT VARIABLE
C    A2          VARIABLE A2                        ANY        OUTPUT VAR
C    B2          VARIABLE B2                        ANY        OUTPUT VAR
      COMMON/CTIME/TIME
      DIMENSION A2T(1),B2T(1)
      NA=A2T(2)
      NB=B2T(2)
      A2=TBLU1(TIME,A2T(4),A2T(NA+4),1,-NA)
      B2=TBLU1(TIME,B2T(4),B2T(NB+4),1,-NB)
      RETURN
      END
```

Table 228: LISTING FOR SUBROUTINE TBL1

```
CTBL1
      FUNCTION TBL1(PR,AZO,DPR,NPTS,IS,NA)
C******
C   VERSION 1.              FEB 1978
C
C     PURPOSE
C        TBL1 PERFORMS LINEAR INTERPOLATION ON 1 INDEPENDENT VARIABLE
C
C   WRITTEN BY J.R.KILNER
C
C   LIMITATIONS  -  USED ONLY BY COMPONENT TK
C
C   FOR INPUTS AND OUTPUTS SEE FUNCTION TBL2
C
      DIMENSION AZO(NA,1)
C
      IF(PR.LE.0.) GO TO 10
      RMAX=DPR*(NPTS-1)
      IF(PR.GE.RMAX) GO TO 20
      II=PR*NPTS
      I=II+1
      TBL1=(AZO(I+1,IS)-AZO(I,IS))*(PR-DPR*II)/DPR+AZO(I,IS)
      GO TO 30
10    TBL1=AZO(1,IS)
      GO TO 30
20    TBL1=AZO(NPTS,IS)
C
30    RETURN
      END
```

Table 229: LISTING FOR SUBROUTINE TBL2

```
CTBL2
      FUNCTION TBL2(PR,ZO,ZOFS,A,DPR,NPTS,IS,NA)
C******
C    VERSION 1.          FEB. 1978
C
C     WRITTEN BY J.R.KILNER
C
C    LIMITATIONS - CALLED ONLY BY COMPONENT TK
C      PURPOSE
C         TBL2 PERFORMS CURVILINEAR TO RECTILINEAR TRANSFORMATION
C         AND LINEAR INTERPOLATION ON 2 INDEPENDENT VARIABLES
C
C      INPUT PARAMETERS
C         PR   - CUSHION TO TRUNK PRESSURE RATIO
C         ZO   - CUSHION HEIGHT
C         ZOFS - CUSHION HEIGHT (FREE SHAPE)
C         A    - DEPENDENT ARRAY
C         DPR  - STEP CHANGE IN PC/PT
C         NPTS - NUMBER OF ROW OR COLUMN ELEMENTS IN A
C         IS   - DIGGES PARAMETER SET
C         NA   - ROW AND COLUMN DIMENSION OF A
C
C     OUTPUT PARAMETERS
C        YO,L1,L3,AS,ACV  ARRAYS FOR TRUNK SHAPES
C
      DIMENSION A(NA,NA,1)
C
      IF(PR.LE.0.) GO TO 10
      RMAX=DPR*(NPTS-1)
      IF(PR.GE.RMAX) GO TO 20
      I=PR*NPTS
      I=I+1
      PR2=PR
      GO TO 30
20    I=NPTS-1
      PR2=RMAX
30    ZB=ZO/ZOFS
      IF(ZO.GE.ZOFS)GOTO 40
      IF(ZB.LE.DPR) GO TO 50
      JX=ZB*NPTS
      GO TO 60
50    JX=1
      ZB=DPR
C
60    J=NPTS-JX
      DZ=ZB-DPR*JX
      DP=PR2-DPR*(I-1)
      A1=(A(I,J,IS)-A(I,J+1,IS))*DZ/DPR+A(I,J+1,IS)
      A2=(A(I+1,J,IS)-A(I+1,J+1,IS))*DZ/DPR+A(I+1,J+1,IS)
      TBL2=(A2-A1)*DP/DPR+A1
      GOTO 99
C
10    IF(ZO.GE.ZOFS)GOTO 12
      ZB=ZO/ZOFS
      IF(ZB.LE.DPR)GOTO 14
      JX=ZB*NPTS
```

Table 229: LISTING FOR SUBROUTINE TBL2 (CONCLUDED)

```
      J=NPTS-JX
      DZ=ZB-DPR*JX
      TBL2=(A(1,J,IS)-A(1,J+1,IS))*DZ/DPR+A(1,J+1,IS)
      GOTO 99
C
12    TBL2=A(1,1,IS)
      GOTO 99
C
14    TBL2=A(1,NPTS,IS)
      GOTO 99
C
40    DP=PR2-DPR*(I-1)
      TBL2=(A(I+1,1,IS)-A(I,1,IS))*DP/DPR+A(I,1,IS)
C
99    RETURN
      END
```

Table 230: LISTING FOR SUBROUTINE TBLU1

```
CTBLU1
      FUNCTION TBLU1(X1,X,F1,NDX,NX)
C******
C     PURPOSE
C        TBLU1 PERFORMS TABLE SEARCH AND LAGRANGIAN POLYNOMIAL
C        INTERPOLATION OF USER-DEFINED DEGREE ON 1 INDEPENDENT
C        VARIABLE
C     USAGE
C        DIMENSION X(NX),F1(NX)
C        V = TBLU1(X1,X,F1,NDX,NX)
C     INPUT PARAMETERS
C        X1 - POINT TO INTERPOLATE FOR
C         X - INDEPENDENT VARIABLE ARRAY
C        F1 - DEPENDENT VARIABLE ARRAY
C       NDX - DEGREE OF INTERPOLATION
C        NX - IABS(NX) IS THE NUMBER OF DATA POINTS IN THE X ARRAY.
C             IF NEGATIVE, NEAREST END POINT IS TO BE USED UPON
C             EXTRAPOLATION
C     OUTPUT PARAMETER
C        V - RESULT OF TABLE SEARCH AND INTERPOLATION
C            SUCCESS  V = INTERPOLATED VALUE
C              ERROR  V = INDEFINITE VALUE WHERE RIGHTMOST DIGIT
C                         DEFINES THE ERROR DETECTED
C                1   DATA VALUES WITHIN X ARE NOT DISTINCT
C                2   NDX IS LESS THAN ZERO
C                3   NX IS ZERO
C******
C
      DIMENSION X(1),Y(1),F1(1)
      INTEGER SEARCH
      DATA ERR2/17770000000000000002B/
      DATA ERR3/17770000000000000003B/
C         TEST FOR USER ERRORS
  100 TBLU1 = 0
      IF (NDX.LT.0) TBLU1 = ERR2
      IF (NX.EQ.0) TBLU1 = ERR3
      IF (TBLU1.NE.0) GO TO 210
C         SEARCH FOR X1 AND TEST FOR EXACTNESS
      MDX = NDX
      IF(SEARCH(X1,X,MDX,NX,I).EQ.0) GO TO 200
      TBLU1 = F1(I)
      GO TO 210
C         INTERPOLATE
  200 TBLU1 = TERP1(X1,X,F1,MDX,I)
  210 RETURN
      END
```

Table 231: LISTING FOR SUBROUTINE TBLU2

```
CTBLU2
      FUNCTION TBLU2(X1,Y1,X,Y,F2,NDX,NDY,NX,NY,MX,MY)
C******
C     PURPOSE
C        TBLU2 PERFORMS TABLE SEARCH AND LAGRANGIAN POLYNOMIAL
C        INTERPOLATION OF USER-DEFINED DEGREE ON 2 INDEPENDENT
C        VARIABLES
C     USAGE
C        DIMENSION X(NX),Y(NY),F2(MX,MY)
C        V = TBLU2(X1,Y1,X,Y,F2,NDX,NDY,NX,NY,MX,MY)
C     INPUT PARAMETERS
C        X1,Y1 - POINT TO INTERPOLATE FOR
C          X,Y - ARRAYS OF INDEPENDENT VARIABLES
C           F2 - 2D ARRAY OF DEPENDENT VARIABLE
C      NDX,NDY - DEGREE OF INTERPOLATION FOR EACH DIMENSION
C        NX,NY - IABS OF EACH IS THE NUMBER OF DATA POINTS IN THE
C                RESPECTIVE X OR Y ARRAY.  IF NEGATIVE, NEAREST
C                END POINT IS TO BE USED UPON EXTRAPOLATION
C        MX,MY - DIMENSIONAL CONSTANTS FOR F2 ARRAY
C     OUTPUT PARAMETERS
C        V - RESULT OF TABLE SEARCH AND INTERPOLATION
C            SUCCESS  V = INTERPOLATED VALUE
C              ERROR  V = INDEFINITE VALUE WHERE RIGHTMOST DIGIT
C                          DEFINES THE ERROR DETECTED
C                1   DATA VALUES WITHIN X OR Y ARE NOT DISTINCT
C                2   EITHER NDX OR NDY IS LESS THAN ZERO
C                3   EITHER NX OR NY IS ZERO
C                4   MX.LT.IABS(NX)
C******
C
      DIMENSION X(1),Y(1),F2(MX,MY)
      INTEGER SEARCH
      DATA ERR2/17770000000000000002B/
      DATA ERR3/17770000000000000003B/
      DATA ERR4/17770000000000000004B/
C         TEST FOR USER ERRORS
  100 TBLU2 = 0
      IF ((NDX.LT.0).OR.(NDY.LT.0)) TBLU2 = ERR2
      IF ((NX.EQ.0).OR.(NY.EQ.0)) TBLU2 = ERR3
      IF (MX.LT.IABS(NX)) TBLU2 = ERR4
      IF (TBLU2.NE.0) GO TO 310
C         SET UP INITIAL PARAMETERS
  120 X2 = X1
      Y2 = Y1
      MDX = NDX
      MDY = NDY
C         SEARCH FOR X1 AND Y1 IN TABLES
      IX = SEARCH(X2,X,MDX,NX,I)
      IY = SEARCH(Y2,Y,MDY,NY,J)
C         TEST FOR EXACTNESS IN 1 OR 2 DIMENSIONS
      IW = IX+IY+IY+1
      GO TO (300,210,220,200),IW
  200 TBLU2 = F2(I,J)
      GO TO 310
  210 X2 = X(I)
      MDX = 0
```

Table 231: LISTING FOR SUBROUTINE TBLU2 (CONCLUDED)

```
      GO TO 300
  220 Y2 = Y(J)
      MDY = 0
C          INTERPOLATE
  300 TBLU2 = TERP2(X2,Y2,X,Y,F2,MDX,MDY,MX,MY,I,J)
  310 RETURN
      END
```

Table 232: LISTING FOR SUBROUTINE TBLU3

```
CTBLU3
      FUNCTION TBLU3(X1,Y1,Z1,X,Y,Z,F3,NDX,NDY,NDZ,NX,NY,NZ,MX,MY,MZ)
C******
C     PURPOSE
C        TBLU3 PERFORMS TABLE SEARCH AND LAGRANGIAN POLYNOMIAL
C        INTERPOLATION OF USER-DEFINED DEGREE ON 3 INDEPENDENT
C        VARIABLES
C     USAGE
C        DIMENSION X(NX),Y(NY),Z(NZ),F3(MX,MY,MZ)
C        V = TBLU3(X1,Y1,Z1,X,Y,Z,F3,NDX,NDY,NDZ,NX,NY,NZ,MX,MY,MZ)
C     INPUT PARAMETERS
C         X1,Y1,Z1 - POINT TO INTERPOLATE FOR
C            X,Y,Z - ARRAYS OF INDEPENDENT VARIABLES
C               F3 - 3D ARRAY OF DEPENDENT VARIABLE
C      NDX,NDY,NDZ - DEGREE OF INTERPOLATION FOR EACH DIMENSION
C         NX,NY,NZ - IABS OF EACH IS THE NUMBER OF DATA POINTS IN
C                    THE RESPECTIVE X, Y OR Z ARRAY.  IF NEGATIVE,
C                    NEAREST END POINT IS TO BE USED UPON
C                    EXTRAPOLATION
C         MX,MY,MZ - DIMENSIONAL CONSTANTS FOR F3 ARRAY
C     OUTPUT PARAMETERS
C        V - RESULT OF TABLE SEARCH AND INTERPOLATION
C            SUCCESS  V = INTERPOLATED VALUE
C              ERROR  V = INDEFINITE VALUE WHERE RIGHTMOST DIGIT
C                         DEFINES THE ERROR DETECTED
C                1   DATA VALUES WITHIN X, Y OR Z ARE NOT DISTINCT
C                2   ONE OF NDX, NDY OR NDZ IS LESS THAN ZERO
C                3   ONE OF NX, NY OR NZ IS ZERO
C                4   EITHER MX.LT.IABS(NX) OR MY.LT.IABS(NY)
C******
C
      DIMENSION X(1),Y(1),Z(1),F3(MX,MY,MZ)
      INTEGER SEARCH
      DATA ERR2/17770000000000000002B/
      DATA ERR3/17770000000000000003B/
      DATA ERR4/17770000000000000004B/
C        TEST FOR USER ERRORS
  100 TBLU3 = 0
      IF ((NDX.LT.0).OR.(NDY.LT.0).OR.(NDZ.LT.0)) TBLU3 = ERR2
      IF ((NX.EQ.0).OR.(NY.EQ.0).OR.(NZ.EQ.0)) TBLU3 = ERR3
      IF ((MX.LT.IABS(NX)).OR.(MY.LT.IABS(NY))) TBLU3 =ERR4
      IF (TBLU3.NE.0) GO TO 310
C        SET UP INITIAL PARAMETERS
  120 X2 = X1
      Y2 = Y1
      Z2 = Z1
      MDX = NDX
      MDY = NDY
      MDZ = NDZ
C        SEARCH FOR X1, Y1 AND Z1 IN TABLES
      IX = SEARCH(X2,X,MDX,NX,I)
      IY = SEARCH(Y2,Y,MDY,NY,J)
      IZ = SEARCH(Z2,Z,MDZ,NZ,K)
C        TEST FOR EXACTNESS IN 1 OR MORE DIMENSIONS
      IW = IX+IY+IZ
      IF (IW.EQ.0) GO TO 300
```

```
      IF (IW.NE.3) GO TO 200
      TBLU3 = F3(I,J,K)
      GO TO 310
  200 IF (IX.EQ.0) GO TO 210
      X2 = X(I)
      MDX = 0
  210 IF (IY.EQ.0) GO TO 220
      Y2 = Y(J)
      MDY = 0
  220 IF (IZ.EQ.0) GO TO 300
      Z2 = Z(K)
      MDZ = 0
C           INTERPOLATE
  300 TBLU3 = TERP3(X2,Y2,Z2,X,Y,Z,F3,MDX,MDY,MDZ,MX,MY,MZ,I,J,K)
  310 RETURN
      END
```

Table 233: LISTING FOR SUBROUTINE TD

```
CTD
      SUBROUTINE TD(V,VD,IV,P,PD,IP,R,RD,IR,ROL,ROLD,IROL,
     1 YAW,YAWD,IYAW,YD,PDOT,RDOT,VDOT,TX,TZ,XXI,ZZI,XZI,PIT)
C   VERSION 2.                              SEPT.29 1977
C  PURPOSE  THREE DEGREE OF FREEDOM RIGID BODY EQUATIONS OF MOTION(LAT)
C  METHOD  EULER ANGLES
C  CALL SEQUENCE
C  *****  OUTPUTS  *****
C   LINEAR VELOCITIES  -- BODY AXES
C      V,VD,IV      - Y AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C   ANGULAR VELOCITIES -- BODY AXES
C      P,PD,IP      - X AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C      R,RD,IR      - Z AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL, DEG/SEC
C   EULER ANGLES  --  EARTH TO BODY - YAW,PITCH,ROLL
C     ROL,ROLD,IROL - ROLL ANGLE,RATE,INT CONTROL,DEG
C     YAW,YAWD,IYAW - YAW ANGLE,RATE,INT CONTROL,DEG
C   POSITION -- EARTH AXES
C      YD           - Y AXIS LINEAR VELOCITY, FT/SEC
C   ANGULAR ACCELERATION  --  BODY AXES
C      PDOT         - X AXIS ANGULAR ACCELERATION, DEG/SEC2
C      RDOT         - Z AXIS ANGULAR ACCELERATION, DEG/SEC2
C  *****  INPUTS  *****
C   LINEAR ACCELERATION  --  BODY AXES
C      VDOT         - Y AXIS LINEAR ACCELERATION, FT/SEC2
C   MOMENTS
C      TX,TZ        - X,Z  AXIS TORQUES, FTLBS
C   MOMENTS OF INERTIA
C      XXI,ZZI      - X,Z  AXIS MOMENTS OF INERTIA, SLUG-FT2
C      XZI          - PRODUCT OF INERTIA, SLUG-FT2
C   EULER ANGLE  --  EARTH TO BODY - PITCH
C     PIT            - PITCH ANGLE,DEG
C    WRITTEN BY   M.K. WAHI                            MARCH 1977
C
      DATA RPD,DPR /.01745329,57.29578/
      CP=COS(PIT*RPD)
      SP=SIN(PIT*RPD)
      CR=COS(ROL*RPD)
      SR=SIN(ROL*RPD)
C ****************  LINEAR VELOCITY EQUATIONS  ***************
      IF(IV.NE.0)VD=VDOT
C ****************  ANGULAR VELOCITY EQUATIONS  *************
      IF(XZI.NE.0..AND.XZI.NE. .99999) GO TO 100
      IF(IP.NE.0)PD=(TX/XXI)*DPR
      IF(IR.NE.0)RD=(TZ/ZZI)*DPR
      GO TO 160
  100 IF(IP+IR.EQ.0)GO TO 160
      TEM=ZZI/XZI
      DIV=XXI*TEM-XZI
      IF(IP.NE.0)PD=((TX*TEM+TZ)/DIV)*DPR
      TEM=XXI/XZI
      IF(IR.NE.0)RD=((TX+TZ*TEM)/DIV)*DPR
C ****************  EULER ANGLE EQUATIONS  *************
  160 IF(CP.NE.0.)PSID=R*CR/CP
      IF(IYAW.NE.0)YAWD=PSID
      IF(IROL.NE.0)ROLD=P+PSID*SP
C ****************  POSITION EQUATIONS  **************
```

Table 233: LISTING FOR SUBROUTINE TD (CONCLUDED)

```
C -----       TEST IF X AND Y  (LATITUDE AND LONGITUDE) ARE BOTH FROZEN
      SPSR=SP*SR
      CY=COS(YAW*RPD)
      SY=SIN(YAW*RPD)
      YD= (CY*CR+SY*SPSR)*V
C ****************  ANGULAR ACCELERATIONS(FOR OUTPUT PURPOSES ONLY) ****
      PDOT=PD
      RDOT=RD
      RETURN
      END
```

Table 234: LISTING FOR SUBROUTINE TERRA

```
CTERRA
      FUNCTION TERRA(X,AMODE,ANR,D,H,Z)
C
C   VERSION 1.                  NOV.21 1977
C
C   PURPOSE  1. TO SIMULATE A ROUGH TERRAIN WITH RANDOM PROFILE
C            2. TO SIMULATE SINUSOIDAL AND (1-COSINE) BUMP PROFILES.
C
C   METHOD   1. THE VERTICAL ELEVATION IS STORED IN TABULAR
C               - FORM AS A FUNCTION OF POSITION.
C            2. DIRECT EVALUTION OF TRIGONOMETERIC EQN.
C
C
C ** INPUTS **
C      IMODE   -INDICATOR FOR TYPE OF SURFACE
C           =0 -DEFINES A FLAT SURFACE OF ZE=0.
C           =1 -DEFINES (1-COSINE) OR SINUSOIDAL SURFACE
C           =2 -DEFINES PROFILE IN TABULAR FORM
C
C   FOR IMODE = 1
C     N        -NUMBER OF SEQUENTIAL (1-COSINE) BUMPS
C     D        -LENGTH OF BUMP, FEET
C     H        -HEIGHT OF BUMP (NEGATIVE H SPECIFIES A DIP),INCHES
C   ANY NUMBER OF SEQUENTIAL (1-COSINE) BUMPS MAY BE SPECIFIED
C   PROFILE BEGINS AT EARTH AXIS ORIGIN
C   CONTINUOUS SINUSOID IS REPRESENTED BY A LARGE NUMBER OF *
C                                          * (1-COSINE) BUMPS
C
C   FOR IMODE = 2
C     N        -NUMBER OF DATA POINTS IN PROFILE DEFINITION
C     D        -INCREMENTAL DISTANCE BETWEEN POINTS, FEET
C     H        -CONSTANT ELEVATION SCALING FACTOR (ZBAR=H*Z(XE))
C   PROFILE DATA MUST BE SPECIFIED AT EVEN INCREMENTS(DL)
C   LINEAR INTERPOLATION BETWEEN POINTS
C   ELEVATION OUTSIDE OF DEFINED REGION IS VALUE OF NEAREST POINT
C     Z(I)     -VECTOR CONTAINING ELEVATION DEFINITION, INCHES
C
C WRITTEN BY  J.R.KILNER
      DIMENSION Z(1)
      N1=ANR
      N2=ANR+3
      IMODE=AMODE
      XE=.08333*X
      IF(IMODE.EQ.0) GO TO 11
C
      IF(IMODE.EQ.1) GO TO 44
C
      IM1=XE/D
      I=IM1+4
C
      IF(I.LT.4) GO TO 22
C
      IF(I.GE.N2) GO TO 33
      XI=IM1*D
      DX=XE-XI
      TERRA=((Z(I+1)-Z(I))*DX/D+Z(I))*H
```

Table 234: LISTING FOR SUBROUTINE TERRA (CONCLUDED)

```
      GO TO 99
C
11    TERRA=0.
      GO TO 99
C
22    TERRA=Z(4)*H
      GO TO 99
C
33    TERRA=Z(N2)*H
      GO TO 99
C
44    IF(XE.LE.0..OR.XE.GE.N1*D) GO TO 11
      TERRA=0.5*H*(1.-COS(6.2832*XE/D))
C
99    CONTINUE
C
      RETURN
      END
```

Table 235: LISTING FOR SUBROUTINE TF

```
CTF
      SUBROUTINE TF(X1,X1DOT,IX1,FO,FODOT,IFO,FIN,Z0,Z1,P0,P1)
C
C    PURPOSE - TO SIMULATE A SECOND ORDER TRANSFER FUNCTION WITH
C              FIRST ORDER NUMERATOR
C
C                            FO       Z1*S + Z0
C                           ----- = -----------
C                            FIN       2
C                                     S + P1*S+P0
C
C    METHOD  - SELF EXPLANATORY
C
C
C    LIMITATIONS   -   NONE
C
C
C    WRITTEN BY       ADAM LLOYD          LATEST REVISION  NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    X1         INTERMEDIATE STATE VARIABLE        ANY       OUTPUT STATE
C    X1DOT      STATE VARIABLE DERIVATIVE          ANY       OUTPUT STATE
C    IX1        INTEGRATOR CONTROL                 ---       PROGRAM VAR
C    FO         TRANSFER FUNCTION OUTPUT           ANY       OUTPUT STATE
C    FODOT      TRANSFER FUNCTION OUTPUT DERIV.    ANY       OUTPUT STATE
C    IFO        INTEGRATOR CONTROL                 ---       PROGRAM VAR
C    FIN        TRANSFER FUNCTION INPUT            ANY       INPUT   VAR
C    Z0         NUMERATOR COEFFICIENT              ANY       INPUT   VAR
C    Z1         NUMERATOR COEFFICIENT              ANY       INPUT   VAR
C    P0         DENOMINATOR COEFFICIENT            1/SEC2    INPUT   VAR
C    P1         DENOMINATOR COEFFICIENT            1/SEC     INPUT   VAR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      IF(IX1.NE.0)X1DOT=Z0*FIN-P0*FO
      IF(IFO.NE.0)FODOT=X1+Z1*FIN-P1*FO
      RETURN
      END
```

Table 236: LISTING FOR SUBROUTINE TG

```
CTG
      SUBROUTINE TG(FX,FY,FZ,TX,TY,TZ,TH,GAMX,GAMY,GAMZ,XO,YO,ZO)
C   VERSION 1.                                    REVISED: SEPT 20 19766
C   PURPOSE:     TRANSFORM ENGINE THRUST INTO BODY AXIS FORCES AND
C                TORQUES
C   CALL SEQUENCE:
C   ******  OUTPUTS  ******
C     FX,FY,FZ        -X,Y,Z AXIS FORCES ABOUT C.G., LBS
C     TX,TY,TZ        -X,Y,Z AXIS TORQUES, FT-LBS
C   ******  INPUTS  ******
C     TH             -ENGINE THRUST IN LBS
C     GAMX,GAMY,GAMZ -X,Y,Z AXIS DIRECTION COSINES
C     XO,YO,ZO       - THRUST LOCATION COMPONENTS , FT
C   DESIGNED BY A.W. WARREN                        SEPT. 1976
C
      FX= TH*GAMX
      FY= TH*GAMY
      FZ= TH*GAMZ
      TX= YO*FZ - ZO*FY
      TY= ZO*FX - XO*FZ
      TZ= XO*FY - YO*FX
C
      RETURN
      END
```

Table 237: LISTING FOR SUBROUTINE TK

```
CTK
      SUBROUTINE TK(ABL,XYZ,DSM,IAL,REL,ZTR,
     1  FXT,FYT,FZT,TXT,TYT,TZT,PT,PTD,IPT,
     2 VT,VTD,IVT,PC,PCD,IPC,VC,VCD,IVC,WTA,WCA,WTC,
     3 ROL,PIT,YAW,X,ALT,U,V,W,PA,WCU,TCU,WTR,TTR,ANE,CDGAP,
     4 ANSET,ANPTS,BST,WLT,CDH1,CDH2,CDA,BSCG,WLCG,TAU,P,Q,R,
     5 AMODE,ANR,DL,H,DMP,EPC,VU,CAV)
C
C   VERSION 6.                                  FEB.17 1978
C
C   PURPOSE - TO DEVELOP AN INELASTIC TYPE ACLS TRUNK MODEL
C
C   METHOD    THE TRUNK HAS BEEN DIVIDED INTO 2*NE SEGMENTS, NE ON THE
C             LEFT HAND SIDE AND NE ON THE RIGHT HAND SIDE.
C             MEMBRANE MODEL
C             CONCEPT(REF: AFFDL-TR-71-50) IS USED FOR MODELING SIDE
C             ELEMENTS WHILE FRONT AND AFT ELEMENTS ARE MODELED WITH A
C             USER OPTION FOR (A)RESTRAINED MEMBRANE MODEL,(B)FROZEN SHAPE
C             MODEL. BOTH HAVE TWO ATTACHMENT POINTS ON THE FUSELAGE
C
      DIMENSION A(6),B(6),LO(6),XA(25),YA(25),ZA(25),D(25),S(25),
     *    BET(25),IS(25),AP(25),LP(25),LH(25),MU(25),REL(1)
      DIMENSION XBA(25),YBA(25),ZBA(25),RA(25),ITYPE(6),SNB(25),CSB(25)
      DIMENSION ABL(1),XYZ(1),DSM(1),IAL(1)
      DIMENSION XMU(4),ZTR(1)
C
      REAL LO,LP,LH,MU,L3P,L3,L1,IAL
C
      COMMON/SECT/AZO(10,6),AYO(10,10,6),AL1(10,10,6),AL3P(10,10,6),
     *              AL3(10,10,6),AACV(10,10,6),AAS(10,10,6)
      COMMON/CIO/IREAD,IWRITE,IDIAG
      COMMON/CTIME/TIME
      COMMON/CXDOT/XD(1)
      COMMON/CDIFS/JST,KIN,TP
      COMMON/COVRLY/INST
C
      DATA RG,NA,TEST2/53.34,10,0./
C
C
C   CALL SEQUENCE
C
C ***** INPUT TABLES OR DATA ARRAYS *****
C     A,B,LO          -ARRAYS OF TRUNK ELEMENT DIMENSIONS; ATTACH
C                     -POINT SPACING,ATTACH POINT HEIGHT AND
C                     -MEMBRANE CIRCUMFERENCE RESPECTIVELY, INCHES
C      ITYPE          -ARRAY OF INDICATOR FOR TRUNK ELEMENT TYPE
C       =0            -FROZEN TRUNK MODEL
C       =1            -MEMBRANE TRUNK MODEL
C
C     XA,YA,ZA        -ARRAYS OF COORDINATES OF TRUNK ELEMENT
C                     -INBOARD ATTACH POINT, INCHES
C     BET             -ARRAY OF ELEMENT ANGLES  ALSO SPECIFIES WHETHER
C                     -ELEMENT IS AN END(FRON,AFT) OR A SIDE ELEMENT, DEG
C
C     D               -ARRAY OF ELEMENT WIDTH, INCHES
C     S               -ARRAY OF ELEMENT SCALING FACTORS
```

```
C      MU                -ARRAY OF ELEMENT COEFICIENTS OF FRICTION
C                        -IN X AND Y AXIS RESPECTIVELY
C
C      IS                -ARRAY OF PARAMETER SET NUMBERS ASSOCIATED
C                        -WITH EACH ELEMENT
C      AP                -ARRAY OF ORIFICE AREA PER UNIT AREA
C                        -(OR POROSITY) OF TRUNK SURFACE
C      LP                -ARRAY OF CIRCUMFERENTIAL DISTANCE FROM OUTBD
C                        -ATTACH POINT TO BEGINNING OF PERFORATIONS, INCHES
C      LH                -ARRAY OF WIDTH OF PERFORATED AREA, INCHES
C
C       REL              -RELIEF VALVE AREA OPENING AS A FUNCTION OF
C                        -TRUNK PRESSURE ,ONE DIM.TABLE,SQ.IN VS PSIA
C
C    ***** OUTPUTS *****
C      FXT,FYT,FZT       -X,Y,Z AXIS,AXIAL,LATERAL AND VERTICAL FORCE
C                        -SUMMATION TERMS, LBS
C      TXT,TYT,TZT       -X,Y,Z AXIS SUMMATION TERMS FOR ROLL,PITCH,
C                        -AND YAW MOMENTS, FT-LB.
C      PT,PTD,IPT        -TRUNK PRESSURE,RATE,INT CONTROL, PSIA
C      VT,VTD,IVT        -TRUNK VOLUME,RATE,INT CONTROL, CU FT
C      PC,PCD,IPC        -CUHSION PRESSURE,RATE,INT CONTROL, PSIA
C      VC,VCD,IVC        -CUSHION VOLUME,RATE,INT CONTROL, CU FT
C      WTA               -AIR FLOW RATE,TRUNK TO ATMOSPHERE, LB/MIN
C      WCA               -AIR FLOW RATE,CUSHION TO ATMOSPHERE, LB/MIN
C      WTC               -AIR FLOW RATE, TRUNK TO CUSION, LB/MIN
C
C    *****  INPUTS  *****
C      ROL,PIT,YAW       -ROLL,PITCH,YAW EULER ANGLES, DEG
C      X,ALT             -X,Z EARTH AXIS POSITIONS
C      U,V,W             -X,Y,Z BODY AXIS LINEAR VELOCITIES, FT/SEC
C      PA                -AMBIENT PRESSURE, PSIA
C      WCU               -SUPPLY AIR FLOW RATE TO CUSHION CAVITY, LB/MIN
C      TCU               -TEMPERATURE OF WCU AIR, DEGR
C      WTR               -SUPPLY AIR FLOW RATE TO TRUNK, LB/MIN
C      TTR               -TEMPERATURE OF WTR AIR, DEGR
C      NE(ANE)           -NUMBER OF ELEMENTS PER TRUNK SIDE
C                        -SYMMETRIC MODEL IF ANE.LT.0
C      NSET              -NUMBER OF ELEMENT SHAPES OR  PARAM. SETS
C      CDGAP             -DISCHARGE COEFF. FOR FLOW THROUGH GAP
C                        -BETWEEN TRUNK AND GROUND
C      CDA               - DISCHARGE COEFF. FOR FLOW THROUGH
C                        -RELIEF VALVE
C      NPTS              -NO. OF ELEMENTS IN A ROW OR COLUMN IN THE
C                        - PARAMETER SET
C      BST,WLT           -BODY STATION AND WATER LINE OF TRUNK AXIS, INCHES
C      CDH1              -ORIFICE DISCHARGE COEFFICIENT FOR FREE
C                        -PORTION OF TRUNK
C      CDH2              -ORIFICE DISCHARGE COEFFICIENT FOR TRUNK AREA
C                        -IN CONTACT WITH THE GROUND
C      BSCG,WLCG         -BODY STATION AND WATER LINE OF C.G., INCHES
C      TAU               -TIME CONSTANT FOR TRUNK AND CUSHION VOLUME
C                        -RATE OF CHANGE, SEC
C      P,Q,R             -X,Y,Z BODY AXIS ANGULAR VELOCITIES,DEG/SEC
C      AMODE,ANR,DL,H-TERRAIN MODEL PARAMETERS,SEE FUNCTION TERRA
C      DMP               -DAMPING COEFFICIENT AS A FUNCTION OF
```

```
C                         -FLATTENED AREA, LB-SEC/IN./SQ IN.
C      EPC                -PRINT CONTROL,  EPC=1. PRINT ELEMENT VARIABLE
C                         -VALUES EVERY PRINT INTERVAL
C      VU                 -BREAK POINT IN MU-VELOCITY CURVE,IN/SEC
C      CAV                EFFECTIVE AREA FOR TRUNK TO CUSHION VENT,SQ IN
C
C *** CONSTANTS ***
C      RG=53.34            -GAS CONSTANT FOR AIR, FT-LB/LB/DEGR
C
C    WRITTEN BY    J.R.KILNER AND M.K.WAHI
C
C  INITIAL CALCULATIONS AND ELEMENT SECTION PROPERTY
C  PARAMETER DATA CALCULATED AT TIME=0
C  IF INPUT DATA ARE UPDATED
C
      IF(TIME.NE.0.)GOTO 11
C
      NSET=ANSET
      NE=ABS(ANE)
      NPTS=ANPTS
      NSET2=4*NSET+3
      TEST=0.
      DO 4 I=4,NSET2
4     TEST=TEST+ABL(I)
C
      NE2=4*NE+3
      DO 6 I=4,NE2
6     TEST=TEST+XYZ(I)+IAL(I)
C
      NE3=3*NE+3
      DO 7 I=4,NE3
7     TEST=TEST+DSM(I)
C
      IF(TEST.EQ.TEST2)GO TO 11
      TEST2=TEST
C
      MS=2
      IF(ANE.LT.0.)MS=1
C
      DO 10 I=1,NSET
      A(I)=ABL(4*I)
      B(I)=ABL(4*I+1)
      LO(I)=ABL(4*I+2)
      ITYPE(I)=ABL(4*I+3)
 10   CONTINUE
C
      DO 20 I=1,NE
      I3=I*3
      I4=I*4
      XA(I)=XYZ(I4)
      YA(I)=XYZ(I4+1)
      ZA(I)=XYZ(I4+2)
      SET(I)=XYZ(I4+3)
      D(I)=DSM(I3+1)
      S(I)=DSM(I3+2)
      MU(I)=DSM(I3+3)
```

```
      IS(I)=IAL(I4)
      AP(I)=IAL(I4+1)
      LP(I)=IAL(I4+2)
      LH(I)=IAL(I4+3)
C
      XBA(I)= BSCG-BST+XA(I)
      YBA(I)= YA(I)
      ZBA(I)= WLCG-WLT+ZA(I)
C
      SNB(I)=SIN(.01745*BET(I))
      CSB(I)=COS(.01745*BET(I))
      RA(I)=YA(I)/CSB(I)
 20   CONTINUE
C
      DSUM=0.
      DO 25 I=1,NE
25    DSUM=DSUM+D(I)
      DSUM=DSUM*2.
C
      XMU(1)=0.
      XMU(2)=VU
      XMU(3)=0.
      XMU(4)=1.
C
C
      WRITE(IWRITE,6000)
      WRITE(IWRITE,6002)(I,XA(I),YA(I),ZA(I),RA(I),BET(I),D(I),S(I),
     *  MU(I),IS(I),AP(I),LP(I),LH(I),I=1,NE)
6000  FORMAT(1H1,31H***** TRUNK PARAMETER DATA ****,16(5H*****)////
     * 37H  ELEMENT        XA          YA          ZA,
     * 8X,23HRA       BET       D        S,
     * 6X,32H MU      IS      AP          LP       LH/)
6002  FORMAT(4X,I2,2X,4F10.2,F8.2,F7.2,F8.3,F8.3,I4,F9.5,2F7.1/)
C
      CALL IC(NSET,NPTS,ITYPE,A,B,LO,DPR)
C
 11   CONTINUE
C
C
C  INITIALIZATION FOR LOOP ITERATION
C
      AGAP=0.
      FXT =0.
      FYT =0.
      FZT=0.
      TXT =0.
      TYT =0.
      TZT =0.
      VCS =0.
      VTS =0.
      AHA1=0.
      AHA2=0.
      AHC1=0.
      AHC2=0.
C
      CR=COS(.01745*ROL)
```

```
      CP=COS(.01745*PIT)
      CY=COS(.01745*YAW)
      SR=SIN(.01745*ROL)
      SP=SIN(.01745*PIT)
      SY=SIN(.01745*YAW)
C
      CPCY=CP*CY
      CRSY=CR*SY
      SRSPCY=SR*SP*CY
      SYSR=SY*SR
      CYCRSP=CY*CR*SP
      CPSR=CP*SR
      CRCP=CR*CP
      SPSR=SP*SR
      SPCR=SP*CR
C
      P1=P*.01745
      Q1=Q*.01745
      R1=R*.01745
C
      U1=U*12.
      V1=V*12.
      W1=W*12.
C
C **  M=1 FOR RIGHT HAND SIDE
C **  M=2 FOR LEFT HAND SIDE   **
C **  I=TRUNK ELEMENT NUMBER   **
C **  E= +1 WHEN M= (RHS)
C **  E= -1 WHEN M=2 (LHS)
C
      DO 30  M=1,MS
      E=1.
      IF(M.EQ.2) E=-1.
      DO 30  I=1,NE
C
C **  TEST FOR SIDE OR END ELEMENT AND CALCULATE CUSHION-
C **  -TO- TRUNK PRESSURE RATIO
      PR =0.
      IF(BET(I).EQ.0.) PR=(PC-PA)/(PT-PA)
C
C ** FS REFERS TO FREE SHAPE VALUES WHICH ARE FUNCTION OF ONLY PR
C
      ZOFSU = TBL1(PR,AZO,DPR,NPTS,IS(I),NA)
      ZOFS =S(I)*ZOFSU
      YOFS =S(I)*TBL2(PR,ZOFSU,ZOFSU,AYO,DPR,NPTS,IS(I),NA)
C
C
      XBT=XBA(I)+YOFS*SNB(I)
      YBT=E*(YBA(I)+YOFS*CSB(I))
      ZBT=ZBA(I)+ZOFS
C
C  DETERMINE X AND Z POSITION OF POINT T IN EARTH COORDINATES
C
      XET=X*12.+XBT*CPCY+YBT*(SRSPCY-CRSY)+ZBT*(SYSR+CYCRSP)
      ZET=-ALT*12.-XBT*SP+YBT*CPSR+ZBT*CRCP
C
```

```
C  DETERMINE TERRAIN ELEVATION AT POINT T
C
      ZEG =TERRA(XET,AMODE,ANR,DL,H,ZTR)
C
C CALCULATE TRUNK-GROUND GAP HEIGHT.
C  NEGATIVE GAP IMPLIES A LOADED TRUNK
C
      ZGAP=-ZEG-ZET
      Z0=Z0FS+ZGAP
      IF(ZGAP.GT.0.) GO TO 44
C
C  LOADED SHAPES
C
C  CALCULATE TRUNK ELEMENT DIMENSIONS AND AREAS FOR A
C  LOADED SHAPE
C
      Z0U=Z0/S(I)
      Y0=S(I)*TBL2(PR,Z0U,Z0FSU,AY0,DPR,NPTS,IS(I),NA)
      L1=S(I)*TBL2(PR,Z0U,Z0FSU,AL1,DPR,NPTS,IS(I),NA)
      L3=S(I)*TBL2(PR,Z0U,Z0FSU,AL3,DPR,NPTS,IS(I),NA)
      L3P=L3
      IT=IS(I)
      IF(ITYPE(IT).EQ.0)L3P=S(I)*TBL2(PR,Z0U,Z0FSU,
     *                        AL3P,DPR,NPTS,IS(I),NA)
      AS=S(I)*S(I)*TBL2(PR,Z0U,Z0FSU,AAS,DPR,NPTS,IS(I),NA)
      ACV=S(I)*S(I)*TBL2(PR,Z0U,Z0FSU,AACV,DPR,NPTS,IS(I),NA)
C
C  TRUNK GROUND REACTION
C
      AT=D(I)*L3
      FTBAR=(PT-PA)*AT
C
C  DETERMINE VELOCITY OF POINT T RELATIVE TO EARTH
C  WITH X AND Y COMPONENTS ORIENTED TO BODY AXIS
C
      XBT=XBA(I)+(Y0+0.5*L3)*SNB(I)
      YBT=E*(YBA(I)+(Y0+0.5*L3)*CSB(I))
      ZBT=ZBA(I)+Z0
C
      XBTD= ZBT*Q1-YBT*R1+U1
      YBTD=-ZBT*P1+XBT*R1+V1
      ZBTD= YBT*P1-XBT*Q1+W1
C
      XTD2=XBTD*CP+YBTD*SPSR+ZBTD*SPCR
      YTD2=YBTD*CR-ZBTD*SR
      ZTD=-XBTD*SP+YBTD*CPSR+ZBTD*CRCP
C
      VET=SQRT(XTD2*XTD2+YTD2*YTD2)
C
C
C  CALCULATE ELEMENT  FRICTION FORCES
C
      IF(VET.EQ.0.)GO TO 34
      UTO=MU(I)*TBLU1(VET,XMU,XMU(3),1,-2)
      UTX=UTO*XTD2/VET
      UTY=UTO*YTD2/VET
```

```
      GO TO 36
34    UTX=0.
      UTY=0.
36    CONTINUE
C
      FFXBAR=-UTX*FTBAR
      FFYBAR=-UTY*FTBAR
C
C  ** CALCULATE ELEMENT DAMPING FORCE
C
      FDBAR=DMP*AT*ZTD
C ** CALCULATE FORCES AND MOMENTS
C
C
      FXT=FXT+FFXBAR
      FYT=FYT+FFYBAR
      FZT=FZT-FTBAR-FDBAR
      TXT=TXT+(-(FTBAR+FDBAR)*YBT-FFYBAR*ZBT)*.08333
      TYT=TYT+((FTBAR+FDBAR)*XBT+FFXBAR*ZBT)*.08333
      TZT=TZT+(FFYBAR*XBT-FFXBAR*YBT)*.08333
      GO TO 66
C
C  CALCULATE TRUNK ELEMENT DIMENSION AND AREAS FOR FREE SHAPE
C
44    CONTINUE
      AGAP=AGAP+ZGAP*D(I)
      YO=S(I)*TBL2(PR,ZOFSU,ZOFSU,AYO,DPR,NPTS,IS(I),NA)
      L1=S(I)*TBL2(PR,ZOFSU,ZOFSU,AL1,DPR,NPTS,IS(I),NA)
      AS=S(I)*S(I)*TBL2(PR,ZOFSU,ZOFSU,AAS,DPR,NPTS,IS(I),NA)
      ACV=S(I)*S(I)*TBL2(PR,ZOFSU,ZOFSU,AACV,DPR,NPTS,IS(I),NA)
      FFXBAR=0.
      FFYBAR=0.
      FDBAR=0.
      FTBAR=0.
      L3=0.
 66   CONTINUE
C
C  TEST FOR END OR SIDE ELEMENT
C
      IF(BET(I).EQ.0.) GO TO 77
C
      YBC=E*(YBA(I)+(.6667*YO-.3333*RA(I))*CSB(I))
      AC=.5*D(I)*(YO+RA(I))**2/(YOFS+RA(I))
      VCS=VCS+(ZO*AC-D(I)*ACV*(.5*YO+RA(I))/(YO+RA(I)))*.0005787
      GO TO 88
C
 77   CONTINUE
      YBC=0.5*E*(YBA(I)+YO)
      AC=D(I)*(YA(I)+YO)
      VCS=VCS+(ZO*AC-D(I)*ACV)*.0005787
 88   CONTINUE
C
      VTS=VTS+D(I)*AS*.0005787
C
      XBC=XBA(I)+(.6667*YO-.3333*RA(I))*SNB(I)
      FCBAR= (PC-PA)*AC
```

```
      FZT=FZT-FCBAR
      TXT=TXT-FCBAR*YBC*.08333
      TYT=TYT+FCBAR*XBC*.08333
C
      IF(AP(I).NE.0.) CALL PERF(ZGAP,L1,L3,L3P,LP(I),LH(I),RA(I),YO,
     1   YOFS,D(I),AP(I),PT,PC,PA,BET(I),AHA1,AHA2,AHC2,AHC1)
C
      IF(EPC.EQ.1.)CALL VPRINT(1,I,M,MS,NE,ZGAP,ZO,YO,L1,L3,VTS,VCS,
     *          FFXBAR,FFYBAR,FDBAR,FTBAR,FCBAR,AGAP)
 30   CONTINUE
C
C CALCULATE FLOW RATES
C
      CACA=CDGAP*AGAP
      CATA=CDH1*AHA1+.6667*CDH2*AHA2
      CATC=CDH1*AHC1+.6667*CDH2*AHC2+CAV
C
C ** TEST FOR SYMMETRIC MODEL
C
      IF(ANE.GT.0.)GOTO 166
      FXT=2.*FXT
      FYT=0.
      FZT=2.*FZT
      TXT=0.
      TYT=2.*TYT
      TZT=0.
      VCS=2.*VCS
      VTS=2.*VTS
      CACA=2.*CACA
      CATA=2.*CATA
      CATC=2.*CATC
166   CONTINUE
C
C
C ** CALCULATE RELIEF VALVE AREA
C
      N=REL(2)
      AREL = TBLU1(PT-PA,REL(4),REL(N+4),1,-N)
      CATA=CATA+CDA*AREL
C
C ** CALCULATE CUSHION VOLUME RATE OF CHANGE
C
      IF(IVC.NE.0) VCD=(VCS-VC)/TAU
C
C ** CALCULATE TRUNK AIR VOLUME RATE OF CHANGE
C
      IF(IVT.NE.0) VTD=(VTS-VT)/TAU
C
      CALL FSFLOW(PC,PA,TCU,CACA,1.,FN,SFN,WCA)
      CALL FNFLOW(PT,PA,TTR,CATA,1.,FN,WTA)
      CALL FNFLOW(PT,PC,TTR,CATC,1.,FN,WTC)
C
C ** CALCULATE CUSHION PRESSURE RATE OF CHANGE
C *** INTEGRATE CUSHION PRESSURE ***
C
      PCD1=(.0001389*RG*TCU*(WCU+WTC-WCA)-1.2*PC*VCD)/VC
```

```
      IF(IPC.NE.0)PCD=PCD1
C
C ** CALCULATE TRUNK AIR PRESSURE RATE OF CHANGE
C *** INTEGRATE TRUNK PRESSURE ***
C
      IF(IPT.NE.0)PTD=(.0001389*RG*TTR*(WTR-WTC-WTA)-1.2*PT*VTD)/VT
C
      IF(EPC.EQ.1.)CALL VPRINT(0,I,M,MS,NE,ZGAP,Z0,Y0,L1,L3,VTS,VCS,
     *          FFXBAR,FFYBAR,FDBAR,FTBAR,FCBAR,AGAP)
C
      RETURN
      END
```

Table 238: LISTING FOR SUBROUTINE TL

```
CTL
      SUBROUTINE TL(U,UD,IU,W,WD,IW,Q,QD,IQ,PIT,PITD,IPIT,XD,
     1 Z,ZD,IZ,QDOT,UDOT,WDOT,TY,YYI,ROL,YAW)
C  VERSION 1.                                   APRIL 28, 1977.
C  PURPOSE   THREE DEGREE OF FREEDOM RIGID BODY EQUATIONS OF MOTION(LON)
C  METHOD   EULER ANGLES
C  CALL SEQUENCE
C  *****  OUTPUTS  *****
C   LINEAR VELOCITIES  --  BODY AXES
C      U,UD,IU      - X AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C      W,WD,IW      - Z AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C   ANGULAR VELOCITIES --  BODY AXES
C      Q,QD,IQ      - Y AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL,DEG/SEC
C   EULER ANGLE   -- EARTH TO BODY - PITCH
C      PIT,PITD,IPIT - PITCH ANGLE,RATE,INT CONTROL, DEG
C   POSITION  --  EARTH AXES
C      XD           - X AXIS LINEAR VELOCITY, FT/SEC
C      Z,ZD,IZ      - -Z AXIS POSITION (ALT),VELOCITY,INT CONTROL, FT
C   ANGULAR ACCELERATION  --  BODY AXES
C      QDOT         - Y AXIS ANGULAR ACCELERATION, DEG/SEC2
C ****************   INPUTS   ****************
C   LINEAR ACCLERATIONS --  BODY AXES
C      UDOT         - X AXIS LINEAR ACCELERATION, FT/SEC2
C      WDOT         - Z AXIS LINEAR ACCELERATION, FT/SEC2
C   MOMENTS
C      TY           - Y AXIS TORQUE, FTLBS
C   MOMENT OF INERTIA
C      YYI          - Y AXIS MOMENTS OF INERTIA, SLUG-FT2
C   EULER ANGLES  -- EARTH TO BODY - ROLL,YAW
C      ROL          - ROLL ANGLE, DEG
C      YAW          - YAW ANGLE, DEG
C     WRITTEN BY    MAHINDER WAHI                     APRIL 1977
C
      DATA RPD,DPR /.01745329,57.29578/
      CP=COS(PIT*RPD)
      SP=SIN(PIT*RPD)
      CR=COS(ROL*RPD)
      SR=SIN(ROL*RPD)
      Q1=Q*RPD
C ****************   LINEAR VELOCITY EQUATIONS   ************
      IF(IU.NE.0)UD=UDOT
      IF(IW.NE.0)WD=WDOT
C ****************   ANGULAR VELOCITY EQUATION   ************
      IF(IQ.NE.0)QD=(TY/YYI)*DPR
C ****************   EULER ANGLE EQUATION   ************
      IF(IPIT.NE.0)PITD=Q*CR
C ****************   POSITION EQUATIONS   ***************
C ----   TEST IF X AND Y (LATITUDE AND LONGITUDE) ARE BOTH BROZEN
      SPCR=SP*CR
      CY=COS(YAW*RPD)
      SY=SIN(YAW*RPD)
      XD=CY*CP*U+(SY*SR+CY*SPCR)*W
      IF(IZ.NE.0)ZD=SP*U-CP*CR*W
C ********** ANGULAR ACCELERATION (FOR OUTPUT PURPOSES ONLY) *****
      QDOT=QD
      RETURN
```

Table 239: LISTING FOR SUBROUTINE TR

```
CTR
      SUBROUTINE TR(PXE,PYE,PZE,PXB,PYB,PZB,ROL,PIT,YAW)
C--VERSION 1  JULY 6 1977
C
C--PURPOSE   TRANSFORM VECTOR QUANTITIES FROM BODY AXES TO EARTH AXES
C
C--METHOD    MATRIX MULTIPLICATION
C
C--CALL SEQUENCE
C  *****  OUTPUTS  *****
C     PXE,PYE,PZE     VECTOR QUANTITIES ALONG EARTH  X,Y AND Z AXES
C  *****  INPUTS  *****
C     PXB,PYB,PZB     VECTOR COMPONENS OF BODY COORDINATE SYSTEM
C     ROL,PIT,YAW     BODY ROLL,PITCH,AND YAW ANGLES
C
C     WRITTEN BY J.J.MCAVOY AND M.K.WAHI
C
C     CONVERT FROM DEGREES TO RADIANS
      ROL=ROL/57.3
      PIT=PIT/57.3
      YAW=YAW/57.3
C----EARTH LONGITUDINAL COMPONENT
      PXE=PXB*COS(PIT)*COS(YAW)+PYB*(-COS(ROL)*SIN(YAW)+SIN(ROL)*
     2 SIN(PIT)*COS(YAW))+PZB*(SIN(YAW)*SIN(ROL)+COS(YAW)*COS(ROL)*
     3 SIN(PIT))
C----EARTH LATERAL COMPONENT
      PYE=-PXB*COS(PIT)*SIN(YAW)-PYB*(COS(YAW)*COS(ROL)+SIN(YAW)*
     2 SIN(ROL)*SIN(PIT))-PZB*(-COS(YAW)*SIN(ROL)+SIN(YAW)*COS(ROL)*
     3 SIN(PIT))
C----EARTH VERTICAL COMPONENT
      PZE=PXB*SIN(PIT)-PYB*COS(PIT)*SIN(ROL)-PZB*COS(ROL)*COS(PIT)
      RETURN
      END
```

Table 240: LISTING FOR SUBROUTINE TRUNK

```
CTRUNK
      SUBROUTINE TRUNK(ISHAPE)
C TRUNK GEOMETRY CALCULATIONS
      REAL L,L1,L2,LS,LP,MASS
      COMMON/GEOMET/A,B,HYI,L,D,LS,LP,SH,NH,AH,NR,PHI1,PHI2,R1,R2,L1,L2,
     1A1,A2,X1,X2,HY
      COMMON/FLAGS/ICLN,IDIF,IFLAG,ISTAT,IPP,IPRV,MM,NSTOP
      DATA RTOL/0.01/
      IF(HY.LE.0.0) GO TO 11
C***********************************************************************
C ITERATION FOR R2
C COMPUTE INNER RADIUS OF CURVATURE
      R2=SQRT(A*A*0.25+HY*HY)
C
C ITERATION LOOP FOR L2,L1,R1,R2
       DO 10 I=1,50
      PHI2=ABS(ACOS(AMAX1(-1.0,AMIN1(1.0,((R2-HY)/R2)))))
       SINPH2=SIN(PHI2)
C COMPUTE OUTER RADIUS OF CURVATURE
      R1=((A-R2*SINPH2)**2+(B+HY)**2)/(2.*(B+HY))
      PHI1=ABS(ACOS(AMAX1(-1.0,AMIN1(1.0,((R1-HY-B)/R1)))))
       XS=A-R2*SINPH2
       IF (XS.LE.0.0) PHI1=6.2831852-PHI1
       L2=L-PHI1*R1
C R2S IS RESULTANT RADIUS FOR COMPUTED L2 IN ITERATION
      IF(ABS(PHI2) .LT.1.0E-2) PHI2=1.0E-2
       R2S=L2/PHI2
C TEST IF TOLERANCE .GT. ERROR
       IF(ABS(R2-R2S).LE.RTOL) GO TO 50
      R2=(R2+R2S)*0.5
10     CONTINUE
C***********************************************************************
C ITERATED 50 TIMES WITHOUT SUCCESS,ERROR RETURN
11    CONTINUE
       WRITE(6,9001)
9001     FORMAT(10X,* INFEASABLE TRUNK GEOMETRY *//)
      ISHAPE=0
      RETURN
C TRUNK OK,RETURN
50     L1=L-L2
      ISHAPE=1
      RETURN
       END
```

Table 241: LISTING FOR SUBROUTINE TS

```
CTS
      SUBROUTINE TS(ABL,XYZ,DM,IAL,END,SPH,STH,PM,REL,BWT,ZTR,
     1  FXT,FYT,FZT,TXT,TYT,TZT,PT,PTD,IPT,
     2 VT,VTD,IVT,PC,PCD,IPC,VC,VCD,IVC,WTA,WCA,WTC,AREL,CPT,
     3 ROL,PIT,YAW,X,ALT,U,V,W,PA,WCU,TCU,WTR,TTR,ANE,CDGAP,
     4 CDH1,CDH2,CDA,TAU,P,Q,R,
     5 DMP,EPC,VU,PTM,CATV,SPB)
C
C
C    VERSION 2.                          REVISED MAR. 1 1979
C
C    WRITTEN BY   G.S.DULEBA
C
C    PURPOSE - TO DEVELOP AN ELASTIC TYPE ACLS TRUNK MODEL
C
C    METHOD - SEE VOLUME 1
C
C
C
C
C    ARGUMENT LIST
C
C ***** INPUT TABLES OR DATA ARRAYS *****
C      ABL             -ELEMENT SET DATA ARRAY; ATTACH POINT SPACING,
C                       ATTACH POINT HEIGHT, MEMBRANE INSTALLED LENGTH,
C                       INITIAL MERIDIAN AND HOOP STRAINS, AND POISSONS
C                       RATIO
C
C      XYZ             -ELEMENT DATA ARRAY; X,Y,Z COORDINATES OF INBOARD
C                       ATTACH POINT, AND ELEMENT ANGLES
C
C      DM              -ELEMENT DATA ARRAY; ELEMENT WIDTHS AND ELEMENT
C                       COEFFICIENTS OF FRICTION
C
C      IAL             -ELEMENT DATA ARRAY; ELEMENT SET NUMBERS ASSOCIATED
C                      WITH EACH ELEMENT, ORIFICE AREA PER UNIT AREA OF
C                      TRUNK SURFACE,DISTANCE FROM OUTBRD. ATTACH POINT TO
C                      BEGINNING OF PERFORATIONS(IN MEMBRANE FREE STATE),
C                      WIDTH OF PERFORATED AREA(IN MEMBRANE FREE STATE)
C
C      END            -END ELEMENT DATA ARRAY; RADIUS OF INBOARD ATTACH
C                      POINT, AND PARAMETER FOR MATERIAL CONSTRUCTION
C                      WHICH AFFECTS THE STRESS/STRAIN RELATIONSHIP
C
C      SPH             -ELEMENT SET INPUT TABLE(TWO DIMENSIONAL);
C                      MERIDIAN LOADS VS. MERIDIAN STRAINS FOR MEMBRANE
C                      (LBS/INCH VS DIMENSIONLESS RATIO)
C      STH             -ELEMENT SET INPUT TABLE(TWO DIMENSIONAL);
C                      HOOP LOADS VS. HOOP STRAINS FOR MEMBRANE
C                      (LBS/INCH VS DIMENSIONLESS RATIO)
```

```
C      PM               -PILLOW ELEMENT DATA ARRAY; ELEMENT NUMBER
C                        ASSOCIATED WITH EACH PILLOW ELEMENT, COEFFICIENT
C                        OF FRICTION, PILLOW INFLATED HEIGHT, AND RATIO OF
C                        INFLATED PILLOW CONTACT WIDTH TO UNINFLATED WIDTH
C
C      REL              -INPUT TABLE(ONE DIMENSION); RELIEF VALVE OPENING
C                        AREA VS TRUNK PRESSURE (SQ.IN VS PSIA)
C
C      BWT              -MISC. DATA ARRAY; BODY STATION AND WATER LINE OF
C                        TRUNK AXIS, BODY STATION AND WATER LINE OF C.G.,
C                        AND TERRAIN MODEL PARAMETERS(SEE FUNCTION TERRA)
C
C      ZTR              -INPUT TABLE FOR DEFINITION OF GROUND ELEVATION
C                        (SEE FUNCTION TERRA)
C
C
C
C    ***** OUTPUTS *****
C      FXT,FYT,FZT      -X,Y,Z AXIS,AXIAL,LATERAL AND VERTICAL FORCE
C                       -SUMMATION TERMS, LBS
C      TXT,TYT,TZT      -X,Y,Z AXIS SUMMATION TERMS FOR ROLL,PITCH,
C                       -AND YAW MOMENTS, FT-LB.
C      PT,PTD,IPT       -TRUNK PRESSURE,RATE,INT CONTROL, PSIA
C      VT,VTD,IVT       -TRUNK VOLUME,RATE,INT CONTROL, CU FT
C      PC,PCD,IPC       -CUHSION PRESSURE,RATE,INT CONTROL, PSIA
C      VC,VCD,IVC       -CUSHION VOLUME,RATE,INT CONTROL, CU FT
C      WTA              -AIR FLOW RATE,TRUNK TO ATMOSPHERE, LB/MIN
C      WCA              -AIR FLOW RATE,CUSHION TO ATMOSPHERE, LB/MIN
C      WTC              -AIR FLOW RATE, TRUNK TO CUSION, LB/MIN
C      AREL             -RELEIF VALVE OPENING AREA (SQ IN)
C      CPT              -CPU TIME (SEC)
C
C    *****  INPUTS   *****
C      ROL,PIT,YAW      -ROLL,PITCH,YAW EULER ANGLES, DEG
C      X,ALT            -X,Z EARTH AXIS POSITIONS
C      U,V,W            -X,Y,Z BODY AXIS LINEAR VELOCITIES, FT/SEC
C      PA               -AMBIENT PRESSURE, PSIA
C      WCU              -SUPPLY AIR FLOW RATE TO CUSHION CAVITY, LB/MIN
C      TCU              -TEMPERATURE OF WCU AIR, DEGR
C      WTR              -SUPPLY AIR FLOW RATE TO TRUNK, LB/MIN
C      TTR              -TEMPERATURE OF WTR AIR, DEGR
C      NE(ANE)          -NUMBER OF ELEMENTS PER TRUNK SIDE
C                       -SYMMETRIC MODEL IF ANE.LT.0
C      CDGAP            -DISCHARGE COEFF. FOR FLOW THROUGH GAP
C                       -BETWEEN TRUNK AND GROUND
C      CDA              - DISCHARGE COEFF. FOR FLOW THROUGH
C                       -RELIEF VALVE
C      CDH1             -ORIFICE DISCHARGE COEFFICIENT FOR FREE
C                       -PORTION OF TRUNK
C      CDH2             -ORIFICE DISCHARGE COEFFICIENT FOR TRUNK AREA
```

```
C                       -IN CONTACT WITH THE GROUND
C      TAU              -TIME CONSTANT FOR TRUNK AND CUSHION VOLUME
C                       -RATE OF CHANGE, SEC
C      P,Q,R            -X,Y,Z BODY AXIS ANGULAR VELOCITIES,DEG/SEC
C      DMP              -DAMPING COEFFICIENT AS A FUNCTION OF
C                       -FLATTENED AREA, LB-SEC/IN./SQ IN.
C      EPC              -PRINT CONTROL,  EPC=1. PRINT ELEMENT VARIABLE
C                       -VALUES EVERY PRINT INTERVAL
C      VU               -BREAK POINT IN MU-VELOCITY CURVE,IN/SEC
C      PTM              -MAXIMUM TRUNK PRESSURE (PSIG) USED TO GENERATE
C                       MEMBRANE DATA ARRAYS
C      CATV             -EFFECTIVE AREA (SQ IN) OF ORIFICE FOR VARYING
C                       TRUNK-CUSHION FLOW (DEFAULT VALVE = 0.)
C
C      SPB              -ACTUATION SIGNAL FOR PILLOW BRAKE ELEMENTS
C                           =0.  BRAKES OFF
C                           =1.  BRAKES FULLY APPLIED
C
C  *** CONSTANTS ***
C      RG=53.34          -GAS CONSTANT FOR AIR, FT-LB/LB/DEGR
C
C
C    CALCULATION OF TRUNK DATA ARRAYS AND TRANSFER OF DATA
C    FROM INPUT DATA ARRAYS IS ACCOMPLISHED AT INITIAL TIME
C
C
      DIMENSION A(8),B(8),LO(8),XA(25),YA(25),ZA(25),D(25),RD(25),
     *    BET(25),IS(25),AP(25),LP(25),LH(25),MU(25),
     *    MB(25),HB(25),GKD(25),EPI(8),ETI(8),NU(8),FRC(8),RAD(8)
      DIMENSION XBA(25),YBA(25),ZBA(25),RA(25),SNB(25),CSB(25)
      DIMENSION ABL(1),XYZ(1),IAL(1),DM(1),END(1),REL(1),ZTR(1)
      DIMENSION SPH(1),STH(1),PM(1),BWT(1)
C
      REAL LO,IAL,LP,LH,MU,MB,NU,L3,L1,L2
C
      COMMON/CIO/IREAD,IWRITE,IDIAG
      COMMON/STRCH/EYO(2560),EL1(2560),EL3(2560),EVC(2560),EVS(2560),
     2 EE1(2560),EE3(2560),EL2(2560),AZO(512)
      COMMON/CPROV/PV(27)
      COMMON/CTIME/TIME
      COMMON/TSTEST/XXX(10)
C
      DATA RG/53.34/
      CALL SECOND(CPT)
      IF(TIME.NE.PV(27)) GO TO 15
C
      NSET=SPH(3)
      NEND=END(2)
      IF(END(2).EQ.1.99999) NEND=0
      NSIDE=NSET-NEND
```

```
      TEST1=0.
      NX=6*NSET+3
      DO 4 I=2,NX
4     TEST1=TEST1+ABL(I)
C
      NX=SPH(2)*(SPH(3)+1.)+SPH(3)+3.
      DO 5 I=2,NX
 5    TEST1=TEST1+SPH(I)
C
      NX=STH(2)*(STH(3)+1.)+STH(3)+3.
      DO 6 I=2,NX
 6    TEST1=TEST1+STH(I)
C
      NX=2*NEND+3
      DO 7 I=2,NX
 7    TEST1=TEST1+END(I)
      TEST1=TEST1+PTM
      IF(ABL(55).EQ.TEST1) GO TO 8
      ABL(55)=TEST1
      DO 10 I=1,NSET
      I6=6*(I-1)
      A(I)=ABL(I6+4)
      B(I)=ABL(I6+5)
      LO(I)=ABL(I6+6)
      EPI(I)=ABL(I6+7)
      ETI(I)=ABL(I6+8)
      NU(I)=ABL(I6+9)
 10   CONTINUE
C
      IF(NEND.EQ.0) GO TO 25
      DO 24 I=1,NEND
      I2=2*(I-1)
      RAD(I)=END(I2+4)
 24   FRC(I)=END(I2+5)
 25   CONTINUE
C
C   CALL ELAS TO COMPUTE TRUNK DATA ARRAYS
C
      CALL ELAS(NSIDE,NEND,SPH,STH,A,B,LO,EPI,ETI,RAD,NU,FRC,PTM)
C
 8    CONTINUE
C
      NPB=PM(2)/2.
      IF(PM(2).EQ.1.99999) NPB=0
      NE=ABS(ANE)
      IF(SPB.EQ.0.99999) SPB=0.
      IF(CATV.EQ.0.99999) CATV=0.
      TEST2=0.
      DO 9 I=2,11
 9    TEST2=TEST2+BWT(I)
```

```
C
      NX=4*NPB+3
      DO 11 I=2,NX
 11   TEST2=TEST2+PM(I)
C
      NX=2*NE+3
      DO 12 I=2,NX
 12   TEST2=TEST2+DM(I)
C
      NX=4*NE+3
      DO 13 I=2,NX
 13   TEST2=TEST2+IAL(I)+XYZ(I)
      IF(ABL(54).EQ.TEST2) GO TO 15
      ABL(54)=TEST2
C
      MS=2
      IF(ANE.LT.0.)MS=1
C
      BST=BWT(4)
      WLT=BWT(5)
      BSCG=BWT(6)
      WLCG=BWT(7)
      AMODE=BWT(8)
      ANR=BWT(9)
      DL=BWT(10)
      H=BWT(11)
C
C
      DO 20 I=1,NE
      RD(I)=0.
      MB(I)=0.
      HB(I)=0.
      GKD(I)=1.
      I4=4*(I-1)
      XA(I)=XYZ(I4+4)
      YA(I)=XYZ(I4+5)
      ZA(I)=XYZ(I4+6)
      BET(I)=XYZ(I4+7)
      IS(I)=IAL(I4+4)
      AP(I)=IAL(I4+5)
      LP(I)=IAL(I4+6)
      LH(I)=IAL(I4+7)
      I2=2*(I-1)
      D(I)=DM(I2+4)
      MU(I)=DM(I2+5)
C
      XBA(I)= BSCG-BST+XA(I)
      YBA(I)= YA(I)
      ZBA(I)= WLCG-WLT+ZA(I)
C
```

```
      SNB(I)=SIN(.01745*BET(I))
      CSB(I)=COS(.01745*BET(I))
      RA(I)=YA(I)/CSB(I)
      IX=IS(I)
      IF(IX.LE.NEND) RD(I)=RAO(IX)
 20   CONTINUE
C
      IF(NPB.EQ.0) GO TO 28
      DO 27 I=1,NPB
      I4=4*(I-1)
      IP=PM(I4+4)
      MB(IP)=PM(I4+5)
      HB(IP)=PM(I4+6)
 27   GKD(IP)=PM(I4+7)
 28   CONTINUE
      DPT=PTM/9.
C
C
C    PRINT INPUT DATA FOR ALL TRUNK ELEMENTS
C
      WRITE(IWRITE,6000)
      WRITE(IWRITE,6002)(I,XA(I),YA(I),ZA(I),RA(I),BET(I),D(I),RD(I),
     *  MU(I),IS(I),AP(I),LP(I),LH(I),HB(I),GKD(I),MB(I),I=1,NE)
6000  FORMAT(1H1,31H***** TRUNK PARAMETER DATA ****,16(5H*****)////
     * 37H  ELEMENT      XA         YA         ZA,
     * 8X,24HRA      BET      D       RD,
     * 6X,49H MU     IS     AP        LP       LH      HB     KD     MB/)
6002  FORMAT(4X,I2,2X,4F10.2,F8.2,F7.2,F8.3,F8.3,I4,F9.5,2F7.1,2F6.1,
     2 F6.2/)
C
      DO 29 I=1,NE
 29   IF(BET(I).NE.0.) D(I)=D(I)/57.29578
C
C    ********************  END OF INITIAL COMPUTATIONS ***************
 15   CONTINUE
C
C
C  INITIALIZATION FOR LOOP ITERATION
C
      AGAP=0.
      FXT =0.
      FYT =0.
      FZT=0.
      TXT =0.
      TYT =0.
      TZT =0.
      VCS =0.
      VTS =0.
      AHA1=0.
      AHA2=0.
```

```
      AHC1=0.
      AHC2=0.
C
      CR=COS(.01745*ROL)
      CP=COS(.01745*PIT)
      CY=COS(.01745*YAW)
      SR=SIN(.01745*ROL)
      SP=SIN(.01745*PIT)
      SY=SIN(.01745*YAW)
C
      CPCY=CP*CY
      CRSY=CR*SY
      SRSPCY=SR*SP*CY
      SYSR=SY*SR
      CYCRSP=CY*CR*SP
      CPSR=CP*SR
      CRCP=CR*CP
      SPSR=SP*SR
      SPCR=SP*CR
C
      P1=P*.01745
      Q1=Q*.01745
      R1=R*.01745
C
      U1=U*12.
      V1=V*12.
      W1=W*12.
C
C **  M=1 FOR RIGHT HAND SIDE
C **  M=2 FOR LEFT HAND SIDE   **
C **  I=TRUNK ELEMENT NUMBER   **
C **  E= +1 WHEN M=1 (RHS)
C **  E= -1 WHEN M=2 (LHS)
C
      DO 30  M=1,MS
      E=1.
      IF(M.EQ.2) E=-1.
      DO 30  I=1,NE
C
C **  TEST FOR SIDE OR END ELEMENT AND CALCULATE CUSHION-
C **  -TO- TRUNK PRESSURE RATIO
      PTG=PT-PA
      PR=(PC-PA)/PTG
C
C ** FS REFERS TO FREE SHAPE VALUES WHICH ARE FUNCTIONS OF PR AND PT
C
      CALL ETB2(IS(I),PR,PTG,DPT,ZOFS)
      CALL ETB3(IS(I),PR,PTG,DPT,1.,EYO,YOFS)
C
C
```

```
      XBT=XBA(I)+YOFS*SNB(I)
      YBT=E*(YBA(I)+YOFS*CSB(I))
      ZBT=ZBA(I)+ZOFS
C
C  DETERMINE X AND Z POSITION OF POINT T IN EARTH COORDINATES
C
      XET=X*12.+XBT*CPCY+YBT*(SRSPCY-CRSY)+ZBT*(SYSR+CYCRSP)
      ZET=-ALT*12.-XBT*SP+YBT*CPSR+ZBT*CRCP
C
C  DETERMINE TERRAIN ELEVATION AT POINT T
C
      ZEG =TERRA(XET,AMODE,ANR,DL,H,ZTR)
C
C CALCULATE TRUNK-GROUND GAP HEIGHT.
C  NEGATIVE GAP IMPLIES A LOADED TRUNK
C
      ZGAP=-ZEG-ZET-HB(I)*SPB*MB(I)/(.0001+MB(I))
      ZO=ZOFS+ZGAP
      ZC=AMIN1(ZO,ZOFS)
      IF(ZGAP.GT.0.) GO TO 44
      ZR=ZO/ZOFS
C
C  LOADED SHAPES
C
C  CALCULATE TRUNK ELEMENT DIMENSIONS AND AREAS FOR A
C  LOADED SHAPE.  CALL ETB3 TO INTEPOLATE FROM TRUNK DATA ARRAYS.
C  LOADED SHAPE VALUES ARE FUNCTIONS OF PR, PT AND ZR.
C
      CALL ETB3(IS(I),PR,PTG,DPT,ZR,EYO,YO)
      CALL ETB3(IS(I),PR,PTG,DPT,ZR,EL1,L1)
      CALL ETB3(IS(I),PR,PTG,DPT,ZR,EL2,L2)
      CALL ETB3(IS(I),PR,PTG,DPT,ZR,EL3,L3)
      LT=L1+L2+L3
      CALL ETB3(IS(I),PR,PTG,DPT,ZR,EVS,AS)
      CALL ETB3(IS(I),PR,PTG,DPT,ZR,EVC,ACV)
C
C  TRUNK GROUND REACTION
C
      DX=D(I)
      IF(BET(I).NE.0.) DX=(YO+RD(I)+.5*L3)*D(I)
      AT=DX*L3
      AT=SPB*(AT*GKD(I)-AT)+AT
      DAGAP=.5*SPB*HB(I)*DX*(1.-GKD(I))
      AGAP=AGAP+DAGAP
 33   CONTINUE
      FTBAR=(PT-PA)*AT
C
C  DETERMINE VELOCITY OF POINT T RELATIVE TO EARTH
C  WITH X AND Y COMPONENTS ORIENTED TO BODY AXIS
C
```

```
      XBT=XBA(I)+(YO+0.5*L3)*SNB(I)
      YBT=E*(YBA(I)+(YO+0.5*L3)*CSB(I))
      ZBT=ZBA(I)+ZO
C
      XBTD= ZBT*Q1-YBT*R1+U1
      YBTD=-ZBT*P1+XBT*R1+V1
      ZBTD= YBT*P1-XBT*Q1+W1
C
      XTD2=XBTD*CP+YBTD*SPSR+ZBTD*SPCR
      YTD2=YBTD*CR-ZBTD*SR
      ZTD=-XBTD*SP+YBTD*CPSR+ZBTD*CRCP
C
      VET=SQRT(XTD2*XTD2+YTD2*YTD2)
C
C
C   CALCULATE ELEMENT  FRICTION FORCES
C
      IF(VET.EQ.0.)GO TO 34
      UTO=MU(I)*VET/VU
      UTO=AMIN1(UTO,MU(I))
C
      IF(MB(I).EQ.0.) GO TO 35
      UT1=MB(I)*VET/VU
      UT1=AMIN1(UT1,MB(I))
      UTO=UTO+SPB*(UT1-UTO)
 35   CONTINUE
      UTX=UTO*XTD2/VET
      UTY=UTO*YTD2/VET
      GO TO 36
34    UTX=0.
      UTY=0.
36    CONTINUE
C
      FFXBAR=-UTX*FTBAR
      FFYBAR=-UTY*FTBAR
C
C   ** CALCULATE ELEMENT DAMPING FORCE
C
      FDBAR=DMP*AT*ZTD
C ** CALCULATE FORCES AND MOMENTS
C
C
      FXT=FXT+FFXBAR
      FYT=FYT+FFYBAR
      FZT=FZT-FTBAR-FDBAR
      DTXT=(-(FTBAR+FDBAR)*YBT-FFYBAR*ZBT)*.08333
      TXT=TXT+DTXT
      DTYT=((FTBAR+FDBAR)*XBT+FFXBAR*ZBT)*.08333
      TYT=TYT+DTYT
      DTZT=(FFYBAR*XBT-FFXBAR*YBT)*.08333
```

```
      TZT=TZT+DTZT
      GO TO 66
C
C  CALCULATE TRUNK ELEMENT DIMENSION AND AREAS FOR FREE SHAPE
C
44    CONTINUE
      RR=1.
      IF(BET(I).NE.0.) RR=RD(I)+YOFS
      DAGAP=RR*ZGAP*D(I)
      AGAP=AGAP+DAGAP
      CALL ETB3(IS(I),PR,PTG,DPT,1.,EYO,YO)
      CALL ETB3(IS(I),PR,PTG,DPT,1.,EL1,L1)
      CALL ETB3(IS(I),PR,PTG,DPT,1.,EL2,L2)
      LT=L1+L2
      CALL ETB3(IS(I),PR,PTG,DPT,1.,EVS,AS)
      CALL ETB3(IS(I),PR,PTG,DPT,1.,EVC,ACV)
      FFXBAR=0.
      FFYBAR=0.
      FDBAR=0.
      FTBAR=0.
      L3=0.
      AT=0.
      DTZT=0.
      ZR=1.
 66   CONTINUE
C
      IQ=IS(I)
      FREEL=LO(IQ)/(1.+EPI(IQ))
      EP=(LT-FREEL)/FREEL
C
C  TEST FOR END OR SIDE ELEMENT
C
      IF(BET(I).EQ.0.) GO TO 77
C
      YBC=E*(YBA(I)+(.6667*YO-.3333*RA(I))*CSB(I))
      AC=.5*D(I)*YO*(YO+2.*RD(I))
      DVCS=.0005787*D(I)*(.5*ZC*(YO+RD(I))**2 - ACV)
      VCS=VCS+DVCS
      GO TO 88
C
 77   CONTINUE
      YBC=0.5*E*(YBA(I)+YO)
      AC=D(I)*(YA(I)+YO)
      DVCS=(ZC*AC-D(I)*ACV)*.0005787
      VCS=VCS+DVCS
 88   CONTINUE
C
      DVTS=D(I)*AS*.0005787
      VTS=VTS+DVTS
C
```

```
      XBC=XBA(I)+(.6667*YO-.3333*RA(I))*SNB(I)
      FCBAR= (PC-PA)*AC
      FZT=FZT-FCBAR
      DTXT=-FCBAR*YBC*.08333
      TXT=TXT+DTXT
      DTYT=FCBAR*XBC*.08333
      TYT=TYT+DTYT
C
C  COMPUTE AREA OF PERFORATIONS
      IF(AP(I).EQ.0.) GO TO 31
      CALL ETB3(IS(I),PR,PTG,DPT,ZR,EE1,E1)
      CALL ETB3(IS(I),PR,PTG,DPT,ZR,EE3,E3)
      SLP=LP(I)*(1.+E1)
      SLH=LH(I)*(1.+E3)
      DX=D(I)
      IF(BET(I).NE.0.) DX=D(I)*(RA(I)+YOFS)
      CALL PERF(ZGAP,L1,L3,L3,SLP,SLH,RA(I),YO,YOFS,DX,AP(I),
     2 PT,PC,PA,BET(I),AHA1,AHA2,AHC2,AHC1)
 31   CONTINUE
C
C  STORE DATA IN XXPRT
      IF(EPC.GT.0.9) CALL XXPRT(0,I,M,MS,NE,ZGAP,ZO,YO,L1,L3,DVTS,DVCS
     2 ,FFXBAR,FFYBAR,FDBAR,FTBAR,FCBAR,L2,DAGAP,EP,DTXT,DTYT,DTZT,AT,
     3 AC)
 30   CONTINUE
C
C CALCULATE FLOW RATES
C
      CACA=CDGAP*AGAP
      CATA=CDH1*AHA1+.6667*CDH2*AHA2
      CATC=CATV+CDH1*AHC1+.6667*CDH2*AHC2
C
C ** TEST FOR SYMMETRIC MODEL
C
      IF(ANE.GT.0.)GOTO 166
      FXT=2.*FXT
      FYT=0.
      FZT=2.*FZT
      TXT=0.
      TYT=2.*TYT
      TZT=0.
      VCS=2.*VCS
      VTS=2.*VTS
      CACA=2.*CACA
      CATA=2.*CATA
      CATC=2.*CATC
166   CONTINUE
C
C
C ** CALCULATE RELIEF VALVE AREA
```

```
C
      N=REL(2)
      AREL = TBLU1(PT-PA,REL(4),REL(N+4),1,-N)
      CATA=CATA+CDA*AREL
C
C ** CALCULATE CUSHION VOLUME RATE OF CHANGE
C
      IF(IVC.NE.0) VCD=(VCS-VC)/TAU
C
C ** CALCULATE TRUNK AIR VOLUME RATE OF CHANGE
C
      IF(IVT.NE.0) VTD=(VTS-VT)/TAU
C
      CACA=AMIN1(3000.,CACA)
      CALL FNFLOW(PC,PA,TCU,CACA,1.,FN,WCA)
      CALL FNFLOW(PT,PA,TTR,CATA,1.,FN,WTA)
      CALL FNFLOW(PT,PC,TTR,CATC,1.,FN,WTC)
C
C
C ** CALCULATE CUSHION PRESSURE RATE OF CHANGE
C *** INTEGRATE CUSHION PRESSURE ***
C
      DD=1.
      IF(TAU.EQ.0.99999) DD=0.
      PCD1=(.0001389*RG*TCU*(WCU+WTC-WCA)-1.2*PC*VCD*DD)/VC
      IF(IPC.NE.0)PCD=PCD1
C
C ** CALCULATE TRUNK AIR PRESSURE RATE OF CHANGE
C *** INTEGRATE TRUNK PRESSURE ***
C
      IF(IPT.NE.0)PTD=(.0001389*RG*TTR*(WTR-WTC-WTA)-1.2*PT*VTD*DD)/VT
C
C
C  PRINT DATA IN XXPRT
      IF(EPC.GT.0.9) CALL XXPRT(1,I,M,MS,NE,ZGAP,ZO,YO,L1,L3,OVTS,OVCS
     2 ,FFXBAR,FFYBAR,FDBAR,FTBAR,FCBAR,L2,DAGAP,EP,DTXT,DTYT,DTZT,AT,
     3 AC)
      RETURN
      END
```

Table 242: LISTING FOR SUBROUTINE TT

```
CTT
      SUBROUTINE TT(W,WD,IW,Q,QD,IQ,PIT,PITD,IPIT,Z,ZD,IZ,
     *              QDOT,WDOT,TY,YYI,ROL,U)
C  PURPOSE   TWO DEGREE OF FREEDOM RIGID BODY EQUATIONS OF MOTION(LONG)
C   VERSION 2.                        AUG.22 1977
C  METHOD   EULER ANGLES
C  CALL SEQUENCE
C  *****  OUTPUTS  *****
C   LINEAR VELOCITIES  --  BODY AXES
C      W,WD,IW       - Z AXIS LINEAR VELOCITY,ACCEL,INT CONTROL, FT/SEC
C   ANGULAR VELOCITIES --  BODY AXES
C      Q,QD,IQ       - Y AXIS ANGULAR VELOCITY,ACCEL,INT CONTROL,DEG/SEC
C   EULER ANGLE   -- EARTH TO BODY - PITCH
C      PIT,PITD,IPIT - PITCH ANGLE,RATE,INT CONTROL, DEG
C   POSITION  --  EARTH AXES
C      Z,ZD,IZ       - -Z AXIS POSITION (ALT),VELOCITY,INT CONTROL, FT
C   ANGULAR ACCELERATION  --  BODY AXES
C      QDOT          - Y AXIS ANGULAR ACCELERATION, DEG/SEC2
C ****************   INPUTS   ****************
C   LINEAR ACCLERATIONS --  BODY AXES
C      WDOT          - Z AXIS LINEAR ACCELERATION, FT/SEC2
C   MOMENTS
C      TY            - Y AXIS TORQUE, FTLBS
C   MOMENT OF INERTIA
C      YYI           - Y AXIS MOMENTS OF INERTIA, SLUG-FT2
C   EULER ANGLES  -- EARTH TO BODY - ROLL,YAW
C      ROL           - ROLL ANGLE, DEG
C      U             - X AXIS LINEAR VELOCITY,FT/SEC
C     WRITTEN BY    MAHINDER WAHI                    APRIL 1977
C
      DATA RPD,DPR /.01745329,57.29578/
      SP=SIN(PIT*RPD)
      CP=COS(PIT*RPD)
      CR=COS(ROL*RPD)
      Q1=Q*RPD
C ****************   LINEAR VELOCITY EQUATIONS   ************
      IF(IW.NE.0)WD=WDOT
C ****************   ANGULAR VELOCITY EQUATION   ************
      IF(IQ.NE.0)QD=(TY/YYI)*DPR
C ****************   EULER ANGLE EQUATION   ************
      IF(IPIT.NE.0)PITD=Q*CR
C ****************  POSITION EQUATIONS   **************
C ----  TEST IF X AND Y (LATITUDE AND LONGITUDE) ARE BOTH FROZEN
      IF(IZ.NE.0)ZD= SP*U-CP*CR*W
C ********** ANGULAR ACCELERATION (FOR OUTPUT PURPOSES ONLY) *****
      QDOT=QD
      RETURN
      END
```

Table 243: LISTING FOR SUBROUTINE TZ

```
CTZ
      SUBROUTINE TZ(X1,X1DOT,IX1,X2,X2DOT,IX2,FO,FIN,Z0,Z1,Z2,P0,P1)
C
C    PURPOSE - TO SIMULATE SECOND ORDER TRANSFER FUNCTION WITH SECOND ORDER
C              NUMERATOR                              2
C                                     FO       Z2*S  + Z1*S + Z0
C                                    ----- = -----------------
C                                     FIN          2
C                                                 S  + P1*S + P0
C
C
C    METHOD   - SELF EXPLANATORY
C
C
C    LIMITATIONS -   NONE
C
C
C    WRITTEN BY  -   ADAM LLOYD           LATEST REVISION  NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    X1         FIRST STATE VARIABLE                  ANY        OUTPUT STATE
C    X1DOT      FIRST STATE VARIABLE DERIVATIVE       ANY        OUTPUT STATE
C    IX1        INTEGRATOR CONTROL                               PROGRAM VAR
C    X2         SECOND STATE VARIABLE                 ANY        OUTPUT STATE
C    X2DOT      SECOND STATE VARIABLE DERIVATIVE      ANY        OUTPUT STATE
C    IX2        INTEGRATOR CONTROL                               PROGRAM VAR
C    FO         TRANSFER FUNCTION OUTPUT              ANY        OUTPUT VAR
C    FIN        TRANSFER FUNCTION INPUT                          INPUT  VAR
C    Z0         NUMERATOR COEFFICIENT                            INPUT  PARAM
C    Z1         NUMERATOR COEFFICIENT                            INPUT  PARAM
C    Z2         NUMERATOR COEFFICIENT                 ANY        INPUT  PARAM
C    P0         DENOMINATOR COEFFICIENT               1/SEC2     INPUT  PARAM
C    P1         DENOMINATOR                           1/SEC      INPUT  PARAM
      COMMON/CIO/IREAD,IWRITE,IDIAG
      FO=X2+FIN*Z2
      IF(IX1.NE.0)X1DOT=Z0*FIN-P0*FO
      IF(IX2.NE.0)X2DOT=X1+Z1*FIN-P1*FO
      RETURN
      END
```

Table 244: LISTING FOR SUBROUTINE VA

```
CVA
      SUBROUTINE VA(UO,VO,WO,PO,QO,RO,DIMO,QWO,RWO,CAS,SAS,AL,ALP,
     1  VBAR,BETA,WP,UP,EU,EV,EW,SIGZ,QCOM,QBAR,XMACH,U,V,W,
     2  P,Q,R,Z,PIT,ROL,DIM,VS,ALS,S,UW,VW,WW,PW,QW,RW,DIG)
      DIMENSION SIG(35),A(35)
C    VERSION 2.                              JULY 13 1977
C    PURPOSE:  COMPUTE AERO VARIABLES FROM STATES
C    CALL SEQUENCE:
C    ******  OUTPUTS  ******
C    OUTPUT VARIABLES= INPUTS - -
C      UO,VO,WO       -X,Y,Z BODY AXIS LINEAR VELOCITY WITH WIND,FT/SEC
C      PO,QO,RO       -X,Y,Z BODY AXIS ANGULAR RATES WITH WIND, DEG/SEC
C      DIMO           -INDICATOR FOR AERO FORCES AND MOMENTS = DIM
C      QWO,RWO        -Q AND R ANGULAR RATE GUSTS, DEG/SEC
C    AERO VARIABLES - -
C      CAS,SAS        -DIRECTION COSINES FOR STABILITY AXIS TRANSFORM
C      AL,ALP         -ANGLE OF ATTACK IN BODY AND STABILITY AXES, DEG
C      VBAR           -TRUE AIRSPEED, FT/SEC
C      BETA           -SIDESLIP ANGLE, DEG
C      WP,UP          -Z AND X STABILITY AXIS VELOCITIES,FT/SEC (DIMENSIONAL
C                     -Z AND X PERTURBATION VELOCITIES (NONDIMENSIONAL)
C      EU,EV,EW       -X,Y,Z BODY AXIS ACCELERATION TERMS FOR UDOT,VDOT,
C                      WDOT SOLUTIONS, FT/SEC**2
C    STANDARD ATMOSPHERE VARIABLES - -
C      SIGZ           -AIR DENSITY RATIO
C      QCOM            -COMPRESSIBLE DYNAMIC PRESSURE, LBS/FT**2
C      QBAR           -DYNAMIC PRESSURE TIMES REFERENCE AREA, LBS
C      XMACH          -MACH NUMBER
C    ******  INPUTS  ******
C    STATE VARIABLES
C      U,V,W          -X,Y,Z BODY AXIS LINEAR VELOCITIES, FT/SEC
C      P,Q,R          -X,Y,Z BODY AXIS ANGULAR RATES, DEG/SEC
C      Z              -ALTITUDE ABOVE SEA-LEVEL, FT
C      PIT,ROL        -PITCH AND ROLL, EARTH TO BODY AXIS EULER ANGLES, DEG
C    INDICATOR FUNCTION FOR AERO FORCES AND MOMENTS
C      DIM = 0        -BODY AXIS,DIMENSIONAL
C            1        -BODY AXIS, NONDIMENSIONAL
C            2        -STABILITY AXIS, DIMENSIONAL
C            3        -STABILITY AXIS, NONDIMENSIONAL
C    CONSTANTS - -
C      VS             -STEADY STATE AIRSPEED, FT/SEC
C      ALS            -STEADY STATE ANGLE OF ATTACK, DEG
C      S              -REFERENCE AREA, FT**2
C    WIND STATES - -
C      UW,VW,WW       -X,Y,Z BODY AXIS WIND VELOCITIES, FT/SEC
C      PW,QW,RW       -X,Y,Z BODY AXIS WIND ANGULAR RATES, DEG/SEC
C    INDICATOR FUNCTION FOR DEGREES OF FREEDOM (DOF)
C      DIG = 2        -TWO DOF LONGITUDINAL (S,Q)
C            3        -THREE DOF LONGITUDINAL (U,W,Q)
C            4        -FOUR DOF  LATERAL(V,P,R) + LONGI.(U)
C            5        -THREE DOF LATERAL (V,P,R)
C            6        -SIX DOF FULL MODEL (U,V,W,P,Q,R)
C
C     WRITTEN BY A.W.WARREN AS COMPONENT *AV* IN
C                FLT.CONTROL LIBRARY      SEPT 1976
C
```

```
C     MODIFIED BY MAHINDER WAHI                              MAY 1977
C
      COMMON /COVRLY/ DUM(3),CPUSEC
      DATA PS,ISW,G,SIG/ .0011884,0,32.174,1.0,.9151,.8359,
     1 .7620,.6932,.6292,.5699,.5150,.4642,.4173,.3741,.3345,
     2 .2981,.2583,.2236,.1936,.1676,.1451,.1256,.10874,.09414,
     3 .08150,.07052,.06081,.05248,.04532,.03915,.03385,.02928,
     4 .02534,.02195,.01902,.01649,.01431,.01242/
      DATA DPR /57.29578/,CPUS /0./
      DATA A /1116.4,1104.9,1093.2,1081.4,1069.4,1057.4,1045.2,
     1 1032.8,1020.3,1007.6,994.9,981.9,968.7,968.1,968.1,968.1,
     2 968.1,968.1,968.1,968.1,968.1,968.1,968.2,970.2,972.2,
     3 974.3,976.3,978.3,980.3,982.3,984.3,986.3,988.3,990.2,992.2/
C
C                   INITIALIZATION
      IF(CPUS.EQ.CPUSEC) GO TO 10
      IF(UW.EQ. .99999) UW=VW=WW=PW=QW=RW=0.
      IF(DIM.EQ. .99999) DIM= 0.
      IF(ALS.EQ. .99999) ALS= 0.
      IF(DIG.EQ.2.)  GO TO 40
      IF(DIG.EQ.3.)  GO TO 50
      IF(DIG.EQ.4.)  GO TO 60
      IF(DIG.EQ.5.)  GO TO 70
      IF(DIG.EQ.6.)  GO TO 80
   40 V =P =R =UW =VW =PW =RW =0.
      GO TO 80
   50 V =P =R =VW =PW =RW =0.
      GO TO 80
   60 Q =WW =QW =0.
      GO TO 80
   70 Q =UW =WW =QW =0.
   80 CONTINUE
      CAS= COS(ALS/DPR)
      SAS= SIN(ALS/DPR)
      US=VS*CAS
      DIMO= DIM
      IF(DIM.GE.2.) GO TO 20
      CAS= 1.
      SAS= 0.
   20 CPUS= CPUSEC
C                   OUTPUT STATES
C
   10 UO= U -UW
      VO= V - VW
      WO= W - WW
      PO= P+ PW
      QO= Q+ QW
      RO= R+ RW
      QWO= QW
      RWO= RW
      IF(DIM.LT.2.) GO TO 30
      P1= PO*CAS+ RO*SAS
      RO= RO*CAS- PO*SAS
      PO= P1
C                   AERO VARIABLES
C
```

Table 244: LISTING FOR SUBROUTINE VA (CONCLUDED)

```
   30 AL= ATAN(WO/UO)*DPR
      ALP= AL - ALS
      VBAR2= UO**2 + VO**2 + WO**2
      VBAR= SQRT(VBAR2)
      BETA= ASIN(VO/VBAR)*DPR
      WP= WO*CAS- UO*SAS
      UP= UO*CAS + WO*SAS
      IF(DIM.EQ.1.) UP= (UO - US)/VS
      IF(DIM.EQ.3.) UP= (UP - VS)/VS
C
      EU= (-Q*W + R*V)/DPR - G*SIN(PIT/DPR)
      G1 = G*COS(PIT/DPR)
      EV= (-R*U + P*W)/DPR +G1*SIN(ROL/DPR)
      EW= (-P*V + Q*U)/DPR +G1*COS(ROL/DPR)
C
C                   ATMOSPHERE VARIABLES
C          LINEAR INTERPOLATION OF AIR DENSITY RATIO AND SOUND VELOCITY
C
      Z1= Z/3000. +1.
      IZ= Z1
      IZ= MIN0(MAX0(1,IZ),34)
      SIGZ= SIG(IZ) + (SIG(IZ+1) - SIG(IZ))*(Z1-IZ)
      AZ= A(IZ) + (A(IZ+1) - A(IZ))*(Z1-IZ)
      DPS= PS*SIGZ*VBAR2
      QBAR= DPS* S
      XMACH= VBAR/AZ
      XM2= XMACH**2
      QCOM= DPS*(1.+ XM2*.25*(1.+ XM2*0.1*(1.+ XM2*.025)))
      IF(XM2.GT.1.) QCOM= DPS*(1.839 -.772/XM2**2 + .035/XM2**3)
C
      RETURN
      END
```

Table 245: LISTING FOR SUBROUTINE VALVE

```
CVALVE
      SUBROUTINE VALVE
C PRESSURE RELIEF VALVE SUBROUTINE
C RELATION BETWEEN VENT AREA AND RELIEF VALVE DISPLACEMENT
C
      COMMON/STATE/PPLM,PCH,PTK,SINKRT,YCG,DPHI,DTHETA,THETAE,PHIE,SIE
     1 ,XV,VV,QFANX
      COMMON/PRV/DPRV,PPLMB,XA,AKPRV,AMPRV,SPRV,NPRV,AVENT
C
      AVENT=0.
      IF(XV.LT.0.0) RETURN
C VALVE OPEN
      AVENT=XV*SPRV
      RETURN
      END
```

Table 246: LISTING FOR SUBROUTINE VLX

```
CVLX
      SUBROUTINE VLX(P1,P2,T,D,DPOPP,OPEN,VAL,W)
C
C    PURPOSE - TO CALCULATE THE WEIGHT FLOW OF AIR ACROSS BUTTERFLY,
C              GATE AND GLOBE TYPE VALUES.
C
C
C    METHOD  -  1. BUTTERFLY VALVES - ASSUMES DISCHARGE COEFF   0.87
C               2. GATE VALVES - USES INPUT TABLE OF K FACTOR VERSUS
C                                FRACTIONAL OPENING
C               3. GLOBE VALVES - CALCULATES GEOMETRIC FLOW AREA AND
C                                 ASSUMES DISCHARGE COEFF = 0.80
C
C
C    WRITTEN BY  -  ADAM LLOYD          LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - FOR GLOBE VALVES, THE POPPET DIAMETER (DPOPP)
C                  MUST BE GREATER THAN OR EQUAL TO THE SEAT DIAMETER (D)
C
C
C    INPUT/OUTPUT LIST
C
C    P1          INLET PRESSURE                        PSIA         INPUT
C    P2          OUTLET PRESSURE                       PSIA         INPUT
C    T           TEMPERATURE                           DEGR         INPUT
C    D           DUCT DIAMETER                         INCH         INPUT
C                (SEAT DIAMETER FOR GLOBE VALVES)
C    DPOPP       POPPET DIAMETER                       INCH         INPUT
C                (REQUIRED FOR GLOBE VALVES (IVAL=3) ONLY)
C    OPEN        VALVE OPENING                         ---          INPUT
C                IVAL=1    DEGREES OPEN
C                IVAL=2,3  FRACTIONAL OPENING (0.LE.OPEN.LE.1.)
C    VAL         CODE IDENTIFYING TYPE OF VALVE        ---          INPUT
C                =1.  BUTTERFLY VALVE
C                =2.  GATE VALVE
C                =3.  GLOBE VALVE
C    W           FLOW RATE                             LB/MIN        OUTPUT
      COMMON/CIO/IREAD,IWRITE,IDIAG
      DIMENSION AKF(21)
C    DATA ARRAY OF K FACTOR FOR GATE VALVE
C    INCREMENT IN FRACTIONAL OPENING IS 0.05
C    ESTIMATED DATA FOR FRACTIONAL OPENING LESS THAN 0.12
      DATA AKF /
     1   1000.    ,600.     ,200.     ,51.      ,27.       ,
     2   16.      ,10.      ,6.5      ,4.4      ,3.0       ,
     3   2.05     ,1.38     ,0.94     ,0.63     ,0.41      ,
     4   0.26     ,0.162    ,0.096    ,0.050    ,0.020     ,
     5   0.010 /
      IVAL=VAL
      GO TO (10,20,30) IVAL
C    BUTTERFLY VALVE ANALYSIS
C    CHECK IF VALVE ANGLE OUTSIDE LIMITS
 10   IF(OPEN.GT.90.)GO TO 14
      AEFF=.87*3.1416*D*D*(1.-COS(OPEN/57.296))/4.
      CALL FNFLOW(P1,P2,T,AEFF,1.,FN,W)
```

Table 246: LISTING FOR SUBROUTINE VLX (CONCLUDED)

```
      IF(OPEN.LT.0.)W=-W
      GO TO 100
 14   AEFF=.87*3.1416*D*D/4.
      CALL FNFLOW(P1,P2,T,AEFF,1.,FN,W)
      W=W*(0.9+.1*OPEN/90.)
      GO TO 100
C   GATE VALVE ANALYSIS
C   CHECK IF VALVE OPENING OUTSIDE LIMITS
 20   IF(OPEN.LT.0.05)GO TO 22
      IF(OPEN.GT.1.)GO TO 24
      X1=OPEN/.05+1.
      X1=AMIN1(AMAX1(X1,1.),29.)
      I=X1
      I=MIN0(MAX0(I,1),28)
      AK=(X1-I)*(AKF(I+1)-AKF(I)) + AKF(I)
      AEFF=.7854*D*D
      CALL FNFLOW(P1,P2,T,AEFF,AK,FN,W)
      GO TO 100
 22   AEFF=.7854*D*D
      CALL FNFLOW(P1,P2,T,AEFF,600.,FN,W)
      W=W*(10.+OPEN)/10.05
      GO TO 100
 24   AEFF=.7854*D*D
      CALL FNFLOW(P1,P2,T,AEFF,.01,FN,W)
      W=W*(0.90+.10*OPEN)
      GO TO 100
C   CLOBE VALVE ANALYSIS
 30   XMAX=(SQRT(.25*(1.+SQRT(1.+4.*(DPOPP/D)**2.))**2.-1.)
     1 - SQRT((DPOPP/D)**2.-1.))*D/2.
C   CHECK IF VALVE OPENING OUTSIDE LIMITS
      IF(OPEN.LT.0.05)GO TO 32
      IF(OPEN.GT.1.)GO TO 34
      X= XMAX*OPEN
      DUM=(2.*X/D + SQRT((DPOPP/D)**2.-1.))**2.+1.
      AGEO=(DUM-(DPOPP/D)**2.)*.7854*D*D/SQRT(DUM)
C   CALCULATE EFFECTIVE AREA AND FLOW
      AEFF=0.80*AGEO
      CALL FNFLOW(P1,P2,T,AEFF,1.,FN,W)
      GO TO 100
 32   X=.05*XMAX
      DUM=(2.*X/D + SQRT((DPOPP/D)**2.-1.))**2.+1.
      AGEO=(DUM-(DPOPP/D)**2.)*.7854*D*D/SQRT(DUM)
      AEFF=.80*AGEO
      CALL FNFLOW(P1,P2,T,AEFF,1.,FN,W)
      W=W*(10.+OPEN)/10.05
      GO TO 100
 34   X=XMAX
      DUM=(2.*X/D + SQRT((DPOPP/D)**2.-1.))**2.+1.
      AGEO=(DUM-(DPOPP/D)**2.)*.7854*D*D/SQRT(DUM)
      AEFF=.80*AGEO
      CALL FNFLOW(P1,P2,T,AEFF,1.,FN,W)
      W=W*(.90+.10*OPEN)
 100  RETURN
      END
```

Table 247: LISTING FOR SUBROUTINE VPRINB

```
CVPRINB
      SUBROUTINE VPRINB(K,I,M,MS,NE,ZGAP,ZO,YO,L1,L3,AS,UTY,FFXBAR,
     *             FFYBAR,FDBAR,FTBAR,CNT,CNTL,CNTF)
C
C   VERSION 2.        REVISED 20 MARCH 1979
C
C    WRITTEN BY   J.R.KILNER
C
C    LIMITATIONS - CALLED ONLY BY COMPONENT AB AND A1
      COMMON/CTIME/T
      COMMON/COVRLY/INST
      COMMON/CSIMUL/D1,IRATE,D2(4),TINC,D3(7)
      COMMON/BMADTS/INT
      COMMON/CIO/IREAD,IWRITE,IDIAG
      REAL L1,L3
      DIMENSION A(50,11)
      DATA NTIM/0/
C
      IF(INST.EQ.27) GO TO 10
      NTIM=0
      IF(INST.NE.26)GOTO 99
      IF(INT.NE.1) GO TO 99
   10 IF(K.EQ.0)GO TO 22
C
C  LOAD PRINT STORAGE ARRAY
C
      J=I+(M-1)*NE
      A(J,1)=ZGAP
      A(J,2)=ZO
      A(J,3)=YO
      A(J,4)=L1
      A(J,5)=L3
      A(J,6)=AS
      A(J,7)=UTY
      A(J,8)=FFXBAR
      A(J,9)=FFYBAR
      A(J,10)=FDBAR
      A(J,11)=FTBAR
      GOTO 99
   22 CONTINUE
      IF(INST.EQ.27.AND.NTIM.GT.0) GO TO 99
C
C  PRINT DATA
C
      WRITE(IWRITE,200)T
      DO 40 MM=1,MS
      DO 40 J=1,NE
      JC=J+(MM-1)*NE
40    WRITE(IWRITE,202)J,(A(JC,JR),JR=1,11)
200   FORMAT(/* TIME=*,F7.4/8X,*ZGAP      ZO         YO          *,
     * *L1         L3         AS         UTY        FFX        *,
     * *FFY        FD         FT         *)
202   FORMAT(1X,I2,2X,11E10.3)
      CNTL=.5*CNTL/NE
      CNTF=.5*CNTF/NE
      WRITE(IWRITE,204)CNT,CNTL,CNTF
```

Table 247: LISTING FOR SUBROUTINE VPRINB (CONCLUDED)

```
204   FORMAT(* CNT=*,E10.3,6X,*CNTL=*,E10.3,6X,*CNTF=*,E10.3)
      CNT=0.
      CNTL=0.
      CNTF=0.
C
C
      NTIM=1
99    RETURN
      END
```

Table 248: LISTING FOR SUBROUTINE VPRINT

```
CVPRINT
      SUBROUTINE VPRINT(K,I,M,MS,NE,ZGAP,ZO,YO,L1,L3,VTS,VCS,FFXBAR,
     *                 FFYBAR,FDBAR,FTBAR,FCBAR,AGAP)
C
C   VERSION 2.         REVISED 20 MARCH 1979
C
C    WRITTEN BY   J.R.KILNER
C
C    LIMITATIONS - CALLED ONLY BY COMPONENT TK
      COMMON/CTIME/T
      COMMON/COVRLY/INST
      COMMON/CSIMUL/D1,IRATE,D2(4),TINC,D3(7)
      COMMON/BMADTS/INT
      COMMON/CIO/IREAD,IWRITE,IDIAG
      REAL L1,L3
      DIMENSION A(50,12)
      DATA NTIM/0/
C
      IF(INST.EQ.27) GO TO 10
      NTIM=0
      IF( INST.NE.26)GOTO 99
      IF(INT.NE.1) GO TO 99
   10 IF(K.EQ.0)GOTO 22
C
C  LOAD PRINT STORAGE ARRAY
C
      J=I+(M-1)*NE
      A(J,1)=ZGAP
      A(J,2)=ZO
      A(J,3)=YO
      A(J,4)=L1
      A(J,5)=L3
      A(J,6)=VTS
      A(J,7)=VCS
      A(J,8)=FFXBAR
      A(J,9)=FFYBAR
      A(J,10)=FDBAR
      A(J,11)=FTBAR
      A(J,12)=FCBAR
      GOTO 99
   22 CONTINUE
      IF(INST.EQ.27.AND.NTIM.GT.0) GO TO 99
C
C  PRINT DATA
C
      WRITE(IWRITE,200)T
      DO 40 MM=1,MS
      DO 40 J=1,NE
      JC=J+(MM-1)*NE
40    WRITE(IWRITE,202)J,(A(JC,JR),JR=1,12)
200   FORMAT(/* TIME=*,F7.4/8X,*ZGAP      ZO         YO         *,
     * *L1        L3         VT         VC         FFX        *,
     * *FFY       FD         FT         FC    *)
202   FORMAT(1X,I2,2X,12E10.3)
      WRITE(IWRITE,204)AGAP
C
```

Table 248: LISTING FOR SUBROUTINE VPRINT (CONCLUDED)

```
 204  FORMAT(2X,*AGAP=*,E10.3)
      NTIM=1
99    RETURN
      END
```

Table 249: LISTING FOR SUBROUTINE WS

```
CWS
      SUBROUTINE WS(TWS,UWS,VWS,WWS,WK,WAN,ALT,PIT,DNI)
      REAL MF
C     VERSION 2.                     MARCH 31   1978
C     PURPOSE   SIMULATE WIND SHEAR OR STEADY WIND COMPONENTS
C     METHOD    WIND MAGNITUDE QUOTED AT TOWER ALTITUDE OF 50 FEET IS
C               MODIFIED BY A NON-LINEAR SHEAR FACTOR TO REFLECT THE
C               CHANGE IN WIND WITH ALTITUDE.WIND VECTOR ASSUMED
C               PARALLEL TO THE GROUND PLANE.THE WIND MAGNITUDE MODIFIED
C               BY SHEAR IS RESOLVED ALONG THE RUNWAY COORDINATES NORTH
C               AND EAST(FWD + SIDE)AND TRANSFORMED INTO BODY AXES.
C
C     CALL SEQUENCE
C     ******  TABLE  ******
C       TWS     -ONE DIMENSIONAL TABLE OF WIND SHEAR FACTOR AS A
C               -FUNCTION OF AIRPLANE CG ALTITUDE
C     ******  OUTPUTS  ******
C     LINEAR VELOCITIES -- BODY AXES
C       UWS     -X AXIS STEADY OR SHEAR WIND VELOCITY, FT/SEC
C       VWS     -Y AXIS STEADY OR SHEAR WIND VELOCITY, FT/SEC
C       WWS     -Z AXIS STEADY OR SHEAR WIND VELOCITY, FT/SEC
C     ******  INPUTS  ******
C       WK      -WIND MAGNITUDE AT 50 FEET (TOWER), FT/SEC
C       WAN     -ANGLE BETWEEN THE WIND VECTOR AND RUNWAY CENTERLINE,DEG
C       ALT     -AIRPLANE CG ALTITUDE, FT
C       PIT            -PITCH ANGLE EARTH TO BODY, DEG
C     INDICATOR FOR STEADY OR SHEAR WIND
C       DNI= 0  SHEAR WIND,TABLE LOOKUP FACTOR
C          = 1  STEADY WIND, FACTOR=1.
C     WRITTEN BY MAHINDER WAHI                                 MAY 1977
      DIMENSION TWS(1)
C       SET DEFAULTS
C
      IF(ISW.EQ.1)GO TO 10
      IF(WK.EQ. .99999) WK=0.
      IF(WAN.EQ. .99999) WAN=0.
      IF(ALT.EQ. .99999) ALT=0.
      IF(PIT.EQ. .99999) PIT=0.
      ISW=1
C     COMPUTE WIND SHEAR FACTOR
   10 IF(DNI.NE.0.) GO TO 20
      NX= TWS(2)
      MF= TBLU1(ALT,TWS(4),TWS(NX+4),1,-NX)
      GO TO 30
   20 MF=1.
C     RESOLVE WIND INTO NORTH AND EAST COMPONENTS
   30 WKN=-WK*MF*COS(WAN*.01745)
      WKE= WK*MF*SIN(WAN*.01745)
C     TRANSFORMATION FROM EARTH TO BODY AXIS
      UWS= WKN*COS(PIT*.01745)
      VWS= WKE
      WWS= -WKN*SIN(PIT*.01745)
      RETURN
      END
```

Table 250: LISTING FOR SUBROUTINE XP

```
CXP
      SUBROUTINE XP(T,P2,Q2,R2,P1,Q1,R1)
C  VERSION: 1.                          REVISED: JUNE 10 1976
C  PURPOSE:  PERFORM STATIC TRANSFORMATION ON THREE VECTOR
C  CALL SEQUENCE:
C       T            -  TRANSFORMATION MATRIX
C       P2,Q2,R2     -  OUTPUT VECTOR COMPONENTS
C       P1,Q1,R1     -  INPUT VECTOR COMPONENTS
C  DESIGNED BY: J.D. BURROUGHS                      JUNE 1976
      DIMENSION T(18)
      P2=P1*T(10)+Q1*T(11)+R1*T(12)
      Q2=P1*T(13)+Q1*T(14)+R1*T(15)
      R2=P1*T(16)+Q1*T(17)+R1*T(18)
      RETURN
      END
```

Table 251: LISTING FOR SUBROUTINE XT

```
CXT
      SUBROUTINE XT(T,P2,Q2,R2,P1,Q1,R1)
C  VERSION: 1.                         REVISED: JUNE 10 1976
C  PURPOSE:  PERFORM STATIC TRANSFORMATION ON THREE VECTOR
C  CALL SEQUENCE:
C       T           -  TRANSFORMATION MATRIX
C       P2,Q2,R2    -  OUTPUT VECTOR COMPONENTS
C       P1,Q1,R1    -  INPUT VECTOR COMPONENTS
C  DESIGNED BY: J.D. BURROUGHS                      JUNE 1976
      DIMENSION T(18)
      P2=P1*T(10)+Q1*T(11)+R1*T(12)
      Q2=P1*T(13)+Q1*T(14)+R1*T(15)
      R2=P1*T(16)+Q1*T(17)+R1*T(18)
      RETURN
      END
```

Table 252: LISTING FOR SUBROUTINE XXPRT

```
CXXPRT
      SUBROUTINE XXPRT(K,I,M,MS,NE,ZGAP,ZO,YO,L1,L3,VTS,VCS,FFXBAR,
     *               FFYBAR,FDBAR,FTBAR,FCBAR,L2,AGAP,EP,TX,TY,TZ,AT,AC)
C
C
C    VERSION 2.                              REVISED MARCH 1979
C
C    WRITTEN BY - GS DULEBA
C
C    PURPOSE - TO STORE AND WRITE VALUES OF ELASTIC TRUNK
C              VARIABLES DURING SIMULATION.
C    LIMITATIONS - CALLED ONLY BY COMPONENT TS
C
C
      COMMON/CTIME/T
      COMMON/COVRLY/INST
      COMMON/CSIMUL/D1,IRATE,D2(4),TINC,D3(7)
      COMMON/BMADTS/INT
      COMMON/CIO/IREAD,IWRITE,IDIAG
      REAL L1,L2,L3
      DIMENSION A(50,10), B(50,10)
      DATA NTIM/0/
C
      IF(INST.EQ.27) GO TO 10
      NTIM=0
      IF(INST.NE.26)GOTO 99
      IF(INT.NE.1) GO TO 99
 10   CONTINUE
C
C  LOAD PRINT STORAGE ARRAY
C
      IF(K.EQ.1) GO TO 22
      J=I+(M-1)*NE
      A(J,1)=ZGAP
      A(J,2)=ZO
      A(J,3)=YO
      A(J,4)=AGAP
      A(J,5)=L1
      A(J,6)=L2
      A(J,7)=L3
      A(J,8)=EP
      A(J,9)=VTS
      A(J,10)=VCS
C
      B(J,1)=FFXBAR
      B(J,2)=FFYBAR
      B(J,3)=FDBAR
      B(J,4)=FTBAR
      B(J,5)=FCBAR
      B(J,6)=TX
      B(J,7)=TY
      B(J,8)=TZ
      B(J,9)=AT
      B(J,10)=AC
      GO TO 99
 22   CONTINUE
```

```
C
      IF(INST.EQ.27.AND.NTIM.GT.0) GO TO 99
C   PRINT DATA
C
      WRITE(IWRITE,200)T
      DO 40 MM=1,MS
      DO 40 J=1,NE
      JC=J+(MM-1)*NE
40    WRITE(IWRITE,202)J,(A(JC,JR),JR=1,10)
 200  FORMAT(/* TIME=*,F7.4/10X, 4HZGAP, 9X, 2HZO, 10X, 2HYO,
     2 8X, 4HAGAP, 10X, 2HL1, 10X, 2HL2, 10X, 2HL3, 10X, 2HEP,
     3 10X, 2HVT, 10X, 2HVC)
202   FORMAT(1X,I2,2X,10E12.4)
      WRITE(IWRITE,300)
      DO 50 MM=1,MS
      DO 50 J=1,NE
      JC=J+(MM-1)*NE
 50   WRITE(IWRITE,202) J,(B(JC,JR),JR=1,10)
 300  FORMAT(//,11X, 3HFFX, 9X, 3HFFY, 9X, 2HFD, 10X, 2HFT, 10X,
     2 2HFC, 10X, 2HTX, 10X, 2HTY, 10X, 2HTZ, 10X, 2HAT,10X,2HAC)
C
      NTIM=1
99    RETURN
      END
```

Table 253: LISTING FOR SUBROUTINE YC

```
CYC
      SUBROUTINE YC(FX,FXDOT,IFX,FY,TX,TZ,ED,TM,ST,SR,C1,C2,SIG,GA,
     1 TC,TH,XA,ZA)
C  VERSION  2.  AUG 1977
C
C--PURPOSE  YAW CONTROL THRUSTER
C
C--METHOD   VECTORED THRUST IS A FUNCTION OF SIG IN A SATURATION FUNCTION,
C           WHERE SIG IS AN INPUT FROM AIRCRAFT CONTROL SYSTEM COMPONENT.
C           IF THE ENGINE DEPENDENCE INDICATOR ED IS 1, VECTORED THRUST
C           CAPABILITY IS DEPENDENT ON ENGINE THRUST  ED=0 SETS AVAILABLE
C           VECTORED THRUST TO A CONSTANT.
C--CALL SEQUENCE
C  *****  OUTPUTS  *****
C    FX         ENGINE THRUST REDUCTION
C    FXDOT,IFX  THRUST REDUCTION RATE,INT CONTROL
C    FY         VECTORED THRUST-SIDE FORCE
C    TX         ROLL MOMENT DUE TO THRUSTER NOT ON X-AXIS
C    TZ         YAW MOMENT DUE TO THRUSTER
C  *****  INPUTS  *****
C    ED         ENGINE DEPENDENCE INDICATOR(ED=1.0,YES ED=0.0,NO)
C    TM         THRUSTER MAXIMUM FORCE FOR ENGINE INDEPENDENT SYSTEM
C    ST         SLOPE FOR MAXIMUM AVAILABLE SIDE  THRUST AS FUNCTION
C               OF ENGINE THRUST
C    SR         SLOPE OF ENGINE THRUST REDUCTION AS FUNCTION OF VECTORED THR
C    C1         SATURATION FUNCTION SLOPE
C    C2         SATURATION SLOPE
C    SIG        AIRCRAFT CONTROL SYSTEM SIGNAL TO THRUSTER
C    GA         FIRST ORDER LAG GAIN
C    TC         FIRST ORDER LAG TIME CONSTANT
C    TH         ENGINE THRUST
C    XA         THRUSTER YAW MOMENT ARM
C    ZA         THRUSTER ROLL MOMENT ARM
C
C  WRITTEN BY JOHN MCAVOY
C
C--SWITCH FOR ENGINE DEPENDENCE
      IF(ED.GT.0.5) GO TO 1
      TVA=TM
      GO TO 2
C--AVAILABLE VECTORED THRUST
    1 TVA=ST*TH
C--SATURATION INTERCEPT
    2 C3=TVA/C1
      C6=-C3
      C4=C1
      C5=C2
C--SATURATION FUNCTION,FY(SIG)
      CALL SA(FY,SIG,C1,C2,C3,C4,C5,C6)
      IF(ED.LT.0.5) GO TO 3
C--ENGINE THRUST REDUCTION FUNCTION
      FR=-SR*ABS(FY)
C--1ST ORDER LAG ON ENGINE RESPONSE
      IF(IFX.NE.0) FXDOT=(FR*GA-FX)/TC
    3 CONTINUE
C--VECTORED THRUST MOMENTS
```

Table 253: LISTING FOR SUBROUTINE YC (CONCLUDED)

```
TX=FY*ZA
TZ=FY*XA
RETURN
END
```

SECTION VII

ANALYSIS OF COMPONENTS

Although the EASY model generation and analysis program is primarily intended for the analysis of systems it is possible to use the program for the analysis of single components. The basic procedures and types of analysis and output are fully described in Reference 1, Volume III.

It should be recognized that there are some constraints in running single components. For example, a controller cannot be used in isolation if there is no feedback. Furthermore types of input variable must comply with the input/output lists of particular components. This means that for most single components, the inlet flow and temperature can be specified but not inlet pressure.

SECTION VIII

USER ADDED COMPONENTS

It is recognized that users of the EASY program will wish to add their own dynamic models. As discussed in Reference 1, Volume III, this can be done by inserting the component model in Fortran directly into the model generation program. Alternatively, the new model can be added to the list of standard components.

Before constructing a model, the user should become familiar with the numerous standard functions and subroutines described in Section 3 of this volume. Use of these routines will save a great deal of unnecessary coding. The user should also become familiar with the required order of specifying inputs and outputs in the subroutine call statement, as described in Reference 1, Volume III.

The only other guidelines are really common sense and apply to any dynamics program. The programming of discontinous functions should be avoided wherever possible. Similarly excessive non-linearities are undesirable since they slow down steady state convergence and simulation. For example, if it is desired to represent the force when an actuator hits a stop, it is preferrable to model this by a spring force over the last 1% or 5% of travel rather than a step input at the limit. The eigenvalues of the model should always be examined, and modified if necessary, to avoid unnecessary high frequency dynamic effects.

APPENDIX A

EASY DOCUMENTATION INDEX

The following index provides a cross reference for the following EASY ACLS documents:

- Volume I
- Volume II
- Volume III
- Reference 2 (Volume III Part 1)
- User's Manual (UM) (Reference 1)

Capitalized words in the index are EASY Command Phrases.

REFERENCES

1. Wahi, M. K., Duleba, G. S., and Perkins P. R., EASY - ACLS Dynamic Analysis User's Manual, AFFDL-TR-79-3106, September 1979 (In Press).

2. Environmental Control System (ECS) Transient Analysis, AFFDL-TR-77-102, (Contract F33615-76-C-3100), July 1977.

3. Keenan, J. H. and Kaye, J., Gas Tables, Wiley, 1948.

4. Keenan, J. H. and Keyes, F. G., Thermodynamic Properties of Steam, Wiley 1936.

5. SAE Aerospace Applied Thermodynamics Manual, 2nd Edition, 1969.

6. Amies, G. E., Clark, C., Jones, C. L. and Sheppard, S. M., Survivable Flight Control System, Interim Report No. 1, Studies, Analysis and Approach, AFFDL-TR-71-20, Suipplement 3, May 1971.

☆U.S.Government Printing Office: 1979 — 657-084/374