11. PROGRAM SOURCE LISTINGS

```
C  ***************************************************************************
C  *                                                                         *
C  *              POWER SYSTEM DYNAMIC SIMULATION PROGRAM                    *
C  *                                                                         *
C  ***************************************************************************
C
C  MAINLINE ROUTINE.
      COMPLEX VT,CT,Y,YFICT
      COMPLEX CMPLX,CONJG
      COMMON /BLOCK1/ TIME,TSTEP
      COMMON /BLOCK2/ PBASE(10),H(10),R(10),XL(10),XD(10),XD1(10),
     1XQ(10),XQ1(10),TD1(10),TQ1(10),DAMP(10),C1(10),C2(10)
      COMMON /BLOCK3/ AVRPRM(10,16)
      COMMON /BLOCK4/ TURPRM(10,16)
      COMMON/BLOCK5/ VT(10),CT(10),EF(10),PM(10)
      COMMON/BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
      COMMON/BLOCK7/ Y(11,11)
      COMMON /BLOCK8/ TYM(200),VAR(200,6),NT,NVAR
      COMMON /BLOCK9/ PRTVAR(10,20)
      NT=0
      TIME=0.0
      TFIN=0.0
      NSTEP=0
      NPRINT=0
C  CLEAR INTEGRATOR ARRAYS
      DO 10 I=1,10
      DO 10 J=1,16
      PLUG(I,J)=0.0
      OUT(I,J)=0.0
      SAVE(I,J)=0.0
10    CONTINUE
      DO 12 I=1,10
      DO 12 J=1,20
12    PRTVAR(I,J)=0.0
      WRITE(6,990)
990   FORMAT('1',T39,'POWER  SYSTEM  RESEARCH  GROUP'//
     1T40,'UNIVERSITY  OF  SASKATCHEWAN'//
     2T32,'POWER  SYSTEM  DYNAMIC  SIMULATION  PROGRAM'//)
      READ(5,1000) NGEN,TSTEP,TPRINT
1000  FORMAT(I5,5X,2F10.4)
      WRITE(6,1005) NGEN,TSTEP,TPRINT
1005  FORMAT('ONO. OF GENERATORS',T20,I5/' TIME STEP',T20,F6.3/
     1' PRINT INTERVAL',T20,F6.3)
      WRITE(6,1008)
1008  FORMAT('0GENERATOR PARAMETERS')
C  READ GENERATOR PARAMETERS.
      DO 20 I=1,NGEN
      READ(5,1010)                PBASE(I),H(I),R(I),XL(I),XD(I),
     1XD1(I),XQ(I),XQ1(I),TD1(I),TQ1(I),DAMP(I),C1(I),C2(I)
1010  FORMAT(8F10.4)
      WRITE(6,1015) I,            PBASE(I),H(I),R(I),XL(I),XD(I),
     1XD1(I),XQ(I),XQ1(I),TD1(I),TQ1(I),DAMP(I),C1(I),C2(I)
1015  FORMAT(1X,I5,8G12.4/6X,8G12.4)
C     CONVERT DATA TO 100 MW BASE.
      C=100.0/PBASE(I)
      H(I)=H(I)/C
      R(I)=R(I)*C
      XL(I)=XL(I)*C
      XD(I)=XD(I)*C
      XD1(I)=XD1(I)*C
      XQ(I)=XQ(I)*C
      XQ1(I)=XQ1(I)*C
      DAMP(I)=DAMP(I)/C
20    CONTINUE
      WRITE(6,1018)
1018  FORMAT('0 EXCITATION SYSTEM PARAMETERS')
C  READ EXCITATION SYSTEM PARAMETERS.
      DO 30 I=1,NGEN
      READ(5,1020) (AVRPRM(I,J),J=1,16)
1020  FORMAT(8F10.4)
      WRITE(6,1025) I,(AVRPRM(I,J),J=1,16)
1025  FORMAT(1X,I5,8G12.4/6X,8G12.4)
30    CONTINUE
      WRITE(6,1028)
1028  FORMAT('0 TURBINE-GOVERNOR PARAMETERS')
C  READ TURBINE AND GOVERNOR PARAMETERS.
      DO 40 I=1,NGEN
      READ(5,1030) (TURPRM(I,J),J=1,16)
1030  FORMAT(8F10.4)
      WRITE(6,1035) I,(TURPRM(I,J),J=1,16)
1035  FORMAT(1X,I5,8G12.4/6X,8G12.4)
40    CONTINUE
      WRITE(6,1038)
1038  FORMAT('0INITIAL GENERATOR TERMINAL CONDITIONS'/
     1T3,'GEN',T15,'MW',T26,'MVAR',T36,'VOLTS',T46,'ANGLE')
C  READ CONDITIONS ON TERMINAL BUSES.
      DO 50 I=1,NGEN
      READ(5,1040)    PT,QT,VMAG,VARG
```

```
1040  FORMAT(       2P2F10.4,0P2F10.4)
      WRITE(6,1045) I,PT,QT,VMAG,VARG
1045  FORMAT(1X,I5,5X,2P2F10.3,0P2F10.5)
      VARG=VARG*3.1416/180.0
      VT(I)=VMAG*CMPLX(COS(VARG),SIN(VARG))
      CT(I)=CONJG(CMPLX(PT,QT)/VT(I))
50    CONTINUE
C  CALL EQUIPMENT SUBROUTINES TO CALCULATE INITIAL CONDITIONS.
C  CALL STATEMENTS MUST BE GENERATED BY USER.
C  ***************************************************************************
      CALL GEN1IC(1)
      CALL GEN1IC(2)
      CALL AVR1IC(1)
C  ***************************************************************************
C  LOOP HERE FOR EACH NEW NETWORK CONDITION.
70    CONTINUE
      TOLD=TFIN
C  READ THE CONTROL CARD.
      READ(5,1050) TFIN
1050  FORMAT(F10.4)
      IF(TFIN .EQ. 0.0) GO TO 150
      WRITE(6,1055) TOLD,TFIN
1055  FORMAT('0',T9,'TERMINAL ADMITTANCE MATRIX FROM',F8.3,' TO',F8.3,
     1' SECS.')
C  READ THE NEW ADMITTANCE MATRIX.
      DO 72 I=1,NGEN
      READ(5,1060) (Y(I,J),J=1,NGEN)
1060  FORMAT(8F10.4)
      WRITE(6,1065) (Y(I,J),J=1,NGEN)
1065  FORMAT((T9 ,8F10.4))
72    CONTINUE
      CALL MATRIX(NGEN)
      WRITE(6,1076)
1076  FORMAT('0')
C  LOOP HERE FOR EACH INTEGRATION STEP.
100   CONTINUE
C  SOLVE THE NETWORK.
      CALL NWSOL(NGEN)
C  CALL EQUIPMENT SUBROUTINES TO CALCULATE STATE VARIABLE DERIVATIVES.
C  CALL STATEMENTS MUST BE GENERATED BY USER.
C  ***************************************************************************
      CALL AVR1(1)
      CALL GEN1(1)
      CALL GEN1(2)
C  ***************************************************************************
C  CHECK FOR OUTPUT.
      IF(NSTEP .EQ. 0) CALL OUTPUT(NGEN)
      IF(NPRINT*TSTEP .LT. TPRINT-.0001) GO TO 125
      CALL OUTPUT(NGEN)
      NPRINT=0
125   CONTINUE
C  PERFORM INTEGRATION STEP.
      CALL INT(NGEN)
      NSTEP=NSTEP+1
      TIME=NSTEP*TSTEP
      NPRINT=NPRINT+1
C  CHECK FOR NEW NETWORK CONDITION.
      IF(TIME .LT. TFIN) GO TO 100
C  LOOP BACK FOR NEW NETWORK CONDITION.
      GO TO 70
C  COME HERE WHEN RUN IS COMPLETED.
150   CONTINUE
      CALL PLOT
      STOP
      END


C
C  SUBROUTINE TO CALCULATE EQUIVALENT Y MATRIX FOR INTERNAL
C  GENERATOR BUSES.
      SUBROUTINE MATRIX(NGEN)
      COMMON /BLOCK2/ PBASE(10),H(10),R(10),XL(10),XD(10),XD1(10),
     1XQ(10),XQ1(10),TD1(10),TQ1(10),DAMP(10),C1(10),C2(10)
      COMMON/BLOCK7/ Y(11,11)
      COMPLEX Y,YFICT,CMPLX,CONJG
C  AUGMENT Y MATRIX WITH GENERATOR BUSES AND ELIMINATE THE
C  TERMINAL BUSES.
      N1=NGEN+1
      DO 80 I=1,NGEN
C  MOVE TERMINAL BUS TO OUTSIDE OF MATRIX.
      Y(N1,N1)=Y(I,I)
      Y(I,I)=(0.0,0.0)
      DO 75 J=1,NGEN
C  MOVE ROW
      Y(N1,J)=Y(I,J)
      Y(I,J)=(0.0,0.0)
C  MOVE COLUMN
      Y(J,N1)=Y(J,I)
```

```
      Y(J,I)=(0.0,0.0)
75     CONTINUE
C  ADD IN GENERATOR BUS
      YFICT=CMPLX(R(I),-(XD1(I)+XQ1(I))/2.0)/(R(I)*R(I)+XD1(I)*XQ1(I))
      Y(I,I)=YFICT
C  CHECK IF TERMINAL BUS IS GROUNDED.
      IF(CABS(Y(N1,N1)) .EQ. 0.0) GO TO 80
      Y(N1,N1)=Y(N1,N1)+YFICT
      Y(I,N1)=-YFICT
      Y(N1,I)=-YFICT
C  ELIMINATE THE TERMINAL BUS.
      DO 76 M=1,NGEN
      DO 76 N=M,NGEN
      Y(M,N)=Y(M,N)-Y(M,N1)*Y(N1,N)/Y(N1,N1)
      Y(N,M)=Y(M,N)
76    CONTINUE
80    CONTINUE
      RETURN
      END

C
C  SUBROUTINE TO SOLVE NETWORK AND ARMATURE EQUATIONS.
      SUBROUTINE NWSOL(NGEN)
      COMPLEX SCALE,ROTATE
      REAL ID,IQ
      COMPLEX CMPLX,CONJG,Y,VT,CT
      COMPLEX VOLD,YFICT
      COMMON /BLOCK2/ PBASE(10),H(10),R(10),XL(10),XD(10),XD1(10),
     1XQ(10),XQ1(10),TD1(10),TQ1(10),DAMP(10),C1(10),C2(10)
      COMMON/BLOCK5/ VT(10),CT(10),EF(10),PM(10)
      COMMON/BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
      COMMON /BLOCK7/ Y(11,11)
      COMPLEX EFICT(10),E(10)
       REAL DEL(10)
      DO 10 I=1,NGEN
      DEL(I)=OUT(I,2)
      EQ=OUT(I,3)
      ED=OUT(I,4)
C  TRANSFORM VOLTAGE TO SYNCHRONOUS REFERENCE.
      THETA=DEL(I)-3.1416/2.0
      E(I)=CMPLX(ED,EQ)*CMPLX(COS(THETA ),SIN(THETA ))
10    CONTINUE
      ITER=0
C  LOOP HERE FOR EACH ITERATION.
15    CONTINUE
      ITER=ITER+1
      DO 20 I=1,NGEN
      THETA=DEL(I)-3.1416/2.0
      SCALE=CMPLX(0.0,(XQ1(I)-XD1(I))*0.5)/CMPLX(R(I),-(XQ1(I)+XD1(I))*
     10.5)
      ROTATE=CMPLX(COS(2.0*THETA),SIN(2.0*THETA))
      EFICT(I)=E(I)+SCALE*CONJG(E(I)-VT(I))*ROTATE
20    CONTINUE
      DO 30 I=1,NGEN
      CT(I)=(0.0,0.0)
      DO 25 J=1,NGEN
25    CT(I)=CT(I)+Y(I,J)*EFICT(J)
30    CONTINUE
      DO 40 I=1,NGEN
      YFICT=CMPLX(R(I),-(XD1(I)+XQ1(I))/2.0)/(R(I)*R(I)+XD1(I)*XQ1(I))
      VT(I)=EFICT(I)-CT(I)/YFICT
40    CONTINUE
C  CHECK FOR CONVERGENCE.
      NFLAG=0
      DO 50 I=1,NGEN
      EQ=OUT(I,3)
      ED=OUT(I,4)
C  TRANSFORM TERMINAL VOLTAGE AND CURRENT TO MACHINE REFERENCE.
      THETA=DEL(I)-3.1416/2.0
      ROTATE=CMPLX(COS(THETA),-SIN(THETA))
      ID=REAL(CT(I)*ROTATE)
      IQ=AIMAG(CT(I)*ROTATE)
      VD=REAL(VT(I)*ROTATE)
      VQ=AIMAG(VT(I)*ROTATE)
      IF(ABS(EQ-R(I)*IQ-XD1(I)*ID-VQ) .GT. .001) NFLAG=1
      IF(ABS(ED-R(I)*ID+XQ1(I)*IQ-VD) .GT. .001) NFLAG=1
      VD1=ED-R(I)*ID+XQ1(I)*IQ
      VQ1=EQ-R(I)*IQ-XD1(I)*ID
50    CONTINUE
      IF(ITER .GE. 10) GO TO 60
      IF(NFLAG .EQ. 1) GO TO 15
      RETURN
60    WRITE(6,1025)
1025  FORMAT('0 SALIENCY ITERATIONS NOT CONVERGED')
      DO 70 I=1,NGEN
      WRITE(6,1010) I,VT(I),CT(I),VD,VQ,VD1,VQ1,ID,IQ
1010  FORMAT(' TERM',I5,10F10.4)
```

```
   70    CONTINUE
         STOP
         END

C
C  MODEL OF SYNCHRONOUS GENERATOR WITH FIELD WINDING IN D AXIS AND
C  DAMPER WINDING IN Q AXIS.
         SUBROUTINE GEN1(I)
         COMMON /BLOCK1/ TIME,TSTEP
         COMMON /BLOCK2/ PBASE(10),H(10),R(10),XL(10),XD(10),XD1(10),
        1XQ(10),XQ1(10),TD1(10),TQ1(10),DAMP(10),C1(10),C2(10)
         COMMON/BLOCK5/ VT(10),CT(10),EF(10),PM(10)
         COMMON/BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
         COMMON /BLOCK9/ PRTVAR(10,20)
         REAL CSAT(10)
         COMPLEX CONJG,CMPLX,VT,CT,EQD,CURR
         REAL ID,IQ
C  ENTER HERE FOR EACH INTEGRATION STEP.
C  DEFINE INTEGRATOR OUTPUTS.
         OME=OUT(I,1)
         DEL=OUT(I,2)
         EQ=OUT(I,3)
         ED=OUT(I,4)
C  TRANSFORM CURRENT TO MACHINE REFERENCE.
         CURR=CT(I)*CMPLX(SIN(DEL),COS(DEL))
         ID=REAL(CURR)
         IQ=AIMAG(CURR)
C  CALCULATE GENERATOR OUTPUT PLUS LOSSES.
         PE=REAL(VT(I)*CONJG(CT(I)))
         QE=AIMAG(VT(I)*CONJG(CT(I)))
         PL=CABS(CURR)**2*R(I)
C  ADJUST REACTANCES AND TIME CONSTANTS TO ACCOUNT FOR SATURATION.
         XDS=CSAT(I)*XD(I)+(1.0-CSAT(I))*XL(I)
         XQS=CSAT(I)*XQ(I)+(1.0-CSAT(I))*XL(I)
         IF(XQ1(I) .EQ. XQ(I))XQS=XQ(I)
         TD1S=TD1(I)*(1.0-(1.0-CSAT(I))*(XD(I)-XD1(I))/(XD(I)-XL(I)))
         TQ1S=TQ1(I)*(1.0-(1.0-CSAT(I))*(XQ(I)-XQ1(I))/(XQ(I)-XL(I)))
         EAQ=EQ-(XD1(I)-XL(I))*ID
         EAD=ED+(XQ1(I)-XL(I))*IQ
         IF(XQ1(I) .EQ. XQ(I))EAD=0.0
         EAT=SQRT(EAQ**2+EAD**2)
         CSAT(I)=1.0/(1.0+C1(I)*EXP(C2(I)*EAT))
C  DEFINE INTEGRATOR INPUTS.
C  SET UP PRINTOUT VARIABLES.
         PLUG(I,1)=(PM(I)-PE-PL-DAMP(I)*OME)/(2.0*H(I))
         PLUG(I,2)=377.0*OME
         PLUG(I,3)=(CSAT(I)*EF(I)-EQ-(XDS-XD1(I))*ID)/TD1S
         PLUG(I,4)=(-ED+(XQS  -XQ1(I))*IQ)/TQ1S
         PRTVAR(I,1)=DEL*180.0/3.142
         PRTVAR(I,2)=OME
         PRTVAR(I,3)=EQ
         PRTVAR(I,4)=ED
         PRTVAR(I,5)=CABS(VT(I))
         PRTVAR(I,6)=PE*100.0/PBASE(I)
         PRTVAR(I,7)=QE*100.0/PBASE(I)
         PRTVAR(I,8)=EF(I)
         PRTVAR(I,9)=PM(I)*100.0/PBASE(I)
         PRTVAR(I,10)=CSAT(I)
         PRTVAR(I,11)=EAT
         RETURN
C  ENTER HERE TO CALCULATE INITIAL CONDITIONS.
         ENTRY GEN1IC(I)
         CSAT(I)=1.0
         DO 100 J=1,3
         XDS=CSAT(I)*XD(I)+(1.0-CSAT(I))*XL(I)
         XQS=CSAT(I)*XQ(I)+(1.0-CSAT(I))*XL(I)
         IF(XQ1(I) .EQ. XQ(I))XQS=XQ(I)
C  CALCULATE ANGLE OF GENERATOR Q AXIS.
         EQD=VT(I)+CMPLX(R(I),XQS  )*CT(I)
         DEL=ATAN2(AIMAG(EQD),REAL(EQD))
C  TRANSFORM CURRENT ONTO GENERATOR REFERENCE.
         CURR=CT(I)*CMPLX(SIN(DEL),COS(DEL))
         ID=REAL(CURR)
         IQ=AIMAG(CURR)
         EQ=CABS(EQD)-(XQS  -XD1(I))*ID
         EF(I)=(EQ+(XDS-XD1(I))*ID)/CSAT(I)
         ED=(XQS  -XQ1(I))*IQ
         EAQ=EQ-(XD1(I)-XL(I))*ID
         EAD=ED+(XQ1(I)-XL(I))*IQ
         IF(XQ1(I) .EQ. XQ(I))EAD=0.0
         EAT=SQRT(EAQ**2+EAD**2)
         CSAT(I)=1.0/(1.0+C1(I)*EXP(C2(I)*EAT))
  100    CONTINUE
         PE=REAL(VT(I)*CONJG(CT(I)))+CABS(CURR)**2*R(I)
         PM(I)=PE
         OUT(I,1)=0.0
         OUT(I,2)=DEL
```

```
      OUT(I,3)=EQ
      OUT(I,4)=ED
      RETURN
      END


C
C  ROTATING EXCITATION SYSTEM - IEEE TYPE 1 MODEL.
      SUBROUTINE AVR1(I)
      COMMON /BLOCK1/ TIME,TSTEP
      COMMON/BLOCK3/          KA(10),KE(10),KF(10),TA(10),TE(10),
     1TF(10),VRMIN(10),VRMAX(10),C1(10),C2(10),DUM(10,6)
      COMMON/BLOCK5/ VT(10),CT(10),EF(10),PM(10)
      COMMON/BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
      COMMON /BLOCK9/ PRTVAR(10,20)
      COMPLEX VT,CT
      REAL KA,KE,KF,VREF(10),SE(10)
C  ENTER HERE FOR EACH INTEGRATION STEP.
C  DEFINE INTEGRATOR OUTPUTS.
      X5=OUT(I,6)
      EF(I)=OUT(I,5)
      X3=OUT(I,7)
C  CALCULATE INTERMEDIATE VARIABLES.
      X1=VREF(I)-CABS(VT(I))
      X2=KF(I)/TF(I)*EF(I)-X3
      X4=X1-X2
      X6=X5
      IF(X6 .LT. VRMIN(I)) X6=VRMIN(I)
      IF(X6 .GT. VRMAX(I)) X6=VRMAX(I)
      X7=SE(I)*EF(I)
      X8=X6-X7
C  DEFINE INTEGRATOR INPUTS.
      PLUG(I,5)=X8/TE(I)-KE(I)/TE(I)*EF(I)
      PLUG(I,6)=KA(I)/TA(I)*X4-X5/TA(I)
      PLUG(I,7)=X2/TF(I)
      RETURN
C  ENTER HERE TO CALCULATE INITIAL CONDITIONS.
      ENTRY AVR1IC(I)
      SE(I)=C1(I)*EXP(C2(I)*EF(I))
      VREF(I)=CABS(VT(I))+(KE(I)+SE(I))*EF(I)/KA(I)
      OUT(I,5)=EF(I)
      OUT(I,6)=EF(I)*(KE(I)+SE(I))
      OUT(I,7)=EF(I)*KF(I)/TF(I)
      IF(OUT(I,6) .LT. VRMIN(I)) WRITE(6,1020) I
      IF(OUT(I,6) .GT. VRMAX(I)) WRITE(6,1020) I
1020  FORMAT('0**** AVR VOLTAGE LIMIT IS EXCEEDED BY INITIAL FIELD ON',
     1' UNIT',I3/)
      RETURN
      END


C
C  STATIC EXCITATION SYSTEM - IEEE TYPE 1S MODEL.
      SUBROUTINE AVR2(I)
      COMMON /BLOCK1/ TIME,TSTEP
      COMMON /BLOCK3/ KA(10),KF(10),TA(10),TF(10),KP(10),DUM(10,11)
      COMMON /BLOCK5/ VT(10),CT(10),EF(10),PM(10)
      COMMON /BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
      COMMON /BLOCK9/ PRTVAR(10,20)
      COMPLEX VT,CT
      REAL KA,KF,KP,VREF(10)
C  ENTER HERE FOR EACH INTEGRATION STEP.
C  DEFINE INTEGRATOR OUTPUTS.
      X5=OUT(I,5)
      X3=OUT(I,6)
C  CALCULATE INTERMEDIATE VARIABLES.
      EF(I)=X5
      VMAG=CABS(VT(I))
      IF(X5 .GT. KP(I)*VMAG) EF(I)=KP(I)*VMAG
      IF(X5 .LT. -KP(I)*VMAG) EF(I)=-KP(I)*VMAG
      X2=EF(I)*KF(I)/TF(I)-X3
      X1=VREF(I)-VMAG
      X4=X1-X2
C  CALCULATE INTEGRATOR INPUTS.
      PLUG(I,5)=X4*KA(I)/TA(I)-X5/TA(I)
      PLUG(I,6)=X2/TF(I)
      RETURN
C  ENTER HERE TO CALCULATE INITIAL CONDITIONS.
      ENTRY AVR2IC(I)
      OUT(I,5)=EF(I)
      OUT(I,6)=EF(I)*KF(I)/TF(I)
      VREF(I)=CABS(VT(I))+EF(I)/KA(I)
C  CHECK IF INITIAL CONDITIONS ARE WITHIN LIMITS.
      VMAG=CABS(VT(I))
      IF(EF(I) .GT. KP(I)*VMAG) WRITE(6,1020) I
      IF(EF(I) .LT. -KP(I)*VMAG) WRITE(6,1020) I
1020  FORMAT('0**** AVR VOLTAGE LIMIT IS EXCEEDED BY INITIAL FIELD ON',
     1' UNIT',I3/)
```

```
      RETURN
      END

C
C  MODEL OF ROTATING EXCITER WITH AUXILIARY STABILIZER.
      SUBROUTINE AVR3(I)
      COMMON /BLOCK1/ TIME,TSTEP
      COMMON /BLOCK2/ PBASE(10),H(10),R(10),XL(10),XD(10),XD1(10),
     1XQ(10),XQ1(10),TD1(10),TQ1(10),DAMP(10),C1(10),C2(10)
      COMMON /BLOCK3/ KA(10),TA1(10),TA2(10),KB(10),KE(10),TE(10),
     1KD(10),TD(10),KC(10),TC(10),T1(10),T3(10),KT(10),TFD(10),T(10),
     2UMAX(10)
      REAL KA,KB,KE,KD,KC,KT,VREF(10)
      COMMON /BLOCK5/ VT(10),CT(10),EF(10),PM(10)
      COMMON /BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
      COMMON /BLOCK9/ PRTVAR(10,20)
      COMPLEX CMPLX,CONJG
      COMPLEX VT,CT,EQD
C  ENTER HERE FOR EACH INTEGRATION STEP.
C  DEFINE STABILIZER VARIABLES.
      OME=OUT(I,1)
      X7=OME
      X8=OUT(I,13)
      X9=OUT(I,14)+T1(I)/T3(I)*X8
      X10=OUT(I,15)+T1(I)/T3(I)*X9
      U=X10-OUT(I,16)
      IF(U .GT. UMAX(I)) U=UMAX(I)
      IF(U .LT. -UMAX(I)) U=-UMAX(I)
C  DEFINE STABILIZER INTEGRATOR INPUTS.
      PLUG(I,13)=(KT(I)*X7-X8)/TFD(I)
      PLUG(I,14)=(X8-X9)/T3(I)
      PLUG(I,15)=(X9-X10)/T3(I)
      PLUG(I,16)=U/T(I)
C  DEFINE AVR VARIABLES.
      EF(I)=OUT(I,5)
      X5=CABS(VT(I))*KC(I)/TC(I)-OUT(I,8)
      X6=CABS(VT(I))+X5
      X4=EF(I)*KB(I)*KD(I)/TD(I)-OUT(I,7)
      X1=VREF(I)+U-X4-X6
      X2=X1*KA(I)*TA1(I)/TA2(I)+OUT(I,6)
C  DEFINE AVR INTEGRATOR INPUTS.
      PLUG(I,5)=X2*KE(I)/KB(I)/TE(I)-EF(I)/TE(I)
      PLUG(I,6)=X1*KA(I)/TA2(I)-X2/TA2(I)
      PLUG(I,7)=X4/TD(I)
      PLUG(I,8)=X5/TC(I)
      PRTVAR(I,12)=U
      RETURN
C  ENTER HERE TO CALCULATE INITIAL CONDITIONS.
      ENTRY AVR3IC(I)
C  SET INITIAL CONDITIONS FOR STABILIZER INTEGRATORS.
      OUT(I,13)=0.0
      OUT(I,14)=0.0
      OUT(I,15)=0.0
      OUT(I,16)=0.0
C  SET INITIAL CONDITIONS FOR AVR.
      VREF(I)=EF(I)*KB(I)/KA(I)/KE(I)+CABS(VT(I))
      OUT(I,5)=EF(I)
      OUT(I,6)=EF(I)*KB(I)/KE(I)*(1.0-TA1(I)/TA2(I))
      OUT(I,7)=EF(I)*K3(I)*KD(I)/TD(I)
      OUT(I,8)=CABS(VT(I))*KC(I)/TC(I)
      RETURN
      END


C
C  STEAM TURBINE AND GOVERNOR MODEL.
      SUBROUTINE TUR1(I)
      COMMON /BLOCK1/ TIME,TSTEP
      COMMON /BLOCK4/ PBASE(10),THP(10),TIP(10),TLP(10),FHP(10),FIP(10),
     1 FLP(10),R(10),TS(10),DGMIN(10),DGMAX(10),GMIN(10),GMAX(10),
     1 DUM(10,3)
      COMMON /BLOCK5/ VT(10),CT(10),EF(10),PM(10)
      COMMON /BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
      COMMON /BLOCK9/ PRTVAR(10,20)
      COMPLEX VT,CT
      REAL WREF(10)
C  ENTER HERE FOR EACH TIME STEP.
C  DEFINE INTEGRATOR OUTPUTS.
      OME=OUT(I,1)
      X4=OUT(I,9)
      X5=OUT(I,10)
      X6=OUT(I,11)
      X7=OUT(I,12)
C  CALCULATE INTERMEDIATE VARIABLES.
      G=X4
      IF(X4 .GT. GMAX(I)) G=GMAX(I)
      IF(X4 .LT. GMIN(I)) G=GMIN(I)
```

```
      X1=(WREF(I)-OME)/R(I)
      X2=(X1-G)/TS(I)
      X3=X2
      IF(X2 .GT. DGMAX(I)) X3=DGMAX(I)
      IF(X2 .LT. DGMIN(I)) X3=DGMIN(I)
      PM(I)=(FHP(I)*X5+FIP(I)*X6+FLP(I)*X7)*PBASE(I)/100.0
C  CALCULATE INTEGRATOR INPUTS.
      PLUG(I,9)=X3
      PLUG(I,10)=(G-X5)/THP(I)
      PLUG(I,11)=(X5-X6)/TIP(I)
      PLUG(I,12)=(X6-X7)/TLP(I)
      PRTVAR(I,12)=G
      RETURN
C  ENTER HERE TO CALCULATE INITIAL CONDITIONS.
      ENTRY TUR1IC(I)
      OME=OUT(I,1)
      G=PM(I)*100.0/PBASE(I)/(FHP(I)+FIP(I)+FLP(I))
      OUT(I,9)=G
      OUT(I,10)=G
      OUT(I,11)=G
      OUT(I,12)=G
C  CALCULATE SET POINT.
      WREF(I)=OME+OUT(I,9)*R(I)
C  CHECK IF INITIAL CONDITIONS ARE WITHIN LIMITS.
      IF(OUT(I,9) .GT. GMAX(I)) WRITE(6,1020) I
      IF(OUT(I,9) .LT. GMIN(I)) WRITE(6,1020) I
1020  FORMAT('0**** TURBINE GATE LIMIT IS EXCEEDED BY INITIAL POWER',
     1' ON UNIT',I3/)
      RETURN
      END


C
C  HYDRAULIC TURBINE AND GOVERNOR MODEL.
      SUBROUTINE TUR2(I)
      COMMON /BLOCK1/ TIME,TSTEP
      COMMON /BLOCK4/ PBASE(10),TW(10),TS(10),TR(10),PDR(10),TDR(10),
     1 DGMIN(10),DGMAX(10),GMIN(10),GMAX(10),DUM(10,6)
      COMMON /BLOCK5/ VT(10),CT(10),EF(10),PM(10)
      COMMON /BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
      COMMON /BLOCK9/ PRTVAR(10,20)
      COMPLEX VT,CT
      REAL WREF(10)
C  ENTER HERE FOR EACH TIME STEP.
C  DEFINE INTEGRATOR OUTPUTS.
      OME=OUT(I,1)
      X4=OUT(I,9)
      X5=OUT(I,10)
      X7=OUT(I,11)
C  CALCULATE INTERMEDIATE VARIABLES.
      G=X4
      IF(G .GT. GMAX(I)) G=GMAX(I)
      IF(G .LT. GMIN(I)) G=GMIN(I)
      X6=G*TDR(I)-X5
      X1=WREF(I)-OME
      X2=(X1-X6-G*PDR(I))/TS(I)
      X3=X2
      IF(X2 .GT. DGMAX(I)) X3=DGMAX(I)
      IF(X2 .LT. DGMIN(I)) X3=DGMIN(I)
      X8=2.0*(X7-G)
      PM(I)=X8*PBASE(I)/100.0
C  CALCULATE INTEGRATOR INPUTS.
      PLUG(I,9)=X3
      PLUG(I,10)=X6/TR(I)
      PLUG(I,11)=(G-X8)/TW(I)
      PRTVAR(I,12)=G
      RETURN
C  ENTER HERE TO CALCULATE INITIAL CONDITIONS.
      ENTRY TUR2IC(I)
      OME=OUT(I,1)
      OUT(I,9)=PM(I)*100.0/PBASE(I)
      OUT(I,10)=PM(I)*100.0 /PBASE(I)*TDR(I)
      OUT(I,11)=1.5*PM(I)*100.0/PBASE(I)
C  CALCULATE SET POINT.
      WREF(I)=OME+OUT(I,9)*PDR(I)
C  CHECK IF INITIAL CONDITIONS ARE WITHIN LIMITS.
      IF(OUT(I,9) .GT. GMAX(I)) WRITE(6,1020) I
      IF(OUT(I,9) .LT. GMIN(I)) WRITE(6,1020) I
1020  FORMAT('0**** TURBINE GATE LIMIT IS EXCEEDED BY INITIAL POWER',
     1' ON UNIT',I3/)
      RETURN
      END


C
C  SUBROUTINE TO CALCULATE STATE VARIABLES FOR NEXT INCREMENT IN TIME.
      SUBROUTINE INT(NGEN)
      COMMON /BLOCK1/ TIME,TSTEP
```

```
      COMMON/BLOCK6/ PLUG(10,16),OUT(10,16),SAVE(10,16)
      IF(TIME .GT. 0.0) GO TO 20
      DO 10 I=1,NGEN
      DO 10 J=1,16
10    SAVE(I,J)=PLUG(I,J)
      NFLAG=1
20    CONTINUE
      DO 30 I=1,NGEN
      DO 30 J=1,16
      OUT(I,J)=OUT(I,J)+PLUG(I,J)*TSTEP+(PLUG(I,J)-SAVE(I,J))*0.5*TSTEP
30    SAVE(I,J)=PLUG(I,J)
      RETURN
      END

C
C  SUBROUTINE TO PRINTOUT SYSTEM VARIABLES.
      SUBROUTINE OUTPUT(NGEN)
      COMMON /BLOCK1/ TIME,TSTEP
      COMMON /BLOCK8/ TYM(200),VAR(200,6),NT,NVAR
      COMMON /BLOCK9/ PRTVAR(10,20)
      IF(TIME .EQ. 0.0) WRITE(6,2000)
2000  FORMAT('1'//T42,'SIMULATED RESPONSES',//T4,'TIME  GEN  ROTOR   ',
     1'ROTOR     EQ''      ED''     TERM      ELEC-POWER     FIELD   ',
     2'MECH     SATN    AIR GAP'/T4,'SECS    NO  ANGLE    SPEED     ',
     3'VOLTS    VOLTS   VOLTS     REAL    IMAG     VOLTS    POWER  ',
     4'FACTOR    VOLTS'/)
      WRITE(6,1010) TIME
1010  FORMAT(' ',F7.3)
      DO 10 I=1,NGEN
      WRITE(6,1000) I,(PRTVAR(I,J),J=1,11)
1000  FORMAT(' ',8X,I3,F8.3,13F8.4)
10    CONTINUE
C  SET UP THE VARIABLES TO BE PLOTTED.
      NT=NT+1
      TYM(NT)=TIME
      VAR(NT,1)=PRTVAR(1,1)
      VAR(NT,2)=PRTVAR(1,5)
      VAR(NT,3)=PRTVAR(1,6)
      VAR(NT,4)=PRTVAR(1,8)
      RETURN
      END

C
C  SUBROUTINE TO PLOT GRAPHS OF SYSTEM VARIABLES.
      SUBROUTINE PLOT
      COMMON /BLOCK8/ TYM(200),VAR(200,6),NT,NVAR
      LOGICAL SYMBOL(6),PLUS,NAME(6,40),ALINE(132),BLANK
      DATA SYMBOL/'A','B','C','D','E','F'/
      DATA PLUS/'+'/,BLANK/' '/
      REAL VMAX(6),VMIN(6)
      REAL YVAL(11)
C  READ GRAPH NAMES AND MINIMUM AND MAXIMUM VALUES.
      DO 3 I=1,7
      IF(I .EQ. 7) GO TO 5
3     READ(5,1000,END=5)  (NAME(I,J),J=1,40),VMIN(I),VMAX(I)
1000  FORMAT(40A1,F10.5,F10.5)
5     NVAR=I-1
      IF(NVAR .EQ. 0) RETURN
      WRITE(6,1060)
1060  FORMAT('1')
C  WRITE HEADINGS FOR EACH GRAPH.
      DO 20 I=1,NVAR
      RANGE=VMAX(I)-VMIN(I)
      LOG=ALOG10(RANGE)
      SCALE=1.0*10.0**LOG
      DO 10J=1,11
10    YVAL(J)=(VMIN(I)+(J-1)*RANGE/10.0)/SCALE
      WRITE(6,1010) SYMBOL(I),(NAME(I,J),J=1,40),SCALE
1010  FORMAT('0',T12,A1,' - ',40A1,T93,'SCALE FACTOR = ',1PE7.1)
      WRITE(6,1020) (YVAL(J),J=1,11)
1020  FORMAT(' ',T10,11(F5.2,5X))
20    CONTINUE
C  WRITE Y AXIS.
      WRITE(6,1030) (PLUS,LOC=1,101)
1030  FORMAT('0     TIME  ',101A1)
C  PLOT GRAPHS.
      DO 50 J=1,NT
      DO 28 LOC=1,101
28    ALINE(LOC)=BLANK
      ALINE(1)=PLUS
      IF(MOD( J,10) .NE. 1) GO TO 32
C  INCLUDE GRID POINTS.
      DO 30 LOC=1,101,10
30    ALINE(LOC)=PLUS
32    CONTINUE
C  INCLUDE GRAPH POINTS.
```

```
      DO 35 I=1,NVAR
      LOC=(VAR(J,I)-VMIN(I))/(VMAX(I)-VMIN(I))*100.0+1.0
      IF(LOC .LT. 1) LOC=1
      IF(LOC .GT. 101) LOC=101
      ALINE(LOC)=SYMBOL(I)
35    CONTINUE
      IF(MOD(J,10) .EQ. 1) WRITE(6,1040) TYM(J),(ALINE(LOC),LOC=1,101)
1040  FORMAT(T4,F7.3,1X,101A1)
      IF(MOD(J,10) .NE. 1) WRITE(6,1045) (ALINE(LOC),LOC=1,101)
1045  FORMAT(T12,101A1)
50    CONTINUE
60    CONTINUE
      WRITE(6,1060)
      RETURN
      END
```

```
C  ************************************************************************
C  *                                                                      *
C  *                        NETWORK REDUCTION PROGRAM                     *
C  *                                                                      *
C  ************************************************************************
      COMPLEX Y(40,40),E(40)
      INTEGER TYPE(40)
      INTEGER GBUS(10)
      COMPLEX YSHUNT,YIJ,ZIJ,YII,YJJ,CURR,S
      COMPLEX CMPLX
      COMPLEX CONJG
C    INITIALIZE VARIABLES
      DO 10 I=1,40
      E(I)=(0.0,0.0)
      TYPE(I)=0
      DO 10 J=1,40
   10 Y(I,J)=(0.0,0.0)
      WRITE(6,2000)
2000  FORMAT('1',T33,'NETWORK  REDUCTION  PROGRAM')
      READ(5,990) NBUS,NGEN,NFAULT,NPUNCH
  990 FORMAT(4I5)
      WRITE(6,995) NBUS,NGEN,NFAULT,NPUNCH
995   FORMAT('0'/' NO. OF BUSES',T19,I4 /' NO. OF GENERATORS',T19,I4/
     1' FAULT BUS',T19,I4/' PUNCH FLAG',T19,I4)
      WRITE(6,2005)
2005  FORMAT('0'/' BUS  DATA'/T37,'LOAD       LOAD       GEN        GEN',
     1'        SHUNT      SHUNT'/T4,'BUS   TYPE    VOLTS      ANGLE      ',
     2'MW         MVAR        MW          MVAR        G          B'/)
C.   LOOP HERE FOR EACH BUS
      N=0
   16 READ(5,1001) I,NTYPE,EMAG,ARG,PL,QL,PG,QG,YSHUNT
 1001 FORMAT(I5,T18,I1,F6.4,F6.2,4F6.1,T67,2F6.3)
      IF(I.EQ.0) GO TO 17
      WRITE(6,1002) I,NTYPE,EMAG,ARG,PL,QL,PG,QG,YSHUNT
1002  FORMAT(' ',2I5,F10.4,F10.3,4F10.2,2F10.4)
      ARG=ARG*3.14159/180.0
      Y(I,I)=Y(I,I)+YSHUNT+CMPLX(PL,-QL)*.01/EMAG**2
      TYPE(I)=NTYPE
      E(I)=EMAG*CMPLX(COS(ARG),SIN(ARG))
      IF(TYPE(I) .EQ. 0) GO TO 16
      N=N+1
      GBUS(N)=I
      GO TO 16
   17 CONTINUE
      NGEN=N
      WRITE(6,2010)
2010  FORMAT('0'/' LINE  DATA'/'   BUS  BUS  RESISTANCE  REACTANCE',
     1'   SUSCEPTANCE'/)
C
C    LOOP HERE FOR EACH LINE TO BE READ
   20 READ(5,1003) I,J,ZIJ,B
 1003 FORMAT(2I5,3F10.5)
      IF(I.EQ.0) GO TO 25
      WRITE(6,1005)I,J,ZIJ,B
1005  FORMAT(' ',2I5,F10.5,2F12.5)
      YIJ=(1.0,0.0)/ZIJ
      Y(I,I)=Y(I,I)+YIJ+CMPLX(0.0,B/2.0)
      Y(J,J)=Y(J,J)+YIJ+CMPLX(0.0,B/2.0)
      Y(I,J)=Y(I,J)-YIJ
      Y(J,I)=Y(J,I)-YIJ
      GO TO 20
   25 CONTINUE
      WRITE(6,2015)
2015  FORMAT('0'/' TRANSFORMER  DATA'/'   BUS  BUS  RESISTANCE  ',
     1'REACTANCE    TAP'/)
C
C    LOOP HERE FOR EACH TRANSFORMER CARD TO BE READ
   30 READ(5,1004)I,J,ZIJ,RATIO
 1004 FORMAT(2I5,2F10.5,F6.4)
      IF(I.EQ.0) GO TO 35
      WRITE(6,1006) I,J,ZIJ,RATIO
1006  FORMAT(2I5,F10.4,2X,F10.4,F10.4)
      YIJ=(1.0,0.0)/ZIJ
      YII=YIJ*(1.0/RATIO-1.0)/RATIO
      YJJ=YIJ*(1.0-1.0/RATIO)
      YIJ=YIJ/RATIO
      Y(I,I)=Y(I,I)+YII+YIJ
      Y(J,J)=Y(J,J)+YJJ+YIJ
      Y(I,J)=Y(I,J)-YIJ
      Y(J,I)=Y(J,I)-YIJ
      GO TO 30
   35 CONTINUE
      IF(NFAULT.EQ.0) GO TO 39
C    ZERO ROW AND COLUMN AT THE FAULTED BUS
      DO 38 I=1,NBUS
      Y(I,NFAULT)=(0.0,0.0)
      Y(NFAULT,I)=(0.0,0.0)
38    CONTINUE
```

```
39    CONTINUE
C  PERFORM KRON ELIMINATION ON LOAD BUSES.
      DO 60 M=1,NBUS
      IF(M .EQ. NFAULT) GO TO 60
      IF(TYPE(M) .EQ. 1) GO TO 60
      IF(CABS(Y(M,M)) .EQ. 0.0) GO TO 60
      TYPE(M)=-1
      DO 50 I=1,NBUS
C  ONLY PROCESS ROWS AND COLUMNS WHICH HAVE NOT BEEN ELIMINATED.
      IF(TYPE(I) .EQ. -1) GO TO 50
      DO 40 J=1,NBUS
      IF(TYPE(J) .EQ. -1) GO TO 40
      Y(I,J)=Y(I,J)-Y(I,M)*Y(M,J)/Y(M,M)
40    CONTINUE
50    CONTINUE
60    CONTINUE
      WRITE(6,2020)
2020  FORMAT('0'/'   COMPUTED GENERATION'/' BUS        MW        MVAR'/)
C    CALCULATE GENERATOR POWERS TO CHECK REDUCTION
      DO 80 I=1,NGEN
      CURR=(0.0,0.0)
      DO 75 J=1,NGEN
75    CURR=CURR+Y(GBUS(I),GBUS(J))*E(GBUS(J))
      S=E(GBUS(I))*CONJG(CURR)
      WRITE(6,1040) GBUS(I),S
1040  FORMAT(' ',I5,2P2F10.2)
   80 CONTINUE
      WRITE(6,2025)
2025  FORMAT('0'/'   EQUIVALENT  GENERATOR  BUS  ADMITTANCE  MATRIX'/)
      DO 85 I=1,NGEN
85    WRITE(6,1050) (Y(GBUS(I),GBUS(J)),J=1,NGEN)
1050  FORMAT((' ',8F10.4))
      IF(NPUNCH .NE. 1) GO TO 100
      DO 90 I=1,NGEN
90    WRITE(7,1060) (Y(GBUS(I),GBUS(J)),J=1,NGEN)
1060  FORMAT(8F10.4)
100   CONTINUE
      WRITE(6,2030)
2030  FORMAT('1')
      STOP
      END
```