DECNCM - A FORTRAN PACKAGE FOR LOAD FLOW ANALYSIS WITH THE AID OF DECOMPOSITION

J.W. Bandler, M.A. El-Kady and J. Wojciechowski

SOS-83-11-L

August 1983

DECNCM - A FORTRAN PACKAGE FOR LOAD FLOW ANALYSIS WITH THE AID OF DECOMPOSITION

J.W. Bandler, M.A. El-Kady and J. Wojciechowski

Abstract

This document contains a listing of the package DECNCM described in [1] for power system load flow analysis. The package implements the Newton complex method and also employs decomposition techniques for solving large systems of sparse linear equations.

The package DECNCM has been developed for the CDC 170/730 system with the NOS 1.4 level 552 operating system and the Fortran Extended (FTN) version 4.8 compiler. The listing contains a total of 862 lines (including 211 comments) constituting thirteen subroutines. The listing does not include the package CSDSLE for solving large systems of decomposed equations.

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant G0647.

The authors are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

M.A. El-Kady is also with Ontario Hydro, Toronto, Canada.

J. Wojciechowski is on leave from the Institute of Electronics Fundamentals, Technical University of Warsaw, Warsaw, Poland.

I. INTRODUCTION

DECNCM is a package of thirteen Fortran subroutines for power system load flow analysis. The package implements the Newton complex method [2,3] and also employs decomposition techniques for solving large sytems of sparse linear equations by an appropriate call to the package CSDSLE [4].

The package DECNCM and documentation have been developed in Fortran IV for use on the CDC 170/730 system with the NOS 1.4 level 552 operating system. The package is available at McMaster University in the form of a library of binary relocatable subroutines. The library is in the group indirect file LIBDNCM accessible under the charge RJWBAND. The package calls subroutine CSDSLE1 of the package CSDSLE for solving large systems of sparse decomposed equations; the package CSDSLE must thus be available when DECNCM is used. The document does not include the package CSDSLE. Information concerning this package is found in [4].

The package DECNCM contains 862 lines of which 211 are comments.

II. REFERENCES

[1] J.W. Bandler, M.A. El-Kady and J. Wojciechowski, "DECNCM - A Fortran package for load flow analysis with the aid of decomposition", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-83-11-L, 1983.

[2] J.W. Bandler and M.A. El-Kady, "A generalized, complex adjoint approach to power network sensitivities", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-80-17-R, 1980.

[3] J.W. Bandler and M.A. El-Kady, "Newton's load flow in complex mode", Proc. ECCTD (The Hague, Holland, August 1981), pp. 500-505.

[4] J.A. Starzyk and J.W. Bandler, "CSDSLE - A Fortran package for the solution of sparse decomposed systems of linear equations", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-83-2-U, 1983.

III. LISTING OF THE DECNCM PACKAGE

Subroutine	Number of lines (source text)	Number of words (compiled code)	Listing from page
LFNCD	101	1026	4
CCURR	19	50	6
DECAK	95	520	7
DECDS	69	234	9
DGR	34	100	11
FORMK	106	457	12
INTRN	29	53	14
MDDTF	108	315	15
RHSLD	39	327	17
FRPR	82	442	18
FORMU	43	200	20
FORMYT	59	315	21
RDAT	78	720	22

```
C
C
C
      SUBROUTINE LFNCD (YT,JRYT,ICYT,BCV,V,BNR,BTYP,DEC,IBV,IBL,CMPWA,IN
     1TWA,NB,NTL,ITEL,TIMEL,VEPS,INPT,OTPT,IWRITE)
C
C     SUBROUTINE LFNCD IS THE HIGHEST LEVEL SUBROUTINE FOR SOLVING
C     LOAD FLOW PROBLEMS USING THE NEWTON COMPLEX METHOD AND
C     THE DECOMPOSITION OF SPARSE SYSTEM OF LINEAR EQUATIONS
C
C     LIBRARY : CSDSLE (SEE SOS-83-2)
C
      INTEGER JRYT(1),ICYT(1),BNR(1),BTYP(1),DEC(1),IBV(1),IBL(1),INTWA(
     11),OTPT
      COMPLEX YT(1),BCV(1),CMPWA(1),V(1),DV
C
      NSUB=DEC(1)
      NBV=DEC(2)
      JDS=1
      JAI=JDS+2*NB-2
      JSOLR=JAI+NB-1
      JAK=JSOLR+4*NB-4
      JJRK=JAK+4*NB+6*NTL
      JIRK=JJRK+NB/2+1
      JICK=JIRK+2*NB+3*NTL
      JDEG=JICK+2*NB+3*NTL
      JINTN=JDEG+NBV
      JLNGTH=JINTN+NB
      JMD=JLNGTH+(NSUB+1)/2
      JIM=JMD+(5*NSUB+2)/2
      JIN=JIM+3
      JCM=JIN+NSUB+15+2**(NSUB-2)
      N=NB-1
      NR=2*N
      ICALL=0
      CALL DGR (DEC,IBV,IBL,BNR,CMPWA(JDEG),N)
      CALL SECOND (TS)
      TT=TS
      TA=0
10    ICALL=ICALL+1
      EPS=0.
      CALL RHSLD (V,YT,JRYT,ICYT,BTYP,BNR,BCV,CMPWA(JDS),CMPWA(JAI),N,OT
     1PT,IWRITE)
      CALL FORMK (YT,JRYT,ICYT,BTYP,BNR,V,CMPWA(JAI),CMPWA(JAK),CMPWA(JJ
     1RK),CMPWA(JIRK),CMPWA(JICK),N,NK,OTPT,IWRITE)
      CALL MDDTF (CMPWA(JIRK),CMPWA(JICK),DEC,CMPWA(JIN),CMPWA(JIM),BNR,
     1IBL,CMPWA(JINTN),CMPWA(JLNGTH),CMPWA(JCM),CMPWA(JMD),NB,NK,ICALL)
      DO 20 ISUB=1,NSUB
      CALL DECDS (DEC,IBL,BNR,CMPWA(JMD),CMPWA(JINTN),CMPWA(JDEG),CMPWA(
     1JDS),CMPWA(JCM),ISUB,N,OTPT,IWRITE)
      CALL DECAK (DEC,IBL,BNR,CMPWA(JMD),CMPWA(JINTN),CMPWA(JDEG),CMPWA(
     1JAK),CMPWA(JJRK),CMPWA(JIRK),CMPWA(JICK),CMPWA(JCM),ISUB,N,1,OTPT,
     2IWRITE)
20    CONTINUE
      CALL CSDSLE1 (CMPWA(JIN),INTWA,CMPWA(JCM),CMPWA(JIM),CMPWA(JSOLR))
      IAR=IFIX(REAL(CMPWA(JIN+1)))
      IF (IAR.EQ.1) GO TO 90
      JJ=JSOLR+1
      DO 30 I=1,N
      DV=CMPWA(JJ)
      R=CABS(DV)
      L=IFIX(REAL(CMPWA(JJ+1)))
      JJ=JJ+4
      L=(L+1)/2
      V(L)=V(L)+DV
```

```
      IF (R.GT.EPS) EPS=R
30    CONTINUE
      IF (IWRITE.LT.2) GO TO 40
      WRITE (OTPT,120)
      WRITE (OTPT,130) (L,V(BNR(L)),L=1,N)
40    CALL SECOND (TIME)
      TT=TIME-TS
      TI=TT-TA
      TA=TT
      IF (IWRITE.GE.1) WRITE (OTPT,110) ICALL,EPS,TI,TT
      IF (EPS.LE.VEPS) GO TO 70
      IF (ICALL.GE.ITEL) GO TO 50
      IF (TT.GE.TIMEL) GO TO 60
      GO TO 10
50    IFLAG=1
      GO TO 70
60    IFLAG=2
70    ITEL=ICALL
      VEPS=EPS
      TIMEL=TT
      IF (IWRITE.LT.1) GO TO 80
      WRITE (OTPT,140) ITEL,VEPS,TIMEL
      WRITE (OTPT,120)
      WRITE (OTPT,130) (L,V(BNR(L)),L=1,N)
80    RETURN
90    WRITE (OTPT,100) IAR
      STOP
100   FORMAT (//" RETURN FLAG FROM CSDSLE:",I2/)
110   FORMAT (//" IT =",I2,6X,"EPS =",E12.6,6X,"ITERATION TIME ",F6.3,6X
     1,"TOTAL TIME ",F6.3/)
120   FORMAT (//" VECTOR OF BUS VOLTAGES"/)
130   FORMAT ((3(3X,I4,1X,2(1X,E13.7))))
140   FORMAT (///" RESULTS OF ANALYSIS"//" NUMBER OF ITERATIONS:",6X,I3/
     1" ACCURACY OBTAINED:",2X,E10.4/" ANALYSIS TIME:",10X,F6.3," SECOND
     2S")
      END
```

```
C
C
C
      COMPLEX FUNCTIONCCURR(V,YT,JRYT,ICYT,M)
C
C     THIS FUNCTION SUBPROGRAM CALCULATES THE VALUE OF CURRENT INJECTED
C     INTO THE MTH BUS FOR THE GIVEN VECTOR V OF BUS VOLTAGES
C
      INTEGER JRYT(1),ICYT(1)
      COMPLEX V(1),YT(1)
C
      K1=JRYT(M)
      K2=JRYT(M+1)-1
      CCURR=(0.,0.)
      DO 10 I=K1,K2
      CCURR=CCURR+YT(I)*V(ICYT(I))
10    CONTINUE
      RETURN
      END
```

```
C
C
C
      SUBROUTINE DECAK (DEC,IBL,BNR,MD,INTN,DEG,AK,JRK,IRK,ICK,CMPLXWA,I
     1SUB,N,ICALL,OTPT,IWRITE)
C
C     SUBROUTINE DECAK INITIALIZES IN THE COMPLEX WORKSPACE VECTOR
C     CMPLXWA DATA DESCRIBING THE SUBSYSTEM ISUB (I.E., VECTORS AK,
C     NROW,NCOL)
C
      INTEGER DEC(1),IBL(1),INTN(1),BNR(1),DEG(1),MD(1),IRK(1),ICK(1),JR
     1K(1),OTPT
      COMPLEX CMPLXWA(1),AK(1)
C
C     NINT IS THE NUMBER OF INTERNAL VERTICES OF A GRAPH REPRESENTING
C     SET OF EQUATIONS
C
      NINT=2*(N-DEC(2))
      CALL INTRN (DEC,IBL,BNR,INTN,2*N,ISUB)
C
C     JST (JLT) IS AN INDEX IN VECTOR IBL OF THE FIRST (THE LAST)
C     VERTEX OF SUBSYSTEM ISUB
C
      JLT=0
      DO 10 I=1,ISUB
      JST=JLT+1
      JLT=JLT+DEC(I+I+1)
10    CONTINUE
      KK=DEC(ISUB+ISUB+2)
C
C     NINTJ IS THE NUMBER OF THE INTERNAL VERTICES OF SUBGRAPH ISUB
C
      NINTJ=KK+KK
      J2=JST+KK-1
      ISB=5*ISUB
      IA1=MD(ISB-3)-1
      IA2=MD(ISB-2)-1
      I1=IA1
      I2=IA2
      IF (IWRITE.LT.3) GO TO 20
      WRITE (OTPT,70) ISUB
20    DO 40 K=JST,J2
C
C     TWO ROWS ASSOCIATED WITH AN INTERNAL NODE OF THE SUBSYSTEM
C     ISUB ARE NOW INITIALISED
C
      L=BNR(IBL(K))
      IST=JRK(L)
      ILT=JRK(L+1)-1
      IS1=I1
      IF (ICALL.EQ.1) IS2=I2
      DO 30 J=IST,ILT
      I1=I1+1
      CMPLXWA(I1)=AK(J)
      IF (ICALL.NE.1) GO TO 30
      I2=I2+1
      CMPLXWA(I2)=CMPLX(FLOAT(INTN(IRK(J))),FLOAT(INTN(ICK(J))))
30    CONTINUE
      IF (IWRITE.LT.3) GO TO 40
      WRITE (OTPT,80) BNR(L)
      NE=I1-IS1
      WRITE (OTPT,90) (CMPLXWA(IS2+J),CMPLXWA(IS1+J),J=1,NE)
40    CONTINUE
      J2=J2+1
      DO 60 K=J2,JLT
```

```
C
C     TWO ROWS ASSOCIATED WITH A PARTION BUS OF THE SUBSYSTEM ISUB
C     ARE NOW INITIALISED
C
      L=BNR(IBL(K))
      IST=JRK(L)
      ILT=JRK(L+1)-1
      IS1=I1
      IF (ICALL.EQ.1)  IS2=I2
      DO 50 J=IST,ILT
      LR=IRK(J)
      LC=ICK(J)
      IF (INTN(LC).EQ.0)  GO TO 50
      I1=I1+1
      CMPLXWA(I1)=AK(J)
      IF (LC.GT.NINT)  CMPLXWA(I1)=CMPLXWA(I1)/DEG(LR-NINT)
      IF (ICALL.NE.1)  GO TO 50
      I2=I2+1
      CMPLXWA(I2)=CMPLX(FLOAT(INTN(LR)),FLOAT(INTN(LC)))
50    CONTINUE
      IF (IWRITE.LT.3)  GO TO 60
      WRITE (OTPT,80)  BNR(L)
      NE=I1-IS1
      WRITE (OTPT,90)  (CMPLXWA(IS2+J),CMPLXWA(IS1+J),J=1,NE)
60    CONTINUE
      RETURN
70    FORMAT (//" MATRIX OF SUBSYSTEM:",I2/)
80    FORMAT (" BUS NO.",I3,":",I3)
90    FORMAT ((3(3X,2F5.0,":",2(1X,E11.5))))
      END
```

```
C
C
C
      SUBROUTINE DECDS (DEC,IBL,BNR,MD,INTN,DEG,DS,CMPLXWA,ISUB,N,OTPT,I
     1WRITE)
C
C     SUBROUTINE  DECDS  INITIALIZES IN THE COMPLEX  WORKSPACE
C     VECTOR CMPLXWA SUBVECTOR OF THE RHS VECTOR DS ASSOCIATED
C     WITH THE SUBSYSTEM ISUB
C
      INTEGER DEC(1),IBL(1),BNR(1),DEG(1),MD(1),OTPT
      COMPLEX CMPLXWA(1),DS(1)
C
      NINT=2*(N-DEC(2))
C
C     JST (JLT) IS THE INDEX IN VECTOR IBL OF THE FIRST (THE LAST)
C     VERTEX OF SUBSYSTEM ISUB
C
      JLT=0
      DO 10 I=1,ISUB
      JST=JLT+1
      JLT=JST+DEC(2*I+1)-1
10    CONTINUE
C
C     J2 IS THE INDEX OF THE LAST INTERNAL BUS OF THE SUBSYSTEM
C     ISUB APPEARING IN VECTOR IBL
C
      KK=ISUB+ISUB+2
      J2=JST+DEC(KK)-1
C
C     IA IS THE STARTING POSITION IN VECTOR CMPLXWA OF THE
C     SUBVECTOR  ISUB  OF  THE RIGHT-HAND  SIDE  VECTOR DS
C
      IA=MD(5*ISUB-1)-1
      IST=IA
C
C     ELEMENTS ASSOCIATED WITH INTERNAL BUSES  OF THE SUBSYSTEM
C     ISUB ARE NOW INITIALIZED
C
      DO 20 K=JST,J2
      L=BNR(IBL(K))
      L=L+L-1
      IA=IA+1
      CMPLXWA(IA)=DS(L)
      IA=IA+1
      CMPLXWA(IA)=DS(L+1)
20    CONTINUE
C
C     ELEMENTS ASSOCIATED WITH PARTITION BUSES OF THE SUBSYSTEM
C     ISUB ARE NOW INITIALIZED
C
      J2=J2+1
      DO 30 K=J2,JLT
      L=BNR(IBL(K))
      L=L+L-1
      IDG=DEG(L-NINT)
      IA=IA+1
      CMPLXWA(IA)=DS(L)/IDG
      IA=IA+1
      CMPLXWA(IA)=DS(L+1)/IDG
30    CONTINUE
      IF (IWRITE.LT.3) RETURN
      L=JLT-JST+1
      JST=JST-1
      WRITE (OTPT,40) (ISUB,(IBL(JST+K),CMPLXWA(IST+2*K-1),K=1,L))
```

```
      RETURN
40    FORMAT (//" RIGHT HAND SIDE VECTOR OF SUBSYSTEM",I2/(1X,4(2X,I3,":
     1",2(1X,E11.5))))
      END
```

```
C
C
C
      SUBROUTINE DGR (DEC,IBV,IBL,BNR,DEG,N)
C
C     THIS SUBROUTINE CALCULATES DEGREES OF PARTITION VERTICES
C     OF A GRAPH REPRESENTING THE SET OF EQUATIONS
C
      INTEGER DEC(1),IBV(1),IBL(1),BNR(1),DEG(1)
C
      NSUB=DEC(1)
      NBV=DEC(2)
      NINT=2*(N-NBV)
C
C     NINT IS THE NUMBER OF INTERNAL GRAPH VERTICES
C
      DO 30 I=1,NBV
      J2=0
      IVER=IBV(I)
      IDGR=0
      DO 20 L=1,NSUB
      K=L+L+1
      J1=J2+DEC(K+1)+1
      J2=DEC(K)+J2
      DO 10 J=J1,J2
      IF (IBL(J).EQ.IVER) IDGR=IDGR+1
10    CONTINUE
20    CONTINUE
      K=2*BNR(IVER)-NINT
      DEG(K)=IDGR
      DEG(K-1)=IDGR
30    CONTINUE
      RETURN
      END
```

```
C
C
C
      SUBROUTINE FORMK (YT,JRYT,ICYT,BTYP,BNR,V,AI,AK,JRK,IRK,ICK,N,NK,O
     1TPT,IWRITE)
C
C     THIS SUBROUTINE FORMULATES MATRIX OF PERTURBED FLOW EQUATIONS
C     (SEE EQUATION (16) OF SOS-80-2). VECTORS AK,IRK,ICK STORE THE
C     THE PERTURBED MATRIX IN A SPARSE FORM
C
      INTEGER JRYT(1),ICYT(1),BTYP(1),IRK(1),ICK(1),BNR(1),JRK(1),OTPT
      COMPLEX YT(1),V(1),AK(1),AI(1),CPX,CPY,CCURR,Z,CM
C
      NK=0
      IF (IWRITE.GE.3) WRITE (OTPT,60)
      DO 50 IR=1,N
C
C     MATRIX YT IS ANALYSED ROW BY ROW IN THIS LOOP
C
      K1=JRYT(IR)
      K2=JRYT(IR+1)-1
      CCURR=AI(IR)
      Z=CONJG(V(IR))
      JE=NK+1
      JR1=IR+IR
      JR=JR1-1
      IF (BTYP(BNR(IR)).EQ.1) GO TO 20
C
C     SETTING UP ELEMENTS OF A ROW OF MATRIX K CORRESPONDING
C     TO LOAD BUS
C
      DO 10 J=K1,K2
      IC=ICYT(J)
      JC1=IC+IC
      JC=JC1-1
      IF (BTYP(BNR(IC)).EQ.2) GO TO 10
      CPX=YT(J)*Z
      NK=NK+1
      AK(NK)=CPX
      IRK(NK)=JR
      ICK(NK)=JC
      NK=NK+1
      AK(NK)=CONJG(CPX)
      IRK(NK)=JR1
      ICK(NK)=JC1
10    CONTINUE
      NK=NK+1
      AK(NK)=CCURR
      IRK(NK)=JR
      ICK(NK)=JR1
      NK=NK+1
      AK(NK)=CONJG(CCURR)
      IRK(NK)=JR1
      ICK(NK)=JR
      GO TO 40
20    DO 30 J=K1,K2
C
C     SETTING UP ELEMENTS OF A ROW OF MATRIX K CORRESPONDING
C     TO GENERATOR BUS
C
      IC=ICYT(J)
      IF (BTYP(BNR(IC)).EQ.2) GO TO 30
      JC1=IC+IC
      JC=JC1-1
      CPX=0.5*YT(J)*Z
```

```
      CPY=CONJG(CPX)
      NK=NK+1
      AK(NK)=CPX
      IRK(NK)=JR
      ICK(NK)=JC
      NK=NK+1
      AK(NK)=CPY
      IRK(NK)=JR
      ICK(NK)=JC1
      NK=NK+1
      AK(NK)=CPX
      IRK(NK)=JR1
      ICK(NK)=JC
      NK=NK+1
      AK(NK)=CPY
      IRK(NK)=JR1
      ICK(NK)=JC1
30    CONTINUE
      CM=CMPLX(0.,1./CABS(Z))
      CPX=(CONJG(CCURR)+CM*Z)/2.
      CPY=(CCURR+CM*V(IR))/2.
      AK(JE)=AK(JE)+CPX
      AK(JE+1)=AK(JE+1)+CPY
      AK(JE+2)=AK(JE+2)+CONJG(CPY)
      AK(JE+3)=AK(JE+3)+CONJG(CPX)
40    JRK(IR)=JE
      IF (IWRITE.LT.3) GO TO 50
      WRITE (OTPT,70) BNR(IR)
      WRITE (OTPT,80) (IRK(J),ICK(J),AK(J),J=JE,NK)
50    CONTINUE
C
C     ON RETURN, JRK(L)  POINTS OUT THE LOCATION OF THE FIRST  ELEMENT
C     OF TWO ROWS OF MATRIX K CORRESPONDING TO THE LTH BUS (L=1,..,N).
C     JRK(N+1)  IS THE FIRST VACANT POSITION IN MATRIX K
C
      JRK(N+1)=NK+1
      RETURN
60    FORMAT (//" MATRIX K OF PERTURBED FLOW EQUATIONS"/)
70    FORMAT (1X,"BUS NO.",I3)
80    FORMAT ((3(3X,I3,1X,I3,":",2(1X,E11.5))))
      END
```

```
C
C
C
      SUBROUTINE INTRN (DEC,IBL,BNR,INTN,NR,ISUB)
C
C     SUBROUTINE INTRN IDENTIFIES AND ORDERS VERTICES OF THE SUBGRAPH
C     ISUB
C
      INTEGER DEC(1),BNR(1),IBL(1),INTN(1)
C
      INDX=0
      DO 10 I=1,NR
      INTN(I)=0
10    CONTINUE
      JLT=0
      DO 20 I=1,ISUB
      JST=JLT+1
      JLT=JST+DEC(2*I+1)-1
20    CONTINUE
      DO 30 I=JST,JLT
      K=2*BNR(IBL(I))-1
      INDX=INDX+1
      INTN(K)=INDX
      K=K+1
      INDX=INDX+1
      INTN(K)=INDX
30    CONTINUE
      RETURN
      END
```

```
C
C
C
      SUBROUTINE MDDTF (IRK,ICK,DEC,IN,IM,BNR,IBL,INTN,LNGTH,CMPLXWA,MD,
     1NB,NK,ICALL)
C
C     SUBROUTINE  MDDTF ORGANIZES COMPLEX WORKSPACE VECTOR CMPLXWA FOR
C     THE DATA DESCRIBING DECOMPOSED POWER SYSTEM AND INITIALIZES SOME
C     ELEMENTS OF CMPLXWA (I.E., N,NINT AND VECTOR NON)
C
      INTEGER IRK(1),ICK(1),DEC(1),BNR(1),IBL(1),IN(1),MD(1),IM(1),LNGTH
     1(1),INTN(1)
      COMPLEX CMPLXWA(1)
C
      IF (ICALL.GT.1)  IN(8)=7
      IF (ICALL.GT.1)  GO TO 50
      NR=2*(NB-1)
      NSUB=DEC(1)
      NBV=2*DEC(2)
      IHIGH=NSUB+NSUB-1
C
C     VECTOR IN IS NOW INITIALIZED
C
      IN(4)=NSUB
      IN(5)=NSUB
      IN(8)=6
      IN(9)=IHIGH
      IN(10)=NSUB
      IN(15)=1
      IN(19)=NR-NBV+1
      IN(20)=1
      IN(22)=0
      IN(26)=NBV
      L=26+IN(9)/2
      L1=L+NSUB
      DO 10 I=1,NSUB
      IN(L+I)=IHIGH
      IN(L1+I)=IHIGH
      IHIGH=IHIGH-1
10    CONTINUE
      CALL INDICAT (IN,IM)
      IAS=3*IN(16)+4*IN(17)+IN(11)+1
      IM(1)=0
      IM(2)=0
      IM(5)=NR
C
C     VECTOR MD IDENTIFIES  THE LOCATIONS OF DATA  SUBVECTORS IN THE
C     WORKSPACE  VECTOR CMPLXWA.  ENTRIES FROM  MD(5*K-4) TO MD(5*K)
C     POINTS OUT THE LOCATIONS OF THE FIRST  ELEMENTS OF VECTORS CE,
C     AK,NROW,V,NON, RESPECTIVELY, ASSOCIATED WITH THE KTH SUBSYSTEM
C
      NSUB=DEC(1)
      DO 20 I=1,NSUB
      LNGTH(I)=0
20    CONTINUE
      DO 40 ISUB=1,NSUB
      CALL INTRN (DEC,IBL,BNR,INTN,NR,ISUB)
      DO 30 I=1,NK
      IF (INTN(IRK(I)).NE.0.AND.INTN(ICK(I)).NE.0) LNGTH(ISUB)=LNGTH(ISU
     1B)+1
30    CONTINUE
40    CONTINUE
50    L=NSUB+NSUB
      INIT=IAS+L
      IND=1
```

```
      DO 70 K=1,NSUB
      K1=5*K
      MD(K1-4)=INIT
      MD(K1-3)=INIT+2
      N=DEC(1+2*K)
      CMPLXWA(INIT)=CMPLX(FLOAT(2*N),FLOAT(2*DEC(2+2*K)))
      R=FLOAT(LNGTH(K))
      CMPLXWA(INIT+1)=CMPLX(R,R)
C
C     N IS THE NUMBER OF BUSES OF SUBSYSTEM. NUMBER OF EQUATIONS IS 2*N
C
      MD(K1-2)=MD(K1-3)+LNGTH(K)
      MD(K1-1)=MD(K1-2)+LNGTH(K)
      LL=MD(K1-1)+N+N
      MD(K1)=LL
      INIT=LL+N-1
      DO 60 KK=LL,INIT
      II=BNR(IBL(IND))
      IND=IND+1
      I2=II+II
      CMPLXWA(KK)=CMPLX(FLOAT(I2-1),FLOAT(I2))
60    CONTINUE
      INIT=INIT+1
70    CONTINUE
      LL=K1+1
      MD(LL)=INIT
      IN(25)=INIT-IAS
C
C     DIRECTORY IS NOW INITIALIZED
C
      IHIGH=IN(9)+1
      IDIR=IAS-1
      IA=IDIR+IN(10)
      DO 80 K=1,NSUB
      CMPLXWA(IDIR+K)=CMPLX(FLOAT(IHIGH-K),1.)
      K1=5*K
      IS=MD(K1-4)
      LG=MD(K1+1)-IS
      IS=IS-IDIR
      CMPLXWA(IA+K)=CMPLX(FLOAT(LG),FLOAT(IS))
80    CONTINUE
      RETURN
      END
```

```
C
C
C
      SUBROUTINE RHSLD (V,YT,JRYT,ICYT,BTYP,BNR,BCV,DS,AI,N,OTPT,IWRITE)
C
C     THIS SUBROUTINE CALCULATES VECTOR DS OF PERTURBED LOAD FLOW
C     EQUATIONS  (SEE EQUATION (17) OF SOS-80-2)  AND BUS CURRENT
C     VECTOR AI
C
      INTEGER JRYT(1),ICYT(1),BTYP(1),BNR(1),OTPT
      COMPLEX V(1),YT(1),BCV(1),DS(1),AI(1),CCURR,S
C
      N1=N+N
      DO 20 I=1,N
      J1=I+I
      J=J1-1
      AI(I)=CCURR(V,YT,JRYT,ICYT,I)
      S=V(I)*CONJG(AI(I))
      IF (BTYP(BNR(I)).EQ.1) GO TO 10
      S=BCV(I)-S
      DS(J)=CONJG(S)
      DS(J1)=S
      GO TO 20
   10 S=CMPLX(REAL(S),CABS(V(I)))
      S=BCV(I)-S
      DS(J)=S
      DS(J1)=CONJG(S)
   20 CONTINUE
      IF (IWRITE.LT.3) RETURN
      WRITE (OTPT,30)
      WRITE (OTPT,40)  (BNR(I),DS(2*I-1),I=1,N)
      WRITE (OTPT,50)
      WRITE (OTPT,40)  (BNR(I),AI(I),I=1,N)
      RETURN
   30 FORMAT (//" RIGHT HAND SIDE VECTOR DS OF PERTURBED FLOW EQUATIONS"
     1/)
   40 FORMAT ((3(3X,I3,":",2(1X,E11.5))))
   50 FORMAT (///" VECTOR AI OF BUS CURRENTS"/)
      END
```

```
C
C
C
      SUBROUTINE FRPR (YT,JRYT,ICYT,BCV,V,BNR,BTYP,IBV,WS,LWS,NB,NTL,NBV
     1,INPT,OTPT,IFLAG,IWRITE)
C
C     SUBROUTINE FORMPR RENUMBERS BUSES OF THE POWER SYSTEM AND
C     FORMULATES THE LOAD FLOW PROBLEM. ON RETURN,THE LOAD
C     PROBLEM IS GIVEN BY VECTORS YT,JRYT,ICYT,BCV,V INDEXED
C     AS DEFINED BY VECTOR BNR
C
      INTEGER BNR(1),IBV(1),JRYT(1),ICYT(1),BTYP(1),OTPT
      REAL BCV(1),WS(1)
      COMPLEX YT(1),V(1)
C
      IFLAG=0
      JBINP=1
      JBOUT=JBINP+NTL
      JINPG=JBOUT+NTL
      JINPB=JINPG+NTL
      JLG=JINPB+NTL
      JLB=JLG+NTL
      JOUTG=JLB+NTL
      JOUTB=JOUTG+NTL
      JTAP=JOUTB+NTL
      JNR=JTAP+NTL
      JVM=JNR+NB
      JVA=JVM+NB
      JGP=JVA+NB
      JLP=JGP+NB
      JLQ=JLP+NB
      JSTL=JLQ+NB
      JMX=JSTL+NB
      IF (JMX.GE.LWS) GO TO 50
      CALL RDAT (WS(JBINP),WS(JBOUT),WS(JINPG),WS(JINPB),WS(JLG),WS(JLB)
     1,WS(JOUTG),WS(JOUTB),WS(JTAP),WS(JNR),BTYP,WS(JVM),WS(JVA),WS(JGP)
     2,WS(JLP),WS(JLQ),WS(JSTL),JRYT,NB,NTL,NLB,INPT,IWRITE)
C
C     BUSES ARE RENUMBERED TO GIVE THE HIGHEST NUMBERS TO PARTITION
C     BUSES
C
      DO 10 I=1,NB
      BNR(I)=I
      JRYT(I)=JRYT(I+1)-JRYT(I)
   10 CONTINUE
      K=NB-1
      DO 20 I=1,NBV
      L=IBV(I)
      BNR(L)=K
      BNR(K)=L
      JR=JRYT(K)
      JRYT(K)=JRYT(L)
      JRYT(L)=JR
      K=K-1
   20 CONTINUE
      L=1
      DO 30 I=1,NB
      M=JRYT(I)
      JRYT(I)=L
      L=M+L
   30 CONTINUE
C
C     JRYT(I) IS AN INDEX OF THE FIRST ELEMENT OF THE ITH ROW OF THE
C     NODAL ADMITTANCE MATRIX (I IS A BUS INDEX DEFINED BY VECTOR BNR,
C     I=1,2,     ,NB)
```

```
C
      CALL FORMYT (WS(JBINP),WS(JBOUT),BNR,IBV,WS(JINPG),WS(JINPB),WS(JL
     1G),WS(JLB),WS(JOUTG),WS(JOUTB),WS(JTAP),WS(JSTL),JRYT,ICYT,YT,NB,N
     2TL,NYT,OTPT,IWRITE)
      CALL FORMU (BTYP,BNR,WS(JVM),WS(JVA),WS(JGP),WS(JLP),WS(JLQ),BCV,N
     1B,OTPT,IWRITE)
      K1=JVM-1
      K2=JVA-1
      DO 40 I=1,NB
      R1=WS(K1+I)
      R2=WS(K2+I)
      V(BNR(I))=CMPLX(R1*COS(R2),R1*SIN(R2))
40    CONTINUE
      RETURN
50    IFLAG=-1
      RETURN
      END
```

```
C
C
C
      SUBROUTINE FORMU (BTYP,BNR,BVMOD,BVARG,BGP,BLP,BLQ,BCV,NB,OTPT,IWR
     1ITE)
C
C     SUBROUTINE FORMU FORMS THE RIGHT HAND SIDE VECTOR BCV OF THE
C     POWER FLOW EQUATIONS
C
      INTEGER BTYP(1),BNR(1),OTPT
      REAL BVMOD(1),BVARG(1),BGP(1),BLP(1),BLQ(1),BCV(1)
C
      DO 40 I=1,NB
      J=2*BNR(I)-1
      IF (BTYP(I)-1) 10,20,30
C
C     SETTING UP ELEMENTS OF BCV CORRESPONDING TO LOAD BUS
C
 10   BCV(J)=BGP(I)-BLP(I)
      J=J+1
      BCV(J)=-BLQ(I)
      GO TO 40
C
C     SETTING UP ELEMENTS OF BCV CORRESPONDING TO GENERATOR BUS
C
 20   BCV(J)=BGP(I)-BLP(I)
      J=J+1
      BCV(J)=BVMOD(I)
      GO TO 40
C
C     SETTING UP ELEMENTS OF BCV CORRESPONDING TO SLACK BUS
C
 30   BCV(J)=BVMOD(I)
      J=J+1
      BCV(J)=BVARG(I)
 40   CONTINUE
      IF (IWRITE.LT.3) GO TO 50
      WRITE (OTPT,60)
      WRITE (OTPT,70) (BNR(I),BCV(2*I-1),BCV(2*I),I=1,NB)
 50   RETURN
 60   FORMAT (//" RHS VECTOR BCV OF POWER FLOW EQUATIONS"/)
 70   FORMAT (3(3X,I3,":",2(1X,E13.7)))
      END
```

```
C
C
C
      SUBROUTINE FORMYT (LBINP,LBOUT,BNR,IBV,LINPG,LINPB,LG,LB,LOUTG,LOU
     1TB,LTAP,BSTL,JRYT,ICYT,YT,NB,NTL,NYT,OTPT,IWRITE)
C
C     SUBROUTINE  FORMYT FORMS THE NODAL ADMITTANCE MATRIX OF POWER
C     SYSTEM AND STORES IT IN A SPARSE FORM BY VECTORS YT,JRYT,ICYT,
C     USING BUS INDICES DEFINED BY VECTOR BNR
C
      INTEGER LBINP(1),LBOUT(1),JRYT(1),ICYT(1),BNR(1),IBV(1),OTPT
      REAL LINPG(1),LINPB(1),LG(1),LB(1),LOUTG(1),LOUTB(1),LTAP(1),BSTL(
     11)
      COMPLEX YT(1),Y
C
      NYT=NB+2*NTL
C
C     ON ENTRY JRYT MUST BE INDEXED AS DEFINED BY VECTOR BNR
C
      DO 10 I=1,NB
      J=JRYT(I)
      YT(J)=CMPLX(0.,BSTL(BNR(I)))
      ICYT(NYT+I)=J
10    CONTINUE
      DO 20 I=1,NTL
      IB1=BNR(LBINP(I))
      IB2=BNR(LBOUT(I))
      Y=CMPLX(LG(I),LB(I))
      L1=JRYT(IB1)
      L2=JRYT(IB2)
      YT(L1)=YT(L1)+CMPLX(LINPG(I),LINPB(I))+Y/(LTAP(I)**2)
      YT(L2)=YT(L2)+CMPLX(LOUTG(I),LOUTB(I))+Y
      ICYT(L1)=IB1
      ICYT(L2)=IB2
      K1=NYT+IB1
      K2=NYT+IB2
      ICYT(K1)=ICYT(K1)+1
      ICYT(K2)=ICYT(K2)+1
      L1=ICYT(K1)
      L2=ICYT(K2)
      Y=-Y/LTAP(I)
      YT(L1)=Y
      YT(L2)=Y
      ICYT(L1)=IB2
      ICYT(L2)=IB1
20    CONTINUE
      IF (IWRITE.LT.3) GO TO 40
      WRITE (OTPT,50)
      DO 30 I=1,NB
      K1=JRYT(I)
      K2=JRYT(I+1)-1
      WRITE (OTPT,60) BNR(ICYT(K1))
      WRITE (OTPT,70) (BNR(ICYT(J)),YT(J),J=K1,K2)
30    CONTINUE
40    RETURN
50    FORMAT (//" BUS ADMITTANCE MATRIX YT"/)
60    FORMAT (/" BUS NO.",I3)
70    FORMAT (3(3X,I3,":",2(1X,E13.7)))
      END
```

```
C
C
C
      SUBROUTINE RDAT (LBINP,LBOUT,LINPG,LINPB,LG,LB,LOUTG,LOUTB,LTAP,BN
     1R,BTYP,BVMOD,BVARG,BGP,BLP,BLQ,BSTL,JRYT,NB,NTL,NLB,INPT,IWRITE)
C
C     SUBROUTINE RDAT READS INPUT DATA DESCRIBING POWER SYSTEM
C
      INTEGER LBINP(1),LBOUT(1),BNR(1),BTYP(1),JRYT(1)
      REAL LINPG(1),LINPB(1),LG(1),LB(1),LOUTG(1),LOUTB(1),LTAP(1),BVMOD
     1(1),BVARG(1),BGP(1),BLP(1),BLQ(1),BSTL(1),HDLN(12)
      COMPLEX Z
C
      READ (INPT,110) (HDLN(I),I=1,12)
      READ (INPT,130) X,NB,X,X,NTL
      DO 10 I=1,NB
      JRYT(I)=1
10    CONTINUE
      IF (IWRITE.LT.1) GO TO 20
      WRITE (6,110) (HDLN(I),I=1,12)
      WRITE (6,150) NB,NTL
20    READ (INPT,110) (HDLN(I),I=1,12)
      IF (IWRITE.LT.2) GO TO 30
      WRITE (6,110) (HDLN(I),I=1,12)
30    READ (INPT,120) (HDLN(I),I=1,12)
      DO 40 I=1,NTL
      READ (INPT,140) LBINP(I),LBOUT(I),LINPG(I),LINPB(I),LG(I),LB(I),LO
     1UTG(I),LOUTB(I),LTAP(I)
C
C     DEGREES OF BOTH BUSES INCIDENT TO THE LINE ARE INCREASED
C     BY ONE
C
      J=LBINP(I)
      JRYT(J)=JRYT(J)+1
      J=LBOUT(I)
      JRYT(J)=JRYT(J)+1
40    CONTINUE
      IF (IWRITE.LT.2) GO TO 60
      WRITE (6,120) (HDLN(I),I=1,12)
      DO 50 I=1,NTL
      WRITE (6,140) LBINP(I),LBOUT(I),LINPG(I),LINPB(I),LG(I),LB(I),LOUT
     1G(I),LOUTB(I),LTAP(I)
50    CONTINUE
60    DO 70 I=1,NTL
      Z=1./CMPLX(LG(I),LB(I))
      LG(I)=REAL(Z)
      LB(I)=AIMAG(Z)
70    CONTINUE
      READ (INPT,110) (HDLN(I),I=1,12)
      IF (IWRITE.GT.1) WRITE (6,110) (HDLN(I),I=1,12)
      READ (INPT,120) (HDLN(I),I=1,12)
      NLB=0
      IX=JRYT(1)
      JRYT(1)=1
      JRYT(NB+1)=0
      DO 80 I=1,NB
      READ (INPT,140) L,BTYP(L),BVMOD(L),BVARG(L),BGP(L),BLP(L),BLQ(L),B
     1STL(L)
      BNR(L)=I
      IF (BTYP(L).EQ.0) NLB=NLB+1
      J=I+1
      IY=JRYT(J)
      JRYT(J)=JRYT(I)+IX
      IX=IY
80    CONTINUE
```

```
      IF (IWRITE.LT.2) GO TO 100
      WRITE (6,120) (HDLN(I),I=1,12)
      DO 90 I=1,NB
      WRITE (6,140) BNR(I),BTYP(I),BVMOD(I),BVARG(I),BGP(I),BLP(I),BLQ(I
     1),BSTL(I)
90    CONTINUE
100   RETURN
110   FORMAT (//12A10/)
120   FORMAT (12A10)
130   FORMAT (1X,A5,I3,2A5,I4)
140   FORMAT (2(1X,I5),2X,7(2X,E13.7))
150   FORMAT (" NB =",I3.3,",      ","NTL =",I4.3)
      END
```