LFLNR - A FORTRAN IMPLEMENTATION OF THE NEWTON-RAPHSON LOAD FLOW TECHNIQUE

J.W. Bandler, M.A. El-Kady, W. Kellermann
and W.M. Zuberek

SOS-83-7-L

July 1983

LFLNR - A FORTRAN IMPLEMENTATION OF THE NEWTON-RAPHSON LOAD FLOW TECHNIQUE

J.W. Bandler, M.A. El-Kady, W. Kellermann and W.M. Zuberek

Abstract

LFLNR is a package of subroutines for solving load flow problems by the well known Newton-Raphson method. The method has been proposed by Van Ness and Griffin, and subsequently improved and practically applied in polar form by Tinney. The implemented version is based on a rectangular formulation of the problem and it uses standard sparse matrix techniques to represent the power system's bus admittance matrix as well as the Jacobian matrix required by the method. The Harwell Package MA28 is called to solve the associated systems of linear equations with real coefficients. The package and documentation have been developed for use on the CDC 170/730 system with the NOS 1.4 level 552 operating system and the Fortran Extended (FTN) version 4.8 compiler. This document contains a Fortran listing of the package.

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.

W.M. Zuberek is on leave from the Institute of Computer Science, Technical University of Warsaw, Warsaw, Poland. He is now with the Department of Electrical Engineering, Texas A & M University, College Station, TX 77843.

I. INTRODUCTION

The Newton-Raphson method for solving load flow problems [1] has been implemented as a package of Fortran IV subroutines for the CDC 170/730 system. The LFLNR package is composed of 2 subroutines. LFLNR1 is the main subroutine of the package and its purpose is to organize workspace provided by the user into a set of vectors used by this subroutine and LFLNR2. It checks formal correctness of some parameters defined by the user and sets the return flag appropriately. It also controls the mode of iteration performed by LFLNR2 and checks the bounds on the number of iterations and on the iteration time, if required by the user.

LFLNR2 performs one iteration of the Newton-Raphson method. It creates and factorizes the sparse matrix of real coefficients (the Jacobian matrix), calculates the right-hand side vector and solves a sparse system of linear equations (using the Harwell package MA28 [2]). It also determines the accuracy of the load flow solution and updates the voltages.

The listing contains 231 lines, 66 of which are comments.

II. REFERENCES

[1] J.W. Bandler, M.A. El-Kady, W. Kellermann and W.M. Zuberek, "LFLNR - a Fortran implementation of the Newton-Raphson load flow technique", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-83-7-U, July 1983.

[2] I.S. Duff, "MA28 - A set of Fortran subroutines for sparse unsymmetric linear equations", Computer Science and Systems Division, AERE, Harwell, Oxfordshire, England, Report R.8730, November 1980.

III. LISTING OF THE LFLNR PACKAGE

Subroutine	Number of Lines (source text)	Number of Words (compiled code)	Listing from Page
LFLNR1	117	345	3
LFLNR2	114	575	4

```
      SUBROUTINE LFLNR1(NB,NZ,INDR,INDC,IBT,Y,YS,V,S,W,LW,ITEL,VEPS,      000001
     1                  TIMEL,MODE,IFLAG)                                 000002
      DIMENSION INDR(1),INDC(1),IBT(1),W(1)                              000003
      COMPLEX Y(1),YS(1),V(1),S(1)                                       000004
C                                                                        000005
C  THIS PACKAGE SOLVES THE LOAD FLOW PROBLEM USING SPARSE MATRIX         000006
C  TECHNIQUES (HARWELL PACKAGE MA28) AND THE NEWTON-RAPHSON METHOD.      000007
C                                                                        000008
C  LIBRARY : HARWELL PACKAGE MA28                                        000009
C                                                                        000010
C     NB    - NUMBER OF BUSES (EXCLUDING SLACK BUS),                   000011
C     NL    - NUMBER OF LOAD BUSES,                                    000012
C     NZ    - NUMBER OF NON-ZEROS IN BUS ADMITTANCE MATRIX,            000013
C     INDR  - ROW INDEX OF SPARSE BUS ADMITTANCE MATRIX,               000014
C     INDC  - COLUMN INDEX OF SPARSE BUS ADMITTANCE MARIX,             000015
C     IBT   - VECTOR OF BUS TYPES (0 - LOAD BUS, 1 - GENERATOR BUS),   000016
C     Y     - SPARSE BUS ADMITTANCE MATRIX,                            000017
C     YS    - VECTOR OF SLACK BUS ADMITTANCES,                         000018
C     V     - COMPLEX BUS VOLTAGES (RECTANGULAR MODE),                 000019
C     S     - COMPLEX BUS INJECTED POWERS,                             000020
C     W     - REAL WORKSPACE,                                          000021
C     LW    - LENGTH OF THE WORKSPACE W,                               000022
C     ITEL  - LIMIT OF ITERATIONS,                                     000023
C     VEPS  - REQUIRED ACCURACY OF SOLUTION,                           000024
C     TIMEL - LIMIT OF ITERATION TIME,                                 000025
C     MODE  - REQUIRED MODE OF ITERATIVE PROCEDURE :                   000026
C             0 - FORM AND DECOMPOSE THE JACOBIAN MATRIX USING         000027
C                 THE BEST PIVOTAL STRATEGY (MA28A IS CALLED),         000028
C             1 - FORM AND FACTORIZE THE JACOBIAN MATRIX (WITH         000029
C                 THE SAME SPARSITY PATTERN) USING THE PIVOTAL         000030
C                 SEQUENCE DETERMINED BY THE EARLIER ENTRY WITH        000031
C                 MODE=0 (MA28B IS CALLED),                            000032
C     IFLAG - RETURN FLAG :                                            000033
C            -3 - ERROR DIAGNOSTICS FROM HARWELL PACKAGE MA28,         000034
C            -2 - INCORRECT USE,                                       000035
C            -1 - INCORRECT PARAMETERS (E.G.,INSUFFICIENT WORKSPACE),  000036
C             0 - NORMAL RETURN (REQUIRED ACCURACY OBTAINED),          000037
C             1 - LIMIT OF ITERATIONS REACHED,                         000038
C             2 - LIMIT OF ITERATION TIME REACHED.                     000039
C                                                                      000040
      DATA MB,MZ,LR,IFL/0,0,0,-1/                                      000041
      CORR=0.0                                                         000042
      IT=0                                                             000043
      CALL SECOND(TIME1)                                               000044
      IF(MODE.LT.0.OR.MODE.GT.1)GO TO 90                               000045
      IF(MODE.EQ.0)GO TO 10                                            000046
      IFLAG=-2                                                         000047
      IF(NB.NE.MB.OR.NZ.NE.MZ.OR.LW.NE.LR.OR.IFL.LT.0)GO TO 95         000048
      GO TO 50                                                         000049
   10 MB=NB                                                            000050
      MZ=NZ                                                            000051
      LR=LW                                                            000052
      NZR=0                                                            000053
C                                                                      000054
C       DETERMINATION OF THE NUMBER OF NON-ZEROS                       000055
C                                                                      000056
      DO 20 I=1,NB                                                     000057
      K=1                                                              000058
      L=0                                                              000059
      IF(I.GT.1) L=INDR(I-1)                                           000060
      IF(IBT(I).EQ.0) K=INDR(I)-L                                      000061
      NZR=NZR+K                                                        000062
   20 CONTINUE                                                         000063
C                                                                      000064
C       DISTRIBUTION OF WORKSPACE                                      000065
```

```
C                                                                       000066
      KZ=NZ+NZ+NZR+NZR                                                  000067
      KN=NB+NB                                                          000068
      LICN=3*KZ                                                         000069
      LIRN=KZ+KN+KN                                                     000070
      JA=1                                                              000071
      JICN=JA+LICN                                                      000072
      JIKEEP=JICN+LICN                                                  000073
      JIW=JIKEEP+5*KN                                                   000074
      JW=JIW+8*KN                                                       000075
      JRHS=JW+KN                                                        000076
      JIRN=JRHS+KN                                                      000077
      MAX1=JIRN+LIRN                                                    000078
      JIVECT=JIRN                                                       000079
      JJVECT=JIVECT+KZ                                                  000080
      MAX2=JJVECT+KZ                                                    000081
      MAX=MAX0(MAX1,MAX2)                                               000082
      IF(MAX.GT.LW)GO TO 90                                             000083
   50 IF(ITEL.EQ.0)GO TO 80                                             000084
      IFLAG=-3                                                          000085
      IMODE=MODE                                                        000086
C                                                                       000087
C       ITERATION LOOP                                                  000088
C                                                                       000089
   60 IT=IT+1                                                           000090
      CALL LFLNR2 (NB,NZ,NZR,INDR,INDC,IBT,Y,YS,V,S,LICN,LIRN,          000091
     1             W(JA),W(JICN),W(JIKEEP),W(JIW),W(JW),W(JRHS),      000092
     2             W(JIRN),W(JIVECT),W(JJVECT),IMODE,ITEL,VEPS,       000093
     3             CORR,IFLG)                                         000094
      IF(IFLG.LT.0)GO TO 95                                             000095
      IMODE=1                                                           000096
      IF(CORR.LE.VEPS)GO TO 85                                          000097
      IF(ITEL.LE.0)GO TO 70                                             000098
      IF(IT.GE.ITEL)GO TO 80                                            000099
   70 CALL SECOND(TIME2)                                                000100
      IF(TIMEL.LE.0.0)GO TO 60                                          000101
      IF(TIMEL.GT.TIME2-TIME1)GO TO 60                                  000102
      IFLAG=2                                                           000103
      GO TO 95                                                          000104
   80 IFLAG=1                                                           000105
      GO TO 95                                                          000106
   85 IFLAG=0                                                           000107
      GO TO 95                                                          000108
   90 IFLAG=-1                                                          000109
   95 ITEL=IT                                                           000110
      VEPS=CORR                                                         000111
      CALL SECOND(TIME2)                                                000112
      TIMEL=TIME2-TIME1                                                 000113
      IF(MODE.EQ.0) IFL=IFLAG                                           000114
      RETURN                                                            000115
      END                                                               000116
C                                                                       000117
C                                                                       000118
      SUBROUTINE LFLNR2 (NB,NZ,NZR,INDR,INDC,IBT,Y,YS,V,S,LICN,LIRN,A,  000119
     1                   ICN,IKEEP,IW,W,RHS,IRN,IVECT,JVECT,MODE,       000120
     2                   ITEL,VEPS,CORR,IFLG)                           000121
      DIMENSION INDR(1),INDC(1),IBT(1),A(1),ICN(1),IKEEP(1),IW(1),W(1), 000122
     1          RHS(1),IRN(1),IVECT(1),JVECT(1)                         000123
      COMPLEX Y(1),YS(1),V(1),S(1)                                      000124
C                                                                       000125
C  THIS SUBROUTINE PERFORMS ONE ITERATION OF THE NEWTON-RAPHSON         000126
C  METHOD.                                                              000127
C                                                                       000128
      COMPLEX CURR,VOLT,CJVOLT,XS,XSB,YY,DELS,VV                        000129
C                                                                       000130
```

```
C       FORM SPARSE MATRIX OF COEFFICIENTS AND RHS                      000131
C                                                                       000132
      NBS=NB+1                                                          000133
      NZG=NZ+NZ                                                         000134
      DO 50 K=1,NB                                                      000135
      IBTP=IBT(K)                                                       000136
C                                                                       000137
C       SET COEFFICIENTS                                                000138
C                                                                       000139
      K1=1                                                              000140
      IF(K.GT.1)K1=INDR(K-1)+1                                          000141
      K2=INDR(K)                                                        000142
      CURR=YS(K)*V(NBS)                                                 000143
      DO 10 I=K1,K2                                                     000144
      J=INDC(I)                                                         000145
   10 CURR=CURR+Y(I)*V(J)                                               000146
      VOLT=V(K)                                                         000147
      CJVOLT=CONJG(VOLT)                                                000148
      VMOD=CABS(VOLT)                                                   000149
      DO 40 I=K1,K2                                                     000150
      YY=Y(I)                                                           000151
      XS=CJVOLT*YY                                                      000152
      J=INDC(I)                                                         000153
      IF(IBTP.EQ.1)GO TO 20                                             000154
      XSB=(0.0,0.0)                                                     000155
      IF(I.EQ.K2) XSB=CURR                                              000156
      GO TO 30                                                          000157
   20 XSB=CONJG(XS)                                                     000158
      IF(I.NE.K2)GO TO 25                                               000159
      VOLT1=REAL(VOLT)/VMOD                                             000160
      VOLT2=AIMAG(VOLT)/VMOD                                            000161
      XS=XS+CONJG(CURR)-CMPLX(VOLT2,VOLT1)                              000162
      XSB=XSB+CURR-CMPLX(-VOLT2,VOLT1)                                  000163
   25 XS=0.5*XS                                                         000164
      XSB=0.5*XSB                                                       000165
   30 XS1=REAL(XS)                                                      000166
      XS2=AIMAG(XS)                                                     000167
      XSB1=REAL(XSB)                                                    000168
      XSB2=AIMAG(XSB)                                                   000169
      L=I                                                               000170
      A(L)=XS1+XSB1                                                     000171
      IVECT(L)=K                                                        000172
      JVECT(L)=J                                                        000173
      L=I+NZ                                                            000174
      A(L)=-XS2+XSB2                                                    000175
      IVECT(L)=K                                                        000176
      JVECT(L)=J+NB                                                     000177
      IF(IBTP.NE.0.AND.I.NE.K2) GO TO 40                                000178
      NZG=NZG+1                                                         000179
      L=NZG                                                             000180
      A(L)=-(XS2+XSB2)                                                  000181
      IVECT(L)=K+NB                                                     000182
      JVECT(L)=J                                                        000183
      L=NZG+NZR                                                         000184
      A(L)=-XS1+XSB1                                                    000185
      IVECT(L)=K+NB                                                     000186
      JVECT(L)=J+NB                                                     000187
   40 CONTINUE                                                          000188
C                                                                       000189
C       SET RHS                                                         000190
C                                                                       000191
      DELS=VOLT*CONJG(CURR)                                             000192
      IF(IBTP.EQ.1)GO TO 45                                             000193
      DELS=S(K)-DELS                                                    000194
      DELS1=REAL(DELS)                                                  000195
```

```
      DELS2=AIMAG(DELS)                                                 000196
      GO TO 49                                                          000197
   45 DELS1=REAL(S(K))-REAL(DELS)                                       000198
      DELS2=AIMAG(S(K))-VMOD                                            000199
   49 RHS(K)=DELS1                                                      000200
      KK=K+NB                                                           000201
      RHS(KK)=DELS2                                                     000202
   50 CONTINUE                                                          000203
C                                                                       000204
C       SOLVE SPARSE SYSTEM OF LINEAR EQUATIONS                         000205
C                                                                       000206
      NS=NB+NB                                                          000207
      NZS=NZ+NZ+NZR+NZR                                                 000208
      IF(MODE.NE.0)GO TO 15                                             000209
      DO 13 I=1,NZS                                                     000210
   13 ICN(I)=JVECT(I)                                                   000211
      U=0.1                                                             000212
      CALL MA28A(NS,NZS,A,LICN,IRN,LIRN,ICN,U,IKEEP,IW,W,IFLG)          000213
      GO TO 26                                                          000214
   15 CALL MA28B(NS,NZS,A,LICN,IVECT,JVECT,ICN,IKEEP,IW,W,IFLG)         000215
   26 IF(IFLG.LT.0) RETURN                                              000216
      CALL MA28C(NS,A,LICN,ICN,IKEEP,RHS,W,1)                           000217
C                                                                       000218
C       UPDATE VOLTAGES                                                 000219
C                                                                       000220
      CORR=0.0                                                          000221
      DO 60 I=1,NB                                                      000222
      J=I+NB                                                            000223
      VV=CMPLX(RHS(I),RHS(J))                                           000224
      VM=CABS(VV)                                                       000225
      IF(VM.GT.CORR) CORR=VM                                            000226
      V(I)=V(I)+VV                                                      000227
   60 CONTINUE                                                          000228
      IFLG=0                                                            000229
      RETURN                                                            000230
      END                                                               000231
```