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-U

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 is a user's manual for 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

In 1961, Van Ness and Griffin [1] described the elimination method of power flow solution, which is the n-dimensional analog of the Newton-Raphson method [2]. Solving relatively small problems, they showed that this method overcomes deficiencies in the successive displacements methods and has other favourable characteristics. Shortly afterwards, the method was tested by Tinney and others who found that the associated computer time and memory requirements increase rapidly with problem size, and for networks of more than 200 nodes, the accelerated displacements methods are more effective. It was, however, evident that the main difficulty was not in the method itself, but in the elimination procedure for solving the simultaneous equations. The development of very efficient sparsity-oriented ordered elimination by Tinney and others [3-5] dramatically improved the computing speed and storage requirements of the Newton-Raphson method, and the method has been widely regarded [6-7] as the pre-eminent general-purpose load-flow approach. In recent years, however, the stimulus of increasing problem sizes, on-line applications and system optimization resulted in newer methods that are more attractive and that gradually replaced the Newton-Raphson method in many applications. However, these newer methods are essentially approximate versions of the exact Newton-Raphson method, where decoupled forms of the Jacobian matrix are employed. Further to the load flow solution, the subsequent use of approximate methods to generate power flow sensitivities leads to erroneous sensitivities. In this respect, an efficient computer implementation of the exact Newton-Raphson method may be advantageous.

The Newton-Raphson method has been implemented as a package of Fortran IV subroutines for the CDC 170/730 system. At McMaster University, it is available in the form of a library of binary relocatable subroutines which is linked with the user's program by the appropriate call of the main subroutine LFLNR1. The library is available in the group indirect file LIBSPWR under the charge RJWBAND. The package calls the subroutines MA28A, MA28B and MA28C from the Harwell Subroutine Library (Harwell Package MA28

[8]); the package MA28 must thus be also available when LFLNR is used. The general sequence of NOS commands to use the LFLNR package may be as follows:

/GET(LIBSPWR, LIBRHSM)	- fetch the libraries,
/LIBRARY(LIBSPWR, LIBRHSM)	- indicate the libraries to the loader,
/FTN(..., GO)	- compile, load and execute the program.

A Fortran listing of the LFLNR package is contained in [9].

II. GENERAL DESCRIPTION

The branches of a power network represent transmission lines, transformers, etc., which are modeled as linear time-invariant RLC elements. The nodes of the network other than the ground node are called buses. They correspond to generation stations and load-center substations. For steady-state analysis, the network is considered as in sinusoidal steady state.

Consider a power network with n buses. There are three types of buses:

i) PQ bus: a bus where the injected real and reactive power are specified;

ii) PV bus: a bus where the injected real power and the voltage magnitude are specified;

iii) Slack bus: a bus whose voltage magnitude and phase angle are specified.

Normally, PQ buses are load buses and PV buses and the slack bus are generator buses. We let subscripts $\ell = 1, 2, \ldots, n_L$, correspond to PQ buses, subscripts $g = n_L + 1, \ldots, n_L + n_G$, correspond to PV buses, and subscript $n = n_L + n_G + 1$ corresponds to the slack bus.

The well-known [7] power network performance equations in the admittance form can be written as

$$I_i = \sum_{k=1}^{n} Y_{ik} V_k, \quad i = 1, 2, \ldots, n-1, \tag{1}$$

where I_i are components of $\mathbf{I}$, which is the complex vector of bus currents, V_k are components of $\mathbf{V}$, which is the complex vector of bus voltages and Y_{ik} are elements of $\mathbf{Y}$ representing the complex bus admittance matrix of the network.

For load buses, the bus loading equations take the form

$$S_\ell = V_\ell I_\ell^*, \ \ell = 1, 2, \ldots, n_L, \tag{2}$$

where S_ℓ is the complex bus power, $S_\ell = P_\ell + jQ_\ell$, and * denotes the complex conjugate. For generator buses, it is convenient [10] to introduce the complex quantities

$$\bar{S}_g \triangleq P_g + j|V_g|, \quad g = n_L + 1, \ldots, n_L + n_G \tag{3}$$

and to derive complex equations similar to (2), i.e., to relate S_g with bus voltages $\mathbf{V}$ and bus currents $\mathbf{I}$. Observe that for all buses

$$V_i I_i^* = P_i + jQ_i, \quad i = 1, 2, \ldots, n-1,$$

and

$$V_i^* I_i = P_i - jQ_i, \quad i = 1, 2, \ldots, n-1,$$

so

$$2P_i = V_i I_i^* + V_i^* I_i, \ i = 1, 2, \ldots, n-1, \tag{4}$$

and substituting (4) into (3)

$$\bar{S}_g = (V_g I_g^* + V_g^* I_g)/2 + j|V_g|, \quad g = n_L + 1, \ldots, n_L + n_G. \tag{5}$$

Now, substituting (1) into (2) and (5) results in

$$S_\ell = V_\ell \sum_{k=1}^{n} Y_{\ell k}^* V_k^*, \quad \ell = 1, 2, \ldots, n_L, \tag{6}$$

and

$$\bar{S}_g = \left(V_g \sum_{k=1}^{n} Y_{gk}^* V_k^* + V_g^* \sum_{k=1}^{n} Y_{gk} V_k \right)/2 + j|V_g|, \quad g = n_L + 1, \ldots, n_L + n_G, \tag{7}$$

respectively.

The system of nonlinear equations (6) and (7) represents the typical load flow problem. In its perturbed form it can be written as

$$\Delta S_\ell = \Delta V_\ell \sum_{k=1}^{n} Y^*_{\ell k} V^*_k + V_\ell \sum_{k=1}^{n} Y^*_{\ell k} \Delta V^*_k, \quad \ell = 1, \ldots, n_L, \tag{8}$$

and

$$\Delta \bar{S}_g = \Big(\Delta V_g \sum_{k=1}^{n} Y^*_{gk} V^*_k + V_g \sum_{k=1}^{n} Y^*_{gk} \Delta V^*_k + \Delta V^*_g \sum_{k=1}^{n} Y_{gk} V_k$$

$$+ V^*_g \sum_{k=1}^{n} Y_{gk} \Delta V_k + j(\Delta V^*_g V_g + V^*_g \Delta V_g)/|V_g| \Big)/2, \; g = n_L + 1, \ldots, n_L + n_G, \tag{9}$$

respectively, where

$$|V_g| = (V_g V^*_g)^{1/2}.$$

We can express (8) and (9) in a more compact form by

$$\Delta S_\ell = \sum_{k=1}^{n} (A_{\ell k} \Delta V_k + B_{\ell k} \Delta V^*_k), \quad \ell = 1, 2, \ldots, n_L, \tag{10}$$

and

$$\Delta \bar{S}_g = \sum_{k=1}^{n} (A_{gk} \Delta V_k + B_{gk} \Delta V^*_k), \quad g = n_L + 1, \ldots, n_L + n_G, \tag{11}$$

respectively, where

$$A_{\ell k} = \begin{cases} 0, & \text{if } \ell = 1, 2, \ldots, n_L \text{ and } k \neq \ell, \\ \sum_{i=1}^{n} Y^*_{\ell i} V^*_i, & \text{if } \ell = 1, 2, \ldots, n_L \text{ and } k = \ell, \end{cases} \tag{12a}$$

$$A_{gk} = \begin{cases} V^*_g Y_{gk}/2, & \text{if } g = n_L + 1, \ldots, n_L + n_G \text{ and } k \neq g, \\ \Big(\sum_{i=1}^{n} Y^*_{gi} V^*_i + V^*_g Y_{gg} + jV^*_g/|V_g| \Big)/2, & \text{if } g = n_L + 1, \ldots, n_L + n_G \text{ and } k = g, \end{cases} \tag{12b}$$

and

$$B_{\ell k} = V_\ell Y^*_{\ell k}, \; \text{if } i = 1, 2, \ldots, n_L, \tag{13a}$$

$$B_{gk} = \begin{cases} V_g Y^*_{gk}/2, & \text{if } g = n_L+1, \ldots, n_L+n_G \text{ and } k \neq g, \\ \Big(\sum_{i=1}^{n} Y_{gi} V_i + V_g Y^*_{gg} + jV_g/|V_g| \Big)/2, & \text{if } g = n_L+1, \ldots, n_L+n_G \text{ and } k = g. \end{cases} \tag{13b}$$

The real form of the Newton-Raphson iterative method can be obtained by separating (10) and (11) into real and imaginary parts. Using the notaion

$$\Delta S_\ell \triangleq \Delta P_\ell + j\,\Delta Q_\ell, \qquad \ell = 1, 2, \ldots, n_L,$$

$$\Delta \bar{S}_g \triangleq \Delta P_g + j\,\Delta|V_g|, \quad g = n_L+1, \ldots, n_L+n_G,$$

$$\Delta V_i \triangleq \Delta V_i^R + j\Delta V_i^I, \quad i = 1, \ldots, n-1,$$

$$\Delta V_i^* \triangleq \Delta V_i^R - j\Delta V_i^I, \qquad i = 1, \ldots, n-1,$$

$$A_{ik} \triangleq A_{ik}^R + jA_{ik}^I, \qquad i,k = 1, \ldots, n-1,$$

$$B_{ik} \triangleq B_{ik}^R + jB_{ik}^I, \qquad i,k = 1, \ldots, n-1,$$

the equations (10) and (11) can be expressed as

$$\Delta P_\ell + j\Delta Q_\ell = \sum_{k=1}^{n} \Big((A_{\ell k}^R + B_{\ell k}^R)\,\Delta V_k^R - (A_{\ell k}^I - B_{\ell k}^I)\,\Delta V_k^I$$

$$+ j((A_{\ell k}^I + B_{\ell k}^I)\,\Delta V_k^R + (A_{\ell k}^R - B_{\ell k}^R)\,\Delta V_k^I)\Big), \quad \ell = 1, 2, \ldots, n_L, \tag{14}$$

$$\Delta P_g + j\Delta|V_g| = \sum_{k=1}^{n}\Big((A^R_{gk} + B^R_{gk})\,\Delta V^R_k - (A^I_{gk} - B^I_{gk})\,\Delta V^I_k$$

$$+\, j\,((A^I_{gk} + B^I_{gk})\,\Delta V^R_k + (A^R_{gk} - B^R_{gk})\,\Delta V^I_k)\Big), \quad g = n_L+1, \ldots, n_L+n_G, \tag{15}$$

and, finally,

$$\Delta P_i = \sum_{k=1}^{n}\Big((A^R_{ik} + B^R_{ik})\,\Delta V^R_k - (A^I_{ik} - B^I_{ik})\,\Delta V^I_k\Big),\ i = 1, \ldots, n_L+n_G, \tag{16}$$

$$\Delta Q_\ell = \sum_{k=1}^{n}\Big((A^I_{\ell k} + B^I_{\ell k})\,\Delta V^R_k + (A^R_{\ell k} - B^R_{\ell k})\,\Delta V^I_k\Big),\ \ell = 1, 2, \ldots, n_L, \tag{17}$$

$$\Delta|V_g| = \sum_{k=1}^{n}\Big((A^I_{gk} + B^I_{gk})\,\Delta V^R_k + (A^R_{gk} - B^R_{gk})\,\Delta V^I_k\Big), \quad g = n_L+1, \ldots, n_L+n_G. \tag{18}$$

Moreover, taking into account (12b) and (13b), it can be observed that for all $g = n_L+1, \ldots, n_L+n_G$ and $k \neq g$,

$$A^R_{gk} = (V^R_g\, Y^R_{gk} + V^I_g\, Y^I_{gk})/2,$$

$$A^I_{gk} = (V^R_g\, Y^I_{gk} - V^I_g\, Y^R_{gk})/2,$$

$$B^R_{gk} = (V^R_g\, Y^R_{gk} + V^I_g\, Y^I_{gk})/2,$$

$$B^I_{gk} = -\,(V^R_g\, Y^I_{gk} - V^I_g\, Y^R_{gk})/2,$$

where $Y_{gk} = Y^R{}_{gk} + jY^I{}_{gk}$ and, therefore, (18) can be simplified to

$$\Delta|V_g| = (A^I_{gg} + B^I_{gg})\,\Delta V^R_g + (A^R_{gg} - B^R_{gg})\,\Delta V^I_g, \ g = n_L+1, \ldots, n_L+n_G. \tag{19}$$

The system of equations (16), (17) and (19) is solved iteratively in the following way. The initial approximation of the $\mathbf{V}^0$ (usually, $\mathbf{V}^0$ is the flat voltage profile) is used to determine the right-hand sides

$$\Delta P_i^1 = P_i - Re(V_i^0 \sum_{k=1}^{n} Y_{ik}^* V_k^{0*}), \quad i = 1, ..., n_L + n_G,$$

$$\Delta Q_\ell^1 = Q_\ell - Im(V_\ell^0 \sum_{k=1}^{n} Y_{\ell k}^* V_k^{0*}), \quad \ell = 1, 2, ..., n_L,$$

$$\Delta|V_g^1| = |V_g| - |V_g^0|, \quad g = n_L + 1, ..., n_L + n_G,$$

as well as the coefficients A_{ik} and B_{ik}, $i,k = 1, 2, ..., n_L + n_G$. The solution of the system of linear equations (16), (17) and (19) provides the corrections $\Delta V^R{}_i$ and $\Delta V^I{}_i$, $i = 1, 2, ..., n\text{-}1$, so

$$V_i^1 = V_i^0 + (\Delta V_i^R + j\Delta V_i^I), \quad i = 1,2, ..., n_L + n_G$$

and the calculation of the coefficients and right-hand sides is repeated for the voltages $\mathbf{V}^1$. This determines new voltage corrections, new voltages $\mathbf{V}^2$, and so on, until the required accuracy is obtained.

III. STRUCTURE OF THE PACKAGE

The 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). It also determines the accuracy of the load flow solution and updates the voltages.

IV. LIST OF ARGUMENTS

There is one general entry to the package:

CALL LFLNR1 (NB, NZ, INDR, INDC, IBT, Y, YS, V, S, W, LW, ITEL, VEPS, TIMEL, MODE, IFLAG)

and the arguments are as follows:

NB is an INTEGER argument that must be set to the number of buses (excluding the slack bus); it must be positive and is not changed by the package.

NZ is an INTEGER argument that must be set to the number of non-zero elements of the sparse bus admittance matrix including the diagonal elements; it must be positive and is not changed by the package.

INDR is an INTEGER vector of length NB that must contain the row index of the sparse bus admittance matrix; the consecutive elements of INDR are equal to the cumulative number of non-zero row elements of the bus admittance matrix up to and including the rows corresponding to the elements, as shown in the example.

INDC is an INTEGER vector of length NZ that must contain the column index of the sparse bus admittance matrix; the consecutive elements of INDC are equal to the column indices of the non-zero elements of the bus admittance matrix, however, in each row, the diagonal element is represented as the last non-zero element of the row, as shown in the example.

IBT is the INTEGER vector of length at least NB that must describe the types of buses; its elements must be equal to 0 for load (or PQ) buses, and equal to 1 for generator (or PV) buses.

Y is a COMPLEX vector of length at least NZ that must contain the sparse bus admittance matrix corresponding to the indices INDR and INDC.

YS is a COMPLEX vector of length at least (NB + 1) that must contain a row of the bus admittance matrix (including zero elements) corresponding to the slack bus.

V is a COMPLEX vector of length at least (NB + 1) that on entry must be set to the initial approximation of bus voltages (in rectangular mode). On exit, V contains the best solution found by the package (in rectangular mode, as well).

S is a COMPLEX vector of length at least NB that must contain the injected bus powers (active and reactive).

W is a REAL vector that is used as workspace by the package; its length is given by LW.

LW is the length of the workspace W; the package requires it to be at most

$$1 + 30*NB + 32*NZ.$$

ITEL is an INTEGER variable that on entry must be set to the bound on the number of iterations; if ITEL is less than zero, the number of iterations performed by the package is not bounded; on exit ITEL contains the number of iterations performed by the package.

VEPS is a REAL variable that on entry must be set to the required accuracy of the load flow solution; the iteration terminates when the maximum of the modulus of the complex bus voltage correction is not greater than VEPS; on exit, VEPS contains the achieved accuracy of the solution.

TIMEL is a REAL variable that on entry must be set to the bound on the iteration time; if TIMEL is less than or equal to zero, the iteration time is not bounded; on exit, TIMEL contains the time spent on iteration.

MODE is an INTEGER argument that must be set to the required mode of the iterative procedure; there are 2 modes defined in the package:

0 -form and decompose the Jacobian matrix using the best pivotal strategy (MA28A is called),

1 -form and factorize the Jacobian matrix (with the same sparsity pattern) using the pivotal sequence determined by the earlier entry with MODE = 0 (MA28B is called).

IFLAG is an INTEGER variable that is used as a return flag describing the solution obtained by the package:

-3 error diagnostics from the Harwell package MA28,

-2 incorrect use of the package (e.g., MODE = 1, which is not preceded by MODE = 0),

-1 incorrect parameters (e.g., insufficient workspace),

0 normal return; required accuracy obtained,

1 limit of iterations reached,

2 limit of iteration time reached.

The contents of the row and column indices INDR and INDC must correspond to the following "conceptual" conversion of the dense COMPLEX bus admittance matrix YY(NB,NB) into the sparse matrix Y(NZ):

```
      L=0
      DO 20 I=1,NB
      DO 10 J=1,NB
      IF (I.EQ.J) GO TO 10
      IF(YY(I,J).EQ.(0.,0.)) GO TO 10
      L=L+1
      INDC(L)=J
      Y(L)=YY(I,J)
10    CONTINUE
      L=L+1
      INDC(L)=I
      INDR(I)=L
      Y(L)=YY(I,I)
20    CONTINUE
```

For example, if the matrix YY(5,5) contains 13 non-zero elements (denoted by *):

```
* 0 0 * 0
0 * * 0 0
0 * * * 0
* 0 * * *
0 0 0 * *
```

then

INDR = [2,4,7,11,13],
INDC = [4,1,3,2,2,4,3,1,3,5,4,4,5].

V. GENERAL INFORMATION

Use of COMMON:	None.
Workspace:	Provided by the user; see arguments W and LW.
Input/Output:	None.
Subroutines:	LFLNR1, LFLNR2 and Harwell package MA28 (MA28A, MA28B, MA28C and their auxiliary subroutines).
Restrictions:	$NB > 0$, $NZ > 0$, $VEPS \geq 0$, $0 \leq MODE \leq 1$.
Date:	September 1982.

VI. EXAMPLES

Example 1

The load flow solution for the test 26-bus power system [11-13] is shown in such a way that the results are printed out after each iteration until the required accuracy VEPS = 10^{-6} is reached. The data file for the 26-bus power system (corresponding to the flat voltage profile) is available as a group indirect file under the charge RJWBAND. The local data file required by the program LFLOW1 has a different name, DATA. In this case, to use the program for the 26-bus test data, the following NOS command can be used to fetch (and rename) the data file

/GET, DATA = D26/GR.

To obtain the load flow solution the user has to write his own program in which he calls the main subroutine LFLNR1. In this example the user's program consists of the main program LFLOW1 and two subroutines, namely FLOW and PRTRES.

The main program LFLOW1 calls the subroutine PWRDS1 from the package PWRDS [14] (available also when the library LIBSPWR is fetched) to read the data and to form the sparse bus admittance matrix. The subroutine FLOW (called by PWRDS1) prepares parameters and calls the main subroutine LFLNR1. It also controls printing of the final results by calling the subroutine PRTRES, which prints the results after each iteration (as in this example) or after the required accuracy is obtained.

```
      PROGRAM LFLOW1 (DATA,OUTPUT,TAPE1=DATA,TAPE6=OUTPUT)
C
C  THIS PROGRAM SOLVES THE LOAD FLOW PROBLEM USING SPARSE MATRIX
C  TECHNIQUES (HARWELL PACKAGE MA28) AND THE NEWTON-RAPHSON METHOD
C  (PACKAGE LFLNR).
C
      DIMENSION W(5000)
      EXTERNAL FLOW
      CALL SECOND(TIME1)
      CALL PWRDS1(FLOW,1,6,W,5000,IRET)
      IF(IRET.NE.0) WRITE(6,111) IRET
  111 FORMAT(/" PWRDS1 RETURN FLAG :",I3)
      CALL SECOND(TIME2)
      EXTIME=TIME2-TIME1
      WRITE(6,222) EXTIME
  222 FORMAT(/" TOTAL EXECUTION TIME :",F7.3," SECONDS")
      STOP
      END
C
C
      SUBROUTINE FLOW (DN,NBS,NS,NTL,NB,NLB,NZ,NLZ,INDR,INDC,IBT,
     1                 Y,YS,V,S,W,LW,LCH,IFLAG)
      DIMENSION INDR(NB),INDC(NZ),IBT(NB),W(LW)
      COMPLEX Y(NZ),YS(NBS),V(NBS),S(NBS)
C
      DO 10 K=1,NB
      IF(IBT(K).EQ.1) S(K)=CMPLX(REAL(S(K)),CABS(V(K)))
   10 CONTINUE
      IFLAG=-5
      ITR=0
      MODE=0
   20 ITL=1
      VEPS=1.E-6
      TIMEX=1.0
      CALL LFLNR1(NB,NZ,INDR,INDC,IBT,Y,YS,V,S,W,LW,ITL,VEPS,
     1            TIMEX,MODE,IRET)
      IF(LCH.GT.0.AND.IRET.LT.0) WRITE(LCH,333) IRET
  333 FORMAT(/" LFLNR1 RETURN FLAG:",I4)
      IF(IRET.LT.0) RETURN
      ITR=ITR+1
      IF(LCH.GT.0) WRITE(LCH,222) ITR,MODE,VEPS,TIMEX
  222 FORMAT(1H1/" ITERATION :                         ",I4
     1          /" ITERATION MODE :                    ",I4
     2          /" ACCURACY OBTAINED :          ",1PE10.3
     3          /" SOLUTION TIME :                 ",0PF6.3," SECONDS")
      CALL PRTRES(NB,NS,INDR,INDC,IBT,Y,YS,V)
      MODE=1
      IF(VEPS.GT.1.E-6) GO TO 20
      IFLAG=0
      RETURN
      END
C
```

```
C
      SUBROUTINE PRTRES(NB,NL,INDR,INDC,IBT,Y,YS,V)
      DIMENSION INDR(1),INDC(1),IBT(1)
      COMPLEX Y(1),YS(1),V(1)
C
C  THIS SUBROUTINE PRINTS FINAL RESULTS OF THE LOAD FLOW SOLUTION
C
C     NB     - NUMBER OF BUSES (EXCLUDING THE SLACK BUS),
C     NL     - INDEX OF THE SLACK BUS,
C     INDR   - ROW INDEX OF THE SPARSE BUS-ADMITTANCE MATRIX,
C     INDC   - COLUMN INDEX OF THE SPARSE BUS-ADMITTANCE MATRIX,
C     IBT    - VECTOR OF BUS TYPES (0 - LOAD, 1 - GENERATOR),
C     Y      - SPARSE COMPLEX BUS-ADMITTANCE MATRIX,
C     YS     - COMPLEX VECTOR OF SLACK ADMITTANCES,
C     V      - COMPLEX VECTOR OF BUS-VOLTAGES (RECTANGULAR MODE).
C
      COMPLEX CC,PW
      FACT=180.0/3.14159265
      NBS=NB+1
      WRITE(6,900) NBS
  900 FORMAT(//10X,"LOAD FLOW SOLUTION BY NEWTON-RAPHSON METHOD",
     1 //20X,I4,"-BUS POWER SYSTEM"///51X,"GENERATOR"/" BUS",
     2 "  COMPLEX BUS VOLTAGE   POLAR BUS VOLTAGE  REACTIVE POWER"/)
      DO 98 I=1,NB
      II=I
      IF(II.GE.NL) II=II+1
      V1=CABS(V(I))
      V2=ATAN2(AIMAG(V(I)),REAL(V(I)))*FACT
      IF(IBT(I).EQ.0) GOTO 97
      J1=1
      IF(I.GT.1) J1=INDR(I-1)+1
      J2=INDR(I)
      CC=YS(I)*V(NBS)
      DO 96 J=J1,J2
      K=INDC(J)
   96 CC=CC+Y(J)*V(K)
      Q=AIMAG(V(I)*CONJG(CC))
      WRITE(6,902) II,V(I),V1,V2,Q
  902 FORMAT(1X,I4,2X,2F9.5,2H*J,2X,F9.5,F9.2,3X,F10.5)
      GOTO 98
   97 WRITE(6,902) II,V(I),V1,V2
   98 CONTINUE
      CC=(0.0,0.0)
      DO 99 I=1,NBS
   99 CC=CC+YS(I)*V(I)
      PW=V(NBS)*CONJG(CC)
      WRITE(6,903) PW
  903 FORMAT(/" COMPLEX SLACK BUS POWER =",3X,2F9.5,2H*J)
      RETURN
      END
```

ITERATION : 1
ITERATION MODE : 0
ACCURACY OBTAINED : 4.022E-01
SOLUTION TIME : .176 SECONDS

LOAD FLOW SOLUTION BY NEWTON-RAPHSON METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.09113	.11125*J	1.09679	5.82	
2	1.08933	.10522*J	1.09440	5.52	
3	1.09021	.08743*J	1.09371	4.59	
4	1.02368	.13888*J	1.03306	7.73	
5	1.01117	.30226*J	1.05537	16.64	
6	1.04069	.06881*J	1.04297	3.78	
7	1.05559	.04579*J	1.05659	2.48	
8	.97005	.08592*J	.97385	5.06	
9	.99168	-.09339*J	.99607	-5.38	
10	1.05910	.08165*J	1.06224	4.41	
11	.92594	-.05460*J	.92754	-3.37	
12	.98315	-.06757*J	.98547	-3.93	
13	1.05216	.01723*J	1.05230	.94	
14	.96810	-.07688*J	.97115	-4.54	
15	.97802	.14122*J	.98817	8.22	
16	1.03882	-.04707*J	1.03989	-2.59	
17	.94734	.08522*J	.95116	5.14	
18	1.07000	.28787*J	1.10805	15.06	-.19791
19	1.05000	.11543*J	1.05633	6.27	.03495
20	1.00000	.28304*J	1.03928	15.80	.66736
21	1.02000	.27365*J	1.05607	15.02	-.21138
22	.89000	-.03149*J	.89056	-2.03	-.36999
23	1.00000	-.02693*J	1.00036	-1.54	-.12066
24	1.00000	.08032*J	1.00322	4.59	-.49429
25	1.00000	.40219*J	1.07785	21.91	.40271

COMPLEX SLACK BUS POWER : -.22384 -.69939*J

ITERATION : 2
ITERATION MODE : 1
ACCURACY OBTAINED : 8.087E-02
SOLUTION TIME : .077 SECONDS

LOAD FLOW SOLUTION BY NEWTON-RAPHSON METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03549	.07792*J	1.03842	4.30	
2	1.06460	.09455*J	1.06879	5.08	
3	1.04464	.05584*J	1.04613	3.06	
4	.98733	.09945*J	.99233	5.75	
5	.97364	.26109*J	1.00804	15.01	
6	1.03270	.05589*J	1.03421	3.10	
7	1.01520	.01874*J	1.01537	1.06	
8	.94518	.04235*J	.94613	2.57	
9	.96252	-.10840*J	.96860	-6.43	
10	1.03720	.06953*J	1.03953	3.83	
11	.90016	-.09726*J	.90540	-6.17	
12	.96761	-.07391*J	.97043	-4.37	
13	1.04637	.01575*J	1.04649	.86	
14	.94038	-.10597*J	.94633	-6.43	
15	.92996	.09830*J	.93514	6.03	
16	1.03527	-.04712*J	1.03635	-2.61	
17	.93338	.03076*J	.93389	1.89	
18	1.03983	.25357*J	1.07030	13.70	-.39959
19	1.04564	.09718*J	1.05015	5.31	.17941
20	.97065	.24250*J	1.00048	14.03	.74065
21	.99397	.23148*J	1.02057	13.11	-.03395
22	.88755	-.08488*J	.89160	-5.46	-.18152
23	.99965	-.02655*J	1.00000	-1.52	-.11442
24	.99936	.04801*J	1.00052	2.75	-.17731
25	.93595	.35282*J	1.00024	20.65	.17082

COMPLEX SLACK BUS POWER : .12334 -.05892*J

ITERATION : 3
ITERATION MODE : 1
ACCURACY OBTAINED : 4.092E-03
SOLUTION TIME : .080 SECONDS

LOAD FLOW SOLUTION BY NEWTON-RAPHSON METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03277	.07730*J	1.03566	4.28	
2	1.06437	.09434*J	1.06854	5.07	
3	1.04237	.05495*J	1.04382	3.02	
4	.98591	.09788*J	.99076	5.67	
5	.97408	.25983*J	1.00814	14.94	
6	1.03245	.05542*J	1.03393	3.07	
7	1.01319	.01806*J	1.01335	1.02	
8	.94412	.04027*J	.94498	2.44	
9	.96138	-.10877*J	.96751	-6.45	
10	1.03697	.06925*J	1.03928	3.82	
11	.89823	-.09921*J	.90369	-6.30	
12	.96705	-.07406*J	.96988	-4.38	
13	1.04633	.01572*J	1.04645	.86	
14	.93882	-.10713*J	.94492	-6.51	
15	.92735	.09701*J	.93241	5.97	
16	1.03526	-.04712*J	1.03633	-2.61	
17	.93177	.02781*J	.93218	1.71	
18	1.03970	.25282*J	1.07000	13.67	-.40043
19	1.04555	.09658*J	1.05000	5.28	.18719
20	.97058	.24080*J	1.00000	13.93	.77937
21	.99384	.22952*J	1.02000	13.00	-.02941
22	.88560	-.08848*J	.89001	-5.71	-.17745
23	.99965	-.02655*J	1.00000	-1.52	-.11439
24	.99895	.04585*J	1.00000	2.63	-.16454
25	.93591	.35223*J	1.00000	20.62	.16912

COMPLEX SLACK BUS POWER : .13337 -.05130*J

ITERATION : 4
ITERATION MODE : 1
ACCURACY OBTAINED : 1.402E-05
SOLUTION TIME : .077 SECONDS

LOAD FLOW SOLUTION BY NEWTON-RAPHSON METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03276	.07729*J	1.03565	4.28	
2	1.06437	.09434*J	1.06854	5.07	
3	1.04236	.05495*J	1.04381	3.02	
4	.98590	.09787*J	.99075	5.67	
5	.97408	.25981*J	1.00814	14.93	
6	1.03244	.05542*J	1.03393	3.07	
7	1.01318	.01806*J	1.01334	1.02	
8	.94412	.04026*J	.94498	2.44	
9	.96137	-.10877*J	.96751	-6.45	
10	1.03697	.06924*J	1.03928	3.82	
11	.89822	-.09922*J	.90368	-6.30	
12	.96704	-.07406*J	.96988	-4.38	
13	1.04633	.01572*J	1.04645	.86	
14	.93882	-.10713*J	.94491	-6.51	
15	.92734	.09701*J	.93240	5.97	
16	1.03526	-.04712*J	1.03633	-2.61	
17	.93176	.02780*J	.93218	1.71	
18	1.03970	.25282*J	1.07000	13.67	-.40042
19	1.04555	.09658*J	1.05000	5.28	.18722
20	.97058	.24078*J	1.00000	13.93	.77951
21	.99384	.22951*J	1.02000	13.00	-.02939
22	.88559	-.08849*J	.89000	-5.71	-.17746
23	.99965	-.02655*J	1.00000	-1.52	-.11439
24	.99895	.04584*J	1.00000	2.63	-.16451
25	.93592	.35222*J	1.00000	20.62	.16913

COMPLEX SLACK BUS POWER : .13341 -.05129*J

ITERATION : 5
ITERATION MODE : 1
ACCURACY OBTAINED : 2.202E-10
SOLUTION TIME : .079 SECONDS

LOAD FLOW SOLUTION BY NEWTON-RAPHSON METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03276	.07729*J	1.03565	4.28	
2	1.06437	.09434*J	1.06854	5.07	
3	1.04236	.05495*J	1.04381	3.02	
4	.98590	.09787*J	.99075	5.67	
5	.97408	.25981*J	1.00814	14.93	
6	1.03244	.05542*J	1.03393	3.07	
7	1.01318	.01806*J	1.01334	1.02	
8	.94412	.04026*J	.94498	2.44	
9	.96137	-.10877*J	.96751	-6.45	
10	1.03697	.06924*J	1.03928	3.82	
11	.89822	-.09922*J	.90368	-6.30	
12	.96704	-.07406*J	.96988	-4.38	
13	1.04633	.01572*J	1.04645	.86	
14	.93882	-.10713*J	.94491	-6.51	
15	.92734	.09701*J	.93240	5.97	
16	1.03526	-.04712*J	1.03633	-2.61	
17	.93176	.02780*J	.93218	1.71	
18	1.03970	.25282*J	1.07000	13.67	-.40042
19	1.04555	.09658*J	1.05000	5.28	.18722
20	.97058	.24078*J	1.00000	13.93	.77951
21	.99384	.22951*J	1.02000	13.00	-.02939
22	.88559	-.08849*J	.89000	-5.71	-.17746
23	.99965	-.02655*J	1.00000	-1.52	-.11439
24	.99895	.04584*J	1.00000	2.63	-.16451
25	.93592	.35222*J	1.00000	20.62	.16913

COMPLEX SLACK BUS POWER : .13341 -.05129*J

TOTAL EXECUTION TIME : 1.869 SECONDS

Example 2

The load flow solution for the test 118-bus power system [14-16] is shown. In this case, the solution is obtained by one call of the package with the required accuracy VEPS = 10^{-6} and the bound on the iteration time TIMEL = 5 seconds.

Because of the size of the workspace, the compiled program LDFLOW is submitted to the batch queue with the following job description:

```
100 /JOB
110 BUS118,JC2
120 /USER
130 /CHARGE
140 GET (LIBSPWR,LIBRHSM,DATA=D118/GR)
150 LIBRARY (LIBSPWR, LIBRHSM)
160 GET(LDFLOW)
170 MAP(OFF)
180 LDFLOW
190 /EOF
```

The data file D118 corresponds to the unoptimized system with flat voltage profile. As in Example 1, the user's program consists of the main program LFLOW2 and two subroutines, namely FLOW and PRTRES. The main program LFLOW2 calls the subroutine PWRDS1 from the package PWRDS to read the data and to form the sparse bus admittance matrix.

The subroutines FLOW and PRTRES have the same function as in Example 1. Moreover, the subroutine PRTRES is identical with that given on p.15.

```
      PROGRAM LFLOW2 (DATA,OUTPUT,TAPE1=DATA,TAPE6=OUTPUT)
C
C  THIS PROGRAM SOLVES THE LOAD FLOW PROBLEM USING SPARSE MATRIX
C  TECHNIQUES (HARWELL PACKAGE MA28) AND THE NEWTON-RAPHSON METHOD
C  (PACKAGE LFLNR).
C
      DIMENSION W(20000)
      EXTERNAL FLOW
      CALL SECOND(TIME1)
      CALL PWRDS1(FLOW,1,6,W,20000,IRET)
      IF(IRET.NE.0) WRITE(6,111) IRET
  111 FORMAT(/" PWRDS1 RETURN FLAG :",I3)
      CALL SECOND(TIME2)
      EXTIME=TIME2-TIME1
      WRITE(6,222) EXTIME
  222 FORMAT(/" TOTAL EXECUTION TIME :",F7.3," SECONDS")
      STOP
      END
C
C
      SUBROUTINE FLOW (DN,NBS,NS,NTL,NB,NLB,NZ,NLZ,INDR,INDC,IBT,
     1                  Y,YS,V,S,W,LW,LCH,IFLAG)
      DIMENSION INDR(NB),INDC(NZ),IBT(NB),W(LW)
      COMPLEX Y(NZ),YS(NBS),V(NBS),S(NBS)
C
      IF(LCH.GT.0) WRITE(LCH,111) DN,NBS,NS,NTL,NZ,LW
  111 FORMAT(/"0DATA-NAME : ",A10
     1       /" NUMBER OF BUSES :                 ",I4
     2       /" SLACK-BUS INDEX :                 ",I4
     3       /" NUMBER OF TRANSMISSION LINES :",I4
     4       /" NUMBER OF NON-ZEROS :             ",I4
     5       /" WORKSPACE USED       :           ",I5/)
      DO 10 K=1,NB
      IF(IBT(K).EQ.1) S(K)=CMPLX(REAL(S(K)),CABS(V(K)))
   10 CONTINUE
      IFLAG=-5
      ITL=-1
      VEPS=1.E-6
      MODE=0
      TIMEX=5.0
      CALL LFLNR1(NB,NZ,INDR,INDC,IBT,Y,YS,V,S,W,LW,ITL,VEPS,
     1            TIMEX,MODE,IRET)
      IF(LCH.GT.0.AND.IRET.LT.0)WRITE(LCH,333) IRET
  333 FORMAT(/" LFLNR1 RETURN FLAG:",I4)
      IF(IRET.LT.0) RETURN
      IF(LCH.GT.0) WRITE(LCH,222) IRET,MODE,ITL,VEPS,TIMEX
  222 FORMAT(/" RETURN FLAG :                     ",I4
     1       /" ITERATION MODE :                  ",I4
     2       /" NUMBER OF ITERATIONS :            ",I4
     3       /" ACCURACY OBTAINED :         ",1PE10.3
     4       /" SOLUTION TIME :                  ",0PF6.3," SECONDS")
      CALL PRTRES(NB,NS,INDR,INDC,IBT,Y,YS,V)
      IFLAG=0
      RETURN
      END
C
```

```
DATA-NAME : DATA-118
NUMBER OF BUSES :                118
SLACK-BUS INDEX :                118
NUMBER OF TRANSMISSION LINES : 179
NUMBER OF NON-ZEROS :            463
WORKSPACE USED       :         17631

RETURN FLAG :                      0
ITERATION MODE :                   0
NUMBER OF ITERATIONS :             5
ACCURACY OBTAINED :        6.601E-08
SOLUTION TIME :                2.785 SECONDS
```

LOAD FLOW SOLUTION BY NEWTON-RAPHSON METHOD

118-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.01438	-.54179*J	1.15000	-28.11	3.89116
2	.90077	-.42455*J	.99580	-25.24	
3	.97038	-.47127*J	1.07876	-25.90	
4	.99951	-.45932*J	1.10000	-24.68	3.92275
5	.96766	-.35304*J	1.03005	-20.04	
6	.83817	-.32782*J	.90000	-21.36	-2.18551
7	.83796	-.33654*J	.90301	-21.88	
8	1.05406	-.27761*J	1.09000	-14.75	4.89627
9	1.00712	-.13173*J	1.01569	-7.45	
10	.91949	.03061*J	.92000	1.91	-3.23178
11	.88233	-.37508*J	.95875	-23.03	
12	.84197	-.34524*J	.91000	-22.30	-4.74042
13	.86496	-.38725*J	.94769	-24.12	
14	.85093	-.36167*J	.92460	-23.03	
15	.89219	-.38066*J	.97000	-23.11	.53879
16	.84823	-.35194*J	.91834	-22.53	
17	.92287	-.34071*J	.98375	-20.26	
18	.89528	-.37333*J	.97000	-22.64	.08458
19	.88291	-.37693*J	.96000	-23.12	-.19968
20	.87913	-.35523*J	.94819	-22.00	
21	.88618	-.32620*J	.94431	-20.21	
22	.90737	-.28394*J	.95076	-17.38	
23	.95359	-.20151*J	.97465	-11.93	
24	.89433	-.16817*J	.91000	-10.65	-1.71564
25	1.09346	-.11974*J	1.10000	-6.25	5.76838
26	1.00707	-.07690*J	1.01000	-4.37	-4.17524
27	.92115	-.30393*J	.97000	-18.26	1.12403
28	.78846	-.26035*J	.83034	-18.27	
29	.81482	-.29571*J	.86682	-19.95	
30	1.01319	-.27064*J	1.04872	-14.96	
31	.86062	-.32516*J	.92000	-20.70	.80291
32	.90808	-.31145*J	.96000	-18.93	.10756
33	.88674	-.36783*J	.96001	-22.53	
34	.87308	-.34832*J	.94000	-21.75	-1.87842
35	.90384	-.36956*J	.97648	-22.24	
36	.90542	-.37498*J	.98000	-22.50	1.86537
37	.90961	-.33135*J	.96809	-20.02	
38	1.00279	-.26512*J	1.03725	-14.81	
39	.88560	-.37545*J	.96189	-22.97	
40	.88779	-.39080*J	.97000	-23.76	-.92445

41	.90848	-.41122*J	.99722	-24.35	
42	1.00959	-.43672*J	1.10000	-23.39	2.42836
43	.88035	-.33242*J	.94102	-20.69	
44	.92386	-.27313*J	.96339	-16.47	
45	.94340	-.23556*J	.97237	-14.02	
46	.98197	-.18906*J	1.00000	-10.90	-.07300
47	1.00114	-.14942*J	1.01223	-8.49	
48	1.00266	-.16071*J	1.01546	-9.11	
49	1.01006	-.14206*J	1.02000	-8.01	.09364
50	.98298	-.16582*J	.99687	-9.57	
51	.94481	-.18991*J	.96371	-11.36	
52	.93310	-.19795*J	.95386	-11.98	
53	.92133	-.19581*J	.94190	-12.00	
54	.93450	-.17093*J	.95000	-10.37	-.77522
55	.92951	-.19624*J	.95000	-11.92	-.04618
56	.93015	-.19317*J	.95000	-11.73	-.29946
57	.94835	-.18798*J	.96680	-11.21	
58	.93528	-.19574*J	.95554	-11.82	
59	.97821	-.15233*J	.99000	-8.85	.69778
60	.98400	-.11590*J	.99080	-6.72	
61	.99697	-.07783*J	1.00000	-4.46	-.13481
62	.99515	-.09841*J	1.00000	-5.65	-.10655
63	1.01472	-.10311*J	1.01994	-5.80	
64	1.01354	-.07609*J	1.01639	-4.29	
65	.99955	-.02989*J	1.00000	-1.71	-5.81631
66	1.04954	-.03101*J	1.05000	-1.69	4.14126
67	1.01779	-.07632*J	1.02065	-4.29	
68	1.03377	-.05118*J	1.03504	-2.83	
69	.94183	-.12098*J	.94957	-7.32	
70	.97124	-.13070*J	.98000	-7.66	-.02294
71	.97397	-.14256*J	.98435	-8.33	
72	.96326	-.18037*J	.98000	-10.61	.31545
73	.97898	-.14728*J	.99000	-8.56	.12903
74	.95044	-.13511*J	.96000	-8.09	-.24539
75	.96132	-.11135*J	.96775	-6.61	
76	.93387	-.10721*J	.94000	-6.55	-.44033
77	1.00988	-.01539*J	1.01000	-.87	.11658
78	.99905	-.02685*J	.99941	-1.54	
79	1.00589	-.01700*J	1.00604	-.97	
80	1.03935	.03690*J	1.04000	2.03	.85596
81	1.00570	.02312*J	1.00596	1.32	
82	.99152	.01070*J	.99158	.62	
83	.98717	.03394*J	.98776	1.97	
84	.98072	.08154*J	.98410	4.75	
85	.98379	.11075*J	.99000	6.42	-.04216
86	.98406	.08825*J	.98801	5.12	
87	1.00546	.09568*J	1.01000	5.44	.07903
88	.97175	.16861*J	.98627	9.84	
89	.96973	.24417*J	1.00000	14.13	-.44583
90	.98129	.13100*J	.99000	7.60	.38645
91	.97105	.13211*J	.98000	7.75	-.18938
92	.97952	.14366*J	.99000	8.34	-.17526
93	.98201	.09075*J	.98619	5.28	
94	.98984	.05340*J	.99128	3.09	
95	.98121	.03105*J	.98170	1.81	
96	.99339	.02187*J	.99363	1.26	
97	1.01167	.02292*J	1.01192	1.30	
98	1.02436	.02292*J	1.02462	1.28	
99	1.00936	.03608*J	1.01000	2.05	-.22816
100	1.01835	.05807*J	1.02000	3.26	.96245
101	.99035	.07861*J	.99346	4.54	
102	.98261	.12011*J	.98993	6.97	
103	1.00982	.01925*J	1.01000	1.09	.39058
104	.96938	-.03459*J	.97000	-2.04	-.15752
105	.95864	-.05106*J	.96000	-3.05	-.62577

106	.95644	-.05876*J	.95825	-3.52	
107	.94432	-.10369*J	.95000	-6.27	-.03307
108	.95950	-.06992*J	.96204	-4.17	
109	.96009	-.07703*J	.96318	-4.59	
110	.96579	-.09030*J	.97000	-5.34	-.45960
111	.97791	-.06397*J	.98000	-3.74	.02020
112	.96893	-.14690*J	.98000	-8.62	.41012
113	.92875	-.34282*J	.99000	-20.26	.32398
114	.90682	-.31662*J	.96051	-19.25	
115	.90953	-.31735*J	.96330	-19.23	
116	.98900	-.14793*J	1.00000	-8.51	.12405
117	.81492	-.36511*J	.89297	-24.13	

COMPLEX SLACK BUS POWER :   4.73875   .39459*J

TOTAL EXECUTION TIME : 6.494 SECONDS

VII. REFERENCES

[1] J.E. Van Ness and J.H. Griffin, "Elimination methods for load flow studies", AIEE Trans. Power Apparatus and Systems, vol. 80, 1961, pp. 299-302.

[2] C.G. Broyden, "A class of methods for solving non-linear simultaneous equations", Mathematics of Computation, vol. 19, 1965, pp. 577-593.

[3] N. Sato and W.F. Tinney, "Techniques for exploiting the sparsity of the network admittance matrix", IEEE Trans. Power Apparatus and Systems, vol. 82, 1963, pp. 944-949.

[4] W.F. Tinney and C.E. Hart, "Power flow solutions by Newton's method", IEEE Trans. Power Apparatus and Systems, vol. 86, 1967, pp. 1449-1460.

[5] W.F. Tinney and J.W. Walker, "Direct solutions of sparse network equations by optimally order triangular factorization", Proc. IEEE, vol. 55, 1967, pp. 1801-1809.

[6] B. Stott, "Review of load flow calculation methods", Proc. IEEE, vol. 62, 1974, pp. 916-929.

[7] G.W. Stagg and A.H. El-Abiad, Computer Methods in Power System Analysis. New York: McGraw-Hill, 1968.

[8] 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.

[9] 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-L, July 1983.

[10] J.W. Bandler and M.A. El-Kady, "Newton's load flow in complex mode", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-80-2, 1980.

[11] J.W. Bandler and M.A. El-Kady, "The adjoint network approach to power flow solution and sensitivities of test power systems: data and results", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-80-15, 1980.

[12] M.S. Sachdev and S.A. Ibrahim, "A fast approximate technique for outage studies in power system planning and operation", IEEE Trans. Power Apparatus and Systems, vol. PAS-93, 1974, pp. 1133-1142.

[13] D26,D118, data sets for test power systems, Simulation Optimization Systems Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, 1982.

[14] PWRDS, reading and preprocessing power systems data, Simulation Optimization Systems Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, 1982.

[15] "American Electric Power 118 Bus Test System", American Electric Power Service Corporation, New York, NY, December 1962.

[16] R.C. Burchett, Electric Utility Systems Engineering Department, General Electric Company, Schenectady, NY, 1981, private communication.