INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-296

LFLFD - A FORTRAN IMPLEMENTATION OF THE FAST DECOUPLED LOAD FLOW TECHNIQUE

J.W. Bandler and W.M. Zuberek

July 1982

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

LFLFD - A FORTRAN IMPLEMENTATION OF THE FAST DECOUPLED LOAD FLOW TECHNIQUE

J.W. Bandler and W.M. Zuberek

Abstract

LFLFD is a package of subroutines for solving load flow problems by the well-known fast decoupled technique. The method has been described by Stott and Alsac, and is implemented with minor modifications only. Sparse matrix techniques are used to represent the power system's bus admittance matrix as well as the approximate Jacobian matrices required by the method, and the Harwell Package MA28 is called to solve the systems of linear equations with real coefficients. The package and documentation have been developed for the CDC 170/730 system with the NOS 1.4 operating system and the Fortran 4.8508 compiler.

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

J.W. Bandler and W.M. Zuberek are with the Group on Simulation, Optimization and Control, and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

W.M. Zuberek is on leave from the Institute of Computer Science, Technical University of Warsaw, Warsaw, Poland.

I. INTRODUCTION

The fast decoupled method proposed by Stott and Alsac [1] has been recognized as a simple, reliable and fast load flow solution technique [2,3] with a wide range of practical applications. The method is derived from the exact generalized Newton-Raphson iterative algorithm but instead of re-evaluating the Jacobian matrix, it uses a constant Jacobian approximation closely corresponding to the initial point (flat voltage profile). Consequently, the initial convergence is very good, and practical approximations of solutions can be obtainend after 5 to 10 iterations. Due to the constant approximation of the Jacobian matrix the convergence rate of the method is, however, linear [1], and if exact solution is required, the number of iterations is usually significantly greater than that of Newton's method, having a quadratic convergence rate. On the other hand, the iteration scheme of the fast decoupled method is very simple, and the much faster iterations more than compensate for the increased number of iterations.

The fast decoupled method has been implemented as the package LFLFD 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 LFLFD1. The library is available in the group indirect file LIBSPWR under the charge RJWBAND. The package calls the subroutines MA28A and MA28C from the Harwell Subroutine Library (Harwell Package MA28) [4]; the package MA28 must thus also be available when LFLFD is used. The general sequence of NOS commands to use the LFLFD package may be as follows:

/GET(LIPSPWR,LIBRHSM/GR) - fetch the libraries,

/LIBRARY(LIBSPWR,LIBRHSM) - indicate the libraries to the loader,

/FTN(...,GO) - compile, load and execute the program.

II. GENERAL DESCRIPTION

The well-known [2] power-mismatch equations in polar form can be written as

$$\Delta P_i = P_i^0 - V_i \sum_{1 \leq k \leq n} (G_{ik}\cos(\delta_i-\delta_k) + B_{ik}\sin(\delta_i-\delta_k))V_k \;, \tag{1}$$

$$\Delta Q_j = Q_j^0 - V_j \sum_{1 \leq k \leq n} (G_{jk}\sin(\delta_j-\delta_k) - B_{jk}\cos(\delta_j-\delta_k))V_k \;, \tag{2}$$

where ΔP_i and ΔQ_j are mismatches of active and reactive power for bus i, $i = 1,\ldots,n$, or bus j, $j = 1,\ldots,n_L$, respectively, n is the number of buses, n_L is the number of load (or P,Q) buses, P_i^0 and Q_j^0 are nominal values of active and reactive power at buses i and j, respectively, V_i and δ_i are the magnitude and the argument of the complex bus voltage, and $G_{ik} + jB_{ik}$ is the (i,k)th element of the complex bus admittance matrix.

The load flow problem consists of determining such bus voltage magnitudes V_i and arguments δ_i, $i = 1,\ldots,n$, that the power mismatches (1) and (2) are equal to zero, i.e., it corresponds to the solution of a set of simultaneous nonlinear equations

$$\Delta P_i = 0 \;, \quad i = 1,\ldots,n, \tag{3}$$

$$\Delta Q_j = 0 \;, \quad j = 1,\ldots,n_L. \tag{4}$$

Usually, the system of nonlinear equations (3) and (4) is solved iteratively (e.g., using the Newton-Raphson method, the decoupled Newton method, etc.), and then the performance of the method is closely associated with the degree of nonlinearity of equations. Therefore, in practical implementations [2,3] the equations (3), (4) are often represented in the form

$$\Delta\overline{P}_i \overset{\Delta}{=} \Delta P_i/V_i = 0 \quad , \quad i = 1,\ldots,n, \tag{5}$$

$$\Delta\overline{Q}_j \overset{\Delta}{=} \Delta Q_j/V_j = 0 \quad , \quad j = 1,\ldots,n_L, \tag{6}$$

where

$$\Delta\overline{P}_i = P_i^0/V_i - \sum_{1\le k\le n} V_k(G_{ik}\cos(\delta_i-\delta_k) + B_{ik}\sin(\delta_i-\delta_k)) \ , \tag{7}$$

$$\Delta\overline{Q}_j = Q_j^0/V_j - \sum_{1\le k\le n} V_k(G_{jk}\cos(\delta_j-\delta_k) - B_{jk}\sin(\delta_j-\delta_k)) \ , \tag{8}$$

in which only one term in each equation is nonlinear.

The generalized Newton-Raphson iterative method uses the square Jacobian matrix that can be represented in a partitioned form [2]

$$\begin{bmatrix} \Delta\underset{\sim}{\overline{P}} \\ \Delta\underset{\sim}{\overline{Q}} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{H} & \underset{\sim}{N} \\ \underset{\sim}{M} & \underset{\sim}{L} \end{bmatrix} \begin{bmatrix} \Delta\underset{\sim}{\delta} \\ \Delta\underset{\sim}{V} \end{bmatrix} \ , \tag{9}$$

where the submatrices $\underset{\sim}{H}$, $\underset{\sim}{N}$, $\underset{\sim}{M}$ and $\underset{\sim}{L}$ represent the negated partial derivatives of (7) and (8) with respect to the relevant δ's and V's.

The fast decoupled method is based on the two approximations of the exact Newton-Raphson scheme:

(1) application of the P-δ/Q-V decoupling principle (i.e., neglecting the coupling submatrices $\underset{\sim}{N}$ and $\underset{\sim}{M}$) that results in two separate systems of equations

$$\Delta\underset{\sim}{\overline{P}} = \underset{\sim}{H} \ \ \Delta\underset{\sim}{\delta} \tag{10}$$

and

$$\Delta \overline{\underset{\sim}{Q}} = \underset{\sim}{L} \, \Delta \underset{\sim}{V} \,, \tag{11}$$

(2) approximation of the Jacobian submatrices $\underset{\sim}{H}$ and $\underset{\sim}{L}$ by the matrices $\overline{\underset{\sim}{A}}$ and $\overline{\underset{\sim}{B}}$ of constant terms

$$\Delta \overline{\underset{\sim}{P}} = \overline{\underset{\sim}{A}} \;\; \Delta \underset{\sim}{\delta} \,, \tag{12}$$

$$\Delta \overline{\underset{\sim}{Q}} = \overline{\underset{\sim}{B}} \;\; \Delta \underset{\sim}{V} \,, \tag{13}$$

where $\overline{\underset{\sim}{B}}$ is the real matrix of negated bus susceptances

$$\overline{B}_{jk} = -B_{jk} \quad , \quad j,k = 1,\ldots,n_L,$$

and $\overline{\underset{\sim}{A}}$ is a real matrix of negated inverted transmission-line reactances

$$\overline{A}_{ik} = - \frac{1}{X_{ik}} \qquad , \quad i,k = 1,\ldots,n, \; i \neq k, \tag{14}$$

$$\overline{A}_{ii} = \sum_{1 \leq k \leq n} \frac{1}{X_{ik}} \quad , \quad i = 1,\ldots,n \;, \; k \neq i. \tag{15}$$

Pursuing the decoupling principle, the elements that primarily affect the Q-V relationship, e.g., line shunt admittances, should not be represented in $\overline{\underset{\sim}{A}}$.

In effect, the fast decoupled method results in the two systems of linear equations with constant coefficients (12), (13). The matrices $\overline{\underset{\sim}{A}}$ and $\overline{\underset{\sim}{B}}$ can thus be evaluated and factorized once only, and the iterative process can be very efficient since it requires the $\Delta P/V$ and $\Delta Q/V$ function evaluations and backward/forward substitutions only.

The systems of equations (12) and (13) can be solved simultaneously at each iteration, but a much better approach is to conduct each iteration cycle independently and to use the updated values of δ's (or V's) to solve the corrections $\Delta \underset{\sim}{V}$ (or $\Delta \underset{\sim}{\delta}$, respectively).

It has been observed [1] that $\underset{\sim}{\delta}$ usually requires more iterations to converge than $\underset{\sim}{V}$. The basic scheme (1δ,1V), i.e., one solution of the system (12), then one solution of the system (13) and so on, has been compared with the schemes (2δ,1V), (2δ,2V), (3δ,2V), (2δ,1V) once and subsequently (1δ,1V), it appeared, however, that the basic scheme is generally the best one. Moreover, different attempts at acceleration, including block successive over-relaxation and a heuristic approach based on testing sign-changes in the bus-mismatches, are reported [1] as unrewarding.

The method used in the LFLFD package basically follows the method of Stott and Alsac, and differs only in a more flexible iteration scheme in which the order of successive P-δ and Q-V iterations is not fixed but depends on the relationship between the accuracies of P-δ (12) and Q-V (13) iterations. Let ε_δ be the accuracy of the last P-δ iteration, and ε_V be the accuracy of the last Q-V iteration. If, after the kth iteration,

$$\varepsilon_\delta \geq 2\varepsilon_V$$

the (k+1)th iteration is the P-δ one. If

$$\varepsilon_V \geq 2\varepsilon_\delta$$

the (k+1)th iteration is the Q-V one. Otherwise the basic (1δ,1V) scheme is followed. The implemented scheme tends to avoid large differences between the two accuracies and over-converging of one of the two systems of equations (12) and (13) which slows down the overall convergence rate [1]. It appeared in several test runs that the additional "compensating" iterations are performed relatively seldom, usually once during a load flow solution.

The package allows one to perform one of the two iterations only (the P-δ or the Q-V iteration) and to control externally (i.e., by one of arguments) the number of iterations performed. Implementation of different iteration schemes is thus very simple, corresponding to a sequence of CALL statements with appropriate parameters.

III. STRUCTURE OF THE PACKAGE

The package is composed of 4 subroutines. Its structure is shown in Fig. 1. LFLFD1 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 the remaining subroutines. It also checks formal correctness of some parameters defined by the user, and sets the return flag appropriately.

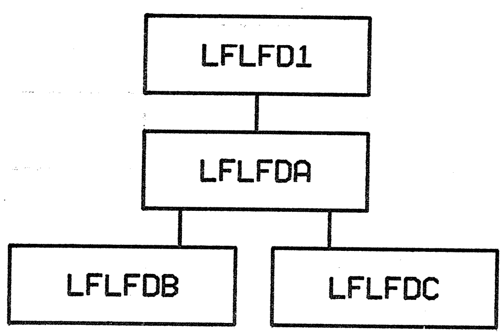

Fig. 1 Structure of the LFLFD package.

LFLFDA creates and factorizes the sparse matrices of real coefficients $\overline{\underset{\sim}{A}}$ and $\overline{\underset{\sim}{B}}$ (using the Harwell package MA28), and controls the iteration scheme performed by LFLFDB and LFLFDC. It also checks the required accuracy of load flow solution as well as bounds on the number of iterations and on the iteration time, if required by the user. The P-δ iterations are performed by LFLFDB, and the Q-V iterations by LFLFDC. Both subroutines use the Harwell package MA28 for solving corresponding systems of linear equations.

IV. LIST OF ARGUMENTS

There is one general entry to the package

```
CALL LFLFD1(NB,NL,NZ,NLZ,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.

NL is an INTEGER argument that must be set to the number of load (or P,Q) buses; it must be positive and not greater than NB; it is not changed by the package.

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

NLZ is an INTEGER argument that must be set to the number of elements of the sparse bus admittance submatrix corresponding to load buses; 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 elements in the corresponding number of initial rows of the bus admittance matrix, 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 indexes 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 an INTEGER vector of length at least NB that must describe the types of buses; its elements must be equal to 0 for load (or P,Q) buses, and equal to 1 for generator (or P,V) buses.

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

YS is a COMPLEX vector of length at least (NB+1) that must contain those elements of the bus admittance matrix which correspond to the slack bus or zeros.

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); it is not changed by the package.

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; it must be at least

1+6*NB+6*NZ+max(NZ+11*NB,5*NL+6*NLZ+max(4*NB,NLZ+11*NL)).

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 EPS; on exit EPS contains the achieved accuracy of the solution.

TIMEL   is the 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 operation; there are 4 modes defined in the package:

0 - evaluate and factorize approximate Jacobian matrices, and perform the P-δ and Q-V iterations unless ITEL = 0,

1 - perform the P-δ iteration only for previously factorized Jacobian matrices,

2 - perform the Q-V iteration only for previously factorized Jacobian matrices,

3 - perform the P-δ and Q-V iteration for previously factorized Jacobian matrices.

Remark: When MODE = 0 or MODE = 3 is used, the P-δ iteration is always performed as the first one, and it is followed by the Q-V iteration even if ITEL = 1 (this is the only situation when the limit of iterations is exceeded by the package).

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

- -2 incorrect use of the package (e.g., singular Jacobian matrices or MODE = 1,2 or 3 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 indexes 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:	LFLFD1, LFLFDA, LFLFDB, LFLFDC and Harwell Package MA28 (MA28A, MA28C and their auxiliary subroutines).
Restrictions:	NB > 0, NL > 0, NL $\leq$ NB, NZ > 0, NLZ > 0, VEPS $\geq$ 0, MODE $\geq$ 0, MODE $\leq$ 3.
Date:	June 1982.

VI. EXAMPLES

Example 1

The load flow solution for the test 26-bus power system [5,6] is shown in such a way that the initial 6 iterations are externally controlled (using MODE = 0,1 and 2) by simulation of the iteration scheme implemented within the package. After each of the initial iterations the results are printed out to show the convergence of the method. After 6 iterations a switch is made to obtain the exact solution (MODE = 3) with the required accuracy VEPS = 10^{-6}, and final results are printed out as well.

The program calls the subroutine PWRDS1 from the package PWRDS [7] to read the data and to form the sparse bus admittance matrix.

```
      PROGRAM LFLOW1 (DATA,OUTPUT,TAPE1=DATA,TAPE6=OUTPUT)
C
C  PROGRAM SOLVES THE LOAD FLOW PROBLEM USING SPARSE MATRIX TECHNIQUES
C  (HARWELL PACKAGE MA28) AND THE FAST-DECOUPLED METHOD (PACKAGE LFLFD)
C
      DIMENSION W(3000)
      EXTERNAL FLOW
      CALL SECOND(TIME1)
      CALL PWRDS1(FLOW,1,6,W,3000,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)
      IFLAG=-5
      ITL=0
      VEPS=0.0
      MODE=0
      TIMEX=0.0
      CALL LFLFD1(NB,NLB,NZ,NLZ,INDR,INDC,IBT,Y,YS,V,S,W,LW,ITL,VEPS,
     1            TIMEX,MODE,IRET)
      IF(IRET.LT.0) RETURN
      CORRA=1.E10
      CORRM=1.E10
      IT=0
   10 MODE=1
      GOTO 30
   20 MODE=2
   30 ITL=1
      VEPS=0.0
      TIMEX=0.0
      CALL LFLFD1(NB,NLB,NZ,NLZ,INDR,INDC,IBT,Y,YS,V,S,W,LW,ITL,VEPS,
     1            TIMEX,MODE,IRET)
      IF(IRET.LT.0) RETURN
      IT=IT+1
      IF(LCH.GT.0) WRITE(LCH,222) IT,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)
      IF(IT.EQ.6) GOTO 50
      IF(MODE.EQ.1) CORRA=VEPS
      IF(MODE.EQ.2) CORRM=VEPS
      IF(CORRA.GT.CORRM+CORRM) GOTO 10
      IF(CORRM.GT.CORRA+CORRA) GOTO 20
      IF(MODE.EQ.1) GOTO 20
      GOTO 10
      VEPS=1.E-6
      MODE=3
      TIMEX=1.5
   50 MODE=3
      ITL=-1
      TIMEX=1.0
      VEPS=1.E-6
      CALL LFLFD1(NB,NLB,NZ,NLZ,INDR,INDC,IBT,Y,YS,V,S,W,LW,ITL,VEPS,
```

```
     1              TIMEX,MODE,IRET)
      IF(IRET.LT.0) RETURN
      IT=IT+ITL
      IF(LCH.GT.0) WRITE(LCH,333) IT,MODE,IRET,VEPS,TIMEX
  333 FORMAT(1H1/" S O L U T I O N"
     1          //" NUMBER OF ITERATIONS :          ",I4
     2           /" ITERATION MODE :                ",I4
     3           /" RETURN FLAG :                   ",I4
     4           /" ACCURACY OBTAINED :      ",1PE10.3
     5           /" SOLUTION TIME :                 ",0PF6.3," SECONDS")
      CALL PRTRES(NB,NS,INDR,INDC,IBT,Y,YS,V)
      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     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 FAST-DECOUPLED 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 : 1
ACCURACY OBTAINED : 3.669E-01
SOLUTION TIME : .017 SECONDS

LOAD FLOW SOLUTION BY FAST-DECOUPLED METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	.99546	.09520*J	1.00000	5.46	
2	.99563	.09334*J	1.00000	5.36	
3	.99744	.07146*J	1.00000	4.10	
4	.99351	.11375*J	1.00000	6.53	
5	.96223	.27223*J	1.00000	15.80	
6	.99811	.06153*J	1.00000	3.53	
7	.99954	.03037*J	1.00000	1.74	
8	.99839	.05671*J	1.00000	3.25	
9	.99551	-.09468*J	1.00000	-5.43	
10	.99754	.07015*J	1.00000	4.02	
11	.99604	-.08890*J	1.00000	-5.10	
12	.99761	-.06909*J	1.00000	-3.96	
13	.99989	.01498*J	1.00000	.86	
14	.99577	-.09192*J	1.00000	-5.27	
15	.99232	.12368*J	1.00000	7.10	
16	.99887	-.04752*J	1.00000	-2.72	
17	.99918	.04042*J	1.00000	2.32	
18	1.03826	.25869*J	1.07000	13.99	.79673
19	1.04557	.09635*J	1.05000	5.26	.89871
20	.96662	.25622*J	1.00000	14.85	-.29627
21	.99051	.24351*J	1.02000	13.81	-.43160
22	.88827	-.05544*J	.89000	-3.57	-.91511
23	.99963	-.02728*J	1.00000	-1.56	-.03018
24	.99793	.06438*J	1.00000	3.69	.14380
25	.93027	.36687*J	1.00000	21.52	.22058

COMPLEX SLACK BUS POWER : .08682 4.37438*J

ITERATION : 2
ITERATION MODE : 2
ACCURACY OBTAINED : 9.333E-02
SOLUTION TIME : .008 SECONDS

LOAD FLOW SOLUTION BY FAST-DECOUPLED METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03176	.09867*J	1.03646	5.46	
2	1.06399	.09975*J	1.06866	5.36	
3	1.04155	.07462*J	1.04422	4.10	
4	.98440	.11271*J	.99083	6.53	
5	.96994	.27442*J	1.00801	15.80	
6	1.03372	.06372*J	1.03568	3.53	
7	1.01300	.03078*J	1.01347	1.74	
8	.94632	.05375*J	.94785	3.25	
9	.97334	-.09257*J	.97773	-5.43	
10	1.03697	.07292*J	1.03953	4.02	
11	.90308	-.08060*J	.90667	-5.10	
12	.97281	-.06737*J	.97514	-3.96	
13	1.04631	.01568*J	1.04643	.86	
14	.94824	-.08753*J	.95227	-5.27	
15	.92606	.11542*J	.93323	7.10	
16	1.03469	-.04922*J	1.03586	-2.72	
17	.93053	.03764*J	.93129	2.32	
18	1.03826	.25869*J	1.07000	13.99	-.40213
19	1.04557	.09635*J	1.05000	5.26	.19289
20	.96662	.25622*J	1.00000	14.85	.76930
21	.99051	.24351*J	1.02000	13.81	-.03372
22	.88827	-.05544*J	.89000	-3.57	-.22909
23	.99963	-.02728*J	1.00000	-1.56	-.11317
24	.99793	.06438*J	1.00000	3.69	-.16737
25	.93027	.36687*J	1.00000	21.52	.17031

COMPLEX SLACK BUS POWER : .09053 -.05063*J

ITERATION : 3
ITERATION MODE : 1
ACCURACY OBTAINED : 2.904E-02
SOLUTION TIME : .016 SECONDS

LOAD FLOW SOLUTION BY FAST-DECOUPLED METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03342	.07939*J	1.03646	4.39	
2	1.06447	.09452*J	1.06866	5.07	
3	1.04265	.05713*J	1.04422	3.14	
4	.98580	.09968*J	.99083	5.77	
5	.97356	.26130*J	1.00801	15.02	
6	1.03418	.05578*J	1.03568	3.09	
7	1.01328	.01962*J	1.01347	1.11	
8	.94695	.04130*J	.94785	2.50	
9	.97162	-.10918*J	.97773	-6.41	
10	1.03720	.06958*J	1.03953	3.84	
11	.90140	-.09766*J	.90667	-6.18	
12	.97230	-.07442*J	.97514	-4.38	
13	1.04631	.01579*J	1.04643	.86	
14	.94629	-.10658*J	.95227	-6.43	
15	.92797	.09889*J	.93323	6.08	
16	1.03479	-.04710*J	1.03586	-2.61	
17	.93084	.02920*J	.93129	1.80	
18	1.03967	.25295*J	1.07000	13.67	-.40247
19	1.04550	.09708*J	1.05000	5.30	.15337
20	.97020	.24232*J	1.00000	14.02	.76641
21	.99348	.23109*J	1.02000	13.09	-.03603
22	.88599	-.08440*J	.89000	-5.44	-.19115
23	.99965	-.02656*J	1.00000	-1.52	-.11330
24	.99885	.04793*J	1.00000	2.75	-.16739
25	.93536	.35370*J	1.00000	20.71	.16991

COMPLEX SLACK BUS POWER : .10870 -.08239*J

ITERATION : 4
ITERATION MODE : 2
ACCURACY OBTAINED : 9.621E-03
SOLUTION TIME : .008 SECONDS

LOAD FLOW SOLUTION BY FAST-DECOUPLED METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03275	.07934*J	1.03579	4.39	
2	1.06446	.09452*J	1.06865	5.07	
3	1.04236	.05711*J	1.04393	3.14	
4	.98583	.09968*J	.99086	5.77	
5	.97367	.26134*J	1.00814	15.02	
6	1.03238	.05568*J	1.03388	3.09	
7	1.01325	.01962*J	1.01344	1.11	
8	.94396	.04117*J	.94486	2.50	
9	.96206	-.10810*J	.96811	-6.41	
10	1.03710	.06957*J	1.03943	3.84	
11	.89845	-.09734*J	.90371	-6.18	
12	.96733	-.07404*J	.97016	-4.38	
13	1.04634	.01579*J	1.04646	.86	
14	.93932	-.10579*J	.94526	-6.43	
15	.92727	.09881*J	.93252	6.08	
16	1.03526	-.04712*J	1.03633	-2.61	
17	.93167	.02923*J	.93213	1.80	
18	1.03967	.25295*J	1.07000	13.67	-.40240
19	1.04550	.09708*J	1.05000	5.30	.18722
20	.97020	.24232*J	1.00000	14.02	.77769
21	.99348	.23109*J	1.02000	13.09	-.02927
22	.88599	-.08440*J	.89000	-5.44	-.18314
23	.99965	-.02656*J	1.00000	-1.52	-.11438
24	.99885	.04793*J	1.00000	2.75	-.16534
25	.93536	.35370*J	1.00000	20.71	.16914

COMPLEX SLACK BUS POWER : .12456 -.05275*J

ITERATION : 5
ITERATION MODE : 1
ACCURACY OBTAINED : 3.719E-03
SOLUTION TIME : .016 SECONDS

LOAD FLOW SOLUTION BY FAST-DECOUPLED METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03289	.07741*J	1.03579	4.29	
2	1.06448	.09430*J	1.06865	5.06	
3	1.04247	.05507*J	1.04393	3.02	
4	.98600	.09797*J	.99086	5.67	
5	.97406	.25989*J	1.00814	14.94	
6	1.03239	.05546*J	1.03388	3.07	
7	1.01328	.01816*J	1.01344	1.03	
8	.94400	.04027*J	.94486	2.44	
9	.96197	-.10889*J	.96811	-6.46	
10	1.03712	.06920*J	1.03943	3.82	
11	.89825	-.09914*J	.90371	-6.30	
12	.96732	-.07416*J	.97016	-4.38	
13	1.04634	.01572*J	1.04646	.86	
14	.93917	-.10717*J	.94526	-6.51	
15	.92745	.09711*J	.93252	5.98	
16	1.03526	-.04712*J	1.03633	-2.61	
17	.93171	.02787*J	.93213	1.71	
18	1.03972	.25276*J	1.07000	13.66	-.40238
19	1.04555	.09662*J	1.05000	5.28	.18791
20	.97056	.24086*J	1.00000	13.94	.77753
21	.99383	.22959*J	1.02000	13.01	-.02926
22	.88563	-.08810*J	.89000	-5.68	-.17806
23	.99965	-.02655*J	1.00000	-1.52	-.11438
24	.99894	.04595*J	1.00000	2.63	-.16533
25	.93589	.35229*J	1.00000	20.63	.16914

COMPLEX SLACK BUS POWER : .13334 -.05406*J

ITERATION : 6
ITERATION MODE : 2
ACCURACY OBTAINED : 6.177E-04
SOLUTION TIME : .008 SECONDS

LOAD FLOW SOLUTION BY FAST-DECOUPLED METHOD

26-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.03276	.07740*J	1.03566	4.29	
2	1.06437	.09429*J	1.06854	5.06	
3	1.04237	.05506*J	1.04382	3.02	
4	.98591	.09796*J	.99076	5.67	
5	.97406	.25989*J	1.00814	14.94	
6	1.03245	.05546*J	1.03393	3.07	
7	1.01319	.01816*J	1.01336	1.03	
8	.94408	.04027*J	.94494	2.44	
9	.96135	-.10882*J	.96749	-6.46	
10	1.03698	.06919*J	1.03928	3.82	
11	.89820	-.09913*J	.90365	-6.30	
12	.96701	-.07414*J	.96985	-4.38	
13	1.04633	.01572*J	1.04645	.86	
14	.93879	-.10713*J	.94489	-6.51	
15	.92734	.09710*J	.93241	5.98	
16	1.03526	-.04712*J	1.03633	-2.61	
17	.93176	.02787*J	.93217	1.71	
18	1.03972	.25276*J	1.07000	13.66	-.40046
19	1.04555	.09662*J	1.05000	5.28	.18716
20	.97056	.24086*J	1.00000	13.94	.77936
21	.99383	.22959*J	1.02000	13.01	-.02930
22	.88563	-.08810*J	.89000	-5.68	-.17803
23	.99965	-.02655*J	1.00000	-1.52	-.11439
24	.99894	.04595*J	1.00000	2.63	-.16458
25	.93589	.35229*J	1.00000	20.63	.16913

COMPLEX SLACK BUS POWER : .13406 -.05132*J

S O L U T I O N

NUMBER OF ITERATIONS : 14
ITERATION MODE : 3
RETURN FLAG : 0
ACCURACY OBTAINED : 1.988E-07
SOLUTION TIME : .082 SECONDS

LOAD FLOW SOLUTION BY FAST-DECOUPLED 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 : 1.443 SECONDS

Example 2

The load flow solution for the test 118-bus power system [8-10] 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 = 2.5 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
```

As in Example 1, the program calls the subroutine PWRDS1 from the package PWRDS to read the data and to form the sparse bus admittance matrix.

```
      PROGRAM LFLOW2 (DATA,OUTPUT,TAPE1=DATA,TAPE6=OUTPUT)
C
C  PROGRAM SOLVES THE LOAD FLOW PROBLEM USING SPARSE MATRIX TECHNIQUES
C  (HARWELL PACKAGE MA28) AND THE FAST-DECOUPLED METHOD (PACKAGE LFLFD)
C
      DIMENSION W(8000)
      EXTERNAL FLOW
      CALL SECOND(TIME1)
      CALL PWRDS1(FLOW,1,6,W,8000,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,NLB,NS,NTL,NZ,NLZ,LW
  111 FORMAT(/" DATA-NAME : ",A10
     1       /" NUMBER OF BUSES :                 ",I4
     2       /" NUMBER OF LOAD-BUSES :            ",I4
     3       /" SLACK-BUS INDEX :                 ",I4
     4       /" NUMBER OF TRANSMISSION-LINES :",I4
     5       /" TOTAL NUMBER OF NON-ZEROS :      ",I4
     6       /" NUMBER OF LOAD NON-ZEROS :       ",I4
     7       /" REMAINING WORKSPACE :          ",I6/)
      IFLAG=-5
      ITL=-1
      VEPS=1.E-6
      MODE=0
      TIMEX=2.5
      CALL LFLFD1(NB,NLB,NZ,NLZ,INDR,INDC,IBT,Y,YS,V,S,W,LW,ITL,VEPS,
     1            TIMEX,MODE,IRET)
      IF(IRET.LT.0) RETURN
      IF(LCH.GT.0) WRITE(LCH,222) IRET,ITL,VEPS,TIMEX
  222 FORMAT(/" RETURN FLAG :                     ",I4
     1       /" NUMBER OF ITERATIONS :            ",I4
     2       /" ACCURACY OBTAINED :           ",1PE10.3
     3       /" SOLUTION TIME :                ",0PF6.3," SECONDS"//)
      CALL PRTRES(NB,NS,INDR,INDC,IBT,Y,YS,V)
      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     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 FAST-DECOUPLED 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
```

DATA-NAME : DATA-118
NUMBER OF BUSES : 118
NUMBER OF LOAD-BUSES : 64
SLACK-BUS INDEX : 118
NUMBER OF TRANSMISSION-LINES : 179
TOTAL NUMBER OF NON-ZEROS : 463
NUMBER OF LOAD NON-ZEROS : 134
REMAINING WORKSPACE : 5631

RETURN FLAG : 0
NUMBER OF ITERATIONS : 16
ACCURACY OBTAINED : 8.320E-07
SOLUTION TIME : 1.104 SECONDS

LOAD FLOW SOLUTION BY FAST-DECOUPLED METHOD

118-BUS POWER SYSTEM

BUS	COMPLEX BUS VOLTAGE		POLAR BUS VOLTAGE		GENERATOR REACTIVE POWER
1	1.01894	-.53317*J	1.15000	-27.62	3.89210
2	.90457	-.41632*J	.99577	-24.71	
3	.97421	-.46325*J	1.07874	-25.43	
4	1.00294	-.45178*J	1.10000	-24.25	3.92318
5	.96995	-.34642*J	1.02996	-19.65	
6	.84080	-.32103*J	.90000	-20.90	-2.18445
7	.84084	-.32927*J	.90301	-21.39	
8	1.05545	-.27227*J	1.09000	-14.47	4.88271
9	1.00777	-.12664*J	1.01569	-7.16	
10	.91932	.03526*J	.92000	2.20	-3.23178
11	.88545	-.36761*J	.95873	-22.55	
12	.84524	-.33715*J	.91000	-21.75	-4.87989
13	.86779	-.38081*J	.94767	-23.69	
14	.86690	-.35191*J	.93561	-22.09	
15	.89371	-.37707*J	.97000	-22.88	.47955
16	.85078	-.34505*J	.91809	-22.08	
17	.92353	-.33918*J	.98385	-20.17	
18	.89611	-.37132*J	.97000	-22.51	.08225
19	.88394	-.37450*J	.96000	-22.96	-.20050
20	.87961	-.35480*J	.94848	-21.97	
21	.88625	-.32722*J	.94472	-20.27	
22	.90697	-.28664*J	.95119	-17.54	
23	.95244	-.20700*J	.97467	-12.26	
24	.89350	-.17251*J	.91000	-10.93	-1.71569
25	1.09273	-.12629*J	1.10000	-6.59	5.77215
26	1.00678	-.08054*J	1.01000	-4.57	-4.18156
27	.91591	-.31940*J	.97000	-19.22	1.30953
28	.78486	-.27077*J	.83025	-19.03	
29	.81216	-.30288*J	.86680	-20.45	
30	1.01390	-.26884*J	1.04894	-14.85	
31	.85820	-.33151*J	.92000	-21.12	.80163
32	.90510	-.31999*J	.96000	-19.47	.14533
33	.88789	-.36517*J	.96005	-22.36	
34	.87376	-.34661*J	.94000	-21.64	-1.87944
35	.90456	-.36782*J	.97648	-22.13	
36	.90615	-.37323*J	.98000	-22.39	1.86468
37	.91028	-.32965*J	.96813	-19.91	
38	1.00339	-.26348*J	1.03741	-14.71	

39	.88625	-.37394*J	.96191	-22.88	
40	.88842	-.38937*J	.97000	-23.67	-.92470
41	.90911	-.40983*J	.99722	-24.27	
42	1.01017	-.43538*J	1.10000	-23.32	2.42765
43	.88093	-.33100*J	.94106	-20.59	
44	.92423	-.27208*J	.96345	-16.40	
45	.94368	-.23465*J	.97241	-13.96	
46	.98213	-.18822*J	1.00000	-10.85	-.07354
47	1.00126	-.14870*J	1.01224	-8.45	
48	1.00280	-.15987*J	1.01546	-9.06	
49	1.01018	-.14122*J	1.02000	-7.96	.09157
50	.98312	-.16500*J	.99687	-9.53	
51	.94497	-.18913*J	.96371	-11.32	
52	.93326	-.19718*J	.95386	-11.93	
53	.92149	-.19505*J	.94190	-11.95	
54	.93464	-.17016*J	.95000	-10.32	-.77522
55	.92967	-.19548*J	.95000	-11.87	-.04617
56	.93031	-.19241*J	.95000	-11.69	-.29946
57	.94851	-.18720*J	.96680	-11.16	
58	.93544	-.19496*J	.95554	-11.77	
59	.97833	-.15153*J	.99000	-8.80	.69778
60	.98410	-.11510*J	.99080	-6.67	
61	.99703	-.07702*J	1.00000	-4.42	-.13481
62	.99523	-.09761*J	1.00000	-5.60	-.10655
63	1.01480	-.10228*J	1.01994	-5.76	
64	1.01360	-.07527*J	1.01639	-4.25	
65	.99958	-.02909*J	1.00000	-1.67	-5.82166
66	1.04957	-.03016*J	1.05000	-1.65	4.14129
67	1.01785	-.07549*J	1.02065	-4.24	
68	1.03374	-.05091*J	1.03499	-2.82	
69	.94173	-.12144*J	.94953	-7.35	
70	.97109	-.13186*J	.98000	-7.73	-.02040
71	.97375	-.14402*J	.98434	-8.41	
72	.96270	-.18331*J	.98000	-10.78	.31584
73	.97876	-.14875*J	.99000	-8.64	.12920
74	.95035	-.13579*J	.96000	-8.13	-.24463
75	.96123	-.11189*J	.96772	-6.64	
76	.93383	-.10756*J	.94000	-6.57	-.44043
77	1.00988	-.01556*J	1.01000	-.88	.11721
78	.99904	-.02700*J	.99941	-1.55	
79	1.00589	-.01714*J	1.00604	-.98	
80	1.03935	.03677*J	1.04000	2.03	.85560
81	1.00569	.02306*J	1.00596	1.31	
82	.99153	.01056*J	.99158	.61	
83	.98718	.03380*J	.98775	1.96	
84	.98073	.08140*J	.98410	4.74	
85	.98380	.11061*J	.99000	6.41	-.04216
86	.98407	.08812*J	.98801	5.12	
87	1.00547	.09554*J	1.01000	5.43	.07903
88	.97177	.16848*J	.98627	9.84	
89	.96977	.24404*J	1.00000	14.12	-.44582
90	.98131	.13087*J	.99000	7.60	.38645
91	.97107	.13198*J	.98000	7.74	-.18938
92	.97954	.14352*J	.99000	8.34	-.17527
93	.98202	.09062*J	.98619	5.27	
94	.98985	.05327*J	.99128	3.08	
95	.98121	.03092*J	.98170	1.81	
96	.99340	.02174*J	.99364	1.25	
97	1.01167	.02279*J	1.01192	1.29	
98	1.02436	.02279*J	1.02462	1.27	
99	1.00936	.03595*J	1.01000	2.04	-.22817
100	1.01835	.05794*J	1.02000	3.26	.96245
101	.99036	.07847*J	.99346	4.53	
102	.98263	.11998*J	.98993	6.96	
103	1.00982	.01912*J	1.01000	1.08	.39058

104	.96938	-.03471*J	.97000	-2.05	-.15752
105	.95863	-.05118*J	.96000	-3.06	-.62577
106	.95643	-.05889*J	.95825	-3.52	
107	.94431	-.10382*J	.95000	-6.27	-.03307
108	.95949	-.07005*J	.96204	-4.18	
109	.96008	-.07715*J	.96318	-4.59	
110	.96578	-.09042*J	.97000	-5.35	-.45960
111	.97790	-.06409*J	.98000	-3.75	.02020
112	.96891	-.14703*J	.98000	-8.63	.41012
113	.92886	-.34253*J	.99000	-20.24	.32005
114	.90170	-.32404*J	.95816	-19.77	
115	.88835	-.33943*J	.95098	-20.91	
116	.98903	-.14770*J	1.00000	-8.49	.12494
117	.81838	-.35728*J	.89297	-23.58	

COMPLEX SLACK BUS POWER : 4.74123 .39423*J

TOTAL EXECUTION TIME : 2.796 SECONDS

VII. REFERENCES

[1] B. Stott and O. Alsac, "Fast decoupled load flow", IEEE Trans. Power Apparatus and Systems, vol. PAS-93, 1974, pp. 859-869.

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

[3] S.N. Talukdar and F.F. Wu, "Computer-aided dispatch for electric power systems", Proc. IEEE, vol. 69, 1981, pp. 1212-1231.

[4] "Document SOC-D1: MA28, sparse matrix techniques", Group on Simulation, Optimization and Control, Faculty of Engineering, McMaster University, Hamilton, Canada, December 1981. Origin of the package: Harwell Subroutine Library, AERE, Harwell, Oxfordshire, England.

[5] 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", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-255, 1980.

[6] M.S. Sachdeva 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.

[7] "Document SOC-D10: PWRDS, reading and preprocessing power systems data", Group on Simulation, Optimization and Control, Faculty of Engineering, McMaster University, Hamilton, Canada, June 1982.

[8] "Document SOC-D19: D6,D23,D26,D118, data sets for test power systems", Group on Simulation, Optimization and Control, Faculty of Engineering, McMaster University, Hamilton, Canada, July 1982.

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

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

APPENDIX

LISTING OF THE LFLFD PACKAGE

Subroutine	Number of Lines (source text)	Number of Words (compiled code)	Listing from Page
LFLFD1	85	313	31
LFLFDA	135	672	32
LFLFDB	46	240	34
LFLFDC	49	221	35

```
      SUBROUTINE LFLFD1 (NB,NL,NZ,NR,INDR,INDC,IBT,Y,YS,V,S,W,LW,ITEL,VE
     1PS,TIMEL,MODE,IFLAG)
      DIMENSION INDR(1), INDC(1), IBT(1), W(1)
      COMPLEX Y(1),YS(1),V(1),S(1)
C
C  THIS PACKAGE SOLVES LOAD FLOW PROBLEMS USING SPARSE MATRIX TECHNIQUES
C  (HARWELL PACKAGE MA28) AND THE FAST DECOUPLED METHOD.
C
C  LIBRARY : HARWELL PACKAGE MA28
C
C     NB    - NUMBER OF BUSES (EXCLUDING SLACK BUS),
C     NL    - NUMBER OF LOAD BUSES,
C     NZ    - NUMBER OF NON-ZEROS IN BUS ADMITTANCE MATRIX,
C     NLZ   - NUMBER OF NON-ZEROS IN LOAD BUS ADMITTANCE SUBMATRIX,
C     INDR  - ROW INDEX OF SPARSE BUS ADMITTANCE MATRIX,
C     INDC  - COLUMN INDEX OF SPARSE BUS ADMITTANCE MATRIX,
C     IBT   - VECTOR OF BUS TYPES (0 - LOAD BUS, 1 - GENERATOR BUS),
C     Y     - SPARSE BUS ADMITTANCE MATRIX,
C     YS    - VECTOR OF SLACK BUS ADMITTANCES,
C     V     - COMPLEX BUS VOLTAGES (RECTANGULAR MODE),
C     S     - COMPLEX BUS INJECTED POWERS,
C     W     - REAL WORKSPACE,
C     LW    - LENGTH OF THE WORSPACE W,
C     ITEL  - LIMIT OF ITERATIONS,
C     VEPS  - REQUIRED ACCURACY OF SOLUTION,
C     TIMEL - LIMIT OF ITERATION TIME,
C     MODE  - MODE OF OPERATION :
C             0 - EVALUATE AND FACTORIZE APPROXIMATE JACOBIAN MATRICES,
C             1 - PERFORM *P-DELTA* ITERATION (FOR FACTORIZED MATRICES),
C             2 - PERFORM *Q-MOD.V* ITERATION (FOR FACTORIZED MATRICES),
C             3 - PERFORM BOTH ITERATIONS (FOR FACTORIZED MATRICES),
C     IFLAG - RETURN FLAG :
C            -2 - INCORRECT USE,
C            -1 - INCORRECT PARAMETERS,
C             0 - NORMAL RETURN (REQUIRED ACCURACY OBTAINED),
C             1 - LIMIT OF ITERATIONS REACHED,
C             2 - LIMIT OF ITERATION TIME REACHED.
C
      DATA NS1,NS2,NZ1,NZ2,IFL/0,0,0,0,-1/
      IF (MODE.LT.0.OR.MODE.GT.3) GO TO 30
      IF (MODE.EQ.0) GO TO 10
      IFLAG=-2
      IF (NB.NE.NS1.OR.NL.NE.NS2.OR.NZ.NE.NZ1.OR.NR.NE.NZ2.OR.IFL.LT.0)
     1RETURN
      GO TO 20
   10 NBS=NB+1
      NS1=NB
      NS2=NL
      NZ1=NZ
      NZ2=NR
      LICN1=3*NZ1
      LICN2=3*NZ2
      LIRN1=NZ1+NS1+NS1
      LIRN2=NZ2+NS2+NS2
      JNRB=1
      JA1=JNRB+NBS
      JICN1=JA1+LICN1
      JIKP1=JICN1+LICN1
      JA2=JIKP1+5*NS1
      JICN2=JA2+LICN2
      JIKP2=JICN2+LICN2
      JRHS=JIKP2+5*NS2
      JW=JRHS+NS1
      JVM=JW+NS1
      JVA=JVM+NS1
```

```
      MXJ=JVA+NS1
      JIRN2=JRHS
      JIW2=JIRN2+LIRN2
      JW2=JIW2+8*NS2
      JIRN1=JA2
      JIW1=JIRN1+LIRN1
      JW1=JIW1+8*NS1
      MXJ=MAX0(MXJ,JW1+NS1,JW2+NS2)
      J=LW-MXJ
      IF (J.LT.0) GO TO 30
   20 CALL LFLFDA (NB,NL,NZ,NR,INDR,INDC,IBT,Y,YS,V,S,LICN1,LIRN1,LICN2,
     1LIRN2,W(JA1),W(JICN1),W(JIKP1),W(JA2),W(JICN2),W(JIKP2),W(JRHS),W(
     2JW),W(JVM),W(JVA),W(JNRB),W(JIRN1),W(JIW1),W(JW1),W(JIRN2),W(JIW2)
     3,W(JW2),MODE,ITEL,VEPS,TIMEL,IFLAG)
      IF (MODE.EQ.0) IFL=IFLAG
      RETURN
   30 IFLAG=-1
      RETURN
      END
C
C
      SUBROUTINE LFLFDA (NB,NLB,NZ,NZR,INDR,INDC,IBT,Y,YS,V,S,LICN1,LIRN
     11,LICN2,LIRN2,A1,ICN1,IKEEP1,A2,ICN2,IKEEP2,RHS,W,VM,VA,NRB,IRN1,I
     2W1,W1,IRN2,IW2,W2,MODE,ITE,VEPS,TIMEX,IERR)
      DIMENSION INDR(1), INDC(1), IBT(1), ICN1(1), ICN2(1), IKEEP1(1), I
     1KEEP2(1), A1(1), A2(1), RHS(1), W(1), W1(1), W2(1), IW1(1), IW2(1)
     2, NRB(1), VM(1), VA(1), IRN1(1), IRN2(1)
      COMPLEX Y(1),YS(1),V(1),S(1)
C
C  THIS SUBROUTINE IMPLEMENTS THE FAST DECOUPLED ITERATIVE METHOD USING
C  THE HARWELL PACKAGE *MA28* FOR SOLVING THE SPARSE SYSTEMS OF LINEAR
C  EQUATIONS WITH REAL COEFFICIENTS.
C
C  REMARK : THE ARRAYS (RHS,W,VM,VA)
C                   AND    (IRN2,IW2,W2)
C           AS WELL AS (A2,ICN2,IKEEP2,IRN2,IW2,W2)
C                   AND    (IRN1,IW1,W1)
C           SHARE THE SAME WORKSPACE.
C
      COMPLEX VV,CC,PW
      LOGICAL SWITCH
      CALL SECOND (TTIME1)
      IERR=-2
      IF (MODE.NE.0) GO TO 70
C
C       SET ORDERING OF BUSES
C
      L=0
      K=NLB
      DO 20 I=1,NB
      IF (IBT(I).NE.0) GO TO 10
      L=L+1
      NRB(I)=L
      GO TO 20
   10 K=K+1
      NRB(I)=K
   20 CONTINUE
C
C       SET AND FACTORIZE SPARSE MATRICES OF REAL COEFFICIENTS
C
      DO 40 I=1,NB
      J1=1
      IF (I.GT.1) J1=INDR(I-1)+1
      J2=INDR(I)-1
      X=0.0
```

```
      IF (AIMAG(YS(I)).NE.0.0) X=1.0/AIMAG(1.0/YS(I))
      DO 30 J=J1,J2
      KK=INDC(J)
      ICN1(J)=KK
      IRN1(J)=I
      XX=1.0/AIMAG(1.0/Y(J))
      X=X+XX
      A1(J)=XX
   30 CONTINUE
      J=INDR(I)
      ICN1(J)=I
      IRN1(J)=I
      A1(J)=-X
   40 CONTINUE
      U=0.1
      CALL MA28A (NB,NZ,A1,LICN1,IRN1,LIRN1,ICN1,U,IKEEP1,IW1,W1,IFLAG)
      IF (IFLAG.LT.0) RETURN
      L=0
      DO 60 I=1,NB
      K=IBT(I)
      IF (K.NE.0) GO TO 60
      LL=NRB(I)
      J1=1
      IF (I.GT.1) J1=INDR(I-1)+1
      J2=INDR(I)
      DO 50 J=J1,J2
      KK=INDC(J)
      IF (IBT(KK).NE.0) GO TO 50
      L=L+1
      ICN2(L)=NRB(KK)
      IRN2(L)=LL
      A2(L)=-AIMAG(Y(J))
   50 CONTINUE
   60 CONTINUE
      U=0.1
      CALL MA28A (NLB,NZR,A2,LICN2,IRN2,LIRN2,ICN2,U,IKEEP2,IW2,W2,IFLAG
     1)
      IF (IFLAG.LT.0) RETURN
C
C        SET INITIAL VALUES AND CHECK LIMIT OF ITERATIONS
C
   70 IT=0
      CMX=0.0
      CALL SECOND (TTIME2)
      IF (ITE.EQ.0) GO TO 140
C
C        CONVERT VOLTAGES TO POLAR FORM
C
      DO 80 I=1,NB
      VV=V(I)
      VM(I)=CABS(VV)
      VA(I)=ATAN2(AIMAG(VV),REAL(VV))
   80 CONTINUE
C
C        ITERATION LOOP
C
      CORRA=0.0
      CORRM=0.0
      IF (MODE.EQ.1) GO TO 90
      IF (MODE.EQ.2) GO TO 100
   90 IT=IT+1
      CALL LFLFDB (NB,NZ,Y,YS,V,VM,VA,S,INDR,INDC,IBT,NRB,A1,LICN1,ICN1,
     1IKEEP1,RHS,W,CORRA)
      SWITCH=.TRUE.
      IF (IT.EQ.1.AND.(MODE.EQ.0.OR.MODE.EQ.3)) GO TO 100
```

```
      GO TO 110
  100 IT=IT+1
      CALL LFLFDC (NB,NLB,NZR,Y,YS,V,VM,VA,S,INDR,INDC,IBT,NRB,A2,LICN2,
     1ICN2,IKEEP2,RHS,W,CORRM)
      SWITCH=.FALSE.
  110 CMX=AMAX1(CORRA,CORRM)
      CALL SECOND (TTIME2)
      IF (CMX.LE.VEPS) GO TO 160
      IF (TIMEX.LE.0.0) GO TO 120
      IF (TTIME2-TTIME1.GE.TIMEX) GO TO 150
  120 IF (ITE.LT.0) GO TO 130
      IF (IT.GE.ITE) GO TO 140
  130 IF (CORRA.GT.2.0*CORRM) GO TO 90
      IF (CORRM.GT.2.0*CORRA) GO TO 100
      IF (SWITCH) GO TO 100
      GO TO 90
  140 IERR=1
      GO TO 170
  150 IERR=2
      GO TO 170
  160 IERR=0
  170 ITE=IT
      TIMEX=TTIME2-TTIME1
      VEPS=CMX
      RETURN
      END
C
C
      SUBROUTINE LFLFDB (NB,NZ,Y,YS,V,VM,VA,S,INDR,INDC,IBT,NRB,A1,LICN1
     1,ICN1,IKEEP1,RHS,W,CORRA)
      DIMENSION VM(1), VA(1), INDR(1), INDC(1), IBT(1), NRB(1), A1(1), I
     1CN1(1), IKEEP1(1), RHS(1), W(1)
      COMPLEX Y(1),YS(1),V(1),S(1)
C
C  THIS SUBROUTINE DETERMINES RIGHT-HAND-SIDES FOR ARGUMENT CORRECTIONS
C  OF THE FAST DECOUPLED METHOD, SOLVES THE SPARSE SYSTEM OF LINEAR
C  EQUATIONS AND UPDATES THE VOLTAGES.
C
      COMPLEX CURR,DELS,PW
      NBS=NB+1
C
C         SET RIGHT-HAND-SIDES
C
      DO 20 I=1,NB
      J1=1
      IF (I.GT.1) J1=INDR(I-1)+1
      J2=INDR(I)
      CURR=YS(I)*V(NBS)
      DO 10 J=J1,J2
      K=INDC(J)
      CURR=CURR+Y(J)*V(K)
   10 CONTINUE
      PW=V(I)*CONJG(CURR)
      DELS=S(I)-PW
C     IF(IBT(I).NE.0) S(I)=CMPLX(REAL(S(I)),AIMAG(PW))
      RHS(I)=REAL(DELS)/VM(I)
   20 CONTINUE
C
C         SOLVE LINEAR EQUATIONS
C
      CALL MA28C (NB,A1,LICN1,ICN1,IKEEP1,RHS,W,1)
C
C         UPDATE VOLTAGES
C
      CORRA=0.0
```

```
      DO 30 I=1,NB
      V1=VM(I)
      DV=ABS(V1*SIN(RHS(I)))
      V2=VA(I)+RHS(I)
      VA(I)=V2
      IF (DV.GT.CORRA) CORRA=DV
      V(I)=CMPLX(V1*COS(V2),V1*SIN(V2))
   30 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE LFLFDC (NB,NLB,NZR,Y,YS,V,VM,VA,S,INDR,INDC,IBT,NRB,A2,
     1LICN2,ICN2,IKEEP2,RHS,W,CORRM)
      DIMENSION VM(1), VA(1), INDR(1), INDC(1), IBT(1), NRB(1), A2(1), I
     1CN2(1), IKEEP2(1), RHS(1), W(1)
      COMPLEX Y(1),YS(1),V(1),S(1)
C
C  THIS SUBROUTINE DETERMINES RIGHT-HAND-SIDES FOR MODULUS CORRECTIONS
C  OF THE FAST DECOUPLED METHOD, SOLVES THE SPARSE SYSTEM OF LINEAR
C  EQUATIONS AND UPDATES THE VOLTAGES.
C
      COMPLEX CURR,DELS,PW
      NBS=NB+1
C
C        SET RIGHT-HAND-SIDES
C
      DO 20 I=1,NB
      IF (IBT(I).NE.0) GO TO 20
      L=NRB(I)
      J1=1
      IF (I.GT.1) J1=INDR(I-1)+1
      J2=INDR(I)
      CURR=YS(I)*V(NBS)
      DO 10 J=J1,J2
      K=INDC(J)
      CURR=CURR+Y(J)*V(K)
   10 CONTINUE
      PW=V(I)*CONJG(CURR)
      DELS=S(I)-PW
      RHS(L)=AIMAG(DELS)/VM(I)
   20 CONTINUE
C
C        SOLVE LINEAR EQUATIONS
C
      CALL MA28C (NLB,A2,LICN2,ICN2,IKEEP2,RHS,W,1)
C
C        UPDATE VOLTAGES
C
      CORRM=0.0
      DO 30 I=1,NB
      IF (IBT(I).NE.0) GO TO 30
      K=NRB(I)
      DV=ABS(RHS(K))
      IF (DV.GT.CORRM) CORRM=DV
      V1=VM(I)
      V2=V1+RHS(K)
      VM(I)=V2
      V(I)=(V2/V1)*V(I)
   30 CONTINUE
      RETURN
      END
```

SOC-296

LFLFD - A FORTRAN IMPLEMENTATION OF THE FAST DECOUPLED LOAD FLOW TECHNIQUE

J.W. Bandler and W.M. Zuberek

July 1982, No. of Pages: 35

Revised:

Key Words: Load flow analysis, fast decoupled method, power systems analysis

Abstract: LFLFD is a package of subroutines for solving load flow problems by the well-known fast decoupled technique. The method has been described by Stott and Alsac, and is implemented with minor modifications only. Sparse matrix techniques are used to represent the power system's bus admittance matrix as well as the approximate Jacobian matrices required by the method, and the Harwell Package MA28 is called to solve the systems of linear equations with real coefficients. The package and documentation have been developed for the CDC 170/730 system with the NOS 1.4 operating system and the Fortran 4.8508 compiler.

Description: Contains Fortran listing, user's manual. Source deck or magnetic tape available for $100.00. The listing contains 322 lines, of which 95 are comments.

Related Work: SOC-283, SOC-293.

Price: $50.00.