



LUND UNIVERSITY

A Low Order Nonlinear Dynamic Model for Drum Boiler-Turbine-Alternator Units

Bell, R.D.; Åström, Karl Johan

1979

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Bell, R. D., & Åström, K. J. (1979). *A Low Order Nonlinear Dynamic Model for Drum Boiler-Turbine-Alternator Units*. (Technical Reports TFRT-7162). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:

2

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

STATE ESTIMATION IN POWER NETWORKS III
PROGRAM description

A.J.M. van OVERBECK

Report 7404(C) April 1974
Lund Institute of Technology
Division of Automatic Control

STATE ESTIMATION IN POWER NETWORKS III

Program description

A.J.M. van Overbeek

ABSTRACT.

This report gives a description of the programs developed to compare the three state estimation methods. It is intended for future users of these programs.

This work has been done as partial fulfillment of the requirements for the Masters Degree in Electrical Engineering at the Eindhoven University of Technology, Eindhoven, The Netherlands.

Contents

	Page
1. Introduction	3
2. Simulation, general	4
3. The common blocks	8
4. The subroutines and mainprograms	15
5. The programlistings	52
6. References	183

1. Introduction

In order to compare the three methods mentioned in /1/*) 41 sub-routines and four mainprograms were written. The purpose of this report is to give a description of these programs for future users.

First a general description is given of the simulation program and the way the data exchange between the different subroutines is organized in common blocks. After a detailed description of these common blocks all subroutines are presented in chapter 4. The last chapter consists of the program listings and two tables showing the relationships between the various subroutines and programs.

The 41 routines and four programs consist of 5630 source-cards. All programming is done in Fortran on the Univac 1108 of the computing center of Lund University.

*) See chapter 6: References

2. Simulation, general

The purpose of the simulation programs is to compare and test the three different methods on the following:

1. errors in the network model used by the estimator
2. the influence of measurement accuracy and bias
3. the influence of the choice of measurement system

How is this implemented in the simulation program?

Fig. 2.1 gives a simplified flow diagram of the main program. The power demand is chosen as the control variable for the true state, since this is closest to reality. In the data read part (part 1 in fig. 2.1) the load pattern is read. The true state is calculated from the power demand by means of a conventional Newton-Raphson load-flow. The load flow program needs as input net bus injections. Therefore there is first done an economic load dispatch to divide the active demand over the generators. The dispatch program needs generator data, these are supplied in the common block /GEN/. The reactive demand is divided over the generators in the same ratio as the active demand. This means that all generators are operated at the same $\cos \phi$. From the true state the other true variables are calculated. These consist of the types in the following table. These type numbers are used in all routines and programs

type	variable
1	bus voltage
2	line current at A-end of line
3	line current at B-end of line
4	line flow at A-end of line
5	line flow at B-end of line
6	bus injection

Table 2.1 variable types

Flow diagram:

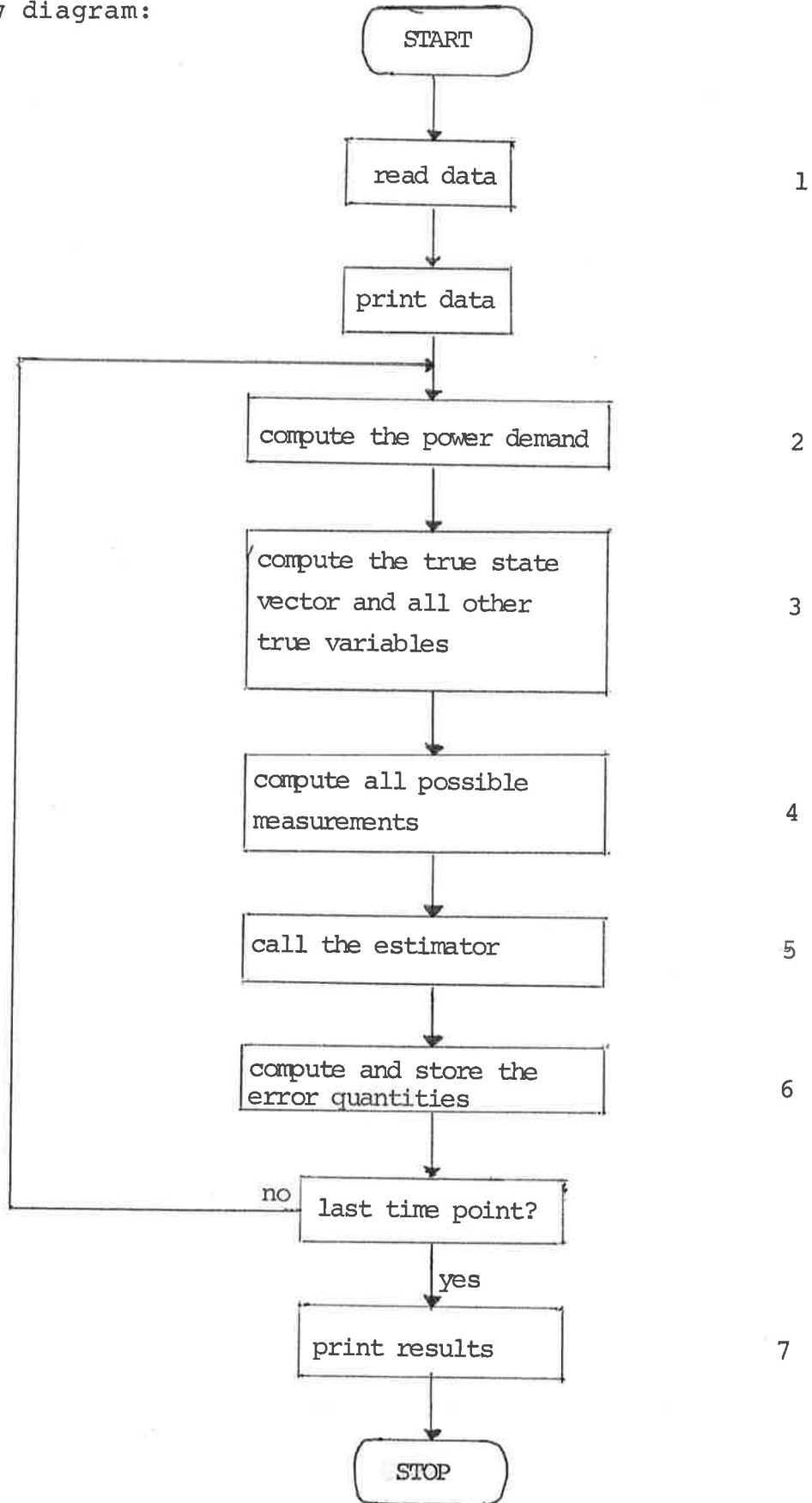


Figure 2.1 Mainprogram for simulation

The true variables are stored in the common block /TRUE/. In all these calculations the true network parameters and structure are used. These are stored in /TNET/. With the true variables given, the measurements are computed (4 in fig. 2.1). All possible measurements are always computed. These are stored in the common block /MSM/. Which measurements are used by the estimator is determined by a mask vector. This has the advantage that for two different measurement systems the same values are generated for those measurements used by both systems. The mask vector and the information on the order of the measurements (necessary for method B) are stored in /MSI/. The measurements are computed by adding bias and noise to the true values. The noise is generated by a random number generator. The noise amplitude for each measurement is given by: $\alpha \times (\text{full scale value}) + \beta \times (\text{true value})$. In this way we can introduce both an absolute (α) and relative (β) error. The α , full scale and β values, the bias and the noise amplitude computed from these quantities are stored in /METT/.

The estimator (5 in fig. 2.1) is not treated in detail here.

It makes use of its own network data: /ENET/.

The estimate and the other variables computed from the estimate are stored in /EST/. All three methods try to minimize the loss function

$$J = \{\underline{y} - g(\underline{x})\}^T W \{\underline{y} - g(\underline{x})\}$$

The weighting factors plus the α , β and full scale values for each measurement used by the estimator are stored in /METE/. So the only coupling between true and estimated values occurs through the common blocks /MSM/ and /MSI/. The error quantities (6 in fig. 2.1): the estimation error, the lineflow error and the measurement index are stored in /EVAL/.

Now we can say which data have to be read in (1)

- true network data
- generator data
- load data
- true meter data
- measurement information

- estimator meter data
- estimator parameters.

The next chapter describes all common blocks while chapter 4 presents all subroutines as they occur in the simulation program and the three simulation programs (one for each method) and the program to plot the results.

The last chapter consists of the program listings.

3. The common blocks

The following dimension parameters are used in the common blocks:

MG maximum number of generators.

MB maximum number of buses.

ML maximum number of lines.

MMB 2MB

MML 2ML

The common block/TNET/: true network parameters.

COMMON /TNET/ NBT,NLT,LTAT(ML),LTBT(ML),YAAT(ML),ZABT(ML),YBBT(ML)

NBT number of buses

NLT number of lines

LTAT(I) A-end of line I is connected to bus LTAT(I)

LTBT(I) B-end of line I is connected to bus LTBT(I)

YAAT(I) shunt admittance at A-end of line I

ZABT(I) series impedance of line I

YBBT shunt admittance at B-end of line I

The common block/TRUE/: the true variables.

COMMON /TRUE/ XT(MB),YT2(ML),YT3(ML),YT4(ML),YT5(ML),YT6(MB)

XT(I) busvoltage I (state)

YT2(I) line current at A-end of line I

YT3(I) line current at B-end of line I

YT4(I) lineflow at A-end of line I

YT5(I) lineflow at B-end of line I

YT6(I) businjection at bus I

all variables are complex.

The common block/GEN/: the generator parameters.

COMMON /GEN/ NG,NGB(MG),PMIN(MG),PMAX(MG)

NG number of generators

NGB(I) generator I is connected to bus I

Al(I) coefficients of the generator

A2(I) cost function: $c = a_1 + a_2 p_{gen}^2$
PMIN(I) min generated active power for generator I
PMAX(I) max generated active power for generator I

The common block/METT/: true meter information.

```
COMMON /METT/ BIAS1(MB),BIAS2(ML),BIAS3(ML),BIAS4(MML),  
X BIAS5(MML),BIAS6(MMB),WN1(MB),WN2(ML),WN3(ML),  
X WN4(MML),WN5(MML),WN6(MMB),ALFT1(MB),ALFT2(ML),  
X ALFT3(ML),ALFT4(MML),ALFT5(MML),ALFT6(MMB),  
X FST1(MB),FST2(ML),FST3(ML),FST4(MML),  
X FST5(MML),FST6(MMB),BETT1(MB),BETT2(ML),BETT3(ML),  
X BETT4(MML),BETT5(MML),BETT6(MMB)
```

BIASX(I) bias for type X measurement
WNX(I) weighting factor for type X measurement noise
ALFTX(I) α value for type X measurement
FSTX(I) full scale value for type X measurement
BETTX(I) β value for type X measurement.

The α , full scale and β values are used by the subroutine CAWN to compute the measurement noise weighting factors. The BIAS and WN values are used by the subroutine ALLMSM to compute the measurements.

The common block/MSM/: all possible measurements.

```
COMMON /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)  
YMX(I) type X measurement
```

The common block/MSI/: measurement system information.

```
COMMON /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),  
X MSK6(MB),NM,NTYP(MM),NMSM(MM)
```

MSKX(I) mask vector for type X measurement

For type 1, 2, and 3 measurements ("modulus measurements") element I = 1 means that the corresponding measurement in /MSM/ is used by the estimator and element I = 0 that the measurement is not used. For type 4, 5, 6 measurements ("power measurements") the meaning of the mask vector elements is given in the following table!

element

- 0 measurement is not used
- 1 only active measurement is used
- 2 only reactive measurement is used
- 3 both active and reactive measurement are used.

NM number of measurements used by the estimator.
active and reactive power measurements are counted separately

NTYP(I) measurement I is of type NTYP(I)

NMSM(I) measurement I is the NMSM(I)-th measurement of type NTYP(I)

examples:

NTYP(I)	NMSMT(I)	
1	5	voltage measurement at bus 5
3	4	line current measurement at B-end of line 4
4	5	active lineflow measurement at A-end of line 3
4	6	reactive lineflow measurement at A-end of line 3

The measurement order information stored in NTYP(I) and NMSM(I) is used by method B.

From this information the subroutine RDMSM computes the mask-vectors.

The common block/RES/: the residues.

```
COMMON /RES/ RES1(MB),RES2(ML),RES3(ML),RES4(MML),RES5(MML),  
X RES6(MMB)
```

RESX(I) The residue $y - g(\underline{x})$ for measurement I of type X.

The common block/METE/: estimator meter information.

```
COMMON /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),  
X             WF6(MMB),ALFE1(MB),ALFE2(ML),ALFE3(ML),ALFE4(MML),  
X             ALFE5(MML),ALFE6(MMB),FSE1(MB),FSE2(ML),FSE3(ML),  
X             FSE4(MML),FSE5(MML),FSE6(MMB),BETE1(MB),BETE2(ML),  
X             BETE3(ML),BETE4(MML),BETE5(MML),BETE6(MMB)
```

WFX(I) weighting factor for measurement I of type X

ALFEX(I) }
FSEX(I) } see corresponding T variables in /METT/
BETEX(I) }

The weighting factors WFX(I) are computed from the α , full scale and β values by the subroutine CAWF.

The common block/ENET/: estimator network data.

```
COMMON /ENET/ XE(MB),YE2(ML),YE3(ML),YE4(ML),YE5(ML),YE6(MB),  
              YAE(ML),YBE(ML),ZABE(ML),YBBE(ML)  
Estimator network data. See /TNET/.
```

The common block/EST/: all estimated variables.

```
COMMON /EST/ XE(MB),YE2(ML),YE3(ML),YE4(ML),YE5(ML),YE6(MB)  
All complex estimated variables. See /TRUE/.
```

The common block/JAC/: the Jacobian.

```
COMMON /JAC/ AJAC1(MB,2),AJAC2(ML,4),AJAC3(ML,4),AJAC4(MML,4),  
X            AJAC5(MML,4),AJAC6(MMB,MMB)
```

The jacobian. Only the non-zero jacobian elements are stored for type 1, 2, 3, 4 and 5 measurements.

type 1: voltage measurement

AJAC1(I,1) element $\frac{\delta V}{\delta e}$

AJAC1(I,2) element $\frac{\delta V}{\delta f}$

type 2 + 3: line current measurement

AJACX(I,1) element $\frac{\delta |I_{ab}|}{\delta e_a}$

AJACX(I,2) element $\frac{\delta |I_{ab}|}{\delta f_a}$

AJACX(I,3) element $\frac{\delta |I_{ab}|}{\delta e_b}$

AJACX(I,4) element $\frac{\delta |I_{ab}|}{\delta f_b}$

X = 2 or 3

type 4 + 5: lineflow measurement

AJACX(I,1) element $\frac{\delta \dots}{\delta e_a}$

X = 4 or 5

AJACX(I,2) element $\frac{\delta \dots}{\delta f_a}$

I odd .. = P_{ab}

AJACX(I,3) element $\frac{\delta \dots}{\delta e_b}$

I even .. = Q_{ab}

AJACX(I,4) element $\frac{\delta \dots}{\delta f_b}$

type 6: bus injection measurement

AJAC6(I,J) all jacobian elements for a bus injection measurement are stored.

The common block/MAT/: Matrices for method A.

COMMON /MAT/ A(MMB,MMB),T(MMB,MMB)

A(I,J) matrix $G^T W G$

T(I,J) triangularized version of A(I,J)

This common block is only used by method A.

The common block/VAR/: diagonal P-matrices for method B.

COMMON /VAR/ COV(MMB),PNEW(MMB)

COV(I) covariance for statevariable I in method B

example: cov(3) = covariance of real part of bus voltage 2.

PNEW(I) new covariance for statevariable I.

This common block is only used by method B.

The common block/MAT/: matrices for method C.

COMMON /MAT/ A(MB,MB),T(MB,MB)

A(I,J) matrix B^TDB of method C

T(I,J) triangularized version of A.

This common block is only used by method C.

The common block/EVL/: evaluation parameters.

COMMON /EVL/ EE(MTMX1),EEM(MTMX1,2),AEE,EEMT(2),EEMMT(3)
X EL(MTMX1),ELM(MTMX1,2),AEL,ELMT(2),ELMMT(3)
X EM(MTMX1),AEM,EMMT(2)

EE(I) estimation error at timepoint I

EEM(I,1) maximum element in EE(I)

EEM(I,2) place of EEM(I,1)

two examples: EEM(I,2) = 4: imaginary part
of bus voltage 2, EEM(I,2) = 5: real part of
bus voltage 3.

AEE average estimation error

EEMT(1) maximum estimation error

EEMT(2) time point of maximum estimation error.

EEMMT(1) maximum element in all estimation errors

EEMMT(2) place of EEMMT(1). See EEM(I,2)

EEMMT(3) time point of EEMMT(1)

EL(I) lineflow error at time point I

ELM(I,1) maximum element in EL(I)

ELM(I,2) place of ELM(I,1)

ELM(I,2) positive means at A-end of line, negative
at B-end of line

examples: $ELM(I,2) = 3$; active flow at A-end of line
2. $ELM(I,2) = -4$: reactive flow at B-end of line 2.

AEL average lineflow error
ELMT(1) maximum lineflow error
ELMT(2) time point of ELMT(1)
ELMMT(1) maximum element in all lineflow errors
ELMMT(2) place of ELMMT(2). See ELM(I,2)
ELMMT(3) time point of ELMMT(3).
EM(I) measurement index at time point I
AEM average measurement index
EMMT(1) maximum measurement index
EMMT(2) time point of EMMT(1)

4. The subroutines and mainprograms.

In this chapter there is given a short description of each subroutine. If necessary a flow diagram and further comments are included. The subroutines are presented about in the order as they occur in the simulation program in the following sections:

- 4.1 The subroutines PRENET and PRTNET.
- 4.2 The data read routines (part 1 in fig. 2.1).
- 4.3 The subroutine CASDB (part 2 in fig. 2.1).
- 4.4 The subroutines for computing all true variables (part 3 in fig. 2.1).
- 4.5 The subroutines for computing the measurements (part 4 in fig. 2.1).
- 4.6 The jacobian routines (used by estimator A and B).
- 4.7 The routines and mainprogram for method A.
- 4.8 The routines and mainprogram for method B.
- 4.9 The routines and mainprogram for method C.
- 4.10 The subroutine EVAL.
- 4.11 The plot program.

Chapter 5 starts with two tables giving the relationships between the various routines and programs.

4.1 The subroutines PRENET and PRTNET.

SUBROUTINE PRENET

Prints the estimator network data stored in the common block /ENET/.

SUBROUTINE PRTNET

Prints the true network data stored in the common block /TNET/.

4.2 The data read routines.

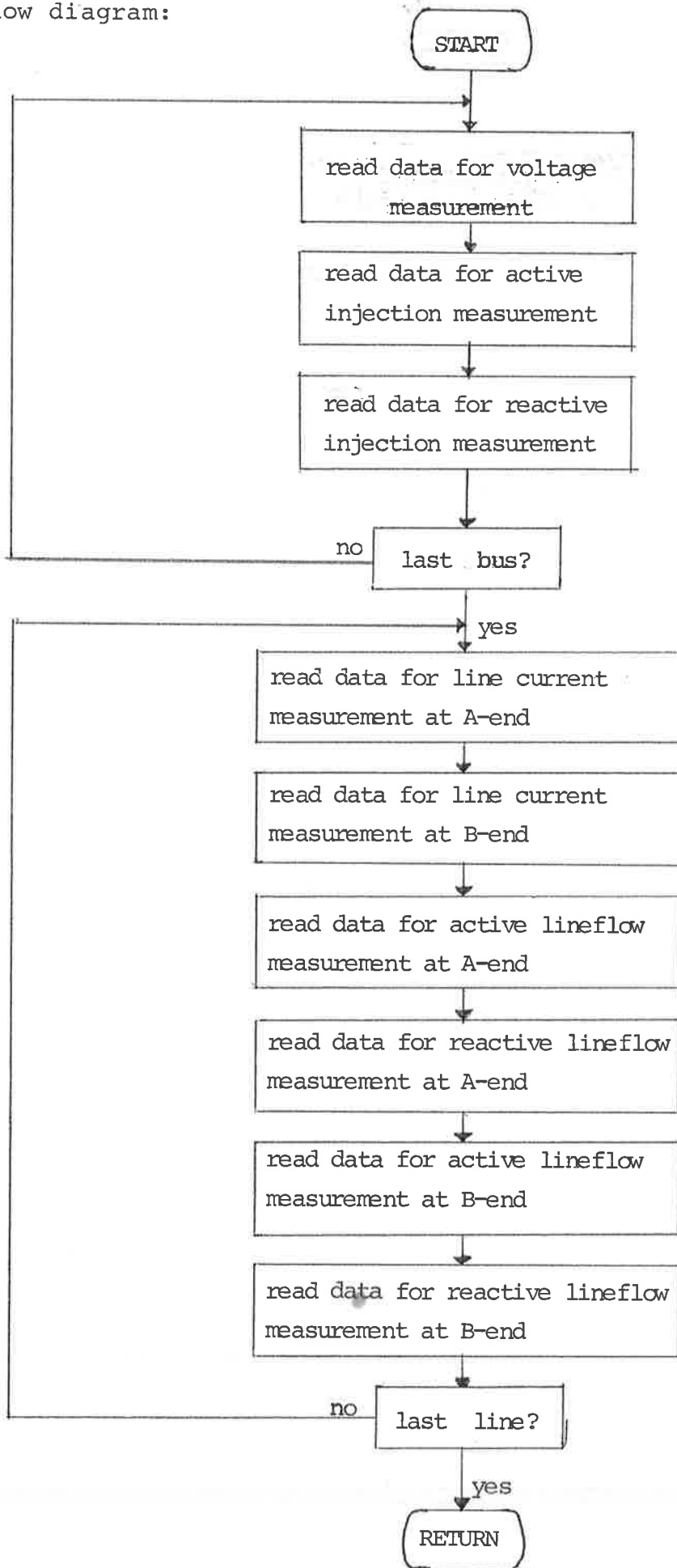
SUBROUTINE RDTNET (IPRINT, IERR)

Reads the true network data into the common block /TNET/ and prints the read data.

SUBROUTINE RDMETT (IPRINT)

Reads for all possible measurements the bias, α , full scale and β values into the common block /METT/ and prints the read data.
flow diagram:

Flow diagram:



SUBROUTINE RDGEN (IPRINT, IERR)

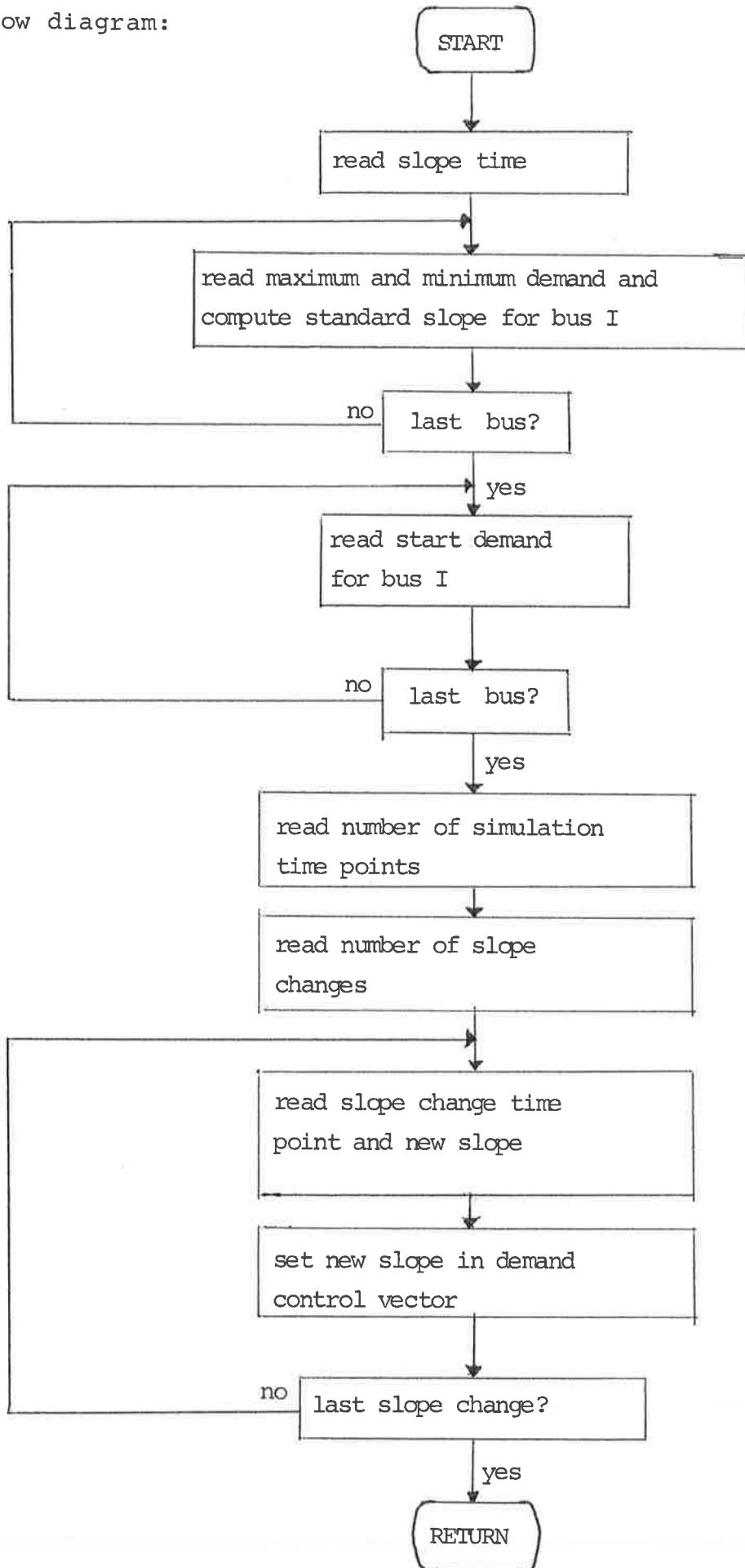
Reads the generator data into the common block /GEN/ and prints the read data.

SUBROUTINE RDLD (IPRINT, IERR)

Reads and prints the demand data necessary for the subroutine CASDB to compute at each time_point the new load demand.

Flow diagram:

Flow diagram:



comments:

The subroutine CASDB computes the new load demand at time point $K + 1$ for bus I as $\text{load}(K+1) = \text{load}(K) + U(K+1) \times \text{standard slope}(I)$. $U(K+1)$ is the demand control vector element. The standard slope for bus I is defined as :

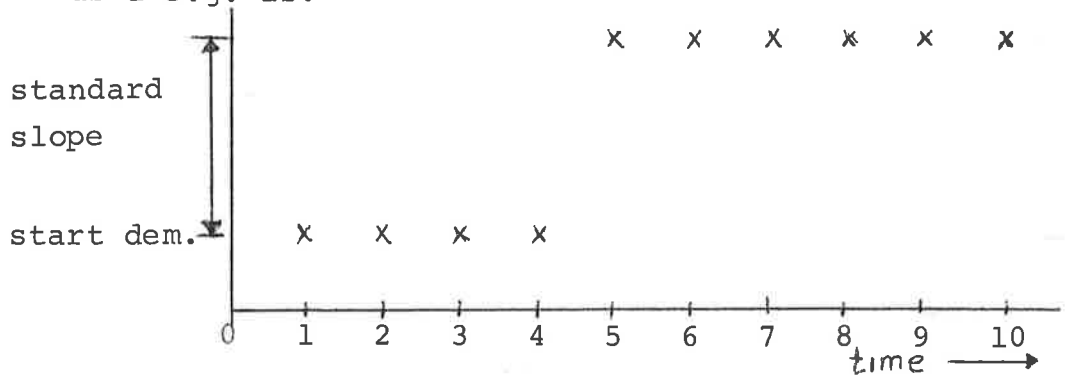
$$\frac{\text{maximum demand} - \text{minimum demand}}{\text{slope time}}$$

example: simulation time = 10
nr of slope changes = 2

slope change time point	new slope
5	1.0
6	0.0

Then U is: 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0

The load for bus 1 e.g. is:



SUBROUTINE RDENET (IPRINT, IERR)

Reads estimator network data into the common block /ENET/ and prints the read data

SUBROUTINE RDMETE (IPRINT)

Reads for all possible measurements the α , full scale and β values into the common block /METE/

flowdiagram: see the subroutine RDMETT

SUBROUTINE RDMSM (IPRINT, IERR)

Reads which of all possible measurements are to be used by the estimator and in which order they are to be processed. This data is read into NM, NTYP(I) and NMSM(I) in the common block /MSI/. From this data the mask vectors are computed. Active and reactive measurements are treated separately. See the description of the common block /MSI/ in the previous chapter.

4.3 The subroutine CASDB.

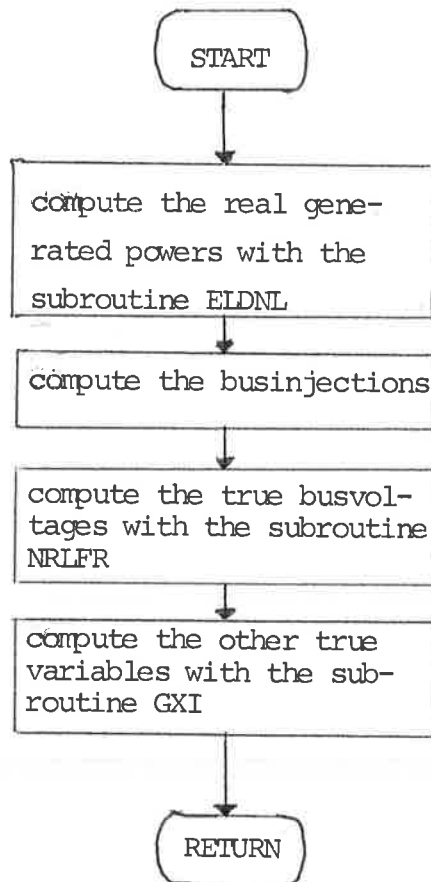
SUBROUTINE CASDB (SDB, SSL, U, IPRINT)

Computes the new load demand for all buses given the old demand, the standard slopes and the demand control vector element. See the subroutine RDL D.

4.4 The subroutines for computing all true variables.

SUBROUTINE TRUEV (SDB, IPRINT, IERR)

Computes all true variables from the load demand flow diagram:



1

comments:

ad 1. the subroutine ELDNL distributes the total active demand over all generators. The total reactive demand is distributed over the generators in the same ratio as the active demand.

SUBROUTINE ELDNL (A1, A2, PMIN, PGEN, PMAX, PDEM, EPS, NG;
IPRINT, IERR)

Performs an economic load dispatch neglecting line losses by minimizing the generator cost function with a Lagrange multiplier method and taking into account minimum and maximum generated power restrictions.

SUBROUTINE NRLFR (NB, NL, LTA, LTB, YAA, ZAB, YBB, SINJ, VB,
EPS, IS, MAXIT, IPRINT, JFAIL)

Performs a conventional Newton-Raphson load flow to compute the true bus voltages. The subroutine TRUEV assumes that bus NB is the slack-bus.

SUBROUTINE DECOM (A, NN, IA, EPS, ISING)

SUBROUTINE SOLVB (B, X, NN, NNB, IA)

Solves the linear system of equations $A \times X = B$ by means of Gauss decomposition of matrix A (DECOM) and forward and backward substitution (SOLVB).

These routines are used in NRLFR and are taken from the program library of the Division of Automatic Control of the Lund Institute of Technology.

SUBROUTINE MPRI (A, M, N, IA, IND, IFORM, IERR)

Prints a matrix.

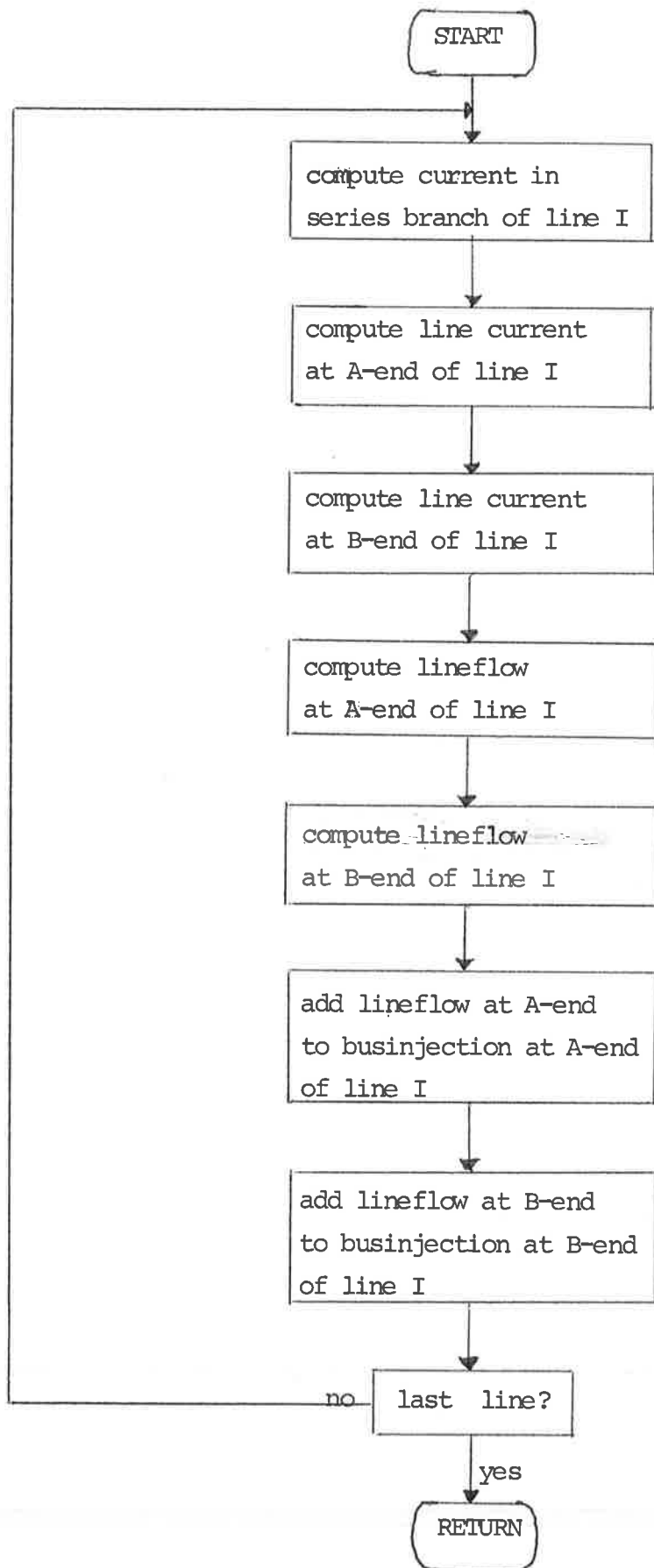
The routine is used in NRLFR and a few other routines. It is also taken from the above mentioned program library.

SUBROUTINE GXT (IPRINT, IERR)

Computes all other true variables from the true state.

Flow diagram:

Flow diagram:



4.5 The subroutines for computing the measurements.

SUBROUTINE CAWN (IPRINT)

Computes for all possible measurements the measurement noise weighting factors WN:

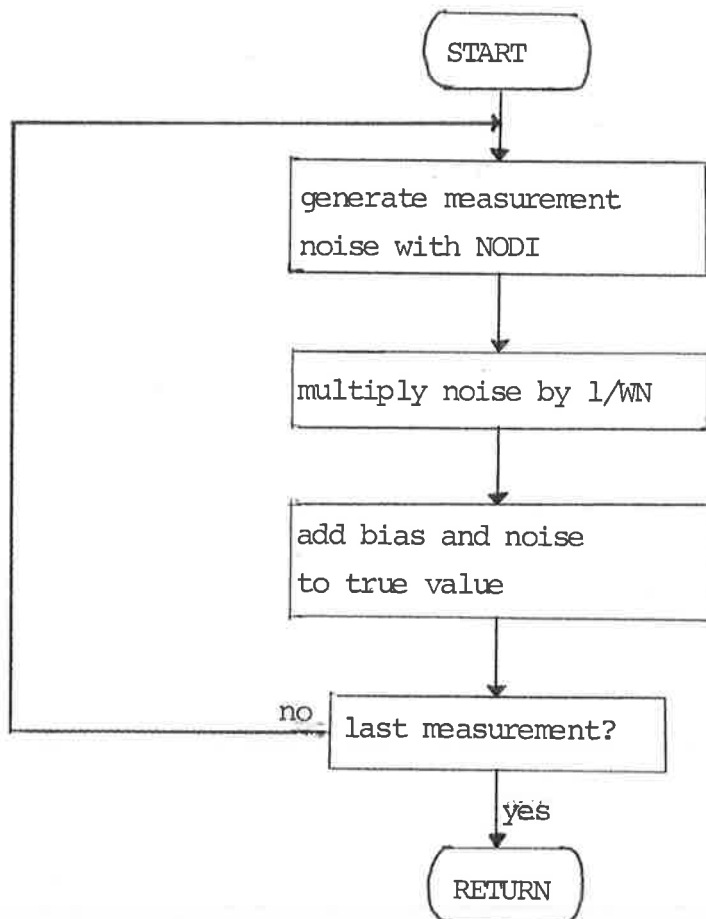
$$WN = \frac{1}{\alpha \times \text{full scale} + \beta \times \text{true value}}$$

The computed weighting factors are stored in the common block /METT/.

SUBROUTINE ALLMSM (NODD, IPRINT)

Computes all possible measurements.

flow diagram:



1

comments: for type 4, 5 and 6 the active and reactive measurements are treated separately.

ad 1. the subroutine NODI needs an odd number to generate random numbers. This is supplied by NODD in the subroutine call to ALLMSM. A good start value for NODD = 19. This must be supplied at the first call to ALLMSM. Successive calls to ALLMSM make use of the value of NODD upon exit of the previous call.

SUBROUTINE NODI (NODD, GAUSS)

function: to provide a random number from a standard normal $N(0,1)$ probability distribution.

The routine is taken from the program library of the Division of Automatic Control and is used in ALLMSM.

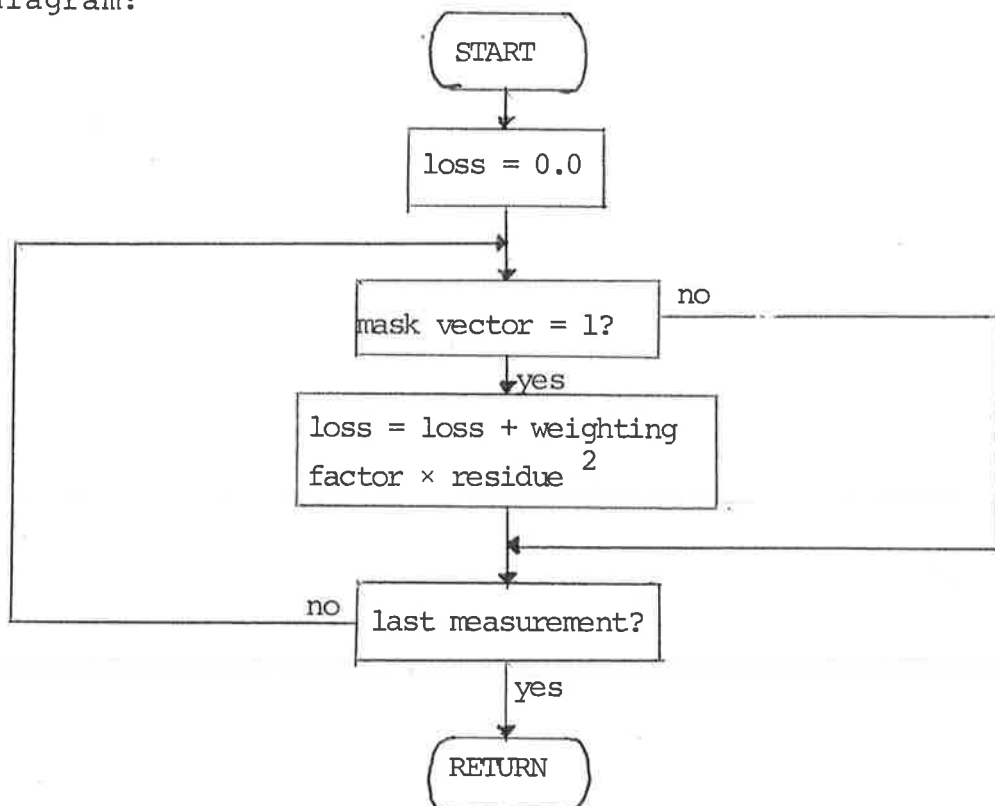
Some routines used by all methods.

FUNCTION ALOSS (IPRINT)

function: to calculate the loss function $\{\underline{y} - g(\underline{x})\}^T W \{\underline{y} - g(\underline{x})\}$

It is assumed that the matrix W is diagonal and that the residues $\underline{y} - g(\underline{x})$ are given,

flow diagram:



comments: for type 4, 5, and 6 active and reactive measurements are treated separately; dependent on the value of the mask vector the active or reactive or both measurements are used in the calculation of the loss function.

SUBROUTINE CARES (IPRINT, IERR)

Computes the residues from the estimated state and the measurements. Before the residues first all estimated variables are computed with the subroutine GXT. The computed residues are stored in the common block /RES/.

SUBROUTINE PRRES

Prints the residues as stored in /RES/, the corresponding used measurements and estimated values.

SUBROUTINE CAWF (IPRINT)

Computes the weighting factors WF for all possible measurements. Only the weighting factors of the measurements processed by the estimator are used in the calculations of the loss function.

SUBROUTINE PRWF

prints all weighting factors as stored in the common block /METE/.

4.6 The_jacobian_routines.

The following four routines make use of the formula's given in the Appendix of /1/ for rectangular coordinates. These routines are used by both method A and B.

SUBROUTINE JACV (U, DU, IR, ID, IERR)

Computes the two non zero jacobian elements for a voltage measurement.

SUBROUTINE JACI (VA, UB, ZAB, YAA, DI, IR, ID, IERR)

Computes the four non zero jacobian elements for a line current measurement.

SUBROUTINE JACLF (M, UA, UB, ZAB, YAA, DP, IRP, DQ, IRQ, ID, IERR)

Computes the non zero jacobian elements for an active, reactive or both lineflow measurements dependent on the value of the mask vector.

SUBROUTINE JACBI (M, IA, DPI, IRP, DQI, IRQ, ID, IERR)

Computes the jacobian row(s) for an active, a reactive or both bus injection measurements, dependent on the value of the mask vector. The row(s) are calculated by computing the jacobian elements of all lineflow measurements at the bus concerned.

SUBROUTINE CAJAC (IPRINT, IERR)

Computes the jacobian for method A using the previous four routines. Only the jacobian elements of the measurements used by estimator A are computed.

SUBROUTINE PRJAC

Prints the jacobian as stored in the common block /JAC/.

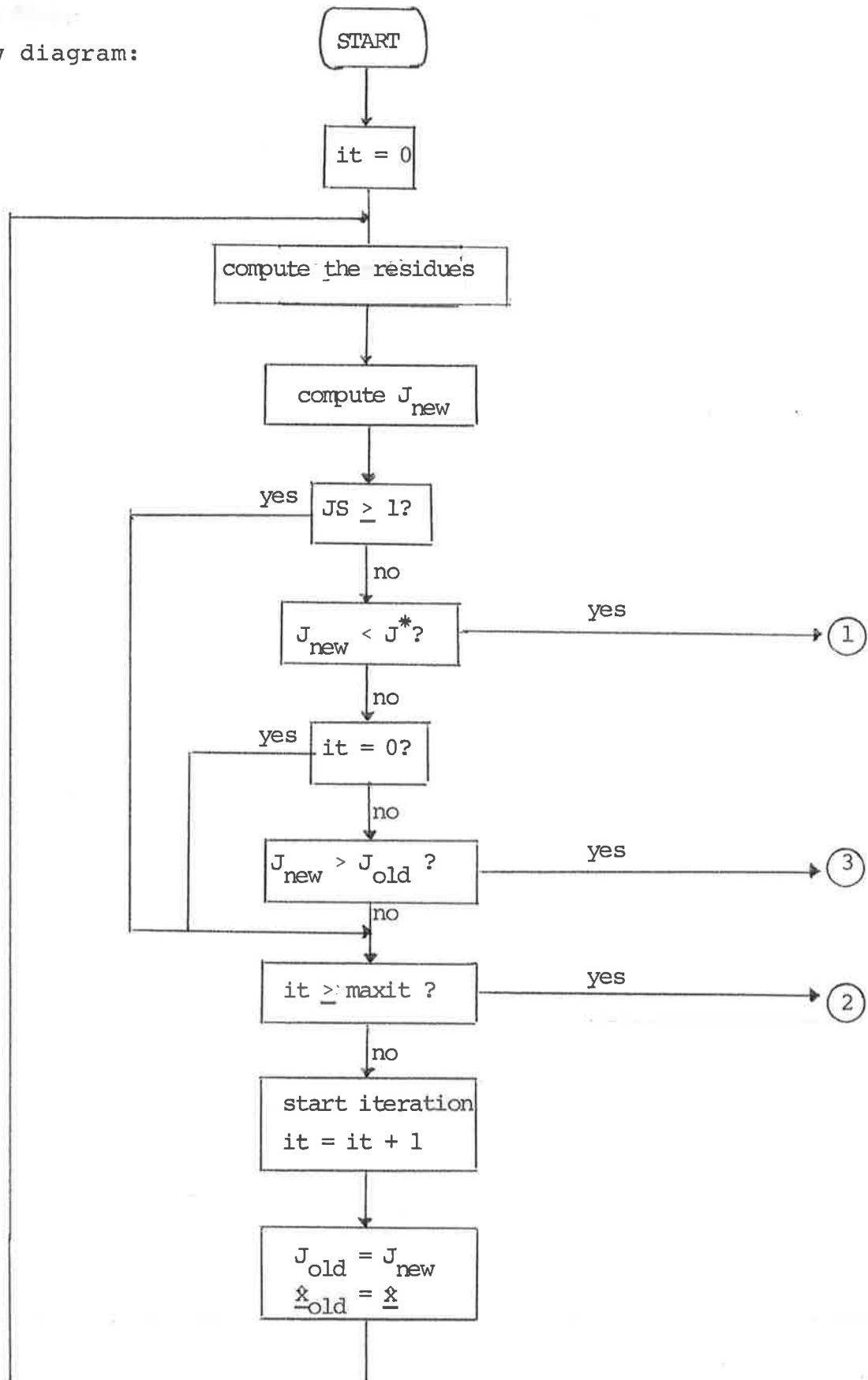
4.7 The routines and mainprogram for method A.

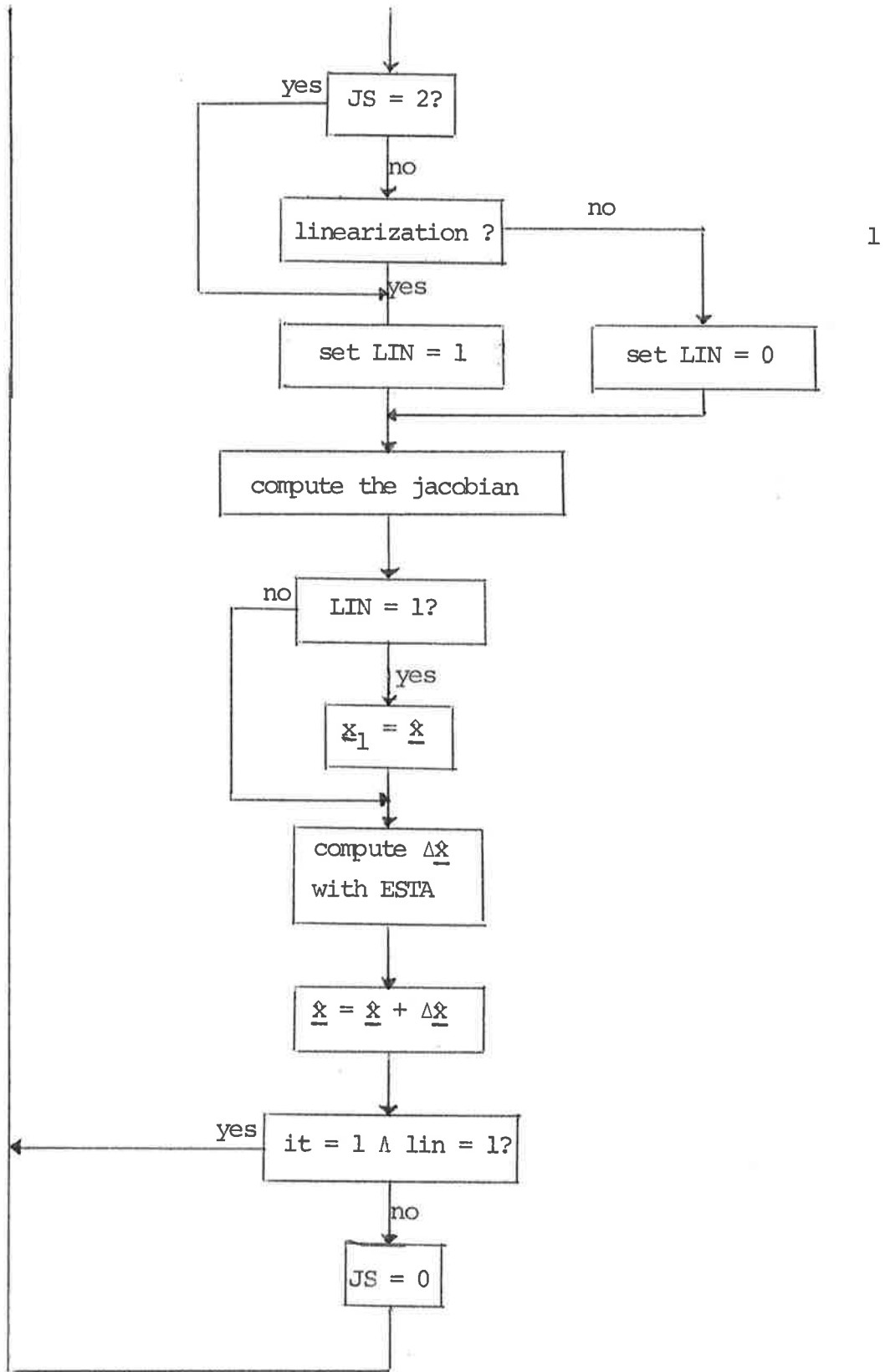
SUBROUTINE ADMA (MAXIT, JS, CRLOSS, XL, CRLIN, IEXIT, TIME, IPRINT)

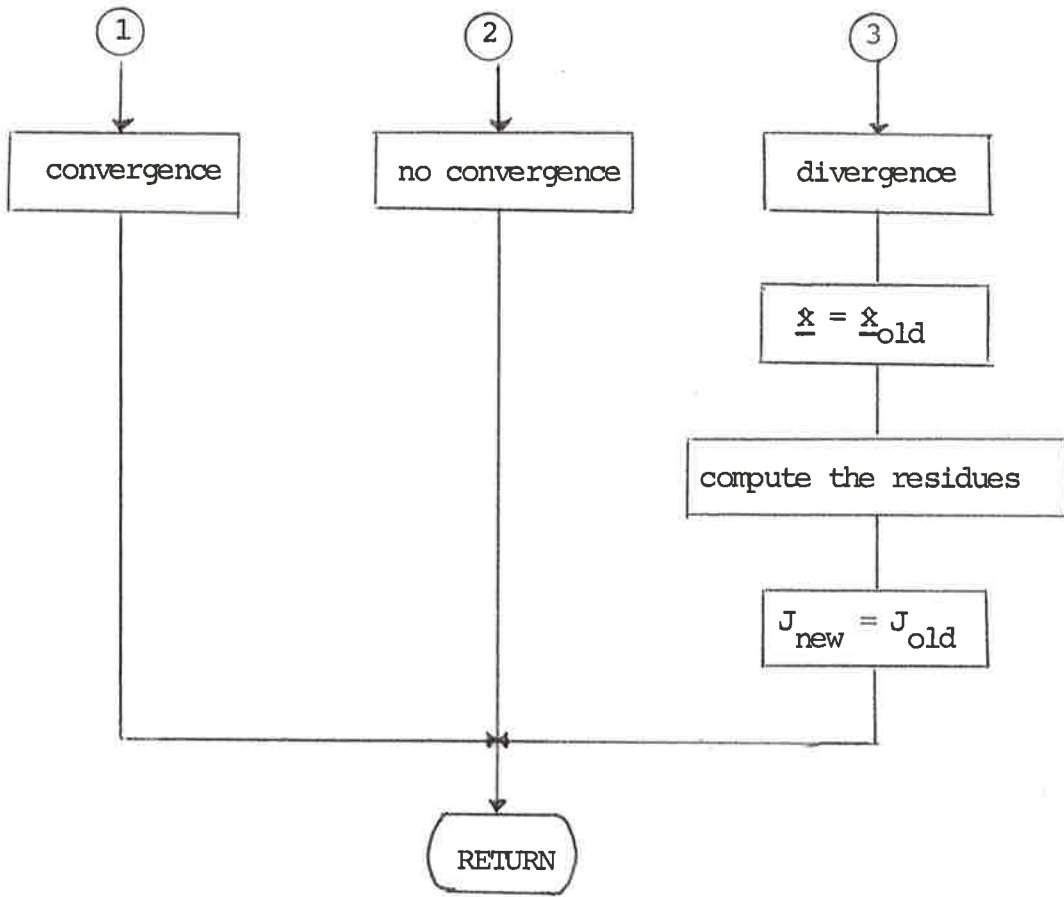
Main routine for method A.

Flow diagram:

Flow diagram:







2

comments:

ad 1. The linearization decision is made by computing the norm of the vector $\underline{x}_1 - \underline{x}$, where \underline{x}_1 is the last linearization point. When this norm comes over a critical value (CRLIN in the subroutine call) a linearization is done around the present estimate.

The estimator can operate in various modes determined by JS

JS	mode
0	Normal operation, iterate till loss function is less than the critical loss J^* (CRLOSS in the subroutine call) or till the maximum number of iterations (MAXIT). Linearize if necessary.
1	Iterate at least once irrespective of the value of the loss function. Upon exit JS = 0.
2	Linearize in two iterations (if maxit permits this). Upon exit (MAXIT \geq 2) is JS = 0.

Table 4-1 Estimator modes for method A.

Examples:

JS = 2, MAXIT = 3: Start up of the estimator from e.g. flat voltage. In the first iteration a linearization is made around the flat voltage profile. In the second around the estimate obtained after the first iteration.

JS = 0, MAXIT = 1: Normal operation under normal conditions.

Ad 2. When after an iteration the loss function has increased the divergence exit is taken. The estimate is set equal to the estimate after the previous iteration. The residues are calculated again and the loss is set equal to the loss after the previous iteration.

SUBROUTINE ESTA (LIN, DX, IPRINT, IERR)

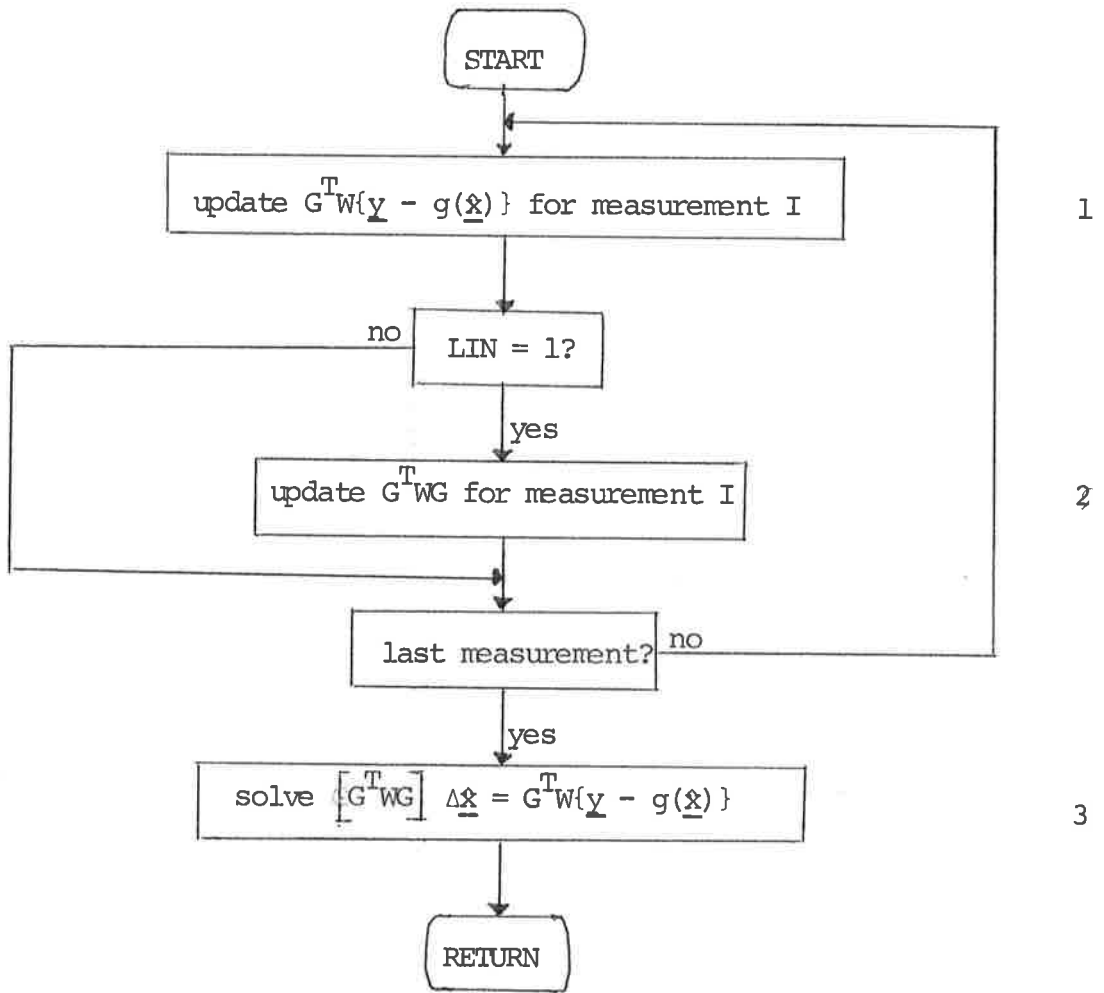
Performs one iteration of method A by solving

$$\left[G^T(\underline{x}_1) W G(\underline{x}_1) \right] \Delta \underline{x} = G^T(\underline{x}) W \{ \underline{y} - g(\underline{x}) \}$$

It is assumed that the jacobian G and the residues $\underline{y} - g(\underline{x})$ are available in the common blocks /JAC/ and /RES/.

Flow diagram:

Flow diagram:



comments:

ad 1. See subroutine UPDBA

ad 2. See subroutine UPDAA

ad 3. This is done by the subroutines DESYM and SOLVS. A mistake in the present realization is that DESYM is also called when there is no linearization ($LIN = 0$). This means that $G^T W G$ is decomposed every time ESTA is called. See the item on computing times in /2/.

SUBROUTINE DESYM (A, G, N, EPS, IRANK, IA)

SUBROUTINE SOLVS (G, B, X, NN, NNB, IA)

Solves the linear system of equations $A \times X = B$ for a symmetric A matrix by triangularization (decomposition) of A (DESYM) and forward and backward substitution (SOLVS). These routines are taken from the program library of the Div. of Automatic Control of the Lund Institute of Technology.

SUBROUTINE UPDAA (INDEX, ELT, NOE, WI, A)

Updates $G^T(\underline{x}_1)WG(\underline{x}_1)$ for a measurement. The NOE non-zero elements of G that have to do with the measurement are given in the vector ELT. The NOE elements in INDEX give the corresponding places of the jacobian elements in the complete jacobian row. WI is the weighting factor for the measurement and A the matrix to be updated.

example:

NOE = 2 INDEX(1) = 1 ELT(1) = 2.0 WI = 2
 INDEX(2) = 3 ELT(2) = 4.0

matrix A before updating

0	0	0
0	0	0
0	0	0

matrix A after updating

8	0	16
0	0	0
16	0	32

In this way multiplications with zero are avoided in computing $G^T W G$, but this implementation also asks much computing time because of the complicated subroutine call. This can be improved by transferring a number of variables (e.g. matrix A) through common blocks.

SUBROUTINE UPDBA (INDEX, ELT, NOE, WI, B, RES)

Updates $G^T(\underline{x}_k)W\{\underline{y} - g(\underline{x}_k)\}$ for a measurement. Since W is diagonal only the places corresponding to non zero jacobian elements are changed in the right hand vector.

INDEX, ELT and NOE are organized in the same way as in UPDAA. WI is again the weighting factor while RES is the residue for the measurement. B is the righthand vector.

example:

NOE = 2 INDEX(1) = 1 ELT(1) = 2.0 WI = 2 RES = 0.25
 INDEX(2) = 3 ELT(2) = 4.0

B before updating

0
0
0

B after updating

1
0
2

Concerning computing time the same can be said as for UPDAA.

Mainprogram MAINA

The three mainprograms for the three methods all are organized as sketched in fig. 2.1. The only differences are the estimator parameters read in part 1 of fig. 2.1 the initialization of the estimator at time point 0 and of course the used estimator: ADMA, ADMB and ADMC respectively.

For method A the estimator parameters that are read are given in the following table:

MAXIT	maximum number of iterations
JS	estimator mode, see ADMA, table 4-1
CRLOSS	the critical value of the loss function: J^*
CRLIN	the linearizing distance: Δx_1

Table 4-2 Estimator parameters for ADMA

Estimator A is initialized by setting the initial estimate at $t = 0$ equal to the true state and linearizing around the true state in one iteration. So we know that the estimator starts with $G^T W G$ computed at the true state.

At the end of the program all evaluation quantities are printed for all time points and the totals. See also the description of the subroutine EVAL and the common block /EVL/.

The data needed for the plotprogram (the estimation error, the lineflow error, and the measurement index for all time points) are written in internal code to a file with internal filename 1. For more details see the programlisting.

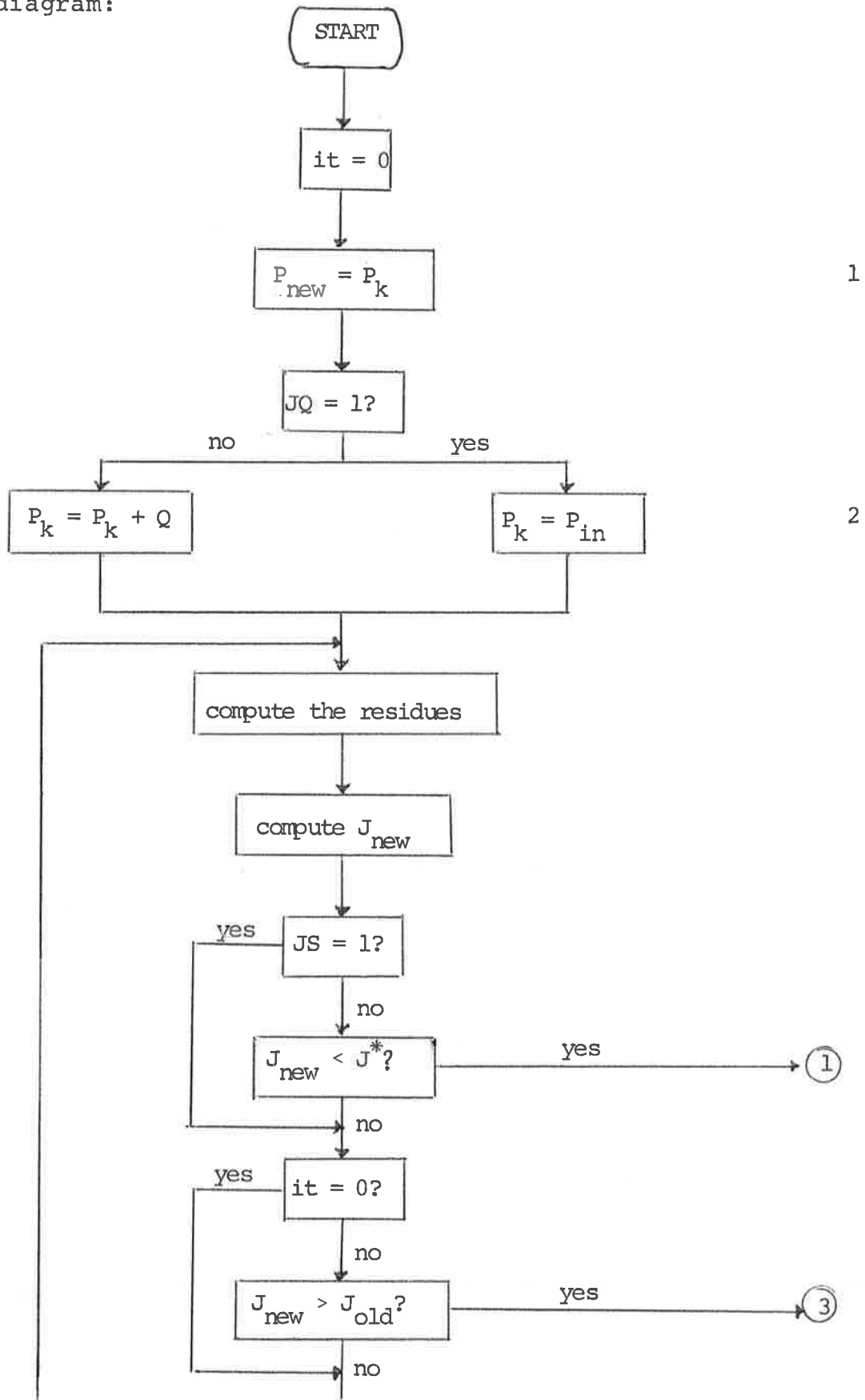
4.8 The routines and mainprogram for method B.

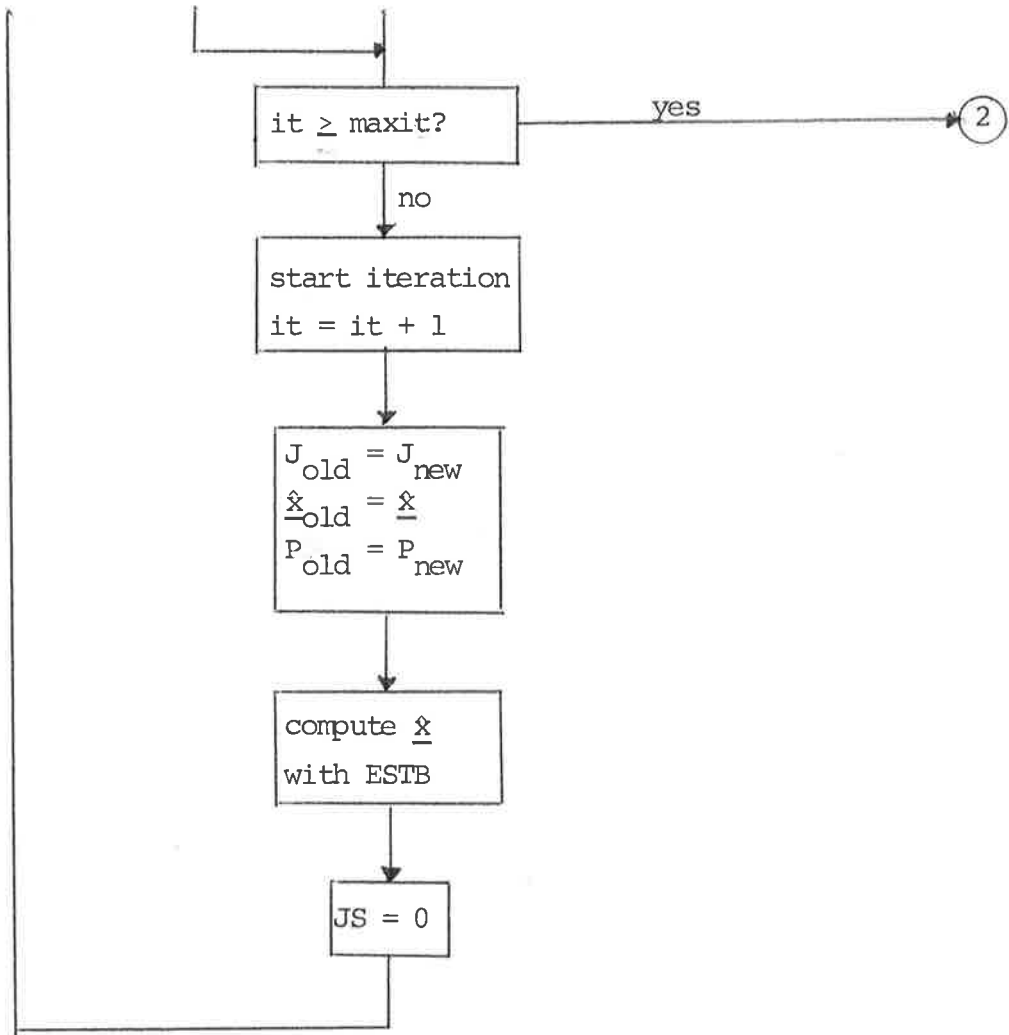
SUBROUTINE ADMB (MAXIT, JS, CRLOSS, JQ, PIN, Q, IEXIT, TIME, IPRINT)

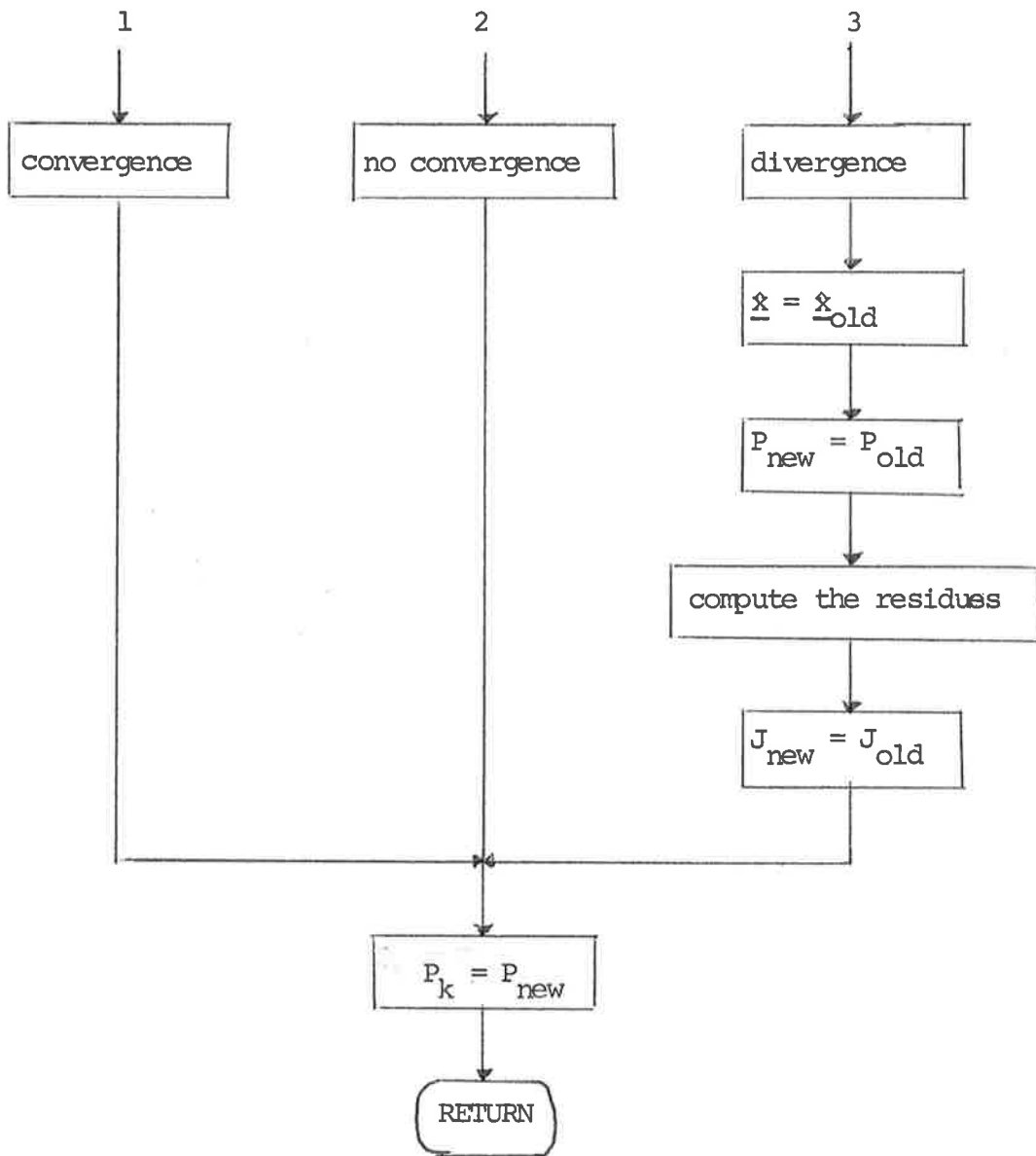
Main routine for method B.

Flow diagram:

Flow diagram:







4

5

comments:

ad 1. P_k and P_{new} are stored in the common block /VAR/ in COV(I) and PNEW(I) respectively.

ad 2. Note that the contents of P_k are not changed during iterations. So each iteration starts with the same P_k .

The estimator can operate in various modes determined by JS and JQ. They are given in the following table.

JS	mode
0	Normal operation, iterate till loss function is less than the critical loss J^* or till the maximum number of iterations.
1	Iterate at least once irrespective of the value of the loss function.

JQ	mode
0	To the final diagonal P-matrix after the previous time point a certain diagonal Q-matrix is added to be used as initial covariance for each iteration.
1	The initial covariance matrix before each iteration has all diagonal elements equal to P_{in} .

Table 4-3 Estimator modes for method B.

Ad 3.-In ESTB P_{new} is computed.

Ad 4. Concerning divergence the same can be said as for ADMA, only here the covariance also has to be restored.

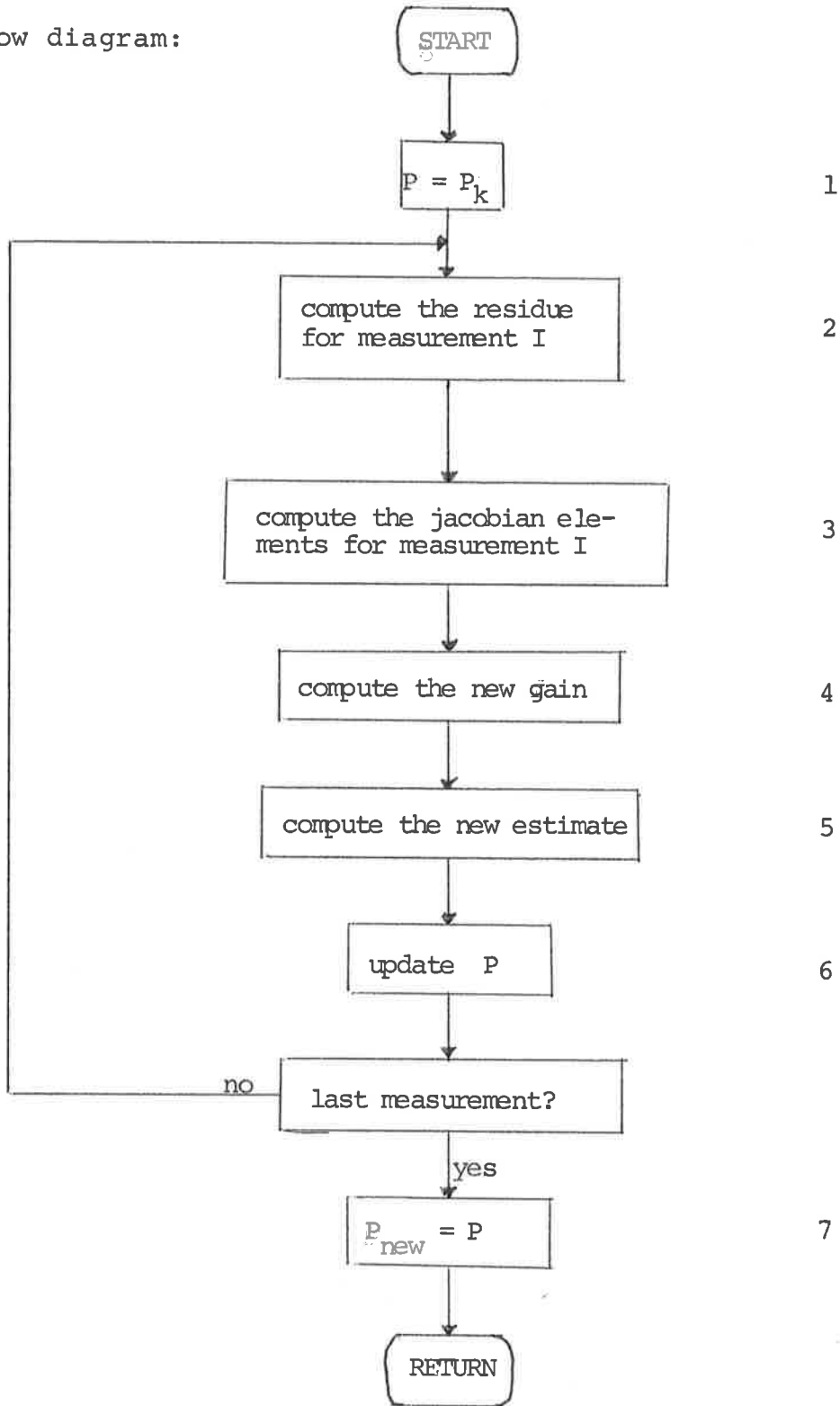
Ad 5. Note that first here P_k (COV(I) in /VAR/) is changed again after 2.

SUBROUTINE ESTB (IPRINT, IERR)

Performs one iteration of method B.

Flow diagram:

Flow diagram:



comments:

Ad 1 + 7. P_k and P_{new} are stored in the common block /VAR/ and PNEW(I) respectively.

Ad 2. The residue for each measurement is computed in ESTB. CARES can not be used in ESTB because the estimate changes during the measurement processing.

Ad 3. The jacobian routines JACV, JACI, JACLF and JACBI are used.

Ad 2 - 6. For the measurement processing use is made of the measurement order information stored in /MSI/.

See the description of this common block.

Ad 4 - 6. For type 2, 3, 4 and 5 measurements only four gain elements are non-zero and only four P- and estimate elements are changed. Therefore there was written a separate subroutine for type 2-- 5 measurements: the subroutine UPDEP.

SUBROUTINE UPDEP (IA, IB, RES, WF, G, P, AK, XE)

See the comments of ESTC.

Mainprogram MAINB.

The general remarks mentioned with MAINA are also valid here.

The estimator parameters read by MAINB are given in the following table:

MAXIT	maximum number of iterations
JS	mode parameters. See ADMB, table 4-3
JQ	mode parameters. See ADMB, table 4-3
CRLOSS	the critical value of the loss function J^*
PIN	the value of all diagonal elements of the initial covariance matrix when $JQ = 0$. See table 4-3
Q(I)	vector of elements to be added to the initial covariance matrix when $JQ = 1$. See table 4-3

Table 4-4 Estimator parameters for ADMB.

The estimator is initialized by setting the initial estimate at $t = 0$ equal to the true state and performing one iteration of ADMB with $JS = 1$, $JQ = 1$ and all elements of the diagonal cov. matrix equal to the value read in PIN.

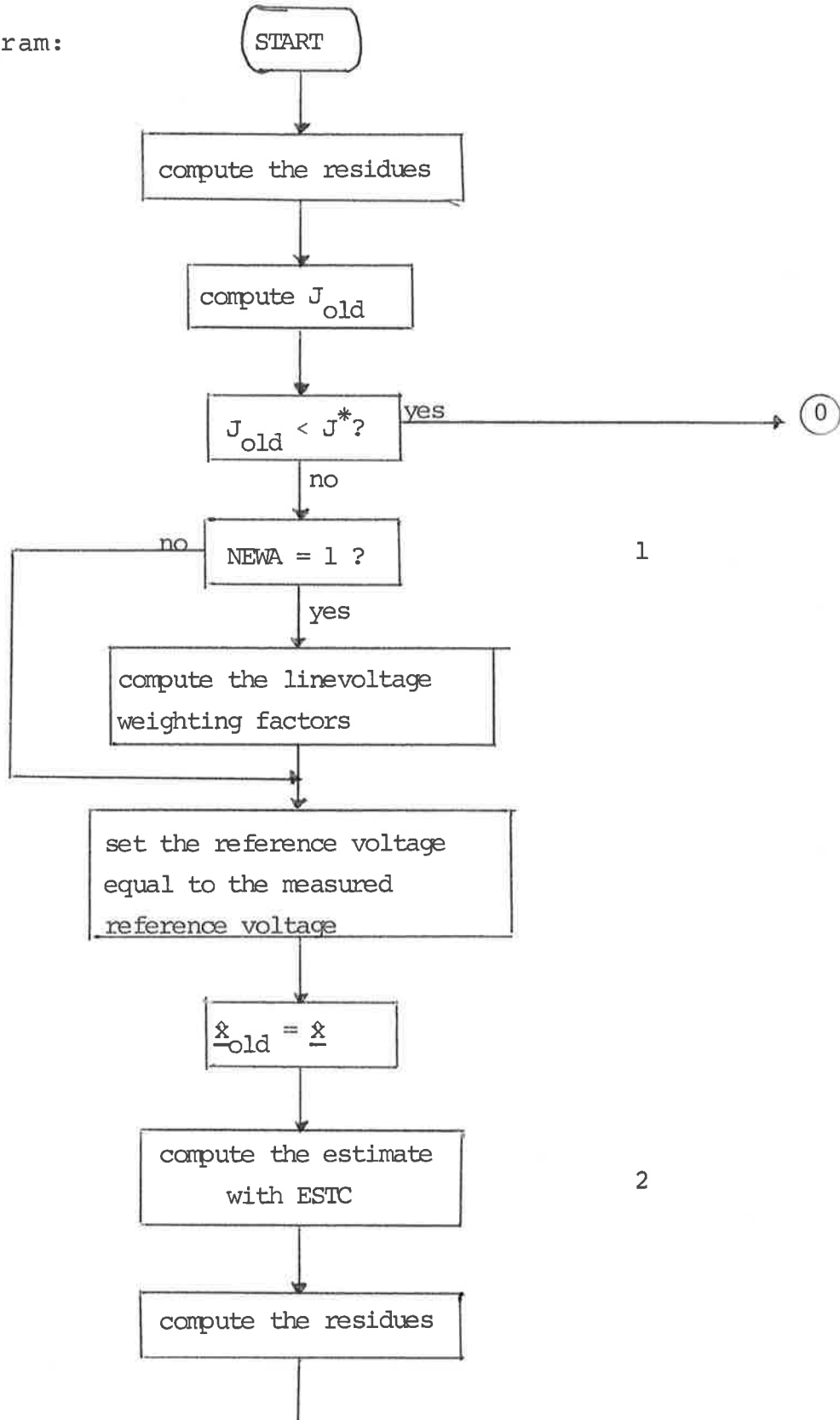
4.9 The routines and mainprogram for method C.

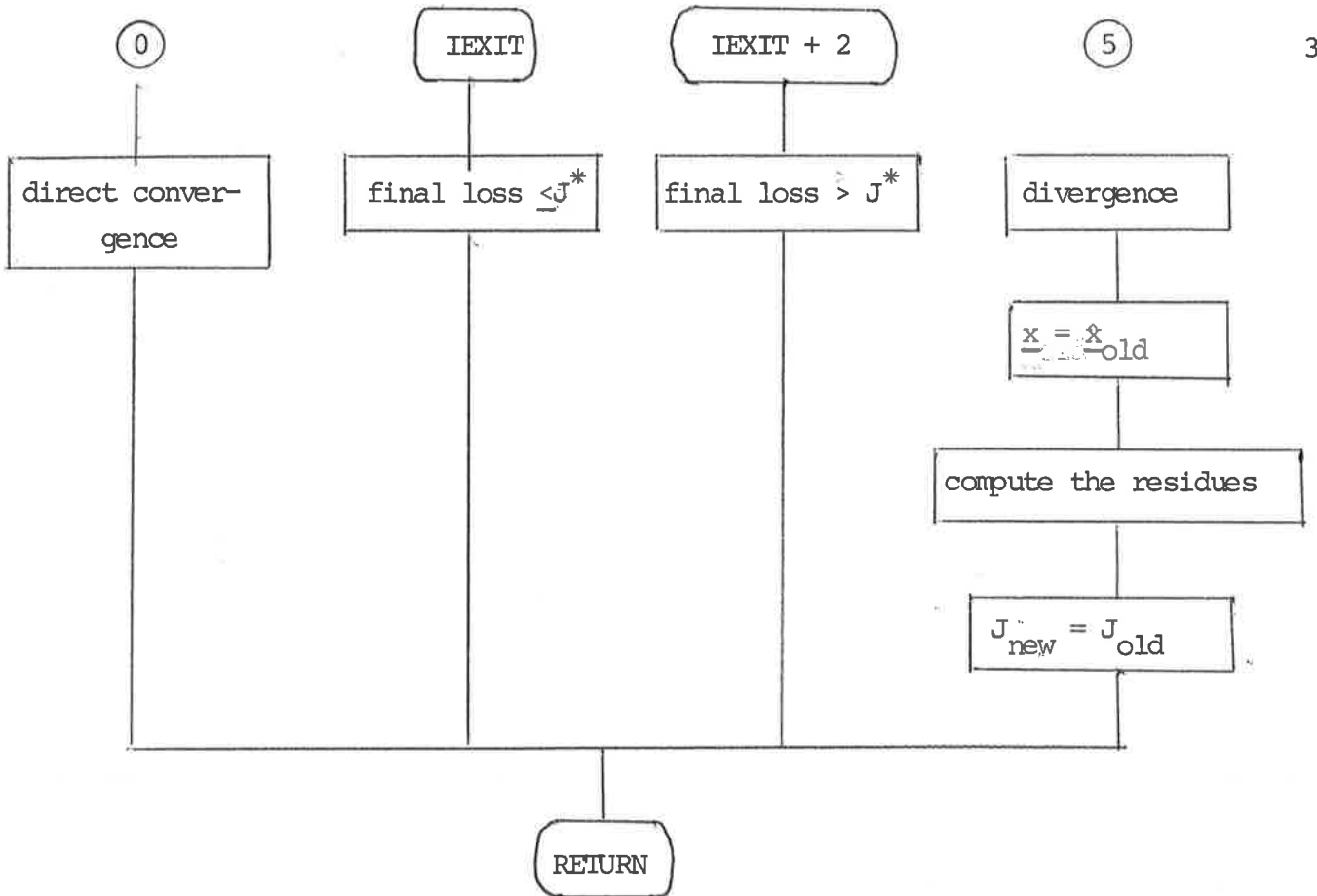
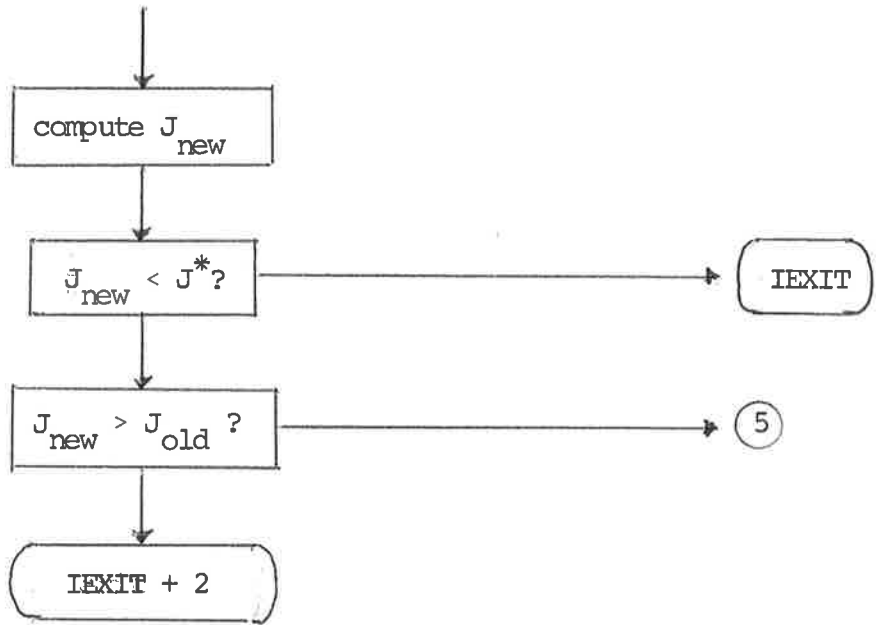
SUBROUTINE ADMC (NEWA, CRLOSS, MAXIT, EPS, IEXIT, TIME, IPRINT)

Main routine for method C.

Flow diagram:

Flow diagram:





comments:

Ad 1 + 2. The subroutine ESTC needs the line voltage weighting factors. These are computed by the subroutine CAD.

Ad 2 + 3. The computation of the estimate in ESTC is done iteratively. This iteration process may or may not convergence. This is represented by IEXIT = 1 or 2 respectively. But this convergence is independent from convergence in the sense of the loss function: $J < J^*$. Therefore there are five exit possibilities. See also ESTC.

SUBROUTINE CAD (DA, DB, IPRINT).

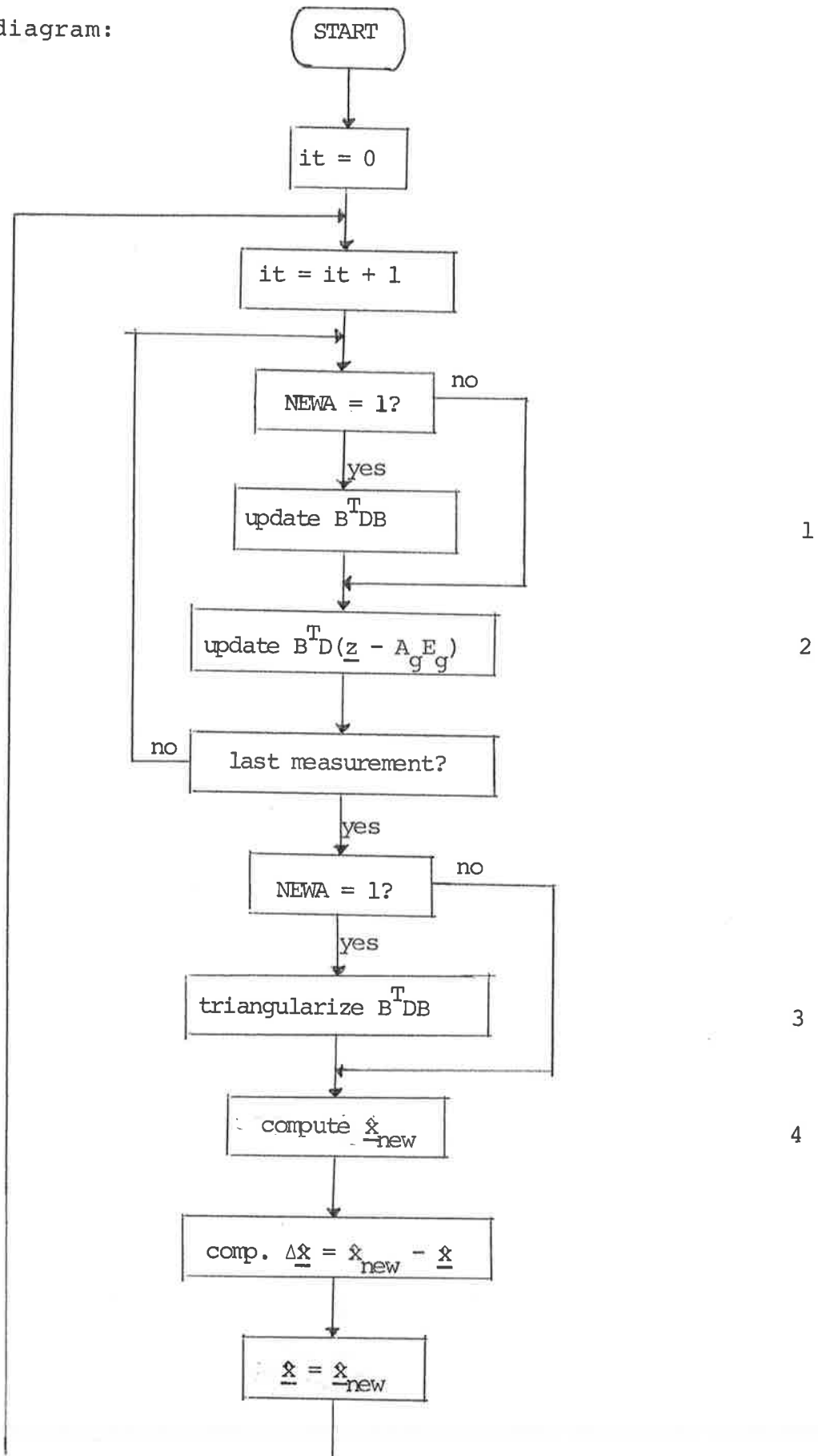
Computes the linevoltage weighting factors for the active line-flow measurements. Only the active weighting factors are used since method C assumes that both the active and reactive measurement are available.

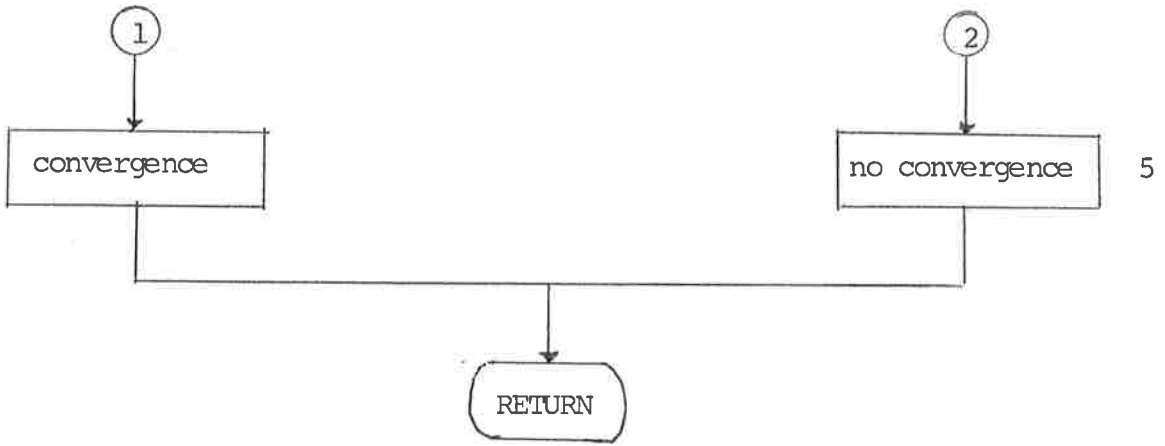
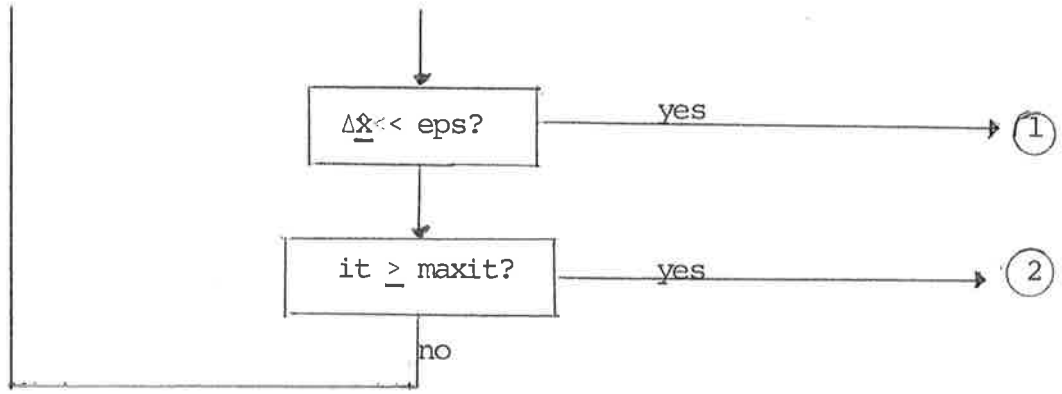
SUBROUTINE ESTC (NR, DA, DB, MAXIT, EPS, NEWA, IEXIT, TIME, IPRINT)

Basis routine for method C.

Flow diagram:

Flow diagram:





Comments:

Ad 1 + 2. The updating of B^TDB is done by the subroutine UPDAC. It is assumed that the linevoltage weighting factors D are already calculated. The updating of the righthand side vector $B^TD(\underline{z} - A_g E_g)$ is done by UPDBC. Note that B and D are real and that $\underline{z} - A_g E_g$ is complex.

Ad 3 + 4. The triangularization is done by DESYM. The solution of $[B^TDB] \hat{E}_{new} = B^TD(\underline{z} - A_g E_g)$ is done by solving the two sets of equations, one for the real parts of \underline{E} and one for the imaginary parts, by SOLVS. Note that the dimension of B^TDB is about half of the corresponding G^TWG -matrix in ESTA.

Ad 5. See the comments on IEXIT in ADMC.

ESTC is written to have more than one reference voltage in E_g but ADMC uses ESTC with only one reference voltage.

SUBROUTINE UPDAC (IA, IB, DA)

Updates the matrix B^TDB for a complex lineflow measurement.

The subroutine functions in the same way as UPDAA. Note that in this routine the matrix to be updated is given in the common block /MAT/.

SUBROUTINE UPDBC (CB, IA, IB, NR, XEA, XEB, ZL, YA, CYM, CLV, DA)

Updates $B^TD(\underline{z} - A_g E_g)$ for a complex lineflow measurement.

The subroutine computes the complex linevoltage CLV from the complex lineflow measurement CYM. It functions in the same way as UPDBA. The only difference is the implementation of the term $-B^TDA_g E_g$. Also here E_g may consist of more than one reference voltage.

See ESTC.

Mainprogram MAINC.

The general remarks in MAINA are also valid here.

The estimator parameters read by MAINC for ADMC are:

NEWA	If NEWA = 1 a new B^{TDB} matrix is computed. This is only necessary when there has been a change in network structure. So normal operation is NEWA = 0.
MAXIT	Maximum number of iterations for ESTC.
CRLOSS	The critical loss J^* for ADMC.
EPS	Convergence criterion for ESTC.

Table 4-5 Estimator parameters for ADMC.

The estimator is initialized by setting the initial estimate at time point 0 equal to the true state. An estimate with method C is done in which B^{TDB} is calculated. Direct convergence is avoided by setting $J^* = 1.0$.

4.10 The subroutine EVAL.

SUBROUTINE EVAL (K, IPRINT)

The subroutine computes the following quantities at a time point:

- the estimation error.
- the maximum element in the estimation error.
- the lineflow error.
- the maximum element in the lineflow error.
- the measurement index.

It updates the following quantities:

- the average estimation error.
- the maximum estimation error.
- the maximum element in all estimation errors.
- the average lineflow error.
- the maximum lineflow error.
- the maximum element in all lineflow errors.
- the average measurement index.
- the maximum measurement index.

All these quantities are stored in /EVL/. The average values AEE, AEL and AEM are updated by adding the corresponding computed errors. To obtain the real average values AEE, AEL and AEM have to be divided by the number of time points. For the organization of the evaluation data see the description of EVL.

4.11 The_plotprogram.

Mainprogram_PLT.

This program is used to plot a number of plots from either MAINA, MAINB or MAINC, It is build around the plotroutine RITA from the program library from the Div. of Automatic Control of the Lund Institute of Technology.

For each plot the data is read from an internal file with internal filename 10+ plotnumber. So if there have to be plotted 4 plots the plotdata have to be supplied to PLT in files with names 11, 12, 13, and 14 respectively.

5. The program listings.

The chapter starts with a table showing the relationships between the routines and mainprograms. The remainder consists of the program listings of the subroutines ordered alphabetically and the listings of the four mainprograms.

<u>ROUTINE/PROGRAM</u>	<u>CALLS:</u>					
ADMA	ALOSS	CAJAC	CARES	ESTA	PRRES	PRWF
ADMB	ALOSS	CARES	ESTB	PRRES	PRWF	
ADMC	ALOSS	CAD	CARES	ESTC	PRRES	
ALLMSM	NODI					
ALOSS	PRRES					
CAD						
CAJAC	JACBI	JACI	JACLF	JACB	PRENET	PRJAC
CARES	GXE	PRRES				
CASDB						
CAWF						
CAWN						
DECOM ¹⁾						
DESYM ¹⁾						
ELDNL						
ESTA	DESYM	MPRI	PRJAC	PRRES	SOLVS	UPDAA
	UPDBA					
ESTB	JACBI	JACI	JACLF	JACV	UPDEP	
ESTC	DESYM	MPRI	SOLVS	UPDAC	UPDBC	
EVAL						
GXE	PRENET					
GXT	PRTNET					
JACBI	JACLF					
JACI						
JACLF						
JACV						
MPRI ¹⁾						
NODI ¹⁾						
NRLFR	DECOM	MPRI	SOLVB			

¹⁾ program library Div. of Autom. Control, Lund Institute of Techn.

Table 5-1 Relationships between routines and programs
(Part 1 of 4).

PRENET							
PRJAC							
PRRES							
PRTNET							
PRWF							
RDENET	PRENET						
RDGEN							
RDLD							
RDMETE							
RDMETT							
RDMSM							
RDTNET	PRTNET						
SOLVB ¹⁾							
SOLVS ¹⁾							
TRUEV	ELDNL	GXT	NRLFR				
UPDAA							
UPDAC							
UPDBA							
UPDBC							
UPDEP							
MAINA	ADMA	ALLMSM	CASDB	CAWF	CAWN	EVAL	
	RDENET	RDGEN	RDLD	RDMETE	RDMETT	RDMSM	
	RDTNET	TRUEV					
MAINB	ADMB	ALLMSM	CASDB	CAWF	CAWN	EVAL	
	RDENET	RDGEN	RDLD	RDMETE	RDMETT	RDMSM	
	RDTNET	TRUEV					
MAINC	ADMC	ALLMSM	CASDB	CAWF	CAWN	EVAL	
	RDENET	RDGEN	RDLD	RDMETE	RDMETT	RDMSM	
	RDTNET	TRUEV					
PLT	PLOT,	PLOTS	RITA				

¹⁾ program library Div. of Autom. Control, Lund Institute of Techn.

Table 5-1 Relationships between routines and programs
(Part 2 of 4).

<u>ROUTINE</u>	<u>CALLED BY:</u>						
ADMA	MAINA						
ADMB	MAINB						
ADMC	MAINC						
ALLMSM	MAINA	MAINB	MAINC				
ALOSS	ADMA	ADMB	ADMC				
CAD	ADMC						
CAJAC	ADMA						
CARES	ADMA	ADMB	ADMC				
CASDB	MAINA	MAINB	MAINC				
CAWF	MAINA	MAINB	MAINC				
CAWN	MAINA	MAINB	MAINC				
DECOM ¹⁾	NRLFR						
DESYM ¹⁾	ESTA	ESTC					
ELDNL	TRUEV						
ESTA	ADMA						
ESTB	ADMB						
ESTC	ADMC						
EVAL	MAINA	MAINB	MAINC				
GXE	CARES						
GXT	TRUEV						
JACBI	CAJAC	ESTB					
JACI	CAJAC	ESTB					
JACLF	CAJAC	ESTB	JACBI				
JACV	CAJAC	ESTB					
MPRI ¹⁾	ESTA	ESTC	NRLFR				
NODI ¹⁾	ALLMSM						
NRLFR	TRUEV						
PRENET	CAJAC	GXE	RDENET				
PRJAC	CAJAC	ESTA					
PRRES	ADMA	ADMB	ADMC	ALOSS	CARES	ESTA	

¹⁾ program library Div. of Autom. Control, Lund Institute of Techn.

Table 5-1 Relationships between routines and programs
(Part 3 of 4).

PRTNET	RDTNET		
PRWF	ADMA	ADMB	
RDENET	<u>MAINA</u>	MAINB	<u>MAINC</u>
RDGEN	MAINA	MAINB	MAINC
RDLD	MAINA	MAINB	MAINC
RDMETE	MAINA	MAINB	MAINC
RDMETT	MAINA	MAINB	MAINC
RDMSM	<u>MAINA</u>	MAINB	MAINC
RDTNET	MAINA	MAINB	MAINC
SOLVB ¹⁾	NRLFR		
SOLVS ¹⁾	ESTA	ESTC	
TRUEV	MAINA	MAINB	MAINC
UPDAA	ESTA		
UPDAC	ESTC		
UPDBA	ESTA		
UPDBC	ESTC		
UPDEP	ESTB		

¹⁾program library Div. of Autom. Control, Lund Institute of Techn.

Table 5-1 Relationships between routines and programs
(Part 4 of 4).

RDMSM	148
RDTNET	151
TRUEV	152
UPDAA	154
UPDAC	155
UPDBA	156
UPDBC	157
UPDEP	159
MAINA	160
MAINB	167
MAINC	174
PLT	181

Table 5-2: The program listings (part 2 of 2).

PROGRAM	page
ADMA	59
ADMB	64
ADMC	69
ALLMSM	73
ALOSS	77
CAD	79
CAJAC	81
CARES	83
CASDB	85
CAWF	86
CAWN	88
ELDNL	91
ESTA	95
ESTB	100
ESTC	106
EVAL	111
GXE	116
GXT	118
JACBI	120
JACI	122
JACLF	123
JACV	125
NRLFR	126
PRENET	131
PRJAC	132
PRRES	135
PRTNET	137
PRWF	138
RDENET	139
RDGEN	140
RDLD	141
RDMETE	144
RDMETT	146

Table 5-2: The program listings (part 1 of 2).

```
      SUBROUTINE ADMA(MAXIT,JS,CRLOSS,XL,CRLIN,IEXIT,TIME,IPRINT)
C
C MAIN ROUTINE FOR ESTIMATORS OF METHOD A
C IT IS ASSUMED THAT BUS NB IS THE SLACK BUS
C
C AUTHOR: TON VAN OVERBEEK 1974-02-08
C
C MAXIT           MAXIMUM NUMBER OF ITERATIONS
C JS=0           NORMAL OPERATION: ITERATE TILL CONVERGENCE
C               OR MAXIT, LINEARIZE IF NECESARRY
C JS=1           ITERATE AT LEAST ONCE, IRRESPECTIVE OF
C               THE VALUE OF THE LOSS FUNCTION
C JS=2           LINEARIZE, IF MAXIT .GE. 2 IN TWO ITERATIONS
C CRLOSS         CONVERGENCE CRITERION: VALUE OF THE LOSS FUNCTION
C XL(*)          VECTOR CONTAINING LAST LINEARIZATION POINT
C CRLIN          LINEARIZATION CRITERION: DISTANCE
C               BETWEEN ESTIMATE AND XL
C IEXIT=-3       ERROR IN ESTA
C IEXIT=-2       ERROR IN CAJAC
C IEXIT=-1       ERROR IN NB OR NL
C IEXIT= 1       CONVERGENCE WITHIN MAXIT ITERATIONS
C IEXIT= 2       NO CONVERGENCE AFTER MAXIT ITERATIONS
C IEXIT= 3       DIVERGENCE: THE LOSS FUNCTION INCREASES
C               IN THE N-TH ITERATION. XE CONTAINS THE
C               ESTIMATE AFTER N - 1 ITERATIONS
C TIME           TOTAL TIME IN MSECS
C IPRINT=5       NO PRINTOUT
C IPRINT=4       INITIAL VALUES OF THE ESTIMATE, THE RESIDUES
C               AND IF JS .EQ. 0 THE LOSS FUNCTION.
C               LINEARIZATION ITERATION NUMBERS.
C               THE RESULTS: NUMBER OF ITERATIONS, TOTAL TIME
C               THE ESTIMATE, THE LOSS FUNCTION AND THE RESIDUES.
C IPRINT=3       SAME + INPUT DATA: MAXIT, JS, CRLOSS, CRLIN
C               AND THE WEIGHTING FACTORS
C IPRINT=2       SAME + IN EACH ITERATION: DX, THE ESTIMATE,
C               THE LOSS FUNCTION AND THE ITERATION TIME
C IPRINT=1       SAME + IN EACH ITERATION THE JACOBIAN AND
C               THE RESIDUES
C IPRINT=0       SAME + IN EACH ITERATION THE A- AND T-MA-
C               TRICES AND THE B-VECTOR
C
C SUBROUTINE REQUIRED
C           ALOSS
C           PRRES
C           CAJAC
C           JACBI
C           JACLF
C           JACI
C           JACLF
C           JACV
C           PRENET
C           PRJAC
```

```
C
C      CARES
C      GXE
C      PNET
C      PRRES
C      ESTA
C      DESYM
C      MPRI
C      PRJAC
C      PRRES
C      SOLVS
C      UPDAA
C      UPDBA
C
C      PRRES
C      PRWF
C
C      PARAMETER MB=10,ML=13
C      PARAMETER MMB=2*MB,MML=2*ML,MBL=MB+MML,MM=3*MBL
C
C      COMPLEX YAA,ZAB,YBB,XE,YE2,YE3,YE4,YE5,YE6,X0(MB),XL(MB),CDX(MB)
C
C      DIMENSION DX(MMB),PR(9)/'(1H0,',0,0,'VERGEN',
1      'CE AFT','ER,IS','11H IT','ERATIO','NS)'/
C
C      INTEGER TIME,TBEG,TEND,TBIT,TEIT,TIT
C
C      COMMON /EST/ XE(MB),YE2(ML),YE3(ML),YE4(ML),YE5(ML),YE6(MB)
X      /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
X      /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
X      /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),
X      WF6(MMB),ALFE1(MB),ALFE2(ML),ALFE3(ML),ALFE4(MML),
X      ALFE5(MML),ALFE6(MMB),FSE1(MB),FSE2(ML),FSE3(ML),
X      FSE4(MML),FSE5(MML),FSE6(MMB),BETE1(MB),BETE2(ML),
X      BETE3(ML),BETE4(MML),BETE5(MML),BETE6(MMB)
C      COMMON /RES/ RES1(MB),RES2(ML),RES3(ML),RES4(MML),RESS(MML),
X      RES6(MMB)
X      /MAT/ A(MMB,MMB),T(MMB,MMB)
X      /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),
X      MSK6(MB),NM,NTYP(MM),NMSM(MM)
X      /JAC/ AJAC1(MB,2),AJAC2(ML,4),AJAC3(ML,4),AJAC4(MML,4),
X      AJAC5(MML,4),AJAC6(MMB,MMB)
C
C      EQUIVALENCE (CDX,DX)
C
C      CHECK NB AND NL
C
C      IF (NB) 20,20,10
10 IF (NB-MB) 40,40,20
20 IEXIT=-1
   WRITE(6,30) NB
30 FORMAT(13H0NB IN ADMA =,15)
   RETURN
C
40 IF (NL) 60,60,50
```

```
50 IF (NL-ML) 80,80,60
60 IEXIT=-1
   WRITE(6,70) NL
70 FORMAT(13HONL IN ADMA =,I5)
   RETURN
```

C
C
C

INITIALIZATION

```
80 IT=0
   IF (IPRINT-5) 90,150,150
90 WRITE(6,100)
100 FORMAT(19HOPRINTOUT FROM ADMA/1X,18(1H*)/
   F15H0INITIAL VALUES/1X,14(1H-))
   IF (IPRINT-4) 110,130,150
110 WRITE(6,120) MAXIT,JS,CRLOSS,CRLIN
120 FORMAT(20H0INPUTDATA: MAXIT =,I5,6H JS =,I2,
   F10H CRLOSS =,F10.5,9H CRLIN =,F10.5)
   CALL PRWF
130 WRITE(6,140)(I,XE(I),I=1,NB)
140 FORMAT(17H0INITIAL ESTIMATE/14H0BUSNR
   F,10HRE(XE)
   F,6HIM(XE)//
   F(I5,5X,2F10.5))
150 TBEG=MSCPU(X)
   TBIT=TBEG
```

C
C
C
C

LAST PART OF THE ITERATION, ALSO USED FOR INITIALIZATION
CALCULATION OF THE RESIDUES AND THE LOSS FUNCTION

```
160 CALL CARES(2,IE)
   IF (IPRINT-4) 170,170,190
170 IF (IT) 180,180,190
180 CALL PRRES
190 IF (IT .EQ. 0 .AND. JS .GT. 0) GO TO 240
   FLOSSN=ALOSS(2)
   IF (IPRINT-4) 200,200,240
200 IF (IT) 220,220,210
210 IF (IPRINT-2) 220,220,240
220 WRITE(6,230) FLOSSN
230 FORMAT(20H0THE LOSS FUNCTION =,F15.5)
240 TEIT=MSCPU(X)
   TIT=TEIT-TBIT
   IF (IPRINT .LE. 2) WRITE(6,250) IT,TIT
250 FORMAT(22H0TIME FOR ITERATION NR,I4,4H IS,I5,6H MSEC5)
```

C
C
C

CONVERGENCE OR MAXIT ?

```
   IF (JS) 260,260,270
260 IF (FLOSSN-ABS(CRLOSS)) 520,264,264
264 IF (IT) 270,270,268
268 IF (FLOSSN-FLOSS0) 270,270,530
270 IF (IT .GE. MAXIT) GO TO 525
```

C

```
C      HERE STARTS THE ITERATION
C
      IT=IT+1
      IF (IPRINT-2) 280,280,300
280  WRITE(6,290) IT
290  FORMAT(13H0ITERATION NR,I5/1X,17(1H*))
300  FLOSS0=FLOSSN
      DO 305 I=1,NB
305  X0(I)=XE(I)
      TBIT=MSCPU(X)
      IF (JS-2) 310,340,340
C
C      LINEARIZATION ?
C
310  XEL=0.0
      DO 320 I=1,NB
320  XEL=XEL+CABS(XE(I)-XL(I))**2
      XEL=SQRT(XEL)
      IF (XEL-ABS(CRLIN)) 330,340,340
330  LIN=0
      GO TO 380
340  LIN=1
      IF (IPRINT-4) 350,350,380
350  WRITE(6,370) IT
370  FORMAT(27H0LINEARIZATION IN ITERATION,I5)
C
C      CALCULATION OF IPRJ(=IPRINT IN THE SUBROUTINE CAJAC)
C      AND IPRA(=IPRINT IN THE SUBROUTINE ESTA)
C
380  IPRJ=IPRINT+1
      IPRA=IPRJ
      IF (IPRINT .EQ. 2) IPRJ=1
      IF (IPRINT .EQ. 0) IPRA=0
C
C      CALCULATION OF THE JACOBIAN
C
      CALL CAJAC(IPRJ,IERR)
      IF (IERR) 410,410,390
390  IEXIT=-2
      WRITE(6,400) IT
400  FORMAT(28H0ERROR IN CAJAC IN ITERATION,I5)
      RETURN
C
C      ESTIMATE DX
C
410  IF (LIN) 440,440,420
420  DO 430 I=1,NB
430  XL(I)=XE(I)
440  CALL ESTA(LIN,DX,IPRA,IERR)
      IF (IERR) 470,470,450
450  IEXIT=-3
      WRITE(6,460) IT
460  FORMAT(27H0ERROR IN ESTA IN ITERATION,I5)
      RETURN
```

```
C
470 DO 480 I=1,NB
480 XE(I)=XE(I)+CDX(I)
      IF (IPRINT-2) 490,490,510
490 WRITE(6,500)(I,XE(I),I=1,NB)
500 FORMAT(13H0THE ESTIMATE/14H0BUSNR
      F,10HRE(XE)
      F,6HIM(XE)//
      F(15,5X,2F10.5))
510 IF (IT .EQ. 1 .AND. LIN .EQ. 1) GO TO 160
      JS=0
      GO TO 160

C
C      END OF ESTIMATION, PRINTOUT OF THE RESULTS
C

520 IEXIT=1
      GO TO 535
525 IEXIT=2
      GO TO 535
530 IEXIT=3
535 TEND=MSCPU(X)
      TIME=TEND-TBEG
      IF (IEXIT .LT. 3) GO TO 540
      PR(2)=' /1H+',
      PR(3)=' 16HDI'
      DO 538 I=1,NB
538 XE(I)=X0(I)
      CALL CARES(2,IE)
      FLOSSN=FLOSS0
      GO TO 548
540 IF (IEXIT .LT. 2) GO TO 542
      PR(2)='3HNO ',
      GO TO 545
542 PR(2)=' /1H+',
545 PR(3)='17HCON'
548 IF (IPRINT-5) 550,990,990
550 WRITE(6,560)
560 FORMAT(19H0ESTIMATION RESULTS/1X,18(1H-))
      WRITE(6,PR) IT
      WRITE(6,570) TIME
570 FORMAT(13H0TOTAL TIME =,I5,6H MSECS)
      WRITE(6,500)(I,XE(I),I=1,NB)
      WRITE(6,580) FLOSSN
580 FORMAT(20H0THE LOSS FUNCTION =,F15.5)
      CALL PRRES

C
990 RETURN
      END
```


SUBROUTINE ADMB(MAXIT,JS,CRLOSS,JQ,PIN,Q,IEXIT,TIME,IPRINT)

MAIN ROUTINE FOR ESTIMATORS OF METHOD B
IT IS ASSUMED THAT BUS NB IS THE SLACK BUS

AUTHOR: TON VAN OVERBEEK 1974-03-05

MAXIT	MAXIMUM NUMBER OF ITERATIONS
JS=0	NORMAL OPERATION: ITERATE TILL CONVERGENCE OR MAXIT
JS=1	ITERATE AT LEAST ONCE, IRRESPECTIVE OF THE VALUE OF THE LOSS FUNCTION
CRLOSS	CONVERGENCE CRITERION: VALUE OF THE LOSS FUNCTION
JQ=1	THE COV MATRIX HAS DIAGONAL ELEMENTS PIN BEFORE EACH ITERATION
JQ=0	THE COV MATRIX HAS DIAGONAL ELEMENTS COV(I,I) + Q(I) BEFORE EACH ITERATION
PIN	VALUE OF ALL DIAGONAL ELEMENTS OF THE INITIAL COV MATRIX WHEN JQ = 0, USUALLY LARGE
Q(*)	VECTOR CONTAINING THE ELEMENTS TO BE ADDED TO THE DIAGONAL ELEMENTS OF THE COV MATRIX WHEN JQ = 1
IEXIT= 1	CONVERGENCE WITHIN MAXIT ITERATIONS
IEXIT= 2	NO CONVERGENCE AFTER MAXIT ITERATIONS
IEXIT= 3	DIVERGENCE: THE LOSS FUNCTION INCREASES IN THE N-TH ITERATION. XE CONTAINS THE ESTIMATE AFTER N - 1 ITERATIONS
IEXIT=-1	ERROR IN NB, NL OR NM
IEXIT=-2	ERROR IN ESTB
TIME	TOTAL TIME IN MSECS
IPRINT=4	NO PRINTOUT
IPRINT=3	THE INITIAL ESTIMATE AND COV.
	THE RESULTS: NUMBER OF ITERATIONS, TOTAL TIME
	THE ESTIMATE AND COV, THE LOSS FUNCTION AND
	THE RESIDUES
IPRINT=2	SAME + INPUT DATA: MAXIT, JS, CRLOSS, JQ, THE WEIGHTING FACTORS AND IF JQ .EQ. 0 PIN ELSE THE Q-VECTOR
IPRINT=1	SAME + AT EACH ITERATION PRINTOUT FROM ESTB
IPRINT=0	SAME + AT EACH ITERATION PRINTOUT FROM ESTB

SUBROUTINE REQUIRED

- ALOSS
- PRRES
- CARES
- GXE
- PRENET
- PRRES
- ESTB
- JACBI
- JACLF
- JACI
- JACLF
- JACV
- UPDEP

```
C          PRRES
C          PRWF
C
C          PARAMETER MB=10,ML=13
C          PARAMETER MMB=2*MB,MML=2*ML,MM=3*(MB+MML)
C
C          COMPLEX YAA,ZAB,YBB,XE,YE2,YE3,YE4,YE5,YE6,X0(MB)
C
C          DIMENSION POLD(MMB),Q(1),PR(9) / ('1H0','0,0','VERGEN',
1          'CE AFT','ER,I5','11H IT','ERATIO','NS') /
C
C          INTEGER TIME,TBEG,TEND,TBIT,TEIT,TIT
C
C          COMMON /EST/ XE(MB),YE2(ML),YE3(ML),YE4(ML),YE5(ML),YE6(MB)
X          /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
X          /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
X          /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),
X          WF6(MMB),ALFE1(MB),ALFE2(ML),ALFE3(ML),ALFE4(MML),
X          ALFE5(MML),ALFE6(MMB),FSE1(MB),FSE2(ML),FSE3(ML),
X          FSE4(MML),FSE5(MML),FSE6(MMB),BETE1(MB),BETE2(ML),
X          BETE3(ML),BETE4(MML),BETE5(MML),BETE6(MMB)
C          COMMON /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),
X          MSK6(MB),NM,NTYP(MM),NMSM(MM)
X          /RES/ RES1(MB),RES2(ML),RES3(ML),RES4(MML),RES5(MML),
X          RES6(MMB)
X          /VAR/ COV(MMB),PNEW(MMB)
C
C          CHECK NB, NL AND NM
C
C          IF (NB) 20,20,10
10 IF (NB-MB) 40,40,20
20 IEXIT=-1
   WRITE(6,30) NB
30 FORMAT(13H0NB IN ADMB =,I5)
   RETURN
C
C          40 IF (NL) 60,60,50
50 IF (NL-ML) 80,80,60
60 IEXIT=-1
   WRITE(6,70) NL
70 FORMAT(13H0NL IN ADMB =,I5)
   RETURN
C
C          80 IF (NM) 100,100,90
90 IF (NM-MM) 120,120,100
100 IEXIT=-1
   WRITE(6,110) NM
110 FORMAT(13H0NM IN ADMB =,I5)
   RETURN
C
C          INITIALIZATION
C
C          120 NNB=2*NB
```

NNB1=NNB-1
IT=0

C

IF (IPRINT-3) 130,130,220
130 WRITE(6,140)
140 FORMAT(19HOPRINTOUT FROM ADMB/1X,18(1H*)/
F15H0INITIAL VALUES/1X,14(1H-))
IF (IPRINT-2) 150,150,220
150 WRITE(6,160) MAXIT,JS,CROSS,JQ
160 FORMAT(12H0INPUT DATA:/8H0MAXIT =,I5,6H JS =,I2,
F10H CROSS =,F10.5,6H JQ =,I3)
IF (JQ) 190,190,170
170 WRITE(6,180) PIN
180 FORMAT(6H0PIN =,F10.5)
GO TO 210
190 WRITE(6,200)(I,Q(2*I-1),Q(2*I),I=1,NB)
200 FORMAT(14H0BUSNR
F,10HQREAL
F,5HQIMAG//((I5,5X,2E10.3))
210 CALL PRWF

C

DO 215 I=1,NNB
215 PNEW(I)=COV(I)
220 IF (JQ .GE. 1) GO TO 240
DO 230 I=1,NNB1
230 COV(I)=COV(I)+Q(I)
GO TO 260
240 DO 250 I=1,NNB1
250 COV(I)=PIN
260 COV(NNB)=0.0

C

IF (IPRINT-3) 270,270,300
270 WRITE(6,280)
280 FORMAT(1H0,19H0INITIAL EST AND COV/
F14H0BUSNR
F,10HRE(XE)
F,15HIM(XE)
F,10HPREAL
F,5HPIMAG/)
WRITE(6,290)(I,XE(I),COV(2*I-1),COV(2*I),I=1,NB)
290 FORMAT(I5,5X,2F10.5,5X,2F10.5)

C

300 TBEG=MSCPU(X)
TBIT=TBEG

C

C

C

C

310 CALL CARES(2,IE)
IF (IPRINT-3) 320,320,340
320 IF (IT) 330,330,340
330 CALL PRRES
340 FLOSSN=ALOSS(2)

```
IF (IPRINT-3) 350,350,390
350 IF (IT) 370,370,360
360 IF (IPRINT-1) 370,370,390
370 WRITE(6,380) FLOSSN
380 FORMAT(20H0THE LOSS FUNCTION.=,F15.5)
390 TEIT=MSCPU(X)
TIT=TEIT-TBIT
IF (IPRINT .LE. 1) WRITE(6,400) IT,TIT
400 FORMAT(22H0TIME FOR ITERATION NR,15,4H IS,15,6H MSECS)
C
C CONVERGENCE OR MAXIT ?
C
IF (JS) 410,410,420
410 IF (FLOSSN-ABS(CRLOSS)) 510,420,420
420 IF (IT) 440,440,430
430 IF (FLOSSN-FLOSS0) 440,440,530
440 IF (IT .GE. MAXIT) GO TO 520
C
C HERE STARTS THE ITERATION
C
IT=IT+1
IF (IPRINT-1) 450,450,465
450 WRITE(6,460) IT
460 FORMAT(13H0ITERATION NR,15/1X,17(1H*))
465 FLOSS0=FLOSSN
DO 470 I=1,NB
470 X0(I)=XE(I)
DO 475 I=1,NNB
475 POLD(I)=PNEW(I)
TBIT=MSCPU(X)
C
C COMPUTE THE ESTIMATE
C
CALL ESTB(IPRINT,IERR)
IF (IERR) 500,500,480
480 WRITE(6,490) IT
490 FORMAT(27H0ERROR IN ESTB IN ITERATION,15)
IEXIT=-2
RETURN
C
500 JS=0
GO TO 310
C
C END OF ESTIMATION, PRINTOUT OF THE RESULTS
C
510 IEXIT=1
GO TO 540
520 IEXIT=2
GO TO 540
530 IEXIT=3
540 TEND=MSCPU(X)
TIME=TEND-TBEG
IF (IEXIT .LT. 3) GO TO 560
```

```
PR(2)=' /1H+',  
PR(3)=' 16HDI'  
DO 550 I=1,NB  
550 XE(I)=X0(I)  
DO 555 I=1,NNB  
555 PNEW(I)=POLD(I)  
CALL CARES(2,IE)  
FLOSSN=FLOSS0  
GO TO 590  
560 IF (IEXIT .LT. 2) GO TO 570  
PR(2)='3HNO '  
GO TO 580  
570 PR(2)=' /1H+',  
580 PR(3)='17HCON'  
590 DO 595 I=1,NNB  
595 COV(I)=PNEW(I)
```

C

```
IF (IPRINT-4) 600,990,990  
600 WRITE(6,610)  
610 FORMAT(19HOESTIMATION RESULTS/1X,18(1H-))  
WRITE(6,PR) IT  
WRITE(6,620) TIME  
620 FORMAT(13HOTOTAL TIME =,I5,6H MSECS)  
WRITE(6,630)  
630 FORMAT(1H0,23X,17HFINAL EST AND COV/  
F14H0BUSNR  
F,10HRE(XE)  
F,15HIM(XE)  
F,10HPREAL  
F,5HPIMAG/)  
DO 640 I=1,NB  
I2=2*I  
I1=I2-1  
640 WRITE(6,650) I,XE(I),COV(I1),COV(I2)  
650 FORMAT(I5,2(5X,2F10.5))  
WRITE(6,660) FLOSSN  
660 FORMAT(20H0THE LOSS FUNCTION =,F15.5)  
CALL PRRES
```

C

```
990 RETURN  
END
```

```

SUBROUTINE ADMC(NEWA,CRLOSS,MAXIT,EPS,IEXIT,TIME,IPRINT)
C
C MAINROUTINE FOR METHOD C. IT IS ASSUMED THAT THERE IS ONLY
C ONE REFERENCE VOLTAGE: THE VOLTAGE AT BUS NB
C
C AUTHOR: TON VAN OVERBEEK 1974-03-14
C
C NEWA=1 NEW LINEVOLTAGE WEIGHTING FACTORS AND A NEW
C A-MATRIX ARE CALCULATED
C NEWA=0 NO NEW WEIGHTING FACTORS AND MATRIX ARE CALCULATED
C CRLOSS CONVERGENCE CRITERION
C IF THE INITIAL LOSS .LE. CRLOSS THEN ESTC
C IS NOT CALLED
C
C MAXIT MAXIMUM NUMBER OF ITERATIONS FOR ESTC
C EPS CONVERGENCE CRITERION FOR ESTC
C IEXIT=-3 ERROR IN MASKVECTOR FOR VOLTAGE MEASUREMENT
C IEXIT=-2 DECOMPOSITION OF A HAS FAILED IN ESTC
C IEXIT=-1 ERROR IN NB OR NL
C IEXIT= 0 INITIAL LOSS .LE. CRLOSS
C IEXIT= 1 CONVERGENCE OF ESTC, FINAL LOSS .LT. CRLOSS
C IEXIT= 2 NO CONVERGENCE OF ESTC, FINAL LOSS .LT. CRLOSS
C IEXIT= 3 CONVERGENCE OF ESTC, FINAL LOSS .GT. CRLOSS
C IEXIT= 4 NO CONVERGENCE OF ESTC, FINAL LOSS .GT. CRLOSS
C IEXIT= 5 DIVERGENCE: THE LOSS FUNCTION INCREASES
C THE ESTIMATE IS SET EQUAL TO THE INITIAL ESTIMATE
C
C IPRINT=4 NO PRINTOUT
C IPRINT=0-3 SEE ESTC
C THE INITIAL AND FINAL RESIDUES AND LOSS FUNCTION
C ARE ALSO PRINTED
C
C SUBROUTINE REQUIRED
C ALOSS
C PRRES
C CAD
C CARES
C GXE
C PRENET
C PRRES
C ESTC
C DESYM
C MPRI
C SOLVS
C UPDAC
C UPDdC
C PRRES
C
C PARAMETER MB=10,ML=13
C PARAMETER MMB=2*MB,MML=2*ML,MM=3*(MB+MML)
C
C INTEGER TIME,TBEG,TEND
C
C DIMENSION DA(ML),DB(ML),PR(6) / ('13HOF', 'INAL L', 'OSS ..', 0,
1 '7H CRL', 'OSS') /
```

```
C      COMPLEX YAA,ZAB,YBB,XE,YE2,YE3,YE4,YE5,YE6,XE0(MB)
C
COMMON /EST/ XE(MB),YE2(ML),YE3(ML),YE4(ML),YE5(ML),YE6(MB)
X      /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
X      /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
X      /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),WF6(MMB)
X      /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),
X          MSK6(MB),Nm,NTYP(MM),NMSM(MM)
X      /RES/ RES1(MB),RES2(ML),RES3(ML),RES4(MML),RES5(MML),
X          RES6(MMB)
X      /MAT/ A(MB,MB),T(MB,MB)
C
C      CHECK NB AND NL
C
      IF (NB) 20,20,10
10     IF (NB-MB) 40,40,20
20     IEXIT=-1
      WRITE(6,30) NB
30     FORMAT(13H0NB IN ADCM =,I5)
      RETURN
C
40     IF (NL) 60,60,50
50     IF (NL-ML) 80,80,60
60     IEXIT=-1
      WRITE(6,70) NL
70     FORMAT(13H0NL IN ADCM =,I5)
      RETURN
C
C      CALCULATE THE INITIAL RESIDUES AND LOSS FUNCTION
C
80     TBEG=MSCPU(X)
      CALL CARES(2,IE)
      FLOSSO=ALOSS(2)
C
      IF (IPRINT-3) 90,90,120
90     WRITE(6,100) CRLOSS
100    FORMAT(19H0PRINTOUT FROM ADCM/1X,18(1H*)/
      F15H0INITIAL VALUES/1X,14(1H-)/9H0CRLOSS =,F10.5)
      CALL PRRES
      WRITE(6,110) FLOSSO
110    FORMAT(20H0THE LOSS FUNCTION =,F15.5)
C
C      DIRECT CONVERGENCE ?
C
120   IF (FLOSSO-ABS(CRLOSS)) 130,130,170
130   IEXIT=0
      TEND=MSCPU(X)
      TIME=TEND-TBEG
C
      IF (IPRINT-3) 140,140,990
140   WRITE(6,150)(I,XE(I),I=1,NB)
150   FORMAT(19H0DIRECT CONVERGENCE/
```

```
F1H0,15X,12HTHE ESTIMATE/  
F14H0BUSNR  
F,10HRE(XE)  
F,6HIM(XE)//  
F(I5,5X,2F10.5))  
WRITE(6,160) TIME  
160 FORMAT(22H0TOTAL TIME FOR ADMC =,I5,6H MSECS)  
GO TO 990  
C  
C CALCULATE THE LINEVOLTAGE WEIGHTING FACTORS IF NECESARRY  
C AND COMPUTE THE ESTIMATE  
C  
170 IF (NEWA) 190,190,180  
180 CALL CAD(DA,DB,IPRINT+1)  
190 IF (MSK1(NB)) 200,200,220  
200 IEXIT=-3  
WRITE(6,210) NB,MSK1(NB)  
210 FORMAT(13H0ERROR IN MSK,5X,5HMSK1(,I5,3H) =,I5)  
GO TO 990  
C  
220 DO 230 I=1,NB  
230 XE0(I)=XE(I)  
XE(NB)=CMPLX(YM1(NB),0.0)  
NR=1  
CALL ESTC(NR,DA,DB,MAXIT,EPS,NEWA,IEXIT,TIME,IPRINT)  
IF (IEXIT) 990,240,240  
C  
C CALCULATE THE FINAL RESIDUES AND LOSS FUNCTION  
C  
240 CALL CARES(2,IE)  
FLOSSN=ALOSS(2)  
IF (FLOSSN-ABS(CRLOSS)) 250,250,260  
250 PR(4)='3HLE.,'  
GO TO 280  
C  
260 IF (FLOSSN-FLOSS0) 270,270,300  
270 IEXIT=IEXIT+2  
PR(4)='3HGT.,'  
C  
280 IF (IPRINT-3) 290,290,350  
290 WRITE(6,PR)  
GO TO 350  
C  
300 IEXIT=5  
DO 310 I=1,NB  
310 XE(I)=XE0(I)  
IF (IPRINT-3) 320,320,340  
320 WRITE(6,330)(I,XE(I),I=1,NB)  
330 FORMAT(11H0DIVERGENCE/1H0,15X,12HTHE ESTIMATE/  
F14H0BUSNR  
F,10HRE(XE)  
F,6HIM(XE)//  
F(I5,5X,2F10.5))
```



```
340 CALL CARES(2,IE)
      FLOSSN=FLOSSO
C
350 TEND=MSCPU(X)
      TIME=TEND-TBEG
      IF (IPRINT-3) 360,360,990
360 WRITE(6,110) FLOSSN
      WRITE(6,160) TIME
C
990 RETURN
      END
```

```

SUBROUTINE ALLMSM(NODD,IPRINT)
C
C COMPUTES ALL POSSIBLE MEASUREMENTS BY ADDING NOISE AND
C BIAS TO THE TRUE VALUES
C
C AUTHOR, TON VAN OVERBEEK 1974-01-16
C
C NODD          PARAMETER FOR THE SUBROUTINE NODI.
C               AT FIRST CALL OF ALLMSM NODD MUST EQUAL AN ODD
C               INTEGER(E.G. 19). NODD IS RETURNED CONTAINING A
C               NEW ODD INTEGER WHICH IS USED BY REPEATED CALLS
C IPRINT=1     NO PRINTOUT
C IPRINT=0     TRUE VALUES, BIAS, NOISE, AND MEASUREMENTS ARE PRINTED
C
C SUBROUTINE REQUIRED
C           NODI
C
C PARAMETER MB=10,ML=13
C PARAMETER MMB=2*MB,MML=2*ML
C
C COMPLEX XT,YT2,YT3,YT4,YT5,YT6
C
C COMMON /TNET/ NB,NL
C 1      /TRUE/ XT(MB),YT2(ML),YT3(ML),YT4(ML),YT5(ML),YT6(MB)
C 2      /MSM/  YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
C 3      /METT/ BIAS1(MB),BIAS2(ML),BIAS3(ML),BIAS4(MML),
C 4          BIAS5(MML),BIAS6(MMB),WN1(MB),WN2(ML),WN3(ML),
C 5          WN4(MML),WN5(MML),WN6(MMB)
C
C MODULUS MEASUREMENTS,TYPE 1,2,3
C
C IF (IPRINT .GE. 1) GO TO 20
C WRITE(6,10)
10 FORMAT(21H1PRINTOUT FROM ALLMSM/1X,20(1H*))//
F14H MSMNR
F,10HTRUE
F,12HBIAS
F,8HWN
F,10HNOISE
F,3HMSM/)
C
C TYPE 1
C
20 IF (IPRINT-1) 25,35,35
25 IT=1
WRITE(6,30) IT
30 FORMAT(5H0TYPE,12/)
35 DO 60 I=1,NB
YT=CABS(XT(I))
CALL NODI(NODD,ERR)
ERR=ERR/WN1(I)
YM1(I)=YT+BIAS1(I)+ERR
IF (IPRINT-1) 40,60,60
```

```
40 WRITE(6,50) I,YT,BIAS1(I),WN1(I),ERR,YM1(I)
50 FORMAT(15,5X,2F10.5,F10.2,2F10.5)
60 CONTINUE
```

C
C
C

TYPE 2

```
IF (IPRINT-1) 61,65,65
61 IT=2
WRITE(6,30) IT
65 DO 80 I=1,NL
YT=CABS(YT2(I))
CALL NODI(NODD,ERR)
ERR=ERR/WN2(I)
YM2(I)=YT+BIAS2(I)+ERR
IF (IPRINT-1) 70,80,80
70 WRITE(6,50) I,YT,BIAS2(I),WN2(I),ERR,YM2(I)
80 CONTINUE
```

C
C
C

TYPE 3

```
IF (IPRINT-1) 81,85,85
81 IT=3
WRITE(6,30) IT
85 DO 100 I=1,NL
YT=CABS(YT3(I))
CALL NODI(NODD,ERR)
ERR=ERR/WN3(I)
YM3(I)=YT+BIAS3(I)+ERR
IF (IPRINT-1) 90,100,100
90 WRITE(6,50) I,YT,BIAS3(I),WN3(I),ERR,YM3(I)
100 CONTINUE
```

C
C
C

ACTIVE AND REACTIVE MEASUREMENTS,TYPE 4,5,6

```
IF (IPRINT .GE. 1) GO TO 120
WRITE(6,110)
110 FORMAT(/1H0,33X
F,53HACTIVE
F,8HREACTIVE/
F14H0MSMNR
F,10HTRUE
F,12HBIAS
F,8HWN
F,10HNOISE
F,15HMSM
F,10HTRUE
F,12HBIAS
F,8HWN
F,10HNOISE
F,3HMSM/)
```

```
C
C   TYPE 4
C
  IF (IPRINT-1) 115,120,120
115 IT=4
  WRITE(6,30) IT
120 DO 150 I=1,NL
  I1=2*I-1
  I2=2*I
  YTR=REAL(YT4(I))
  YTI=AIMAG(YT4(I))
  CALL NODI(NODD,ERRR)
  CALL NODI(NODD,ERRI)
  ERRR=ERRR/WN4(I1)
  ERRI=ERRI/WN4(I2)
  YM4(I1)=YTR+BIAS4(I1)+ERRR
  YM4(I2)=YTI+BIAS4(I2)+ERRI
  IF (IPRINT-1) 130,150,150
130 WRITE(6,140) I,YTR,BIAS4(I1),WN4(I1),ERRR,YM4(I1),
  1      YTI,BIAS4(I2),WN4(I2),ERRI,YM4(I2)
140 FORMAT(I5,2(5X,2F10.5,F10.2,2F10.5))
150 CONTINUE
```

```
C
C   TYPE 5
C
  IF (IPRINT-1) 161,165,165
161 IT=5
  WRITE(6,30) IT
165 DO 170 I=1,NL
  I1=2*I-1
  I2=2*I
  YTR=REAL(YT5(I))
  YTI=AIMAG(YT5(I))
  CALL NODI(NODD,ERRR)
  CALL NODI(NODD,ERRI)
  ERRR=ERRR/WN5(I1)
  ERRI=ERRI/WN5(I2)
  YM5(I1)=YTR+BIAS5(I1)+ERRR
  YM5(I2)=YTI+BIAS5(I2)+ERRI
  IF (IPRINT-1) 160,170,170
160 WRITE(6,140) I,YTR,BIAS5(I1),WN5(I1),ERRR,YM5(I1),
  1      YTI,BIAS5(I2),WN5(I2),ERRI,YM5(I2)
170 CONTINUE
```

```
C
C   TYPE 6
C
  IF (IPRINT-1) 181,185,185
181 IT=6
  WRITE(6,30) IT
185 DO 190 I=1,NB
  I1=2*I-1
  I2=2*I
  YTR=REAL(YT6(I))
  YTI=AIMAG(YT6(I))
```

```
CALL NODI(NODD,ERRR)
CALL NODI(NODD,ERRI)
ERRR=ERRR/WN6(I1)
ERRI=ERRI/WN6(I2)
YM6(I1)=YTR+BIAS6(I1)+ERRR
YM6(I2)=YTI+BIAS6(I2)+ERRI
IF (IPRINT-1) 180,190,190
180 WRITE(6,140) I,YTR,BIAS6(I1),WN6(I1),ERRR,YM6(I1),
1          YTI,BIAS6(I2),WN6(I2),ERRI,YM6(I2)
190 CONTINUE
C
RETURN
END
```

FUNCTION ALOSS(IPRINT)

C
C CALCULATES THE LOSS FUNCTION GIVEN THE RESIDUES AND THE WEIGHTING
C FACTORS
C

C AUTHOR: TON VAN OVERBEEK 1974-01-23
C

C IPRINT=2 NO PRINTOUT
C IPRINT=1 THE VALUE OF THE LOSS FUNCTION IS PRINTED
C IPRINT=0 SAME + THE USED MEASUREMENTS, THE CORRESPONDING
C ESTIMATED VALUES AND THE RESIDUES
C

C SUBROUTINE REQUIRED
C PRRES
C

C PARAMETER MB=10, ML=13
C PARAMETER MMB=2*MB, MML=2*ML, MBL=MB+MML, MMBL=2*MBL
C

C DIMENSION WFA(MBL), WFB(MMBL), MSKA(MBL), MSKB(MBL), RESA(MBL),
1 RESB(MMBL)
C

C COMPLEX XE, YE2, YE3, YE4, YE5, YE6, YEA(MBL), YEB(MBL)
C

C COMMON /ENET/ NB, NL

1 /EST/ XE(MB), YE2(ML), YE3(ML), YE4(ML), YE5(ML), YE6(MB)
2 /METE/ WF1(MB), WF2(ML), WF3(ML), WF4(MML), WF5(MML), WF6(MMB)
3 /MSI/ MSK1(MB), MSK2(ML), MSK3(ML), MSK4(ML), MSK5(ML), MSK6(MB)
4 /RES/ RES1(MB), RES2(ML), RES3(ML), RES4(MML), RES5(MML),
5 RES6(MMB)
6 /MSM/ YM1(MB), YM2(NL), YM3(ML), YM4(MML), YM5(MML), YM6(MMB)
C

C EQUIVALENCE (YEA, XE), (YEB, YE4), (WFA, WF1), (WFB, WF4), (MSKA, MSK1),
1 (MSKB, MSK4), (RESA, RES1), (RESB, RES4)
C

C ALOSS=0.0
C DO 60 I=1, MBL
C I2=2*I
C I1=I2-1
C IF (MSKA(I)) 20, 20, 10
10 ALOSS=ALOSS+(WFA(I)*RESA(I))**2
20 MSB=MSKB(I)
C IF (MSB) 60, 60, 30
30 IF (MSB-2) 40, 50, 40
40 ALOSS=ALOSS+(WFB(I1)*RESB(I1))**2
C IF (MSB-1) 60, 60, 50
50 ALOSS=ALOSS+(WFB(I2)*RESB(I2))**2
60 CONTINUE
C

C PRINTOUT
C

C IF (IPRINT-2) 70, 990, 990
70 WRITE(6, 80)
80 FORMAT(20H, OPRINTOUT FROM ALOSS/1X, 19(1H*))
C

```
      IF (IPRINT-1) 90,100,990
    90 CALL PRRES
    100 WRITE(6,110) ALOSS
    110 FORMAT(20H0THE LOSS FUNCTION =,F15.5)
C
    990 RETURN
      END
```

```

SUBROUTINE CAD(DA,DB,IPRINT)
C
C HELPROUTINE FOR ESTC
C CALCULATES THE WEIGHTING FACTORS DA(*) AND DB(*) FOR THE
C LINEVOLTAGES FROM THE ORIGINAL WEIGHTING FACTORS WF4(*) AND
C WF5(*) FOR THE LINE FLOW MEASUREMENTS. SINCE METHOD C ASSUMES
C COMPLEX LINE FLOW MEASUREMENTS ONLY THE WEIGHTING FACTORS FOR
C THE ACTIVE MEASUREMENTS ARE USED
C
C
C AUTHOR, TON VAN OVERBEEK 1974-03-11
C
C DA(*)          WEIGHTING FACTOR FOR LINEVOLTAGE AT A-END
C DB(*)          WEIGHTING FACTOR FOR LINEVOLTAGE AT B-END
C IPRINT=1      NO PRINTOUT
C IPRINT=0      THE ORIGINAL AND COMPUTED WEIGHTING FACTORS
C               ARE PRINTED
C
C SUBROUTINE REQUIRED
C               NONE
C
C PARAMETER MB=10,ML=13
C PARAMETER MMB=2*MB,MML=2*ML
C
C COMPLEX YAA,ZAB,YBB
C
C DIMENSION DA(1),DB(1)
C
C COMMON /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
C X      /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),WF6(MMB)
C
C   IF (NL) 20,20,10
C 10 IF (NL-ML) 40,40,20
C 20 WRITE(6,30) NL
C 30 FORMAT(12H0NL IN CAD =,15)
C   RETURN
C
C 40 IF (IPRINT) 50,50,70
C 50 WRITE(6,60)
C 60 FORMAT(18H0PRINTOUT FROM CAD/1X,17(1H*)/
C   F14H0 LINE
C   F,11HWF4
C   F,14H0A
C   F,11HWF5
C   F,2H0B/)
C
C 70 DO 100 I=1,NL
C   WF4A=WF4(2*I-1)
C   WF5A=WF5(2*I-1)
C   ZLM2=CABS(ZAB(1))**2
C   DA(I)=WF4A/ZLM2
C   DB(I)=WF5A/ZLM2
C
```



```
IF (IPRINT) 80,80,100
80 WRITE(6,90) I,WF4A,DA(I),WF5A,DB(I)
90 FORMAT(15,2(5X,F10.5,F10.2))
C
100 CONTINUE
C
RETURN
END
```

SUBROUTINE CAJAC(IPRINT,IERR)

CALCULATES THE JACOBIAN FOR ESTIMATION METHOD A GIVEN THE ESTIMATE AND ESTIMATOR NETWORK DATA

AUTHOR, TON VAN OVERBEEK 1974-01-28

IPRINT=2 NO PRINTOUT
IPRINT=1 THE JACOBIAN IS PRINTED
IPRINT=0 SAME + ESTIMATE AND ESTIMATOR NETWORK DATA
IERR=1 ERROR IN JACV, JACI, JACLF OR JACBI
IERR=0 NO ERROR

SUBROUTINE REQUIRED

JACBI
JACLF
JACI
JACLF
JACV
PRENET
PRJAC

PARAMETER MB=10,ML=13
PARAMETER MMB=2*MB,MML=2*ML

COMPLEX YAA,ZAB,YBB,YA,ZL,YB,X,XA,XB

COMMON /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)

1 /EST/ X(MB)
2 /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),MSK6(MB)
3 /JAC/ AJAC1(MB,2),AJAC2(ML,4),AJAC3(ML,4),AJAC4(MML,4),
4 AJAC5(MML,4),AJAC6(MMB,MMB)

IF (IPRINT=2) 10,50,50
10 WRITE(6,20)
20 FORMAT(20H0PRINTOUT FROM CAJAC/1X,19(1H*))
IF (IPRINT) 30,30,50
30 CALL PRENET
WRITE(6,40)(I,X(I),I=1,NB)
40 FORMAT(13H0THE ESTIMATE/1X,12(1H-)/14H0BUSNR
F,10HRE(XE)
F,6HIM(XE)//(I5,5X,2F10.5))

TYPE 1 AND 6

50 DO 110 I=1,NB
IF (MSK1(I)) 90,90,60
60 CALL JACV(X(I),AJAC1,I,MB,IERR)
IF (IERR) 90,90,70
70 WRITE(6,80) I
80 FORMAT(16H0ERROR AT BUS NR,I5)
RETURN

```
C
  90 IF (MSK6(I)) 110,110,100
 100 CALL JACBI(MSK6(I),I,AJAC6,2*I-1,AJAC6,2*I,MMB,IERR)
      IF (IERR) 110,110,70
 110 CONTINUE
C
C   TYPE 2, 3, 4 AND 5
C
      DO 210 I=1,NL
      IA=LTA(I)
      IB=LTB(I)
      XA=X(IA)
      XB=X(IB)
      YA=YAA(I)
      ZL=ZAB(I)
      YB=YBB(I)
      I2=2*I
      I1=I2-1
      IF (MSK2(I)) 150,150,120
 120 CALL JACI(XA,XB,ZL,YA,AJAC2,I,ML,IERR)
      IF (IERR) 150,150,130
 130 WRITE(6,140) I,IA,IB
 140 FORMAT(17HOERROR AT LINE NR,I5,7H LTA =,I5,7H LTB =,I5)
      RETURN
C
 150 IF (MSK3(I)) 170,170,160
 160 CALL JACI(XB,XA,ZL,YB,AJAC3,I,ML,IERR)
      IF (IERR) 170,170,130
C
 170 IF (MSK4(I)) 190,190,180
 180 CALL JACLF(MSK4(I),XA,XB,ZL,YA,AJAC4,I1,AJAC4,I2,MML,IERR)
      IF (IERR) 190,190,130
C
 190 IF (MSK5(I)) 210,210,200
 200 CALL JACLF(MSK5(I),XB,XA,ZL,YB,AJAC5,I1,AJAC5,I2,MML,IERR)
      IF (IERR) 210,210,130
C
 210 CONTINUE
C
      IF (IPRINT-2) 220,990,990
 220 CALL PRJAC
C
 990 RETURN
      END
```

```

SUBROUTINE CARES(IPRINT,IERR)
C
C   CALCULATES THE RESIDUES GIVEN THE MEASUREMENTS AND THE ESTIMATED
C   STATE.  WHEN THE MASK VECTOR ELEMENTS FOR TYPE 4, 5 AND 6 MEA-
C   SUREMENTS ARE 1 OR 2 THE RESIDUES FOR BOTH THE ACTIVE AND REACTIVE
C   MEASUREMENT ARE COMPUTED
C
C   AUTHOR, TON VAN OVERBEEK 1974-01-23
C
C   IPRINT=2      NO PRINTOUT
C   IPRINT=1      THE USED MEASUREMENTS, THE CORRESPONDING ESTIMATED
C                 VALUES AND THE RESIDUES ARE PRINTED
C   IPRINT=0      SAME + ESTIMATOR NETWORK DATA
C   IERR=1        ERROR IN NB OR NL
C   IERR=0        NO ERROR
C
C   SUBROUTINE REQUIRED
C       GXE
C       PRENET
C       PRRES
C
C   PARAMETER MB=10,ML=13
C   PARAMETER MMB=2*MB,MML=2*ML,MBL=MB+MML,MMBL=2*MBL
C
C   DIMENSION YMA(MBL),MSKA(MBL),MSKB(MBL),RESA(MBL)
C
C   COMPLEX YAA,ZAB,YBB,XE,YE2,YE3,YE4,YE5,YE6,YEA(MBL),YEB(MBL),
1     YMB(MBL),RESB(MBL)
C
C   COMMON /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
1     /EST/ XE(MB),YE2(ML),YE3(ML),YE4(ML),YE5(ML),YE6(MB)
2     /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
3     /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),
4     MSK6(MMB)
5     /RES/ RES1(MB),RES2(ML),RES3(ML),RES4(MML),RES5(MML),
6     RES6(MMB)
C
C   EQUIVALENCE (YEA,XE),(YEB,YE4),(YMA,YM1),(YMB,YM4),
1     (MSKA,MSK1),(MSKB,MSK4),(RESA,RES1),(RESB,RES4)
C
C   IF (IPRINT-2) 10,30,30
10  WRITE(6,20)
20  FORMAT(20H0PRINTOUT FROM CARES/1X,19(1H*))
30  DO 40 I=1,MBL
    RESA(I)=0.0
40  RESB(I)=(0.0,0.0)
    CALL GXE(IPRINT,IERR)
    IF (IERR-1) 50,990,990
C
50  DO 90 I=1,MBL
    IF (MSKA(I)) 70,70,60
60  RESA(I)=YMA(I)-CABS(YEA(I))
70  IF (MSKB(I)) 90,90,80
```

80 RESB(I)=YMB(I)-YEB(I)
90 CONTINUE

C

IF (IPRINT-2) 100,990,990
100 CALL PRRES

C

990 RETURN
END

SUBROUTINE CASDB(SDB,SSL,U,IPRINT)

COMPUTES THE NEW LOAD DEMAND GIVEN THE OLD DEMAND, THE
STANDARD SLOPES AND THE LOAD CONTROL VECTOR ELEMENT U
ACCORDING TO: $SDB(*) = SDB(*) + U*SSL(*)$

AUTHOR, TON VAN OVERBEEK 1974-02-11

SDB(*) COMPLEX LOAD DEMAND AT BUS *
SSL(*) COMPLEX STANDARD SLOPE FOR BUS *
U LOAD CONTROL VECTOR ELEMENT, SEE SUBROUTINE RDLD
IPRINT=1 NO PRINTOUT
IPRINT=0 THE VALUE OF U, THE STANDARD SLOPES AND
 THE NEW LOAD DEMAND ARE PRINTED

SUBROUTINE REQUIRED
 NONE

COMPLEX SDB(1),SSL(1)

COMMON /TNET/ NB

DO 10 I=1,NB
10 SDB(I)=SDB(I)+U*SSL(I)
 IF (IPRINT) 20,20,99
20 WRITE(6,30) U
30 FORMAT(20HOPRINTOUT FROM CASDB/1X,19(1H*)/
 F4HOU =,F10.5)
 WRITE(6,40)(I,SSL(I),SDB(I),I=1,NB)
40 FORMAT(13HOBUSNR
 F,10HRE(SSL)
 F,16HIM(SSL)
 F,10HPDEM
 F,4HQDEM//
 F(I5,5X,2F10.5,5X,2F10.5))

99 RETURN
END

SUBROUTINE CAWF(IPRINT)

COMPUTES THE WEIGHT FACTORS WF FOR THE ESTIMATORS:
WF=1/(ALFA*FULL SCALE VALUE + BETA*MEASUREMENT)

AUTHOR, TON VAN OVERBEEK 1974-01-21

IPRINT=1 NO PRINTOUT
IPRINT=0 ALFA, FULL SCALE, BETA, MEASUREMENT AND WF
VALUES ARE PRINTED

SUBROUTINE REQUIRED
NONE

PARAMETER MB=10,ML=13
PARAMETER MMB=2*MB,MML=2*ML

COMMON /ENET/ NB,NL
1 /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),
2 WF6(MMB),ALFE1(MB),ALFE2(ML),ALFE3(ML),ALFE4(MML),
3 ALFE5(MML),ALFE6(MMB),FSE1(MB),FSE2(ML),FSE3(ML),
4 FSE4(MML),FSE5(MML),FSE6(MMB),BETE1(MB),BETE2(ML),
5 BETE3(ML),BETE4(MML),BETE5(MML),BETE6(MMB)
6 /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)

DO 10 I=1,NB
I2=2*I
I1=I2-1
WF1(I)=1/(ALFE1(I)*FSE1(I)+BETE1(I)*YM1(I))
WF6(I1)=1/(ALFE6(I1)*FSE6(I1)+BETE6(I1)*YM6(I1))
10 WF6(I2)=1/(ALFE6(I2)*FSE6(I2)+BETE6(I2)*YM6(I2))
DO 20 I=1,NL
I2=2*I
I1=I2-1
WF2(I)=1/(ALFE2(I)*FSE2(I)+BETE2(I)*YM2(I))
WF3(I)=1/(ALFE3(I)*FSE3(I)+BETE3(I)*YM3(I))
WF4(I1)=1/(ALFE4(I1)*FSE4(I1)+BETE4(I1)*YM4(I1))
WF4(I2)=1/(ALFE4(I2)*FSE4(I2)+BETE4(I2)*YM4(I2))
WF5(I1)=1/(ALFE5(I1)*FSE5(I1)+BETE5(I1)*YM5(I1))
20 WF5(I2)=1/(ALFE5(I2)*FSE5(I2)+BETE5(I2)*YM5(I2))

PRINTOUT

IF (IPRINT-1) 30,990,990
30 WRITE(6,40)
40 FORMAT(19HOPRINTOUT FROM CAWF/1X,18(1H*)/
F14H0MSMNR
F,6HALFA
F,14HFULL SCALE
F,10HBETA
F,12HMSMT
F,2HWF/
F,7H0TYPE 1/
F,14H0MSMNR

```
WRITE(6,50)(I,ALFE1(I),FSE1(I),BETE1(I),YM1(I),WF1(I),I=1,NB)
50 FORMAT(15,5X,4F10.5,F10.2)
WRITE(6,60)
60 FORMAT(7H0TYPE 2/)
WRITE(6,50)(I,ALFE2(I),FSE2(I),BETE2(I),YM2(I),WF2(I),I=1,NL)
WRITE(6,70)
70 FORMAT(7H0TYPE 3/)
WRITE(6,50)(I,ALFE3(I),FSE3(I),BETE3(I),YM3(I),WF3(I),I=1,NL)
WRITE(6,80)
80 FORMAT(/1H0,33X
F,53HACTIVE
F,8HREACTIVE/
F,14H0MSMNR
F,6HALFA
F,14HFULL SCALE
F,10HBETA
F,12HMSMT
F,13HWF
F,6HALFA
F,14HFULL SCALE
F,10HBETA
F,12HMSMT
F,2HWF/
F,7H0TYPE 4/)
DO 90 I=1,NL
I2=2*I
I1=I2-1
90 WRITE(6,100) I,ALFE4(I1),FSE4(I1),BETE4(I1),YM4(I1),WF4(I1),
1 ALFE4(I2),FSE4(I2),BETE4(I2),YM4(I2),WF4(I2)
100 FORMAT(15,2(5X,4F10.5,F10.2))
WRITE(6,110)
110 FORMAT(7H0TYPE 5/)
DO 120 I=1,NL
I2=2*I
I1=I2-1
120 WRITE(6,100) I,ALFE5(I1),FSE5(I1),BETE5(I1),YM5(I1),WF5(I1),
1 ALFE5(I2),FSE5(I2),BETE5(I2),YM5(I2),WF5(I2)
WRITE(6,130)
130 FORMAT(7H0TYPE 6/)
DO 140 I=1,NB
I2=2*I
I1=I2-1
140 WRITE(6,100) I,ALFE6(I1),FSE6(I1),BETE6(I1),YM6(I1),WF6(I1),
1 ALFE6(I2),FSE6(I2),BETE6(I2),YM6(I2),WF6(I2)
C
990 RETURN
END
```


SUBROUTINE CAWN(IPRINT)

COMPUTES THE STANDARD DEVIATIONS WN FOR THE MEASUREMENT
NOISE: $WN=1/(ALFA*FULL\ SCALE\ VALUE + BETA*TRUE\ MEASUREMENT)$

AUTHOR, TON VAN OVERBEEK 1974-01-21

IPRINT=1 NO PRINTOUT
IPRINT=0 ALFA, FULL SCALE, BETA, TRUE MEASUREMENT AND WN
VALUES ARE PRINTED

SUBROUTINE REQUIRED
NONE

PARAMETER MB=10,ML=13
PARAMETER MMB=2*MB,MML=2*ML

COMPLEX XT,YT2,YT3,YT4,YT5,YT6

DIMENSION YA1(MB),YA2(ML),YA3(ML),YR4(ML),YI4(ML),
1 YR5(ML),YI5(ML),YR6(MB),YI6(MB)

COMMON /TNET/ NB,NL
1 /METT/ BIAS1(MB),BIAS2(ML),BIAS3(ML),BIAS4(MML),
2 BIAS5(MML),BIAS6(MMB),WN1(MB),WN2(ML),WN3(ML),
3 WN4(MML),WN5(MML),WN6(MMB),ALFT1(MB),ALFT2(ML),
4 ALFT3(ML),ALFT4(MML),ALFT5(MML),ALFT6(MMB),
5 FST1(MB),FST2(ML),FST3(ML),FST4(MML),
6 FST5(MML),FST6(MMB),BETT1(MB),BETT2(ML),BETT3(ML),
7 BETT4(MML),BETT5(MML),BETT6(MMB)
8 /TRUE/ XT(MB),YT2(ML),YT3(ML),YT4(ML),YT5(ML),YT6(MB)

DO 10 I=1,NB
I2=2*I
I1=I2-1
YA1(I)=CABS(XT(I))
YR6(I)=REAL(YT6(I))
YI6(I)=AIMAG(YT6(I))
WN1(I)=1/(ALFT1(I)*FST1(I)+BETT1(I)*YA1(I))
WN6(I1)=1/(ALFT6(I1)*FST6(I1)+BETT6(I1)*YR6(I))
10 WN6(I2)=1/(ALFT6(I2)*FST6(I2)+BETT6(I2)*YI6(I))
DO 20 I=1,NL
I2=2*I
I1=I2-1
YA2(I)=CABS(YT2(I))
YA3(I)=CABS(YT3(I))
YR4(I)=REAL(YT4(I))
YI4(I)=AIMAG(YT4(I))
YR5(I)=REAL(YT5(I))
YI5(I)=AIMAG(YT5(I))
WN2(I)=1/(ALFT2(I)*FST2(I)+BETT2(I)*YA2(I))
WN3(I)=1/(ALFT3(I)*FST3(I)+BETT3(I)*YA3(I))
WN4(I1)=1/(ALFT4(I1)*FST4(I1)+BETT4(I1)*YR4(I))

```
WN4(I2)=1/(ALFT4(I2)*FST4(I2)+BETT4(I2)*YI4(I));  
WN5(I1)=1/(ALFT5(I1)*FST5(I1)+BETT5(I1)*YR5(I));  
20 WN5(I2)=1/(ALFT5(I2)*FST5(I2)+BETT5(I2)*YI5(I))
```

C
C
C

```
PRINTOUT
```

```
IF (IPRINT-1) 30,990,990  
30 WRITE(6,40)  
40 FORMAT(19H0PRINTOUT FROM CAWN/1X,18(1H*)/  
F14H0MSMNR  
F,6HALFA  
F,14HFULL SCALE  
F,8HBETA  
F,14HTRUE MSM  
F,2HWN/  
F,7H0TYPE 1/)  
F,14H0MSMNR  
WRITE(6,50)(I,ALFT1(I),FST1(I),BETT1(I),YA1(I),WN1(I),I=1,NB)  
50 FORMAT(I5,5X,4F10.5,F10.2)  
WRITE(6,60)  
60 FORMAT(7H0TYPE 2/)  
WRITE(6,50)(I,ALFT2(I),FST2(I),BETT2(I),YA2(I),WN2(I),I=1,NL)  
WRITE(6,70)  
70 FORMAT(7H0TYPE 3/)  
WRITE(6,50)(I,ALFT3(I),FST3(I),BETT3(I),YA3(I),WN3(I),I=1,NL)  
WRITE(6,80)  
80 FORMAT(/1H0,33X  
F,53HACTIVE  
F,8HREACTIVE/  
F,14H0MSMNR  
F,6HALFA  
F,14HFULL SCALE  
F,8HBETA  
F,14HTRUE MSM  
F,13HWN  
F,6HALFA  
F,14HFULL SCALE  
F,8HBETA  
F,14HTRUE MSM  
F,2HWN/  
F,7H0TYPE 4/)  
DO 90 I=1,NL  
I2=2*I  
I1=I2-1  
90 WRITE(6,100) I,ALFT4(I1),FST4(I1),BETT4(I1),YR4(I),WN4(I1),  
1 ALFT4(I2),FST4(I2),BETT4(I2),YI4(I),WN4(I2)  
100 FORMAT(I5,2(5X,4F10.5,F10.2))  
WRITE(6,110)  
110 FORMAT(7H0TYPE 5/)  
DO 120 I=1,NL  
I2=2*I  
I1=I2-1  
120 WRITE(6,100) I,ALFT5(I1),FST5(I1),BETT5(I1),YR5(I),WN5(I1),  
1 ALFT5(I2),FST5(I2),BETT5(I2),YI5(I),WN5(I2)
```

```
WRITE(6,130)
130 FORMAT(7H0TYPE 6/)
DO 140 I=1,NB
  I2=2*I
  I1=I2-1
140 WRITE(6,100) I,ALFT6(I1),FST6(I1),BETT6(I1),YR6(I),WN6(I1),
  1 ALFT6(I2),FST6(I2),BETT6(I2),YI6(I),WN6(I2)
C
990 RETURN
  END
```

SUBROUTINE ELDNL(A1,A2,PMIN,PGEN,PMAX,PDEM,EPS,NG,IPRINT,IERR)

COMPUTES A SOLUTION TO

* THE ECONOMIC LOAD DISPATCH PROBLEM *
* NEGLECTING THE TRANSMISSION LOSSES *

REFERENCE, L.K. KIRCHMAYER, 'ECONOMIC OPERATIONS OF POWER SYSTEMS'

AUTHOR, STURE LINDAHL 1972-03-12

REVISED FOR SEIPS, TON VAN OVERBEEK 1974-01-11

A1(I) COEFFICIENTS IN THE GENERATOR COST FUNCTION
A2(I) $F(PG)=A1(I)*PG(I)+A2(I)*PG(I)**2$
PMIN(I) MINIMUM PERMISSIBLE ACTIVE POWER AT GENERATOR I
PGEN(I) COMPUTED ACTIVE POWER AT GENERATOR I
PMAX(I) MAXIMUM PERMISSIBLE ACTIVE POWER AT GENERATOR I
PDEM TOTAL DEMAND OF ACTIVE POWER
EPS THE ITERATION IS TERMINATED WHEN
THE POWER MISMATCH IS LESS THAN EPS*PDEM
NG NUMBER OF GENERATORS
IPRINT=0 MAXIMUM PRINTOUT FROM ELDNL
IPRINT=1 INPUT DATA AND RESULTS ARE PRINTED
IPRINT=2 NO PRINTOUT
IERR=0 A SOLUTION HAS BEEN COMPUTED
IERR=1 ERROR IN NG
IERR=2 TOTAL DEMAND OUTSIDE LOAD BOUNDARIES

SUBROUTINE REQUIRED
NONE

DIMENSION A1(1),A2(1),PMIN(1),PGEN(1),PMAX(1)

DATA LP/6/

IF(NG) 10,10,30
10 WRITE(LP,20) NG
20 FORMAT(4H NG=,I5,9H IN ELDNL)
IERR=1
GO TO 990

30 IF(IPRINT-1) 40,40,100
40 WRITE(LP,50)
50 FORMAT(20HOPRINTOUT FROM ELDNL/1X,19(1H*))/
WRITE(LP,60)
60 FORMAT(26H GENERATOR CHARACTERISTICS/1X,25(1H-)/
110H GENERATOR
2,15H PMIN
3,15H PMAX
4,15H A1

```
5,15H      A2
6)
  DO 70 I=1,NG
70 WRITE(LP,80) I,PMIN(I),PMAx(I),A1(I),A2(I)
80 FORMAT(17,2F15.1,F15.3,F15.5)
  WRITE(LP,90) PDEM
90 FORMAT(15H TOTAL DEMAND =,F22.1)

  COMPUTE MINIMUM AND MAXIMUM CAPACITY,INITIAL PGA AND PGB
100 PGMIN=0.0
  PGMAX=0.0
  DO 110 I=1,NG
  PGMIN=PGMIN+PMIN(I)
110 PGMAX=PGMAX+PMAx(I)
  PGB=PGMAX-PDEM
  PGA=PGMIN-PDEM
  IF(PGB) 130,150,150
150 WRITE(LP,140)
140 FORMAT(59H POWER DEMAND GREATER THAN SUM OF MAXIMUM PERMISSIBLE POWER)
  IERR=2
  GO TO 990

150 IF(PGA) 200,200,160
160 WRITE(LP,170)
170 FORMAT(56H POWER DEMAND LESS THAN SUM OF MINIMUM PERMISSIBLE POWER)
  IERR=2
  GO TO 990

  SOLVE THE PROBLEM IF ONLY ONE GENERATOR
200 IF(NG-1) 210,210,220
210 PGEN(1)=PDEM
  GO TO 400

  COMPUTE INCREMENTAL COST AT MAXIMUM AND MINIMUM LOAD
  INITIAL ALA AND ALB
220 ALB=A1(1)+2.0*A2(1)*PMAx(1)
  ALA=A1(1)+2.0*A2(1)*PMIN(1)
  DO 230 I=2,NG
  ALB=AMAX1(ALB,(A1(I)+2.0*A2(I)*PMAx(I)))
230 ALA=AMIN1(ALA,(A1(I)+2.0*A2(I)*PMIN(I)))

  PREPARE FIRST ITERATION.

  ITER=0
  ALN=ALA
  PNM=PGA
```

```
C
C   HERE STARTS THE ITERATION
C
C   240 ITER=ITER+1
C
C   COMPUTE A NEW LAMBDA
C
C   IF(PMM) 250,250,260
250 ALA=ALN
    PGA=PMM
    GO TO 270
260 ALB=ALN
    PGB=PMM
270 ALN=ALA+PGA*(ALB-ALA)/(PGA-PGB)
    ALN=AMAX1(ALN,ALA+0.1*(ALB-ALA))
    ALN=AMIN1(ALN,ALB-0.1*(ALB-ALA))
C
C   DETERMINE FEASIBLE LOADS AND POWER MISMATCH
C
C   PGN=0.0
    DO 345 I=1,NG
    PGEN(I)=0.5*(ALN-A1(I))/A2(I)
    IF(PGEN(I)-PMIN(I)) 310,320,320
310 PGEN(I)=PMIN(I)
    GO TO 340
320 IF(PMAX(I)-PGEN(I)) 330,340,340
330 PGEN(I)=PMAX(I)
340 PGN=PGN+PGEN(I)
345 CONTINUE
    PMM=PGN-PDEM
C
C   ITERATION PRINTOUT
C
C   IF(IPRINT) 350,350,380
350 WRITE(LP,360) ITER,ALN,PMM
360 FORMAT(15H ITERATION NR =,I22/19H INCREMENTAL COST =,F18.5/
    117H POWER MISMATCH =,F20.5)
    WRITE(LP,370) (PGEN(I),I=1,NG)
370 FORMAT(22H COMPUTED ACTIVE POWER/(7X,5F15.3))
C
C   TEST ON CONVERGENCY
C
C   380 IF(EPS*PDEM-ABS(PMM)) 240,240,400
C
C   PRINTOUT OF FINAL RESULTS
C
C   400 IERR=0
    IF(IPRINT-1) 410,410,990
410 WRITE(LP,420)
420 FORMAT(/23H RESULT OF OPTIMIZATION/1X,22(1H-)/
    110H GENERATOR,10H      PGEN)
    DO 430 I=1,NG
```

```
430 WRITE(LP,440) I,PGEN(I)
440 FORMAT(I7,F15.3)
      WRITE(LP,450) ALN,PMM
450 FORMAT(19H INCREMENTAL COST =,F18.5/
117H POWER MISMATCH =,F20.5)
990 RETURN
      END
```

SUBROUTINE ESTA(LIN,DX,IPRINT,IERR)

PERFORMS ONE ITERATION OF METHOD A BY SOLVING THE EQUATION A*DX=B. IF LIN=1 A NEW A-MATRIX IS COMPUTED. A=(JACOBIAN)**T*WF*JACOBIAN. THE RIGHT-HAND VECTOR B IS COMPUTED EACH TIME ESTA IS CALLED. B=(JACOBIAN)**T*WF*RES. IT IS ASSUMED THAT THE JACOBIAN AND THE RESIDUES ARE ALREADY COMPUTED AND AVAILABLE IN THE COMMON BLOCKS /JAC/ AND /RES/.

AUTHOR, TON VAN OVERBEEK 1974-02-01

LIN=1 RELINEARIZATION, A NEW A-MATRIX IS COMPUTED
LIN=0 NO RELINEARIZATION
DX THE SOLUTION: DX=NEW ESTIMATE - OLD ESTIMATE
DX(2*NB) = 0.0 = IMAGINARY PART OF SLACKBUS VOLTAGE
IPRINT=4 NO PRINTOUT
IPRINT=3 DX IS PRINTED
IPRINT=2 SAME + JACOBIAN, THE USED MEASUREMENTS, THE CORRESPONDING ESTIMATED VALUES AND THE RESIDUES
NOTE: ONLY THE RESIDUES ARE USED IN THE CALCULATIONS
IPRINT=1 SAME + B-VECTOR
IPRINT=0 SAME + A- AND T-MATRICES
IERR=1 DECOMPOSITION OF A- IN T-MATRIX HAS FAILED
IERR=0 NO DECOMPOSITION ERROR

SUBROUTINE REQUIRED

- DESYM
MPRI
PRJAC
PRRES
SOLVS
UPDAA
UPDBA

PARAMETER MB=10,ML=13
PARAMETER MMB=2*MB,MML=2*ML,MM=3*(MB+MML)

COMPLEX XE, YE2, YE3, YE4, YE5, YE6, YAA, ZAB, YBB

DIMENSION DX(1), INDEX(MMB), ELT(MMB), B(MMB)

COMMON /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
1 /EST/ XE(MB),YE2(ML),YE3(ML),YE4(ML),YE5(ML),YE6(MB)
2 /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
3 /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),
4 WF6(MMB)
5 /RES/ RES1(MB),RES2(ML),RES3(ML),RES4(MML),RES5(MML),
6 RES6(MMB)
7 /MAT/ A(MMB,MMB),T(MMB,MMB)
8 /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),
9 MSK6(MB)
1 /JAC/ AJAC1(MB,2),AJAC2(ML,4),AJAC3(ML,4),AJAC4(MML,4),
2 AJAC5(MML,4),AJAC6(MMB,MMB)


```
C
  IF (IPRINT-4) 10,40,40
10 WRITE(6,20)
20 FORMAT(19HOPRINTOUT FROM ESTA/1X,18(1H*))
  IF (IPRINT-3) 30,40,40
30 CALL PRRES
  CALL PRJAC
```

```
C
C
C   UPDATING OF A-MATRIX AND B-VECTOR
```

```
40 NNB=2*NB
  DO 55 I=1,NNB
    B(I)=0.0
    IF (LIN) 55,55,45
45 DO 50 J=1,NNB
50 A(I,J)=0.0
55 CONTINUE
```

```
C
C
C   TYPE 1
```

```
  DO 80 I=1,NB
    IF (MSK1(I)) 80,80,60
60 INDEX(1)=2*I-1
  INDEX(2)=2*I
  ELT(1)=AJAC1(I,1)
  ELT(2)=AJAC1(I,2)
  CALL UPDBA(INDEX,ELT,2,WF1(I),B,RES1(I))
  IF (LIN) 80,80,70
70 CALL UPDAA(INDEX,ELT,2,WF1(I),A)
80 CONTINUE
```

```
C
C
C   TYPE 2
```

```
  DO 120 I=1,NL
    IF (MSK2(I)) 120,120,90
90 IA=LTA(I)
  IB=LTB(I)
  INDEX(1)=2*IA-1
  INDEX(2)=2*IA
  INDEX(3)=2*IB-1
  INDEX(4)=2*IB
  DO 100 J=1,4
100 ELT(J)=AJAC2(I,J)
  CALL UPDBA(INDEX,ELT,4,WF2(I),B,RES2(I))
  IF (LIN) 120,120,110
110 CALL UPDAA(INDEX,ELT,4,WF2(I),A)
120 CONTINUE
```

```
C
C
C   TYPE 3
```

```
  DO 160 I=1,NL
    IF (MSK2(I)) 160,160,130
130 IA=LTA(I)
```

```
IB=LTB(I)
INDEX(1)=2*IB-1
INDEX(2)=2*IB
INDEX(3)=2*IA-1
INDEX(4)=2*IA
DO 140 J=1,4
140 ELT(J)=AJAC3(I,J)
CALL UPDBA(INDEX,ELT,4,WF3(I),B,RES3(I))
IF (LIN) 160,160,150
150 CALL UPDAA(INDEX,ELT,4,WF3(I),A)
160 CONTINUE
C
C TYPE 4
C
DO 250 I=1,NL
M=MSK4(I)
IF (M) 250,250,170
170 I2=2*I
I1=I2-1
IA=LTA(I)
IB=LTB(I)
INDEX(1)=2*IA-1
INDEX(2)=2*IA
INDEX(3)=2*IB-1
INDEX(4)=2*IB
C
IF (M-2) 210,180,180
180 DO 190 J=1,4
190 ELT(J)=AJAC4(I2,J)
CALL UPDBA(INDEX,ELT,4,WF4(I2),B,RES4(I2))
IF (LIN) 210,210,200
200 CALL UPDAA(INDEX,ELT,4,WF4(I2),A)
C
210 IF (M-2) 220,250,220
220 DO 230 J=1,4
230 ELT(J)=AJAC4(I1,J)
CALL UPDBA(INDEX,ELT,4,WF4(I1),B,RES4(I1))
IF (LIN) 250,250,240
240 CALL UPDAA(INDEX,ELT,4,WF4(I1),A)
C
250 CONTINUE
C
C TYPE 5
C
DO 340 I=1,NL
M=MSK5(I)
IF (M) 340,340,260
260 I2=2*I
I1=I2-1
IA=LTA(I)
IB=LTB(I)
INDEX(1)=2*IB-1
INDEX(2)=2*IB
```

```
INDEX(3)=2*IA-1
INDEX(4)=2*IA
C
  IF (M-2) 300,270,270
270 DO 280 J=1,4
280 ELT(J)=AJAC5(I2,J)
  CALL UPDBA(INDEX,ELT,4,WF5(I2),B,RES5(I2))
  IF (LIN) 300,300,290
290 CALL UPDAA(INDEX,ELT,4,WF5(I2),A)
C
300 IF (M-2) 310,340,310
310 DO 320 J=1,4
320 ELT(J)=AJAC5(I1,J)
  CALL UPDBA(INDEX,ELT,4,WF5(I1),B,RES5(I1))
  IF (LIN) 340,340,330
330 CALL UPDAA(INDEX,ELT,4,WF5(I1),A)
C
340 CONTINUE
C
C
C
  TYPE 6
C
  NNB1=NNB-1
  DO 350 I=1,NNB1
350 INDEX(I)=I
  DO 440 I=1,NNB1
  M=MSK6(I)
  IF (M) 440,440,360
360 I2=2*I
  I1=I2-1
C
  IF (M-2) 400,370,370
370 DO 380 J=1,NNB1
380 ELT(J)=AJAC6(I2,J)
  CALL UPDBA(INDEX,ELT,NNB1,WF6(I2),B,RES6(I2))
  IF (LIN) 400,400,390
390 CALL UPDAA(INDEX,ELT,NNB1,WF6(I2),A)
C
400 IF (M-2) 410,440,410
410 DO 420 J=1,NNB1
420 ELT(J)=AJAC6(I1,J)
  CALL UPDBA(INDEX,ELT,NNB1,WF6(I1),B,RES6(I1))
  IF (LIN) 440,440,430
430 CALL UPDAA(INDEX,ELT,NNB1,WF6(I1),A)
C
440 CONTINUE
C
C
C
  PRINTOUT
C
  IF (IPRINT) 450,450,470
450 WRITE(6,460)
460 FORMAT(9HOMATRIX A/1X,8(1H-)//)
  CALL MPRI(A,NNB1,NNB1,MMB,8,0,IE)
C
```

```
C      SOLVE A*DX=B
C
470 CALL DESYM(A,T,NNB1,1.E-7,IRANK,MMB)
      IF (IRANK) 480,480,500
480 IERR=1
      WRITE(6,490)
490 FORMAT(34H0DECOMPOSITION OF A FAILED IN ESTA)
      RETURN
C
500 IERR=0
      CALL SOLVS(T,B,DX,NNB1,1,MMB)
      DX(NNB)=0.0
C
C      PRINTOUT
C
      IF (IPRINT-4) 505,990,990
505 IF (IPRINT-1) 510,530,550
510 WRITE(6,520)
520 FORMAT(9H0MATRIX T/1X,8(1H-)//)
      CALL MPRI(T,NNB1,NNB1,MMB,8,0,IE)
530 WRITE(6,540)(B(I),I=1,NNB1)
540 FORMAT(1H0,6X,8HB-VECTOR/7X,8(1H-)//(F11.3,F10.3))
550 WRITE(6,560)(I,DX(2*I-1),DX(2*I),I=1,NB)
560 FORMAT(14H0BUSNR
      F,10HRE(DX)
      F,6HIM(DX)//
      F(15,5X,2F10.5))
C
990 RETURN
      END
```

SUBROUTINE ESTB(IPRINT,IERR)

PERFORMS ONE ITERATION OF METHOD B

AUTHOR: TON VAN OVERBEEK 1974-03-04

IPRINT=2 NO PRINTOUT
IPRINT=1 THE INITIAL ESTIMATE, COVARIANCE AND THE FINAL
ESTIMATE AND COVARIANCE ARE PRINTED
IPRINT=0 SAME + FOR EACH MEASUREMENT THE OLD AND NEW
ESTIMATE AND COVARIANCE ELEMENTS AND THE
CORRESPONDING FOUR GAIN ELEMENTS
IERR=1 ERROR IN MASK VECTOR
IERR=0 NO ERROR

SUBROUTINE REQUIRED

JACBI
JACLF
JACI
JACLF
JACV
UPDEP

PARAMETER MB=10,ML=13
PARAMETER MMB=2*MB,MML=2*ML,MM=3*(MB+MML)

DIMENSION G(MMB),P(MMB),PP(4),AK(MMB)

COMPLEX YAA,YA,ZAB,ZL,YBB,XE,XEA,XEB,CGXE

COMMON /EST/ XE(MB)

X /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
X /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),WF6(MMB)
X /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
X /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),
X MSK6(MB),NM,NTYP(MM),NMSM(MM)
X /VAR/ COV(MMB),PNEW(MMB)

FORMAT STATEMENTS

10 FORMAT(24H0TYPE MSMNR MSM
F,6HWEIGHT//I4,16,2F10.5)
20 FORMAT(24H0TYPE MSMNR ACTIVE
F,6HWEIGHT//I4,16,2F10.5)
30 FORMAT(24H0TYPE MSMNR REACTIVE
F,6HWEIGHT//I4,16,2F10.5)
40 FORMAT(1H0,24X,15HOLD EST AND COV)
50 FORMAT(14H0BUSNR
F,10HRE(XE)
F,15HIM(XE)
F,10HPREAL
F,5HPIMAG//(I5,5X,2F10.5,5X,2F10.5))
60 FORMAT(1H0,36X,15HNEW EST AND COV/

```
F,14HOBUSNR
F,10HRE(XE)
F,15HIM(XE)
F,10HREGAIN
F,15HIMGAIN
F,10HPREAL
F,5HPIMAG// (I5,5X,2F10.5,5X,2F10.5,5X,2F10.5))
70 FORMAT(13H0ERROR IN MSK,5X,5HMSK =,I5,
F7H TYPE =,I5,8H MSMNR =,I5)
```

C

```
NNB=2*NB
NNB1=NNB-1
IERR=0
DO 75 I=1,NNB1
75 P(I)=COV(I)
P(NNB)=0.0
IF (IPRINT-1) 80,80,100
80 WRITE(6,90)
90 FORMAT(1H0,22X,19HINITIAL EST AND COV)
WRITE(6,50)(I,XE(I),P(2*I-1),P(2*I),I=1,NB)
```

C

SEQUENTIAL PROCESSING OF MEASUREMENTS

C

```
100 DO 620 I=1,NM
IF (IPRINT .LE. 0) WRITE(6,105) I
105 FORMAT(13H0MEASUREMENT ,I5/1X,17(1H*))
NT=NTYP(I)
IM=NMSM(I)
GO TO (110,170,210,290,320,450),NT
```

C

TYPE 1

C

```
110 IF (MSK1(IM)) 120,120,130
120 WRITE(6,70) MSK1(IM),NT,IM
IERR=1
RETURN
```

C

```
130 IM2=2*IM
IM1=IM2-1
YM=YM1(IM)
XEA=XE(IM)
IF (IPRINT) 140,140,150
140 WRITE(6,10) NT,IM,YM,WF1(IM)
WRITE(6,40)
WRITE(6,50) IM,XEA,P(IM1),P(IM2)
150 GXE=CABS(XEA)
RES=YM-GXE
CALL JACV(XEA,G,1,1,IE)
AK1=P(IM1)*G(1)
AK2=P(IM2)*G(2)
DEN=AK1*G(1)+AK2*G(2)+1/WF1(IM)
AK1=AK1/DEN
AK2=AK2/DEN
```

```
IF (IM .EQ. NB) AK2=0.0
XE(IM)=XEA+CMPLX(AK1,AK2)*RES
P(IM1)=P(IM1)*(1-AK1*G(1))
P(IM2)=P(IM2)*(1-AK2*G(2))
IF (IPRINT) 160,160,620
160 WRITE(6,60) IM,XE(IM),AK1,AK2,P(IM1),P(IM2)
GO TO 620
```

C
C
C

```
TYPE 2 AND 3
170 IF (MSK2(IM)) 180,180,190
180 WRITE(6,70) MSK2(IM),NT,IM
IERR=1
RETURN
```

C

```
190 IA=LTA(IM)
IB=LTB(IM)
YA=YAA(IM)
YM=YM2(IM)
WF=WF2(IM)
IF (IPRINT) 200,200,250
200 WRITE(6,10) NT,IM,YM,WF
GO TO 250
```

C

```
210 IF (MSK3(IM)) 220,220,230
220 WRITE(6,70) MSK3(IM),NT,IM
IERR=1
RETURN
```

C

```
230 IA=LTB(IM)
IB=LTA(IM)
YA=YBB(IM)
YM=YM3(IM)
WF=WF3(IM)
IF (IPRINT) 240,240,250
240 WRITE(6,10) NT,IM,YM,WF
```

C

```
250 IA2=2*IA
IA1=IA2-1
IB2=2*IB
IB1=IB2-1
PP(1)=P(IA1)
PP(2)=P(IA2)
PP(3)=P(IB1)
PP(4)=P(IB2)
XEA=XE(IA)
XEB=XE(IB)
IF (IPRINT) 260,260,270
260 WRITE(6,40)
WRITE(6,50) IA,XEA,PP(1),PP(2),
1 IB,XEB,PP(3),PP(4)
270 ZL=ZAB(IM)
GXE=CABS((XEA-XEB)/ZL+YA*XEA)
```

```
RES=YM-GXE
CALL JAC1(XEA,XEB,ZL,YA,G,1,1,IE)
CALL UPDEP(IA,IB,RES,WF,G,PP,AK,XE)
P(IA1)=PP(1)
P(IA2)=PP(2)
P(IB1)=PP(3)
P(IB2)=PP(4)
IF (IPRINT) 280,280,620
280 WRITE(6,60) IA,XE(IA),AK(1),AK(2),PP(1),PP(2),
1 IB,XE(IB),AK(3),AK(4),PP(3),PP(4)
GO TO 620
```

```
C
C TYPE 4 AND 5
C
```

```
290 M=MOD(IM,2)
II=IM/2+M
IF (MSK4(II)) 300,300,310
300 WRITE(6,70) MSK4(II),NT,IM
IERR=1
RETURN
```

```
C
310 IA=LTA(II)
IB=LTB(II)
YA=YAA(II)
YM=YM4(IM)
WF=WF4(IM)
GO TO 350
```

```
C
320 M=MOD(IM,2)
II=IM/2+M
IF (MSK5(II)) 330,330,340
330 WRITE(6,70) MSK5(II),NT,IM
IERR=1
RETURN
```

```
C
340 IA=LTB(II)
IB=LTA(II)
YA=YBB(II)
YM=YM5(IM)
WF=WF5(IM)
```

```
C
350 MSK=2-M
IA2=2*IA
IA1=IA2-1
IB2=2*IB
IB1=IB2-1
PP(1)=P(IA1)
PP(2)=P(IA2)
PP(3)=P(IB1)
PP(4)=P(IB2)
XEA=XE(IA)
XEB=XE(IB)
ZL=ZAB(II)
```



```
CALL JACLF(MSK,XEA,XEB,ZL,YA,G,1,G,1,1,IE)
CGXE=XEA*CONJG((XEA-XEB)/ZL+XEA*YA)
IF (M) 390,390,360
C
C ACTIVE MEASUREMENT
C
360 IF (IPRINT) 370,370,380
370 WRITE(6,20) NT,IM,YM,WF
WRITE(6,40)
WRITE(6,50) IA,XEA,PP(1),PP(2),
1 IB,XEB,PP(3),PP(4)
380 RES=YM-REAL(CGXE)
GO TO 420
C
C REACTIVE MEASUREMENT
C
390 IF (IPRINT) 400,400,410
400 WRITE(6,30) NT,IM,YM,WF
WRITE(6,40)
WRITE(6,50) IA,XEA,PP(1),PP(2),
1 IB,XEB,PP(3),PP(4)
410 RES=YM-AIMAG(CGXE)
C
420 CALL UPDEP(IA,IB,RES,WF,G,PP,AK,XE)
IF (IPRINT) 430,430,440
430 WRITE(6,60) IA,XE(IA),AK(1),AK(2),PP(1),PP(2),
1 IB,XE(IB),AK(3),AK(4),PP(3),PP(4)
440 P(IA1)=PP(1)
P(IA2)=PP(2)
P(IB1)=PP(3)
P(IB2)=PP(4)
GO TO 620
C
C TYPE 6
C
450 M=MOD(IM,2)
IA=IM/2+M
IF (MSK6(IA)) 460,460,470
460 WRITE(6,70) MSK6(IA),NT,IM
IERR=1
RETURN
C
470 MSK=2-M
YM=YM6(IM)
WF=WF6(IM)
XEA=XE(IA)
CALL JACBI(MSK,IA,G,1,G,1,1,IE)
CGXE=(0.0,0.0)
DO 500 J=1,NL
IF (LTA(J) .NE. IA) GO TO 480
IB=LTB(J)
YA=YAA(J)
GO TO 490
480 IF (LTB(J) .NE. IA) GO TO 500
```

```
      IB=LTA(J)
      YA=YBB(J)
490  XEB=XE(1B)
      ZL=ZAB(J)
      CGXE=CGXE+XEA*CONJG((XEA-XEB)/ZL+YA*XEA)
500  CONTINUE
      IF (M) 540,540,510
C
C   ACTIVE MEASUREMENT
C
510  IF (IPRINT) 520,520,530
520  WRITE(6,20) NT,IM,YM,WF
      WRITE(6,40)
      WRITE(6,50)(J,XE(J),P(2*J-1),P(2*J),J=1,NB)
530  RES=YM-REAL(CGXE)
      GO TO 570
C
C   REACTIVE MEASUREMENT
C
540  IF (IPRINT) 550,550,560
550  WRITE(6,30) NT,IM,YM,WF
      WRITE(6,40)
      WRITE(6,50)(J,XE(J),P(2*J-1),P(2*J),J=1,NB)
560  RES=YM-AIMAG(CGXE)
C
570  DEN=0.0
      DO 580 J=1,NNB1
          AK(J)=P(J)*G(J)
580  DEN=DEN+AK(J)*G(J)
      DEN=DEN+1/WF
      AK(NNB)=0.0
      DO 590 J=1,NB
          J2=2*J
          J1=J2-1
          AK(J1)=AK(J1)/DEN
          AK(J2)=AK(J2)/DEN
590  XE(J)=XE(J)+CMPLX(AK(J1),AK(J2))*RES
      DO 600 J=1,NNB1
600  P(J)=P(J)*(1-AK(J)*G(J))
      P(NNB)=0.0
      IF (IPRINT) 610,610,620
610  WRITE(6,60)(J,XE(J),AK(2*J-1),AK(2*J),P(2*J-1),P(2*J),J=1,NB)
C
620  CONTINUE
C
      DO 625 I=1,NNB
625  PNEW(I)=P(I)
      IF (IPRINT-1) 630,630,990
630  WRITE(6,640)
640  FORMAT(1H0,23X,17HFINAL EST AND COV)
      WRITE(6,50)(J,XE(J),P(2*J-1),P(2*J),J=1,NB)
C
990  RETURN
      END
```

SUBROUTINE ESTC(NR,DA,DB,MAXIT,EPS,NEWA,IEXIT,TIME,IPRINT)

BASIS ROUTINE FOR METHOD C. IT ESTIMATES THE BUSVOLTAGES THAT DON'T BELONG TO THE REFERENCE VECTOR. THE REFERENCE VOLTAGES AND THE LINEVOLTAGE WEIGHTING FACTORS DA(*) AND DB(*) ARE ASSUMED TO BE GIVEN

AUTHOR, TON VAN OVERBEEK 1974-03-12

NR NUMBER OF COMPLEX REFERENCE VOLTAGES
DA(*) WEIGHTING FACTOR FOR LINEVOLTAGE AT A-END
DB(*) IDEM FOR B-END
MAXIT MAXIMUM NUMBER OF ITERATIONS
EPS THE ITERATING IS TERMINATED WHEN
CABS(XE(IT + 1) - XE(IT)) .LE. EPS
NEWA=1 A NEW A-MATRIX IS CALCULATED AND DECOMPOSED
IN THE FIRST ITERATION
NEWA=0 NO NEW A-MATRIX IS CALCULATED
IEXIT=-2 DECOMPOSITION OF A HAS FAILED
IEXIT=-1 ERROR IN NB, NL OR NR
IEXIT= 1 CONVERGENCE WITHIN MAXIT ITERATIONS
IEXIT= 2 NO CONVERGENCE AFTER MAXIT ITERATIONS
TIME TOTAL TIME IN MSECS
IPRINT=4 NO PRINTOUT
IPRINT=3 INITIAL ESTIMATE, THE NUMBER OF ITERATIONS,
TOTAL TIME AND THE ESTIMATE ARE PRINTED
IPRINT=3 SAME + THE INPUT DATA: NR, DA, DB, MAXIT AND EPS
IPRINT=1 SAME + IN EACH ITERATION: THE CALCULATED
LINEVOLTAGES, THE B-VECTOR, THE ESTIMATE AND DELX
IPRINT=0 SAME + IN THE FIRST ITERATION THE A- AND
T-MATRICES

SUBROUTINE REQUIRED

DESYM
MPRI
SOLVS
UPDAC
UPDBC

PARAMETER MB=10,ML=13
PARAMETER MMB=2*MB,MML=2*ML

INTEGER TIME,TBEG,TEND

DIMENSION DA(1),DB(1),X(MB,2),B(MB,2),
1 PR(9)/'(1H0,',0,'17HCON','VERGEN','CE AFT',
2 'ER,15,',11H IT','ERATIO','NS)'/

COMPLEX YAA,ZAB,YBB,XE,XEA,XEB,ZL,CYM4(ML),CYM5(ML),
1 CLVA(ML),CLVB(ML),CB(MB),XEN(MB)

COMMON /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
X /EST/ XE(MB)

```

X      /MAT/ A(MB,MB),T(MB,MB)
X      /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
X      /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),MSK6(MB)

```

C

```

EQUIVALENCE (CYM4,YM4),(CYM5,YM5)

```

C

C

C

```

CHECK NB, NL AND NR

```

```

IF (NB) 20,20,10

```

```

10 IF (NB-MB) 40,40,20

```

```

20 IEXIT=-1

```

```

WRITE(6,30) NB

```

```

30 FORMAT(13H0NB IN ESTC =,I5)

```

```

RETURN

```

C

```

40 IF (NL) 60,60,50

```

```

50 IF (NL-ML) 80,80,60

```

```

60 IEXIT=-1

```

```

WRITE(6,70) NL

```

```

70 FORMAT(13H0NL IN ESTC =,I5)

```

```

RETURN

```

C

```

80 IF (NR) 100,100,90

```

```

90 IF (NR-NB) 120,120,100

```

```

100 IEXIT=-1

```

```

WRITE(6,110) NR

```

```

110 FORMAT(13H0NR IN ESTC =,I5)

```

```

RETURN

```

C

```

120 IF (IPRINT-2) 130,130,160

```

```

130 WRITE(6,140) NR,MAXIT,EPS,NEWA

```

```

140 FORMAT(12H0INPUT DATA:/5H0NR =,I5,5X,7HMAXIT =,I5,5X,

```

```

6SHEPS =,E7.1,5X,6HNEWA =,I5)

```

```

WRITE(6,150)(I,MSK4(I),DA(I),MSK5(I),DB(I),I=1,NL)

```

```

150 FORMAT(1H0,5X,33HALL LINEVOLTAGE WEIGHTING FACTORS/

```

```

F12H0 LINE

```

```

F,20HMSK4 DA

```

```

F,10HMSK5 DB//

```

```

F(I5,I10,F10.2,I10,F10.2))

```

```

160 IF (IPRINT-3) 170,170,190

```

```

170 WRITE(6,180)(I,XE(I),I=1,NB)

```

```

180 FORMAT(1H0,13X,16HINITIAL ESTIMATE/

```

```

F14H0BUSNR

```

```

F,10HRE(XE)

```

```

F,6HIM(XE)//

```

```

F(I5,5X,2F10.5))

```

C

C

C

```

INITIALIZATION

```

```

190 IT=0

```

```

NE=NB-NR

```

```

IF (NEWA) 220,220,200

```

```

200 DO 210 I=1,NE

```

```
DO 210 J=1,NE
210 A(I,J)=0.0
220 TBEG=MSCPU(X)
C
C   HERE STARTS THE ITERATION
C
230 IT=IT+1
DO 240 I=1,NB
240 CB(I)=(0.0,0.0)
C
C   CALCULATION OF B AND A
C
DO 320 I=1,NL
IA=LTA(I)
IB=LTB(I)
XEA=XE(IA)
XEB=XE(IB)
ZL=ZAB(I)
IF (MSK4(I)-2) 280,280,250
250 IF (NEWA) 270,270,260
260 CALL UPDAC(IA,IB,DA(I))
270 CALL UPDBC(CB,IA,IB,NR,XEA,XEB,ZL,YAA(I),CYM4(I),CLVA(I),DA(I))
280 IF (MSK5(I)-2) 320,320,290
290 IF (NEWA) 310,310,300
300 CALL UPDAC(IB,IA,DB(I))
310 CALL UPDBC(CB,IB,IA,NR,XEB,XEA,ZL,YBB(I),CYM5(I),CLVB(I),DB(I))
320 CONTINUE
C
C   ITERATION PRINTOUT
C
IF (IPRINT-1) 330,330,450
330 WRITE(6,340) IT
340 FORMAT(13H0ITERATION NR,15/1X,17(1H*)/
F1H0,21X,25HTHE COMPUTED LINEVOLTAGES/
F13H0 LINE
F,10HRE(LVA)
F,15HIM(LVA)
F,10HRE(LVB)
F,7HIM(LVB)/)
DO 410 I=1,NL
WRITE(6,350) I
350 FORMAT(I5)
IF (MSK4(I)-2) 380,380,360
360 WRITE(6,370) CLVA(I)
370 FORMAT(1H+,9X,2F10.5)
380 IF (MSK5(I)-2) 410,410,390
390 WRITE(6,400) CLVB(I)
400 FORMAT(1H+,34X,2F10.5)
410 CONTINUE
WRITE(6,420)(I,CB(I),I=1,NE)
420 FORMAT(1H0,15X,8HB-VECTOR/16X,8(1H-)//
F(15,5X,2F10.2))
IF (IT-1) 430,430,520
```

```

430 IF (IPRINT) 440,440,450
440 WRITE(6,445)
445 FORMAT(9HOMATRIX-A/1X,8(1H-)/)
CALL MPRI(A,NE,NE,NB,8,0,IE)

```

C
C
C

SOLUTION OF $A * E = B$

```

450 IF (NEWA) 520,520,460
460 CALL DESYM(A,T,NE,1.0E-7,IRANK,MB)
IF (IRANK) 470,470,490
470 WRITE(6,480)
480 FORMAT(21HODECOMPOSITION FAILED)
IEXIT=-2
RETURN

```

C

```

490 IF (IPRINT) 500,500,520
500 WRITE(6,510)
510 FORMAT(9HOMATRIX-T/1X,8(1H-)/)
CALL MPRI(T,NE,NE,MB,8,0,IE)

```

C

```

520 DO 530 I=1,NE
B(I,1)=REAL(CB(I))
530 B(I,2)=AIMAG(CB(I))
CALL SOLVS(T,B,X,NE,2,MB)
DELX=0.0
DO 540 I=1,NE
XEN(I)=CMPLX(X(I,1),X(I,2))
DELX=DELX+CABS(XEN(I)-XE(I))
540 XE(I)=XEN(I)

```

C

```

IF (IPRINT-1) 550,550,580
550 WRITE(6,560)(I,XE(I),I=1,NB)
560 FORMAT(1H0,15X,12HTHE ESTIMATE/
F14H0BUSNR
F,10HRE(XE)
F,6HIM(XE)//
F(15,5X,2F10.5))
WRITE(6,570) DELX
570 FORMAT(7HODELX =E10.3)

```

C

```

580 IF (DELX-ABS(EPS)) 600,600,590
590 IF (IT .GE. MAXIT) GO TO 610
NEWA=0
GO TO 230

```

C

C

END OF ESTIMATION, PRINTOUT OF THE RESULTS

C

```

600 IEXIT=1
PR(2)='/1H+',
GO TO 620
610 IEXIT=2
PR(2)='3HNO ,'
620 TEND=MSCPU(X)

```

TIME=TEND-TBEG

C

IF (IPRINT-3) 630,630,990

630 WRITE(6,640)

640 FORMAT(19H0ESTIMATION RESULTS/1X,18(1H-))

WRITE(6,PR) IT

WRITE(6,650) TIME

650 FORMAT(22H0TOTAL TIME FOR ESTC =,15,6H MSEC)

WRITE(6,560)(I,XE(I),I=1,NB)

C

990 RETURN

END

SUBROUTINE EVAL(K, IPRINT)

```
C
C
C      THIS SUBROUTINE COMPUTES FOR TIMEPOINT K THE FOLLOWING QUANTITIES:
C      - THE ESTIMATION ERROR
C      - THE MAXIMUM ELEMENT IN THE ESTIMATION ERROR
C      - THE LINE FLOW ERROR
C      - THE MAXIMUM ELEMENT IN THE LINE FLOW ERROR
C      - THE MEASUREMENT QUALITY INDEX
C      IT UPDATES THE FOLLOWING QUANTITIES:
C      - THE AVERAGE ESTIMATION ERROR
C      - THE MAXIMUM ESTIMATION ERROR
C      - THE MAXIMUM ELEMENT IN ALL ESTIMATION ERRORS
C      - THE AVERAGE LINE FLOW ERROR
C      - THE MAXIMUM LINE FLOW ERROR
C      - THE MAXIMUM ELEMENT IN ALL LINE FLOW ERRORS
C      - THE AVERAGE MEASUREMENT QUALITY INDEX
C      - THE MAXIMUM MEASUREMENT QUALITY INDEX
C      ALL THESE QUANTITIES ARE STORED IN THE COMMON BLOCK /EVL/
C
C      AUTHOR: TON VAN OVERBEEK 1974-02-18
C
C      K              TIME POINT
C      IPRINT=1      NO PRINTOUT
C      IPRINT=0      THE ABOVE MENTIONED QUANTITIES ARE PRINTED
C
C      SUBROUTINE REQUIRED
C              NONE
C
C      PARAMETER MB=10, ML=13, MTMX=360
C      PARAMETER MMB=2*MB, MML=2*ML, MBL=MB+MML, MMBL=2*MBL, MTMX1=MTMX+1
C
C      DIMENSION AXE(MMB), AXT(MMB), AYE4(MML), AYT4(MML), AYES(MML),
C      1             AYT5(MML), YMA(MBL), MSKA(MBL), YMB(MMBL), MSKB(MBL)
C
C      COMPLEX XE, YE2, YE3, YE4, YE5, YE6, XT, YT2, YT3, YT4, YT5, YT6,
C      1             YEA(MBL), YEB(MBL), YTA(MBL), YTB(MBL), AME
C
C      COMMON /EST/ XE(MB), YE2(ML), YE3(ML), YE4(ML), YE5(ML), YE6(MB)
C      X            /TRUE/ XT(MB), YT2(ML), YT3(ML), YT4(ML), YT5(ML), YT6(MB)
C      X            /ENET/ NB, NL
C      X            /MSI/ MSK1(MB), MSK2(ML), MSK3(ML), MSK4(ML), MSK5(ML), MSK6(MB)
C      X            /MSM/ YM1(MB), YM2(ML), YM3(ML), YM4(MML), YM5(MML), YM6(MMB)
C      X            /EVL/ EE(MTMX1), EEM(MTMX1,2), AEE, EEMT(2), EEMMT(3),
C      X             EL(MTMX1), ELM(MTMX1,2), AEL, ELMT(2), ELMMT(3),
C      X             EM(MTMX1), AEM, EMMT(2)
C
C      EQUIVALENCE (AXE, XE), (AXT, XT), (AYE4, YE4), (AYT4, YT4), (AYE5, YE5),
C      1            (AYT5, YT5), (YEA, XE), (YTA, XT), (MSKA, MSK1), (YMA, YM1),
C      2            (YEB, YE4), (YTB, YT4), (MSKB, MSK4), (YMB, YM4)
C
C      K1=K+1
C
```


C ESTIMATION ERROR

C
C
EE(K1)=0.0
EEM(K1,1)=0.0
NNB1=2*NB-1
DO 30 I=1,NNB1
AE=ABS(AXE(I)-AXT(I))
E=AE**2
EE(K1)=EE(K1)+E
IF (EEM(K1,1)-AE) 10,10,30
10 EEM(K1,1)=AE
EEM(K1,2)=FLOAT(I)
IF (EEMMT(1)-AE) 20,20,30
20 EEMMT(1)=AE
EEMMT(2)=FLOAT(I)
EEMMT(3)=FLOAT(K)
30 CONTINUE
EE(K1)=SQRT(EE(K1))
IF (EEMT(1)-EE(K1)) 40,40,50
40 EEMT(1)=EE(K1)
EEMT(2)=FLOAT(K)
50 AEE=AEE+EE(K1)

C C C LINE FLOW ERROR

EL(K1)=0.0
ELM(K1,1)=0.0
NNL=2*NL
DO 110 I=1,NNL
AE4=ABS(AYE4(I)-AYT4(I))
AE5=ABS(AYE5(I)-AYT5(I))
E4=AE4**2
E5=AE5**2
EL(K1)=EL(K1)+E4+E5
IF (ELM(K1,1)-AE4) 60,60,80
60 ELM(K1,1)=AE4
ELM(K1,2)=FLOAT(I)
IF (ELMMT(1)-AE4) 70,70,80
70 ELMMT(1)=AE4
ELMMT(2)=ELM(K1,2)
ELMMT(3)=FLOAT(K)
80 IF (ELM(K1,1)-AE5) 90,90,110
90 ELM(K1,1)=AE5
ELM(K1,2)=-FLOAT(I)
IF (ELMMT(1)-AE5) 100,100,110
100 ELMMT(1)=AE5
ELMMT(2)=ELM(K1,2)
ELMMT(3)=FLOAT(K)
110 CONTINUE
EL(K1)=SQRT(EL(K1))
IF (ELMT(1)-EL(K1)) 120,120,130
120 ELMT(1)=EL(K1)
ELMT(2)=FLOAT(K)
130 AEL=AEL+EL(K1)

C
C
C

MEASUREMENT QUALITY INDEX

```
ANOM=0.0
DENOM=0.0
DO 150 I=1,MBL
IF (MSKA(I)) 150,150,140
140 AYTA=CABS(YTA(I))
ANOM=ANOM+(CABS(YEA(I))-AYTA)**2
DENOM=DENOM+(YMA(I)-AYTA)**2
150 CONTINUE
DO 180 I=1,MBL
AME=YEB(I)-YTB(I)
IF (MSKB(I)-1) 180,170,160
160 ANOM=ANOM+AIMAG(AME)**2
DENOM=DENOM+(YMB(2*I)-AIMAG(YTB(I)))**2
IF (MSKB(I)-2) 170,180,170
170 ANOM=ANOM+REAL(AME)**2
DENOM=DENOM+(YMB(2*I-1)-REAL(YTB(I)))**2
180 CONTINUE
EM(K1)=SQRT(ANOM/DENOM)
IF (EMMT(1)-EM(K1)) 190,190,200
190 EMMT(1)=EM(K1)
EMMT(2)=FLOAT(K)
200 AEM=AEM+EM(K1)

C
C
C PRINTOUT
IF (IPRINT) 210,210,990
210 WRITE(6,220)
220 FORMAT(19H0PRINTOUT FROM EVAL/1X,18(1H*))

C
C
C ESTIMATION ERROR
WRITE(6,230) EE(K1)
230 FORMAT(17H0ESTIMATION ERROR/1X,16(1H-)//
F6X,18HESTIMATION ERROR =,F10.7)
I=INT(EEM(K1,2))
M=MOD(I,2)
J=I/2+M
IF (M) 240,240,250
240 ROI=4HIMAG
GO TO 260
250 ROI=4HREAL
260 WRITE(6,270) EEM(K1,1),ROI,J
270 FORMAT(6X,18HMAXIMUM ELEMENT =,F10.7,4H IN ,A4
F,20H PART OF BUSVOLTAGE ,I2)
IF (K .LE. 0) GO TO 335
AEEK=AEE/K
WRITE(6,280) AEEK
280 FORMAT(/6X,21HCURRENT TOTAL VALUES:/
F6X,27HAVERAGE ESTIMATION ERROR =,F10.7)
KT=INT(EEMT(2))
```

```
WRITE(6,290) EEMT(1),KT
290 FORMAT(6X,27HMAXIMUM ESTIMATION ERROR =,F10.7
F,7H AT T =,I4)
I=INT(EEMT(2))
M=MOD(I,2)
J=I/2+M
IF (M) 300,300,310
300 ROI=4HIMAG
GO TO 320
310 ROI=4HREAL
320 KT=INT(EEMT(3))
WRITE(6,330) EEMT(1),KT,ROI,J
330 FORMAT(6X,27HMAX ELEMENT IN ALL ERRORS =,F10.7
F,7H AT T =,I4,4H IN /8X,A4,20H PART OF BUSVOLTAGE ,I2)
```

C
C
C

```
LINE FLOW ERROR
335 WRITE(6,340) EL(K1)
340 FORMAT(16HOLINE FLOW ERROR/1X,15(1H-))//
F6X,17HLINE FLOW ERROR =,F10.7)
I=INT(ELM(K1,2))
IF (I) 350,350,360
350 AOB=1HB
GO TO 370
360 AOB=1HA
370 I=IABS(I)
M=MOD(I,2)
J=I/2+M
IF (M) 380,380,390
380 AOR=6HREACT.
GO TO 400
390 AOR=6HACTIVE
400 WRITE(6,410) ELM(K1,1),AOR,AOB,J
410 FORMAT(6X,17HMAXIMUM ELEMENT =,F10.7,4H IN ,A6
F,9H FLOW AT ,A1,12H-END OF LINE ,I2)
IF (K .LE. 0) GO TO 505
AELK=AEL/K
WRITE(6,420) AELK
420 FORMAT(/6X,21HCURRENT TOTAL VALUES:/
F6X,27HAVERAGE LINE FLOW ERROR =,F10.7)
KT=INT(ELMT(2))
WRITE(6,430) ELMT(1),KT
430 FORMAT(6X,27HMAXIMUM LINE FLOW ERROR =,F10.7
F,7H AT T =,I4)
I=INT(ELMT(2))
IF (I) 440,440,450
440 AOB=1HB
GO TO 460
450 AOB=1HA
460 I=IABS(I)
M=MOD(I,2)
J=I/2+M
IF (M) 470,470,480
```

```
470 AOR=6HREACT.  
    GO TO 490  
480 AOR=6HACTIVE  
490 KT=INT(ELMMT(3))  
    WRITE(6,500) ELMMT(1),KT,AOR,AOB,J  
500 FORMAT(6X,27HMAX ELEMENT IN ALL ERRORS =,F10.7  
    F,7H AT T =,I4,4H IN /8X,A6,9H FLOW AT ,A1,13H-END OF LINE ,I2)
```

C
C
C

```
    MEASUREMENT QUALITY INDEX  
505 WRITE(6,510) EM(K1)  
510 FORMAT(20HMEASUREMENT QUALITY/1X,19(1H-)//  
    F6X,19HMEASUREMENT INDEX =,F10.7)  
    IF (K .LE. 0) GO TO 990  
    AEMK=AEM/K  
    WRITE(6,520) AEMK  
520 FORMAT(/6X,21HCURRENT TOTAL VALUES:/  
    F6X,20HAVERAGE MSMT INDEX =,F10.7)  
    KT=INT(EMMT(2))  
    WRITE(6,530) EMMT(1),KT  
530 FORMAT(6X,20HMAXIMUM MSMT INDEX =,F10.7  
    F,7H AT T =,I4)
```

C

```
990 RETURN  
    END
```

SUBROUTINE GXE(IPRINT,IERR)

```
C
C COMPUTES YE FROM ESTIMATOR NETWORK DATA AND ESTIMATED STATE
C AUTHOR: TGN VAN OVERBEEK 1973-12-21
C
C IPRINT=2      NO PRINTOUT
C IPRINT=1      XE AND YE ARE PRINTED
C IPRINT=0      XE, YE, AND TRUE NETWORK ARE PRINTED
C IERR=1        ERROR IN NB OR NL
C IERR=0        NO ERROR
C SUBROUTINE REQUIRED
C PRENET
C
C PARAMETER MB=10,ML=13
C
C COMPLEX YAA,ZAB,YBB,X,Y2,Y3,Y4,Y5,Y6,DI
C
C COMMON /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
C 1 /EST/ X(MB),Y2(ML),Y3(ML),Y4(ML),Y5(ML),Y6(MB)
C
C IERR=0
C IF(NB .GT. 0 .AND. NB .LE. MB) GO TO 20
C IERR=1
C WRITE(6,10) NB
C 10 FORMAT(12H0NB IN GXE =,I5)
C RETURN
C
C 20 IF (NL .GT. 0 .AND. NL .LE. ML) GO TO 40
C IERR=1
C WRITE(6,30) NL
C 30 FORMAT(12H0NL IN GXE =,I5)
C RETURN
C
C 40 IF(IPRINT .GE. 2) GO TO 100
C WRITE(6,60)
C 60 FORMAT(18H0PRINTOUT FROM GXE/1X,17(1H*))
C IF (IPRINT .GE. 1) GO TO 100
C CALL PRENET
C
C 100 DO 110 I=1,NB
C 110 Y6(I)=(0.0,0.0)
C DO 120 I=1,NL
C IA=LTA(I)
C IB=LTB(I)
C DI=(X(IA)-X(IB))/ZAB(I)
C Y2(I)=DI+X(IA)*YAA(I)
C Y3(I)=-DI+X(IB)*YBB(I)
C Y4(I)=X(IA)*CONJG(Y2(I))
C Y5(I)=X(IB)*CONJG(Y3(I))
C Y6(IA)=Y6(IA)+Y4(I)
C Y6(IB)=Y6(IB)+Y5(I)
C 120 CONTINUE
```

C

```
      IF (IPRINT .GE. 2) GO TO 990
      WRITE(6,130)
130  FORMAT(//14H0BUSNR
      F,10HRE(XE)
      F,10HIM(XE)
      F,10HPINJ
      F,4HQINJ/)
      WRITE(6,140)(I,X(I),Y6(I),I=1,NB)
140  FORMAT(15,5X,4F10.5)
      WRITE(6,150)
150  FORMAT(20H0 LINE  A-END B-END
      F,10H RE(IAB)
      F,10H IM(IAB)
      F,10HMOD(IAB)
      F,10H RE(IBA)
      F,10H IM(IBA)
      F,10HMOD(IBA)
      F,10H  PAB
      F,10H  QAB
      F,10H  PBA
      F,5H  QBA/)
      DO 170 I=1,NL
      CMY2=CABS(Y2(I))
      CMY3=CABS(Y3(I))
      WRITE(6,160) I,LTA(I),LTB(I),Y2(I),CMY2,Y3(I),CMY3,Y4(I),Y5(I)
160  FORMAT(15,1X,2I6,10F10.5)
170  CONTINUE
C
990  RETURN
      END
```

```

SUBROUTINE GXT(IPRINT,IERR)
C
C COMPUTES YT FROM TRUE NETWORK DATA AND TRUE STATE
C AUTHOR: TON VAN OVERBEEK 1973-12-21
C
C IPRINT=2      NO PRINTOUT
C IPRINT=1      XT AND YT ARE PRINTED
C IPRINT=0      XT, YT, AND TRUE NETWORK ARE PRINTED
C IERR=1        ERROR IN NB OR NL
C IERR=0        NO ERROR
C SUBROUTINE REQUIRED
C          PRTNET
C
C PARAMETER MB=10,ML=13
C
C COMPLEX YAA,ZAB,YBB,X,Y2,Y3,Y4,Y5,Y6,DI
C
C COMMON /TNET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
1  /TRUE/ X(MB),Y2(ML),Y3(ML),Y4(ML),Y5(ML),Y6(MB)
C
C IERR=0
C IF(NB .GT. 0 .AND. NB .LE. MB) GO TO 20
C IERR=1
C WRITE(6,10) NB
10 FORMAT(12H0NB IN GXT =,15)
C RETURN
C
C 20 IF (NL .GT. 0 .AND. NL .LE. ML) GO TO 40
C IERR=1
C WRITE(6,30) NL
30 FORMAT(12H0NL IN GXT =,15)
C RETURN
C
C 40 IF(IPRINT .GE. 2) GO TO 100
C WRITE(6,60)
60 FORMAT(18H0PRINTOUT FROM GXT/1X,17(1H*))
C IF (IPRINT .GE. 1) GO TO 100
C CALL PRTNET
C
C 100 DO 110 I=1,NB
110 Y6(I)=(0.0,0.0)
C DO 120 I=1,NL
C IA=LTA(I)
C IB=LTB(I)
C DI=(X(IA)-X(IB))/ZAB(I)
C Y2(I)=DI+X(IA)*YAA(I)
C Y3(I)=-DI+X(IB)*YBB(I)
C Y4(I)=X(IA)*CONJG(Y2(I))
C Y5(I)=X(IB)*CONJG(Y3(I))
C Y6(IA)=Y6(IA)+Y4(I)
C Y6(IB)=Y6(IB)+Y5(I)
120 CONTINUE

```

```
C
  IF (IPRINT .GE. 2) GO TO 990
  WRITE(6,130)
130 FORMAT(//14H0BUSNR
  F,10HRE(XT)
  F,10HIM(XT)
  F,10HPINJ
  F,4HQINJ/)
  WRITE(6,140)(I,X(I),Y6(I),I=1,NB)
140 FORMAT(15,5X,4F10.5)
  WRITE(6,150)
150 FORMAT(20H0 LINE  A-END B-END
  F,10H RE(IAB)
  F,10H IM(IAB)
  F,10HMOD(IAB)
  F,10H RE(IBA)
  F,10H IM(IBA)
  F,10HMOD(IBA)
  F,10H PAB
  F,10H QAB
  F,10H PBA
  F,5H QBA/)
  DO 170 I=1,NL
  CMY2=CABS(Y2(I))
  CMY3=CABS(Y3(I))
  WRITE(6,160) I,LTA(I),LTB(I),Y2(I),CMY2,Y3(I),CMY3,Y4(I),Y5(I)
160 FORMAT(15,1X,216,10F10.5)
170 CONTINUE
C
990 RETURN
  END
```



```

SUBROUTINE JACBI(M,IA,DPI,IRP,DQI,IRQ,ID,IERR)
C
C COMPUTES THE JACOBIAN ROW(S) FOR AN INJECTION MEASUREMENT
C AT BUS IA
C
C AUTHOR, TON VAN OVERBEEK 1974-01-24
C
C M=1 ACTIVE MEASUREMENT ONLY
C M=2 REACTIVE MEASUREMENT ONLY
C M=3 BOTH MEASUREMENTS
C IA BUS INJECTION MEASUREMENT AT BUS IA
C DPI JACOBIAN ROW FOR ACTIVE MEASUREMENT, LENGTH 2*NB
C IRP ROWNR FOR DPI
C DQI JACOBIAN ROW FOR REACTIVE MEASUREMENT, LENGTH 2*NB
C IRQ ROWNR FOR DQI
C ID DIMENSION PARAMETER
C IERR=3 ERROR IN M
C IERR=2 ERROR IN IA
C IERR=1 DIMENSION ERROR
C IERR=0 NO ERROR
C
C SUBROUTINE REQUIRED
C JACLF
C
C PARAMETER MB=10,ML=13
C
C COMPLEX X,YAA,ZAB,YBB
C
C DIMENSION DP(4),DQ(4),DPI(ID,1),DQI(ID,1)
C
C COMMON /EST/ X(MB)
C 1 /ENET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
C
C IERR=0
C IF (M) 20,20,10
10 IF (M-3) 40,40,20
20 IERR=3
C WRITE(6,30) M
30 FORMAT(13H0M IN JACBI =,I5)
C RETURN
C
40 IF (IA) 60,60,50
50 IF (IA-NB) 80,80,60
60 IERR=2
C WRITE(6,70) IA,NB
70 FORMAT(14H0IA IN JACBI =,I5,5X,13HNB IN JACBI =,I5)
C RETURN
C
80 IF (IRP) 100,100,90
90 IF (IRP-ID) 120,120,100
100 IERR=1
C WRITE(6,110) IRP,ID
110 FORMAT(15H0IRP IN JACBI =,I5,5X,13HID IN JACBI =,I5)
C RETURN
```

```
C
120 IF (IRQ) 140,140,130
130 IF (IRQ-ID) 160,160,140
140 IERR=1
    WRITE(6,150) IRQ,ID
150 FORMAT(15H0IRQ IN JACBI =,I5,5X,13HID IN JACBI =,I5)
    RETURN
C
160 NNB=2*NB
    DO 170 I=1,NNB
        DPI(IRP,I)=0.0
170 DQI(IRQ,I)=0.0
C
    DO 230 I=1,NL
        IF (IA .EQ. LTA(I)) GO TO 180
        IF (IA .EQ. LTB(I)) GO TO 190
        GO TO 230
C
180 IB=LTB(I)
    CALL JACLF(M,X(IA),X(IB),ZAB(I),YAA(I),DP,1,DQ,1,1,IE)
    GO TO 200
C
190 IB=LTA(I)
    CALL JACLF(M,X(IA),X(IB),ZAB(I),YBB(I),DP,1,DQ,1,1,IE)
C
200 IA2=2*IA
    IA1=IA2-1
    IB2=2*IB
    IB1=IB2-1
    IF (M-1) 220,220,210
210 DQI(IRQ,IA1)=DQI(IRQ,IA1)+DQ(1)
    DQI(IRQ,IA2)=DQI(IRQ,IA2)+DQ(2)
    DQI(IRQ,IB1)=DQ(3)
    DQI(IRQ,IB2)=DQ(4)
    IF (M-2) 220,230,220
220 DPI(IRP,IA1)=DPI(IRP,IA1)+DP(1)
    DPI(IRP,IA2)=DPI(IRP,IA2)+DP(2)
    DPI(IRP,IB1)=DP(3)
    DPI(IRP,IB2)=DP(4)
C
230 CONTINUE
C
    RETURN
    END
```

```

SUBROUTINE JACI(UA,UB,ZAB,YAA,DI,IR,ID,IERR)
C
C COMPUTES THE JACOBIAN ELEMENTS FOR A LINE CURRENT MEASUREMENT
C AT LINE END A
C
C AUTHOR, TON VAN OVERBEEK 1974-01-24
C
C UA          BUSVOLTAGE AT A END
C UB          BUSVOLTAGE AT B END
C ZAB         LINE SERIES IMPEDANCE
C YAA        LINE SHUNT ADMITTANCE AT A END
C DI(IR,1)   ELEMENT DI/DEA
C DI(IR,2)   ELEMENT DI/DFB
C DI(IR,3)   ELEMENT DI/DEB
C DI(IR,4)   ELEMENT DI/DFB
C IR         ROW IN DI
C ID         DIMENSION PARAMETER
C IERR=1     DIMENSION ERROR
C IERR=0     NO ERROR
C
C SUBROUTINE REQUIRED
C          NONE
C
C COMPLEX UA,UB,ZAB,YAA,Y,X
C
C DIMENSION DI(ID,1)
C
C IERR=0
C 1F (IR) 20,20,10
10 IF (IR-ID) 40,40,20
20 IERR=1
C WRITE(6,30) IR, ID
30 FORMAT(13H0IR IN JACI =,I5,5X,12HID IN JACI =,I5)
C RETURN
C
40 Y=(UA-UB)/ZAB+UA*YAA
C AMY=CABS(Y)
C X=CONJG(Y)/(ZAB*AMY)
C DI(IR,3)=-REAL(X)
C DI(IR,4)=AIMAG(X)
C X=X+CONJG(Y)*YAA/AMY
C DI(IR,1)=REAL(X)
C DI(IR,2)=-AIMAG(X)
C RETURN
C END
```

```

SUBROUTINE JACLF(M,UA,UB,ZAB,YAA,DP,IRP,DQ,IRQ,ID,IERR)
C
C COMPUTES THE JACOBIAN ELEMENTS FOR A LINE FLOW MEASUREMENT
C AT LINE END A
C
C AUTHOR, TON VAN OVERBEEK 1974-01-24
C
C M=1 ACTIVE MEASUREMENT ONLY
C M=2 REACTIVE MEASUREMENT ONLY
C M=3 BOTH MEASUREMENTS
C UA BUSVOLTAGE AT A END
C UB BUSVOLTAGE AT B END
C ZAB LINE SERIES IMPEDANCE
C YAA LINE SHUNT ADMITTANCE AT A END
C DP(IRP,1) ELEMENT DP/DEA
C DP(IRP,2) ELEMENT DP/DFA
C DP(IRP,3) ELEMENT DP/DEB
C DP(IRP,4) ELEMENT DP/DFB
C IRP ROW IN DP
C DQ(IRQ,1) ELEMENT DQ/DEA
C DQ(IRQ,2) ELEMENT DQ/DFA
C DQ(IRQ,3) ELEMENT DQ/DEB
C DQ(IRQ,4) ELEMENT DQ/DFB
C IRQ ROW IN DQ
C ID DIMENSION PARAMETER
C IERR=2 ERROR IN M
C IERR=1 DIMENSION ERROR
C IERR=0 NO ERROR
C
C SUBROUTINE REQUIRED
C NONE
C
C COMPLEX UA,UB,ZAB,YAA,CX1,CX2,CX,Y
C
C DIMENSION DP(ID,1),DQ(ID,1)
C
C IERR=0
C IF (IRP) 20,20,10
10 IF (IRP-ID) 40,40,20
20 IERR=1
C WRITE(6,30) IRP,ID
30 FORMAT(15H0IRP IN JACLF =,I5,5X,13HID IN JACLF =,I5)
C RETURN
C
C 40 IF (IRQ) 60,60,50
50 IF (IRQ-ID) 80,80,60
60 IERR=1
C WRITE(6,70) IRQ,ID
70 FORMAT(15H0IRQ IN JACLF =,I5,5X,13HID IN JACLF =,I5)
C RETURN
C
C 80 IF (M) 100,100,90
90 IF(M-3) 120,120,100
```

```
100 IERR=2
    WRITE(6,110) M
110 FORMAT(13H0M IN JACLF =,I5)
    RETURN
C
120 CX1=-UB/ZAB
    CX2=CONJG(UA)/ZAB
    Y=1/ZAB+YAA
    IF (M-1) 140,140,130
C
C    REACTIVE LINE FLOW
C
130 CX=-2*UA*AIMAG(Y)
    DQ(IRQ,1)=-AIMAG(CX1)+REAL(CX)
    DQ(IRQ,2)=REAL(CX1)+AIMAG(CX)
    DQ(IRQ,3)=AIMAG(CX2)
    DQ(IRQ,4)=REAL(CX2)
    IF (M-2) 140,990,140
C
C    ACTIVE LINE FLOW
C
140 CX=2*UA*REAL(Y)
    CX=CX+CX1
    DP(IRP,1)=REAL(CX)
    DP(IRP,2)=AIMAG(CX)
    DP(IRP,3)=-REAL(CX2)
    DP(IRP,4)=AIMAG(CX2)
C
990 RETURN
    END
```

```

SUBROUTINE JACV(U,DU,IR,ID,IERR)
C
C COMPUTES THE JACOBIAN ELEMENTS FOR A VOLTAGE MEASUREMENT
C
C AUTHOR, TON VAN OVERBEEK 1974-01-24
C
C U          BUSVOLTAGE
C DU(IR,1)   ELEMENT DU/DE
C DU(IR,2)   ELEMENT DU/DF
C IR         ROW IN DU
C ID        DIMENSION PARAMETER
C IERR=1     DIMENSION ERROR
C IERR=0     NO ERROR
C
C SUBROUTINE REQUIRED
C          NONE
C
C COMPLEX U,X
C
C DIMENSION DU(ID,1)
C
C IERR=0
C IF (IR) 20,20,10
10 IF (IR-ID) 40,40,20
20 IERR=1
C WRITE(6,30) IR,ID
30 FORMAT(13H0IR IN JACV =,I5,5X,12HID IN JACV =,I5)
C RETURN
C
40 X=U/CABS(U)
C DU(IR,1)=REAL(X)
C DU(IR,2)=AIMAG(X)
C RETURN
C END
```

SUBROUTINE NRLFR(NB,NL,LTA,LTB,YAA,ZAB,YBB,SINJ,VB,EPS,
IIS,MAXIT,IPRINT,JFAIL)

COMPUTES A SOLUTION TO

* THE LOAD-FLOW PROBLEM USING *
* THE NEWTON-RAPHSON METHOD *
* AND RECTANGULAR COORDINATES *

REFERENCE, G.W. STAGG AND A.H. EL-ABIAD
'COMPUTER METHODS IN POWER SYSTEM ANALYSIS'
CHAPTER 8, NEW YORK, 1968

AUTHOR, STURE LINDAHL 1972-03-12
REVISED FOR SEIPS, TON VAN OVERBEEK 1974-01-14

YAA(*) SHUNT ADMITTANCE AT ENDPOINT A
ZAB(*) LINE IMPEDANCE BETWEEN ENDPOINT A AND B
YBB(*) SHUNT ADMITTANCE AT ENDPOINT B
LTA(*) ENDPOINT A OF LINE I IS CONNECTED TO BUS LTA(I)
LTB(*) ENDPOINT B OF LINE I IS CONNECTED TO BUS LTB(I)
SINJ(*) COMPLEX POWER INJECTION AT BUS *
VB(*) COMPLEX BUSVOLTAGES
EPS THE ITERATION IS TERMINATED WHEN
THE MAXIMUM APPARENT POWER MISMATCH IS LESS THAN EPS
NB NUMBER OF BUSES (MAX 10)
NL NUMBER OF LINES (NO MAX)
IS SLACK BUS NUMBER
MAXIT MAXIMUM NUMBER OF ITERATIONS
IPRINT=3 NO PRINTOUT
IPRINT=2 INPUT DATA AND THE FINAL RESULT IS PRINTED
IPRINT=1 SAME + APPARENT POWER MISMATCH AND
BUSVOLTAGES AT EACH ITERATION
IPRINT=0 SAME + Y-BUS MATRIX AND AT EACH ITERATION
THE JACOBIAN AND VOLTAGE CORRECTIONS
JFAIL=-1 NO CONVERGENCE AFTER MAXIT ITERATIONS
JFAIL=0 THE SOLUTION IS FOUND
JFAIL=1 ERROR IN NB OR NL
JFAIL=2 THE JACCBIAN IS SINGULAR

SUBROUTINE REQUIRED

DECOM
MPRI
SOLVB

PARAMETER MB=10, MX=2*(MB-1), MMB=2*MB

COMPLEX YAA(1),ZAB(1),YBB(1),SINJ(1),VB(1),
IY(MB,MB),SMM(MB),DV(MB),IB(MB),SA,SB,SI,SL,ETT,NOLL

```

C   DIMENSION LTA(1),LTB(1),INDEX(MB),A(MX,MX),B(MX),X(MX)
C   DATA NOLL/(0.0,0.0)/,ETT/(1.0,0.0)/,LP/6/,JJAC/1/,EPSJ/1.0E-7/
C   IF(NB.GT.MB) GO TO 10
C   IF(NB) 10,10,30
10  WRITE(LP,20) NB
20  FORMAT(5H NB =,15,9H IN NRLFR)
C   JFAIL=1
C   GO TO 990
C
30  IF(NL) 40,40,60
40  WRITE(LP,50) NL
50  FORMAT(5H NL =,15,9H IN NRLFR)
C   JFAIL=1
C   GO TO 990
C
60  IF(IPRINT-2) 70,70,200
70  WRITE(LP,80)
80  FORMAT(20HOPRINTOUT FROM NRLFR/1X,19(1H*)//
210H      LINE
3,10H     A-END
4,10H     B-END,5X
5,15H     GAA
6,15H     BAA
7,15H     RAB
8,15H     XAB
9,15H     GBB
1,15H     BBB
2)
DO 90 I=1,NL
90  WRITE(LP,100) I,LTA(I),LTB(I),YAA(I),ZAB(I),YBB(I)
100  FORMAT(3I10,6F15.5)
C   WRITE(LP,110)
110  FORMAT(42H0INITIAL BUS VOLTAGES AND POWER INJECTIONS/
210H      BUS
2,15H     REAL(VB(I))
3,15H     IMAG(VB(I))
4,15H     PINJ(I)
5,15H     QINJ(I)
6)
DO 160 I=1,NB
IF(I-15) 120,140,120
120  WRITE(LP,130) I,VB(I),SINJ(I)
130  FORMAT(I10,4F15.5)
C   GO TO 160
140  WRITE(LP,150) I,VB(I)
150  FORMAT(I10,2F15.5,5X,10H SLACK BUS)
160  CONTINUE
```



```
C
C   FORM YBUS-MATRIX
C
200 DO 210 I=1,NB
    DO 210 J=1,NB
210  Y(I,J)=NOLL
    DO 220 I=1,NL
        SL=ETT/ZAB(I)
        SA=SL+YAA(I)
        SB=SL+YBB(I)
        II=LTA(I)
        JJ=LTB(I)
        Y(II,JJ)=Y(II,JJ)-SL
        Y(JJ,II)=Y(JJ,II)-SL
        Y(II,II)=Y(II,II)+SA
220  Y(JJ,JJ)=Y(JJ,JJ)+SB
        IF(IPRINT) 230,230,250
230  WRITE(LP,240)
240  FORMAT(12H0YBUS-MATRIX/1X,11(1H-)/
F54HONOTE: REAL PART OF Y(I,J) IS LISTED ON PLACE 2*I-1,J/
F52H          IMAG PART OF Y(I,J) IS LISTED ON PLACE 2*I,J//)
        NNB=2*NB
        CALL MPRI(Y,NNB,NB,MMB,8,0,IE)
C
C   COMPUTE BUS INDEX
C
250  II=1
    DO 270 I=1,NB
        IF(I-IS) 260,270,260
260  INDEX(II)=I
        II=II+1
270  CONTINUE
        IJAC=JJAC-1
        JFAIL=0
        NX=NB-1
        NXX=2*NX
        DV(IS)=NOLL
C
C   START THE ITERATION
C
    DO 570 K=1,MAXIT
C
C   COMPUTE APPARENT POWER MISMATCH
C
        SMMM=0.0
        DO 340 I=1,NB
            SL=NOLL
            DO 310 J=1,NB
310  SL=SL+Y(I,J)*VB(J)
                IB(I)=SL
                SI=VB(I)*CONJG(SL)
                IF(I-IS) 330,320,330
320  SINJ(I)=SI
```

```
330 SMM(I)=SINU(I)-SI
340 SMMM=AMAX1(SMMM,CABS(SMM(I)))
   IF(IPRINT-1) 350,350,390
350 WRITE(LP,360) K,(VB(I),I=1,NB)
360 FORMAT(/19H ITERATION NUMBER =,I5/1X,18(1H-)/
110X,12HBUS VOLTAGES/(10X,10F12.5))
   WRITE(LP,370) (SMM(I),I=1,NB)
370 FORMAT(/10X,23HAPPARENT POWER MISMATCH/(10X,10F12.5))
   WRITE(LP,380) SMMM
380 FORMAT(/10X,31HMAXIMUM APPARENT POWER MISMATCH/10X,2F12.5)
390 IF(SMMM-EPS) 600,600,400
```

C
C
C

 CALCULATE THE ELEMENTS OF THE JACCOBIAN IF IJAC=JJAC

```
400 IJAC=IJAC+1
   IF(JJAC-IJAC) 410,410,470
410 DO 440 I=1,NX
   II=INDEX(I)
   DO 430 J=1,NX
   JJ=INDEX(J)
   SL=VB(II)*CONJG(Y(II,JJ))
   A(I,J)=REAL(SL)
   A(I,J+NX)=AIMAG(SL)
   A(I+NX,J)=AIMAG(SL)
   A(I+NX,J+NX)=-REAL(SL)
   IF(I-J) 430,420,430
420 A(I,1)=A(I,I)+REAL(IB(II))
   A(I,I+NX)=A(I,1+NX)+AIMAG(IB(II))
   A(I+NX,1)=A(I+NX,1)-AIMAG(IB(II))
   A(I+NX,I+NX)=A(I+NX,I+NX)+REAL(IB(II))
430 CONTINUE
440 CONTINUE
   IJAC=0
   IF(IPRINT) 450,450,470
450 WRITE(LP,460)
460 FORMAT(1H0,9X,13HTHE JACCOBIAN/10X,13(1H-)/)
   CALL MPRI(A,NXX,NXX,MX,8,0,IE)
470 DO 480 I=1,NX
   II=INDEX(I)
   B(I)=REAL(SMM(II))
480 B(I+NX)=AIMAG(SMM(II))
```

C
C
C

 SOLVE THE EQUATION A*DV=SMMM

```
500 CALL DECOM(A,NXX,MX,EPSU,ISING)
   IF(ISING) 510,530,510
510 JFAIL=2
   WRITE(LP,520)
520 FORMAT(26H THE JACCOBIAN IS SINGULAR)
   GO TO 990
C
530 CALL SOLVB(B,X,NXX,1,MX)
   DO 540 I=1,NX
```

```
      II=INDEX(I)
      DV(II)=CMPLX(X(I),X(I+NX))
540  VB(II)=VB(II)+DV(II)
      IF(IPRINT) 550,550,570
550  WRITE(LP,560) (DV(I),I=1,NB)
560  FORMAT(1H0,9X,19HVOLTAGE CORRECTIONS/(10X,10F12.5))
570  CONTINUE
      WRITE(LP,580) MAXIT
580  FORMAT(22H NO CONVERGENCE AFTER,I5,11H ITERATIONS)
      JFAIL=-1
      GO TO 990
```

C
C
C

```
      PRINT OUT THE RESULTS
600  IF(IPRINT-2) 610,610,990
610  WRITE(LP,620)
620  FORMAT(33HORESULT OF LOAD-FLOW CALCULATIONS/1X,32(1H-)/
      110H      BUS
      2,15H      REAL(VB(I))
      3,15H      IMAG(VB(I))
      4,15H      PINJ(I)
      5,15H      QINJ(I)
      6,15H      DELP(I)
      7,15H      DELQ(I)
      8)
      DO 630 I=1,NB
630  WRITE(LP,640) I,VB(I),SINJ(I),SMM(I)
640  FORMAT(110,6F15.5)
990  RETURN
      END
```



```

SUBROUTINE PRJAC
C
C PRINTS THE JACOBIAN AS STORED IN THE COMMON BLOCK /JAC/
C
C AUTHOR, TON VAN OVERBEEK 1974-01-24
C
C SUBROUTINE REQUIRED
C     NONE
C
C PARAMETER MB=10,ML=13
C PARAMETER MMB=2*MB,MML=2*ML
C
C COMMON /JAC/ AJAC1(MB,2),AJAC2(ML,4),AJAC3(ML,4),AJAC4(MML,4),
1             AJAC5(MML,4),AJAC6(MMB,MMB)
2             /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),MSK6(MB)
3             /ENET/ NB,NL,LTA(ML),LTB(ML)
C
C WRITE(6,10)
10 FORMAT(13H0THE JACOBIAN/1X,12(1H*))
C
C TYPE 1
C
C WRITE(6,20)
20 FORMAT(7H0TYPE 1/14H0MSMNR
F,10HDV/DE
F,5HDV/DF/)
DO 50 I=1,NB
IF (MSK1(I)) 50,50,30
30 WRITE(6,40) I,(AJAC1(I,J),J=1,2)
40 FORMAT(I5,5X,2F10.5)
50 CONTINUE
C
C TYPE 2
C
C WRITE(6,60)
60 FORMAT(7H0TYPE 2/24H0MSMNR LTA LTB
F,10HDI/DEA
F,10HDI/DFA
F,10HDI/DEB
F,6HDI/DFB/)
DO 90 I=1,NL
IF (MSK2(I)) 90,90,70
70 WRITE(6,80) I,LTA(I),LTB(I),(AJAC2(I,J),J=1,4)
80 FORMAT(3I5,5X,4F10.5)
90 CONTINUE
C
C TYPE 3
C
C WRITE(6,100)
100 FORMAT(7H0TYPE 3/24H0MSMNR LTB LTA
F,10HDI/DEB
F,10HDI/DFB
F,10HDI/DEA
```

```
F,6HDI/DFA/)  
DO 120 I=1,NL  
IF (MSK3(I)) 120,120,110  
110 WRITE(6,80) I,LTB(I),LTA(I),(AJAC3(I,J),J=1,4)  
120 CONTINUE
```

C
C
C

```
TYPE 4  
  
WRITE(6,130)  
130 FORMAT(7H0TYPE 4/24H0MSMNR LTA LTB  
F,10HDP/DEA  
F,10HDP/DFA  
F,10HDP/DEB  
F,15HDP/DFB  
F,10HDQ/DEA  
F,10HDQ/DFA  
F,10HDQ/DEB  
F,6HDQ/DFB/)  
DO 200 I=1,NL  
I2=2*I  
I1=I2-1  
MS=MSK4(I)  
IF (MS) 200,200,140  
140 WRITE(6,150) I,LTA(I),LTB(I)  
150 FORMAT(3I5)  
IF (MS-2) 160,180,160  
160 WRITE(6,170)(AJAC4(I1,J),J=1,4)  
170 FORMAT(1H+,19X,4F10.5)  
IF (MS-1) 200,200,180  
180 WRITE(6,190)(AJAC4(I2,J),J=1,4)  
190 FORMAT(1H+,64X,4F10.5)  
200 CONTINUE
```

C
C
C

```
TYPE 5  
  
WRITE(6,210)  
210 FORMAT(7H0TYPE 5/24H0MSMNR LTB LTA  
F,10HDP/DEB  
F,10HDP/DFB  
F,10HDP/DEA  
F,15HDP/DFA  
F,10HDQ/DEB  
F,10HDQ/DFB  
F,10HDQ/DEA  
F,6HDQ/DFA/)  
DO 250 I=1,NL  
I2=2*I  
I1=I2-1  
MS=MSK5(I)  
IF (MS) 250,250,220  
220 WRITE(6,150) I,LTB(I),LTA(I)  
IF (MS-2) 230,240,230  
230 WRITE(6,170)(AJAC5(I1,J),J=1,4)
```

```
      IF (MS-1) 250,250,240
240 WRITE(6,190)(AJAC5(I2,J),J=1,4)
250 CONTINUE
C
C   TYPE 6
C
      WRITE(6,260)
260 FORMAT(7H0TYPE 6/)
      DO 350 I=1,NB,5
      IA=2*I-1
      IB=MIN(IA+9,2*NB)
      I4=MIN(I+4,NB)
      WRITE(6,270)(J,J,J=I,I4)
270 FORMAT(11H0D/DE D/DF:,10I10/8H   MSMNR)
      DO 340 J=1,NB
      J2=2*J
      J1=J2-1
      MS=MSK6(J)
      IF (MS) 340,340,280
280 WRITE(6,290)
290 FORMAT(1H )
      IF (MS-2) 300,320,300
300 WRITE(6,310) J,(AJAC6(J1,K),K=IA,IB)
310 FORMAT(15,4HACT:,6X,10F10.5)
      IF (MS-1) 340,340,320
320 WRITE(6,330) J,(AJAC6(J2,K),K=IA,IB)
330 FORMAT(15,6HREACT:,4X,10F10.5)
340 CONTINUE
350 CONTINUE
C
      RETURN
      END
```

SUBROUTINE PRRES

PRINTS THE USED MEASUREMENTS, THE CORRESPONDING ESTIMATED VALUES
AND THE RESIDUES.

AUTHOR, TON VAN OVERBEEK 1974-01-22

SUBROUTINE REQUIRED
NONE

PARAMETER MB=10,ML=13
PARAMETER MMB=2*MB,MML=2*ML

COMPLEX XE, YE2, YE3, YE4, YE5, YE6

COMMON /ENET/ NB,NL

1 /EST/ XE(MB), YE2(ML), YE3(ML), YE4(ML), YE5(ML), YE6(MB)
2 /MSM/ YM1(MB), YM2(ML), YM3(ML), YM4(MML), YM5(MML), YM6(MMB)
3 /MSI/ MSK1(MB), MSK2(ML), MSK3(ML), MSK4(ML), MSK5(ML), MSK6(MB)
4 /RES/ RES1(MB), RES2(ML), RES3(ML), RES4(MML), RES5(MML),
5 RES6(MMB)

WRITE(6,10)
10 FORMAT(13H0THE RESIDUES/1X,12(1H*)/14H0MSMNR
F,10HMSM
F,10HEST
F,3HRES/
F7H0TYPE 1/)
DO 40 I=1,NB
IF (MSK1(I)) 40,40,20
20 YA=CABS(XE(I))
WRITE(6,30) I, YM1(I), YA, RES1(I)
30 FORMAT(15,5X,3F10.5)
40 CONTINUE
WRITE(6,50)
50 FORMAT(7H0TYPE 2/)
DO 70 I=1,NL
IF (MSK2(I)) 70,70,60
60 YA=CABS(YE2(I))
WRITE(6,30) I, YM2(I), YA, RES2(I)
70 CONTINUE
WRITE(6,80)
80 FORMAT(7H0TYPE 3/)
DO 100 I=1,NL
IF (MSK3(I)) 100,100,90
90 YA=CABS(YE3(I))
WRITE(6,30) I, YM3(I), YA, RES3(I)
100 CONTINUE
WRITE(6,110)
110 FORMAT(1H0,23X,33HACTIVE
F,8HREACTIVE/
F14H0MSMNR
F,10HMSM


```
F,10HEST
F,15HRES
F,10HMSM
F,10HEST
F,3HRES/
F7HOTYPE 4/)
DO 180 I=1,NL
IF (MSK4(I)) 180,180,120
120 I2=2*I
I1=I2-1
YR=REAL(YE4(I))
YI=AIMAG(YE4(I))
WRITE(6,130) I
130 FORMAT(I5)
IF (MSK4(I)-2) 140,160,140
140 WRITE(6,150) YM4(I1),YR,RES4(I1)
150 FORMAT(1H+,9X,3F10.5)
IF (MSK4(I)-1) 180,180,160
160 WRITE(6,170) YM4(I2),YI,RES4(I2)
170 FORMAT(1H+,44X,3F10.5)
180 CONTINUE
WRITE(6,185)
185 FORMAT(7HOTYPE 5/)
DO 220 I=1,NL
IF (MSK5(I)) 220,220,190
190 I2=2*I
I1=I2-1
YR=REAL(YE5(I))
YI=AIMAG(YE5(I))
WRITE(6,130) I
IF (MSK5(I)-2) 200,210,200
200 WRITE(6,150) YM5(I1),YR,RES5(I1)
IF (MSK5(I)-1) 220,220,210
210 WRITE(6,170) YM5(I2),YI,RES5(I2)
220 CONTINUE
WRITE(6,225)
225 FORMAT(7HOTYPE 6/)
DO 260 I=1,NB
IF (MSK6(I)) 260,260,230
230 I2=2*I
I1=I2-1
YR=REAL(YE6(I))
YI=AIMAG(YE6(I))
WRITE(6,130) I
IF (MSK6(I)-2) 240,250,240
240 WRITE(6,150) YM6(I1),YR,RES6(I1)
IF (MSK6(I)-1) 260,260,250
250 WRITE(6,170) YM6(I2),YI,RES6(I2)
260 CONTINUE
```

C

```
RETURN
END
```

```
      SUBROUTINE PRTNET
C
C PRINTS TRUE NETWORK DATA
C AUTHOR: TON VAN OVERBEEK 1973-12-12
C
C SUBROUTINE REQUIRED
C      NONE
C
C PARAMETER ML=13
C
C COMPLEX YAA,ZAA,YBB
C
C COMMON /TNET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
C
C IF (NL .GT. 0 .AND. NL .LE. ML) GO TO 20
C WRITE (6,10) NL
10  FORMAT(22H0ERROR IN PRTNET  NL =,I5)
C GO TO 99
C
20  WRITE(6,30) NB,NL
30  FORMAT(18H0TRUE NETWORK DATA/1X,17(1H-))//
C     F5H NB =,I3,6H NL =,I3//
C     WRITE (6,40)
40  FORMAT(11H0 LINE
C     F,10HA-END
C     F,13HB-END
C     F,10HGAA
C     F,10HBAA
C     F,10HRAB
C     F,10HXAB
C     F,10HGBB
C     F,3HBBB/)
C     WRITE (6,50)(I,LTA(I),LTB(I),YAA(I),ZAB(I),YBB(I),I=1,NL)
50  FORMAT(I5,2I10,5X,6F10.5)
C
99  RETURN
C
END
```

```

SUBROUTINE PRWF
C
C PRINTS ALL WEIGHTING FACTORS
C
C AUTHOR, TON VAN OVERBEEK 1974-01-22
C
C SUBROUTINE REQUIRED
C     NONE
C
C PARAMETER MB=10,ML=13
C PARAMETER MMB=2*MB,MML=2*ML,MM=3*(MB+MML)
C
C COMMON /ENET/ NB,NL
C 1     /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),WF6(MMB)
C
C     WRITE(6,10)(I,WF1(I),WF6(2*I-1),WF6(2*I),I=1,NB)
10  FORMAT(22H0ALL WEIGHTING FACTORS/1X,21(1H-)/
      F17H0MSMNR
      F,7HWF1
      F,16HWF6ACT WF6REACT//
      F(I5,5X,3F10.2))
      WRITE(6,20)
20  FORMAT(/17H0MSMNR
      F,10HWF2
      F,7HWF3
      F,20HWF4ACT WF4REACT
      F,16HWF5ACT WF5REACT/)
      DO 30 I=1,NL
      I2=2*I
      I1=I2-1
30  WRITE(6,40) I,WF2(I),WF3(I),WF4(I1),WF4(I2),WF5(I1),WF5(I2)
40  FORMAT(I5,5X,6F10.2)
C
C RETURN
C END
```



```

SUBROUTINE RDGEN(IPRINT,IERR)
C
C   READS GENERATOR DATA IN COMMON BLOCK /GEN/
C
C   AUTHOR: TON VAN OVERBEEK 1974-01-15
C
C   IPRINT=1      NO PRINTOUT
C   IPRINT=0      GENERATOR DATA IS PRINTED
C   IERR=1        ERROR IN NG
C   IERR=0        NO ERROR
C
C   SUBROUTINE REQUIRED
C           NGNE
C
C   PARAMETER MG=14
C
C   COMMON /GEN/ NG, NGB(MG), A1(MG), A2(MG), PMIN(MG), PMAX(MG)
C
C   READ (5,10) NG
10  FORMAT(15)
   IF (NG) 30,30,20
20  IF (NG-MG) 50,50,30
30  IERR=1
   WRITE (6,40) NG
40  FORMAT(14HONG IN RDGEN =,15)
   RETURN
C
50  IERR=0
   READ (5,60)(NGB(I),A1(I),A2(I),PMIN(I),PMAX(I),I=1,NG)
60  FORMAT(15,4F10.5)
   IF (IPRINT) 70,70,99
70  WRITE (6,80)(NGB(I),A1(I),A2(I),PMIN(I),PMAX(I),I=1,NG)
80  FORMAT(15H0GENERATOR DATA/1X,14(1H-)/
   F14H0  NGB
   F,10HA1
   F,10HA2
   F,10HPMIN
   F,4HPMAX//
   F(15,5X,4F10.5))
C
99  RETURN
   END
```

SUBROUTINE RDL D(SDB,SSL,KIMX,U,IPRINT,IERR)

READS THE SIMULATION TIME, DEMAND CONTROL DATA AND THE
START DEMAND. IT USES THE DEMAND CONTROL DATA TO COMPUTE
THE STANDARD SLOPES AND THE DEMAND CONTROL VECTOR. THESE
ARE USED BY THE SUBROUTINE CASDB.

AUTHOR, TON VAN OVERBEEK 1974-02-08

SDB(*) COMPLEX DEMAND AT BUS *
RDL D READS THE START DEMAND INTO SDB(*)
SSL(*) COMPLEX STANDARD SLOPE FOR BUS *
SSL=(SDMAX(*)-SDMIN(*))/KTSL.
SDMAX(*) IS THE MAXIMUM COMPLEX DEMAND AT BUS *,
SDMIN(*) THE MINIMUM DEMAND. KTSL IS THE SLOPE
TIME. KTSL, SDMIN(*) AND SDMAX(*) ARE READ BY RDL D
KTMX SIMULATION TIME
U(*) DEMAND CONTROL VECTOR. SEE SUBROUTINE CASDB
IPRINT=1 NO PRINTOUT
IPRINT=0 THE SLOPE TIME, THE MINIMUM, MAXIMUM AND
START DEMANDS, THE STANDARD SLOPES, THE DEMAND
CONTROL DATA AND THE DEMAND CONTROL VECTOR
ARE PRINTED
IERR=1 ERROR IN KIMX,NOE OR KTCH
IERR=0 NO ERROR

SUBROUTINE REQUIRED
NONE

PARAMETER MB=10,MTMX=360

DIMENSION U(1)

COMPLEX SDB(1),SSL(1),SDMIN(MB),SDMAX(MB)

COMMON /TNET/ NB

READ SLOPE TIME, MINIMUM AND MAXIMUM DEMANDS AND
COMPUTE THE STANDARD SLOPES

READ(5,10) KTSL
10 FORMAT(I5)
DO 30 I=1,NB
READ(5,20) SDMIN(I),SDMAX(I)
20 FORMAT(4F10.5)
30 SSL(I)=(SDMAX(I)-SDMIN(I))/KTSL

READ START DEMAND

READ(5,35)(SDB(I),I=1,NB)
35 FORMAT(2F10.5)

```
C      PRINTOUT
C
      IF (IPRINT) 40,40,70
40 WRITE(6,50) KTSL
50 FORMAT(10H0LOAD DATA/1X,9(1H-)/13HOSLOPE TIME =,I5)
      WRITE (6,60)(I,SDMIN(I),SUMAX(I),SSL(I),SDB(I),I=1,NB)
60 FORMAT(14H0BUSNR
      F,10HPDMIN
      F,15HQDMIN
      F,10HPDMAX
      F,14HQDMAX
      F,10HRE(SSL)
      F,16HIM(SSL)
      F,10HPDEM
      F,4HQDEM//
      F(I5,F15.5,F10.5,F15.5,F10.5,F15.5,F10.5,F15.5,F10.5))
C
C      READ DEMAND CONTROL DATA AND COMPUTE DEMAND CONTROL VECTOR
C
      70 READ(5,10) KTMX
      IF(KTMX) 90,90,80
      80 IF (KTMX-MTMX) 110,110,90
      90 IERR=1
      WRITE(6,100) KTMX
100 FORMAT(15H0KTMX IN RDL D =,I5)
      RETURN
C
      110 READ(5,10) NOE
      IF (NOE) 130,120,120
      120 IF (NOE-KTMX) 150,150,130
      130 IERR=1
      WRITE(6,140) NOE,KTMX
      140 FORMAT(14H0NOE IN RDL D =,I5,5X,6HKTMX =,I5)
      RETURN
C
      150 DO 160 I=1,MTMX
      160 U(I)=0.0
      IF (IPRINT) 170,170,190
      170 WRITE(6,180) KTMX,NOE
      180 FORMAT(/13H0CONTROL DATA/1X,12(1H-)/
      F7H0KTMX =,I5,5X,5HNOE = I5)
      IF (NOE) 320,320,190
      190 IF (IPRINT) 200,200,220
      200 WRITE(6,210)
      210 FORMAT(15H0TCHANGE
      F,4HUNEW/)
      220 K=1
      UOLD=0.0
      DO 300 I=1,NOE
      READ(5,230) KTCH,UNEW
      230 FORMAT(I5,F10.5)
      IF (KTCH) 250,250,240
      240 IF (KTCH-KTMX) 270,270,250
```

```
250 IERR=1
    WRITE(6,260) I,KTCH,KTMX
260 FORMAT(22H0ERROR IN CONTROL DATA/4H0I =,I5,
    F5X,6HKTCH =,I5,5X,6HKTMX =,I5)
    RETURN
```

```
C
270 IF (IPRINT .LT. 1) WRITE(6,280) KTCH,UNEW
280 FORMAT(I6,F15.5)
    DO 290 J=K,KTCH
290 U(J)=UOLD
    K=KTCH
    UOLD=UNEW
300 CONTINUE
    DO 310 I=KTCH,KTMX
310 U(I)=UNEW
```

```
C
C      PRINT THE DEMAND CONTROL VECTOR
```

```
C
320 IF (IPRINT) 330,330,990
330 WRITE(6,340)(I,I=1,10)
340 FORMAT(/13H0THE U-VECTOR,10(I10,1X)/1X,12(1H-))
    DO 360 I=1,KTMX,10
    IA=I
    IB=MIN(IA+9,KTMX)
    WRITE(6,350) IA,IB,(U(J),J=IA,IB)
350 FORMAT(I6,1H-,I5,5X,10F11.5)
360 CONTINUE
```

```
C
990 IERR=0
    RETURN
    END
```



```

SUBROUTINE RDMETE(IPRINT)
C
C READS ESTIMATOR METER DATA TO BE USED FOR THE CALCULATION OF
C THE WEIGHT FACTORS IN COMMON BLOCK /METE/
C
C AUTHOR: TON VAN OVERBEEK 1974-01-18
C
C IPRINT=1      NO PRINTOUT
C IPRINT=0      ESTIMATOR METER DATA IS PRINTED
C
C SUBROUTINE REQUIRED
C           NONE
C
C PARAMETER MB=10,ML=13
C PARAMETER MMB=2*MB,MML=2*ML
C
C COMMON /ENET/ NB,NL
C 1      /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),
C 2          WF6(MMB),ALFE1(MB),ALFE2(ML),ALFE3(ML),ALFE4(MML),
C 3          ALFE5(MML),ALFE6(MMB),FSE1(MB),FSE2(ML),FSE3(ML),
C 4          FSE4(MML),FSE5(MML),FSE6(MMB),BETE1(MB),BETE2(ML),
C 5          BETE3(ML),BETE4(MML),BETE5(MML),BETE6(MMB)
C
C DO 10 I=1,NB
C   I2=2*I
C   I1=I2-1
C 10 READ(5,20) ALFE1(I),FSE1(I),BETE1(I),
C 1           ALFE6(I1),FSE6(I1),BETE6(I1),
C 2           ALFE6(I2),FSE6(I2),BETE6(I2)
C 20 FORMAT(3F10.5)
C   DO 30 I=1,NL
C   I2=2*I
C   I1=I2-1
C 30 READ(5,20) ALFE2(I),FSE2(I),BETE2(I),
C 1           ALFE3(I),FSE3(I),BETE3(I),
C 2           ALFE4(I1),FSE4(I1),BETE4(I1),
C 3           ALFE4(I2),FSE4(I2),BETE4(I2),
C 4           ALFE5(I1),FSE5(I1),BETE5(I1),
C 5           ALFE5(I2),FSE5(I2),BETE5(I2)
C   IF (IPRINT-1) 40,990,990
C 40 WRITE(6,50)
C 50 FORMAT(21H1ESTIMATOR METER DATA/1X,20(1H*)/
C F14HOMETER
C F,10HALFA
C F,10HFSC
C F,4HBETA/
C F,7HATYPE 1/)
C   WRITE(6,60)(I,ALFE1(I),FSE1(I),BETE1(I),I=1,NB)
C 60 FORMAT(15,5X,3F10.5)
C   WRITE(6,70)
C 70 FORMAT(7HATYPE 2/)
C   WRITE(6,60)(I,ALFE2(I),FSE2(I),BETE2(I),I=1,NL)
```

```
WRITE(6,80)
80 FORMAT(7H0TYPE 3/)
WRITE(6,60)(I,ALFE3(I),FSE3(I),BETE3(I),I=1,NL)
WRITE(6,90)
90 FORMAT(/1H0,22X
F,34HACTIVE
F,8HREACTIVE/
F14HOMETER
F,10HALFA
F,10HFSC
F,15HBETA
F,10HALFA
F,10HFSC
F,4HBETA/
F,7H0TYPE 4/)
DO 110 I=1,NL
I2=2*I
I1=I2-1
WRITE(6,100) I,ALFE4(I1),FSE4(I1),BETE4(I1),
1
,ALFE4(I2),FSE4(I2),BETE4(I2)
100 FORMAT(I5,5X,3F10.5,5X,3F10.5)
110 CONTINUE
WRITE(6,120)
120 FORMAT(7H0TYPE 5/)
DO 130 I=1,NL
I2=2*I
I1=I2-1
130 WRITE(6,100) I,ALFE5(I1),FSE5(I1),BETE5(I1),
1
,ALFE5(I2),FSE5(I2),BETE5(I2)
WRITE(6,140)
140 FORMAT(7H0TYPE 6/)
DO 150 I=1,NB
I2=2*I
I1=I2-1
150 WRITE(6,100) I,ALFE6(I1),FSE6(I1),BETE6(I1),
1
,ALFE6(I2),FSE6(I2),BETE6(I2)
C
990 RETURN
END
```



```
WRITE(6,70)
70 FORMAT(7H0TYPE 2/)
WRITE(6,60)(I,BIAS2(I),ALFT2(I),FST2(I),BETT2(I),I=1,NL)
WRITE(6,80)
80 FORMAT(7H0TYPE 3/)
WRITE(6,60)(I,BIAS3(I),ALFT3(I),FST3(I),BETT3(I),I=1,NL)
WRITE(6,90)
90 FORMAT(/1H0,27X
F,43HACTIVE
F,8HREACTIVE/
F14HOMETER
F,10HBIAS
F,10HALFA
F,10HFSC
F,15HBETA
F,10HBIAS
F,10HALFA
F,10HFSC
F,4HBETA/
F,7H0TYPE 4/)
DO 110 I=1,NL
I2=2*I
I1=I2-1
WRITE(6,100) I,BIAS4(I1),ALFT4(I1),FST4(I1),BETT4(I1),
1 BIAS4(I2),ALFT4(I2),FST4(I2),BETT4(I2)
100 FORMAT(I5,5X,4F10.5,5X,4F10.5)
110 CONTINUE
WRITE(6,120)
120 FORMAT(7H0TYPE 5/)
DO 130 I=1,NL
I2=2*I
I1=I2-1
130 WRITE(6,100) I,BIAS5(I1),ALFT5(I1),FST5(I1),BETT5(I1),
1 BIAS5(I2),ALFT5(I2),FST5(I2),BETT5(I2)
WRITE(6,140)
140 FORMAT(7H0TYPE 6/)
DO 150 I=1,NL
I2=2*I
I1=I2-1
150 WRITE(6,100) I,BIAS6(I1),ALFT6(I1),FST6(I1),BETT6(I1),
1 BIAS6(I2),ALFT6(I2),FST6(I2),BETT6(I2)
C
990 RETURN
END
```

SUBROUTINE RDMSM(IPRINT,IERR)

SUBROUTINE TO READ WHICH MEASUREMENTS ARE TO BE USED AND IN WHICH ORDER THEY ARE TO BE TREATED BY THE ESTIMATORS. DATA IS READ INTO THE COMMON BLOCK /MSI/. FROM THE DATA READ INTO NTPY AND NMSM THE MASKVECTORS ARE COMPUTED. ACTIVE AND REACTIVE MEASUREMENTS ARE TREATED SEPARATELY IN NMSM. ODD NUMBERS REFER TO ACTIVE, EVEN NUMBERS TO REACTIVE MEASUREMENTS.

AUTHOR, TON VAN OVERBEEK 1974-01-22

IPRINT=2 NO PRINTOUT
IPRINT=1 THE MASKVECTORS ARE PRINTED
IPRINT=0 SAME + NTPY AND NMSM
IERR=3 ERROR IN NMSM
IERR=2 ERROR IN NTPY
IERR=1 ERROR IN NM
IERR=0 NO ERROR

SUBROUTINE REQUIRED
NONE

PARAMETER MB=10,ML=13
PARAMETER MMB=2*MB,MML=2*ML,MM=3*(MB+MML)

COMMON /ENET/ NB,NL
1 /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),
2 MSK6(MB),NM,NTPY(MM),NMSM(MM)

INITIALIZE MASKVECTORS, NTPY AND NMSM

DO 10 I=1,MB
MSK1(I)=0.0
10 MSK6(I)=0.0
DO 20 I=1,ML
MSK2(I)=0.0
MSK3(I)=0.0
MSK4(I)=0.0
20 MSK5(I)=0.0
DO 30 I=1,MM
NTPY(I)=0.0
30 NMSM(I)=0.0
MMX=3*NB+6*NL

READ AND CHECK NM

READ(5,40) NM
40 FORMAT(I5)
IF (NM) 60,60,50
50 IF (NM-MMX) 80,80,60
60 IERR=1
WRITE(6,70) NM
70 FORMAT(14H0NM IN RDMSM =,I5)

```
      RETURN
C
80 DO 300 I=1,NM
C
C   READ AND CHECK NTYP AND NMSM
C
      READ(5,90) NTI,NMI
90  FORMAT(2I5)
      IF (NTI) 110,110,100
100 IF (NMI-6) 130,130,110
110 IERR=2
      WRITE(6,120) I,NTI
120 FORMAT(30H0TYPE ERROR IN RDMSM   MSMNR =,I5,8H  TYPE =,I5)
      RETURN
C
130 NTYP(1)=NTI
      GO TO (140,150,150,160,160,170),NTI
140 NMAX=NB
      GO TO 180
150 NMAX=NL
      GO TO 180
160 NMAX=2*NL
      GO TO 180
170 NMAX=2*NB
180 IF (NMI) 200,200,190
190 IF (NMI-NMAX) 220,220,200
200 IERR=3
      WRITE(6,210) I,NTI,NMI
210 FORMAT(47H0TYPE MEASUREMENT NUMBER OUT OF BOUNDS IN RDMSM/
      F,8H MSMNR =,I5,8H  TYPE =,I5,14H  TYPE MSMNR =,I5)
      RETURN
C
220 NMSM(I)=NMI
C
C   COMPUTE MASK VECTOR
C
      IERR=0
      GO TO (230,240,250,260,260,260),NTI
230 MSK1(NMI)=1.0
      GO TO 300
240 MSK2(NMI)=1.0
      GO TO 300
250 MSK3(NMI)=1.0
      GO TO 300
260 MD=MOD(NMI,2)
      J=NMI/2+MD
      NTI=NTI-3
      GO TO (270,280,290),NTI
270 MSK4(J)=MSK4(J)+2-MD
      GO TO 300
280 MSK5(J)=MSK5(J)+2-MD
      GO TO 300
290 MSK6(J)=MSK6(J)+2-MD
300 CONTINUE
```

C
C
C

PRINTOUT

```
IF (IPRINT-2) 310,990,990
310 WRITE(6,320)
320 FORMAT(20H0PRINTOUT FROM RDMMSM/1X,19(1H*))
IF (IPRINT-1) 330,350,350
330 WRITE(6,340)(I,NTYP(I),NMSM(I),I=1,NM)
340 FORMAT(12H0MSMNR
F,10HTYPE
F,4HNMSM//
F(15,2I10))
350 WRITE(6,360)(I,MSK1(I),MSK6(I),I=1,NB)
360 FORMAT(/17H0THE MASK VECTORS/1X,16(1H-)/
F,12H0 MSM
F,10HMSK1
F,4HMSK6//
F(15,2I10))
WRITE(6,370)(I,MSK2(I),MSK3(I),MSK4(I),MSK5(I),I=1,NL)
370 FORMAT(/12H0 MSM
F,10HMSK2
F,10HMSK3
F,10HMSK4
F,4HMSK5//
F(15,4I10))
990 RETURN
END
```

```

C      SUBROUTINE RDTNET(IPRINT,IERR)
C      READS TRUE NETWORK DATA IN COMMON BLOCK /TNET/
C      AUTHOR: TON VAN OVERBEEK 1974-01-14
C      IPRINT=1      NO PRINTOUT
C      IPRINT=0      TRUE NETWORK DATA IS PRINTED
C      IERR=1        ERROR IN NB OR NL
C      IERR=0        NO ERROR
C      SUBROUTINE REQUIRED
C          PRTNET
C
C      PARAMETER MB=10,ML=13
C
C      COMPLEX YAA,ZAB,YBB
C
C      COMMON /TNET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
C
C      READ (5,10) NB,NL
10  FORMAT(2I5)
    IF (NB) 30,30,20
20  IF (NB-MB) 50,50,30
30  IERR=1
    WRITE (6,40) NB
40  FORMAT(15H0NB IN RDTNET =,I5)
    RETURN
C
50  IF (NL) 70,70,60
60  IF (NL-ML) 90,90,70
70  IERR=1
    WRITE (6,80) NL
80  FORMAT(15H0NL IN RDTNET =,I5)
    RETURN
C
90  IERR=0
    READ (5,100)(LTA(I),LTB(I),YAA(I),ZAB(I),YBB(I),I=1,NL)
100 FORMAT(2I5,6F10.5)
    IF (IPRINT-1) 110,990,990
110 CALL PRTNET
C
990 RETURN
    END
```



```

SUBROUTINE TRUEV(SDB,IPRINT,IERR)
C
C COMPUTES THE TRUE VALUES OF ALL VARIABLES GIVEN TRUE
C NETWORK DATA AND A POWER DEMAND.
C
C AUTHOR, TON VAN OVERBEEK 1974-01-16
C
C SDB(*)          COMPLEX LOAD AT BUS *
C IERR=3          ERROR IN NRLFR
C IERR=2          ERROR IN ELDNL
C IERR=1          ERROR IN NB,NL OR NG
C IERR=0          YT HAS BEEN COMPUTED
C IPRINT=4        NO PRINTOUT
C IPRINT=3        POWER DEMAND AND YT ARE PRINTED
C IPRINT=2        SAME + TRUE NETWORK AND GENERATOR DATA
C IPRINT=1        SAME + OUTPUT FROM THE SUB-
C IPRINT=0        ROUTINES ELDNL AND NRLFR
C
C SUBROUTINE REQUIRED
C           ELDNL
C           GXT
C           PRTNET
C           NRLFR
C           DECOM
C           MPRI
C           SOLVB
C
C PARAMETER MB=10,ML=13,MG=14
C
C COMPLEX YAA,ZAB,YBB,XT,YT2,YT3,YT4,YT5,YT6
C 1,SDB(1),SDEM,SGEN(MG),SINJ(MB)
C
C DIMENSION PGEN(MG)
C
C COMMON /TNET/ NB,NL,LTA(ML),LTB(ML),YAA(ML),ZAB(ML),YBB(ML)
C 1 /GEN/ NG,NGB(MG),A1(MG),A2(MG),PMIN(MG),PMAX(MG)
C 2 /TRUE/ XT(MB),YT2(ML),YT3(ML),YT4(ML),YT5(ML),YT6(MB)
C
C IERR=0
C IF (NB .GT. 0 .AND. NB .LE. MB) GO TO 20
C IERR=1
C WRITE (6,10) NB
C 10 FORMAT(14H0NB IN TRUEV =,I5)
C RETURN
C
C 20 IF (NL .GT. 0 .AND. NL .LE. ML) GO TO 40
C IERR=1
C WRITE (6,30) NL
C 30 FORMAT(14H0NL IN TRUEV =,I5)
C RETURN
C
C 40 IF (NG .GT. 0 .AND. NG .LE. MG) GO TO 60
C IERR=1
```

```
WRITE (6,50) NG
50 FORMAT(14HONG IN TRUEV =,I5)
RETURN
```

C
C
C

```
PRINT INPUT DATA

60 IF (IPRINT .GE. 4) GO TO 100
WRITE (6,70) (1,SDB(I),I=1,NB)
70 FORMAT(20H1PRINTOUT FROM TRUEV/1X,19(1H*)/
F16H0THE LOAD DEMAND/
F14H0BUSNR
F,8HACTIVE
F,8HREACTIVE//
F(15,5X,2F10.5))
IF (IPRINT .GE. 3) GO TO 100
WRITE (6,80) NG
80 FORMAT(/15H0GENERATOR DATA/5HONG =,I3)
WRITE (6,90) (1,NGB(I),A1(I),A2(I),PMIN(I),PMAX(I),I=1,NG)
90 FORMAT(19H0GENNR NGB
F,10HA1
F,10HA2
F,10HPMIN
F,4HPMAX//
F(215,5X,4F10.5))
```

C
C
C

COMPUTE REAL GENERATED POWERS WITH SUBROUTINE ELDNL

```
100 SDEM=(0.0,0.0)
DO 110 I=1,NB
SINJ(I)=-SDB(I)
110 SDEM=SDEM+SDB(I)
CALL ELDNL(A1,A2,PMIN,PGEN,PMAX,REAL(SDEM),0.001,NG,IPRINT,IERR)
IF (IERR) 120,130,120
120 IERR=2
RETURN
```

C
C
C

COMPUTE BUSINJECTIONS

```
130 FACTOR =AIMAG(SDEM)/REAL(SDEM)
DO 140 I=1,NG
IB=NGB(I)
SGEN(I)=CMPLX(PGEN(I),PGEN(I)*FACTOR)
140 SINJ(IB)=SINJ(IB)+SGEN(I)
```

C
C
C

COMPUTE BUSVOLTAGES FROM BUSINJECTIONS

```
CALL NRLFR(NB,NL,LTA,LTB,YAA,ZAB,YBB,SINJ,XT,0.01,NB,5,
1IPRINT,JFAIL)
IF (JFAIL .EQ. 0) GO TO 150
IERR=3
RETURN
```

C
C
C

COMPUTE OTHER VARIABLES WITH SUBROUTINE GXT

```
150 CALL GXT(IPRINT-2,I)
RETURN
END
```

```

SUBROUTINE UPDAA(INDEX,ELT,NOE,WI,A)
C
C SUBROUTINE FOR ESTA
C UPDATES MATRIX A FOR MEASUREMENT I ACCORDING TO
C  $A = A + (\text{JACOBIAN ROW I}) ** T * \text{WI} * (\text{JACOBIAN ROW I})$ 
C
C AUTHOR: TON VAN OVERBEEK 1974-01-30
C
C INDEX(*)          PLACE OF ELT(*) IN JACOBIAN ROW I
C ELT(*)           JACOBIAN ELEMENT
C NOE              NUMBER OF ELEMENTS IN ELT AND INDEX, MAX=MMB
C WI              WEIGHTING FACTOR FOR MEASUREMENT I
C A               MATRIX TO BE UPDATED
C
C SUBROUTINE REQUIRED
C NONE
C
C PARAMETER MB=10
C PARAMETER MMB=2*MB
C
C DIMENSION INDEX(1),ELT(1),A(MMB,1)
C
C IF (NOE) 20,99,10
10 IF (NOE-MMB) 40,40,20
20 WRITE (6,30) NOE
30 FORMAT(25HOCALL TO UPDAA WITH NOE =,15)
RETURN
C
40 DO 70 I=1,NOE
   K=INDEX(I)
   A(K,K)=A(K,K)+ELT(I)**2*WI
   IF (I-1) 70,70,50
50 I1=I-1
   DO 60 J=1,I1
   L=INDEX(J)
   A(K,L)=A(K,L)+ELT(I)*ELT(J)*WI
   A(L,K)=A(K,L)
60 CONTINUE
70 CONTINUE
C
99 RETURN
END
```

```

SUBROUTINE UPDAC(IA,IB,DA)
C
C HELPROUTINE FOR ESTC
C UPDATES MATRIX A FOR A COMPLEX LINE FLOW MEASUREMENT AT
C A-END OF THE LINE
C
C AUTHOR: TON VAN OVERBEEK 1974-03-11
C
C IA          A-END OF LINE CONNECTED TO BUS IA
C IB          B-END OF LINE CONNECTED TO BUS IB
C DA          WEIGHTING FACTOR FOR THE LINEVOLIAGE
C
C SUBROUTINE REQUIRED
C          NONE
C
C PARAMETER MB=10
C
C COMMON /ENET/ NB
C X      /MAT/ A(MB,MB)
C
C   IF (IA) 20,20,10
10 IF (IA-NB) 40,40,20
20 WRITE(6,30) IA
30 FORMAT(14H0IA IN UPDAC =,I5)
   RETURN
C
C 40 IF (IB) 60,60,50
50 IF (IB-NB) 80,80,60
60 WRITE(6,70) IB
70 FORMAT(14H0IB IN UPDAC =,I5)
   RETURN
C
C 80 A(IA,IA)=A(IA,IA)+DA
   A(IB,IB)=A(IB,IB)+DA
   A(IA,IB)=A(IA,IB)-DA
   A(IB,IA)=A(IA,IB)
C
C RETURN
C END
```

```

SUBROUTINE UPDBA(INDEX,ELT,NOE,WI,B,RES)
C
C SUBROUTINE FOR ESTA
C UPDATES THE B VECTOR FOR MEASUREMENT I ACCORDING TO
C  $B = B + (JACOBIAN\ ROW\ I) * T * WI * RES$ 
C
C AUTHOR, TON VAN OVERBEEK 1974-01-30
C
C INDEX(*)          PLACE OF ELT(*) IN JACOBIAN ROW I
C ELT(*)           JACOBIAN ELEMENT
C NOE              NUMBER OF ELEMENTS IN ELT AND INDEX, MAX=MMB
C WI              WEIGHTING FACTOR FOR MEASUREMENT I
C B               VECTOR TO BE UPDATED
C RES            RESIDU OF MEASUREMENT I
C
C SUBROUTINE REQUIRED
C             NONE
C
C PARAMETER MB=10
C PARAMETER MMB=2*MB
C
C DIMENSION INDEX(1),ELT(1),B(1)
C
C IF (NOE) 20,99,10
10 IF (NOE=MMB) 40,40,20
20 WRITE (6,30) NOE
30 FORMAT(25H0CALL TO UPDBA WITH NOE =,15)
   RETURN
C
40 WIR=WI*RES
   DO 50 I=1,NOE
     K=INDEX(I)
50 B(K)=B(K)+ELT(I)*WIR
C
99 RETURN
   END
```

```
      SUBROUTINE UPDBC(CB,IA,IB,NR,XEA,XEB,ZL,YA,CYM,CLV,DA)
C
C      HELPROUTINE FOR ESTC
C      UPDATES THE RIGHT HAND B-VECTOR OF THE EQUATION A*E=B FOR
C      A COMPLEX LINE FLOW MEASUREMENT
C
C      AUTHOR, TON VAN OVERBEEK 1974-03-11
C
C      CB(*)          COMPLEX B-VECTOR
C      IA            A-END OF LINE CONNECTED TO BUS IA
C      IB            B-END OF LINE CONNECTED TO BUS IB
C      NR            NUMBER OF ELEMENTS IN THE REFERENCE VECTOR
C      XEA           COMPLEX BUSVOLTAGE AT A-END OF LINE
C      XEB           COMPLEX BUSVOLTAGE AT B-END OF LINE
C      ZL            LINE SERIES IMPEDANCE
C      YA            LINE SHUNT ADMITTANCE AT A-END
C      CYM           COMPLEX LINE FLOW MEASUREMENT AT A-END
C      CLV           COMPUTED COMPLEX LINEVOLTAGE
C      DA            WEIGHTING FACTOR FOR LINEVOLTAGE AT A-END
C
C      SUBROUTINE REQUIRED
C              NONE
C
C      PARAMETER MB=10
C
C      COMPLEX CB(1),XEA,XEB,ZL,YA,CYM,CLV
C
C      COMMON /ENET/ NB
C
C      IF (IA) 20,20,10
10  IF (IA-NB) 40,40,20
20  WRITE(6,30) IA
30  FORMAT(14H0IA IN UPDBC =,I5)
   RETURN
C
C      40 IF (IB) 60,60,50
50  IF (IB-NB) 80,80,60
60  WRITE(6,70) IB
70  FORMAT(14H0IB IN UPDBC =,I5)
   RETURN
C
C      80 IF (NR) 100,100,90
90  IF (NR-NB) 120,120,100
100 WRITE(6,110) NR
110 FORMAT(14H0NR IN UPDBC =,I5)
   RETURN
C
C      120 CLV=ZL*(CONJG(CYM/XEA)-YA*XEA)
      NE=NB-NR
C
C      IF (IA-NE) 130,130,140
130 CB(IA)=CB(IA)+DA*CLV
      GO TO 150
140 CB(IB)=CB(IB)+DA*XEA
```

```
C
150 IF (IB-NE) 160,160,170
160 CB(IB)=CB(IB)-DA*CLV
    GO TO 990
170 CB(IA)=CB(IA)+DA*XEB
C
990 RETURN
    END
```



```
C      NXOP          NUMBER OF TIMEPERIODS DURING WHICH OUTPUT
C      KXB(*)       IS PRODUCED IRRESPECTIVE OF THE VALUE OF IOUT
C      KXE(*)       START TIMEPOINT FOR PERIOD *
C      END TIMEPOINT FOR PERIOD *
C
C      READ(5,10) IOUT,NXOP
10  FORMAT(2I5)
    IF (NXOP) 30,20,20
20  IF (NXOP-MTMX) 50,50,30
30  WRITE(6,40) NXOP
40  FORMAT(7HONXOP =,15)
    GO TO 990
C
50  M=IABS(IOUT)
    IF (IOUT .LE. 0) M=MTMX1
    DO 60 I=1,MTMX
    IPR(I)=1
    IF (I/M*M .EQ. 1) IPR(I)=0
60  CONTINUE
    IF (NXOP) 100,100,70
70  READ(5,10)(KXB(I),KXE(I),I=1,NXOP)
    DO 90 I=1,NXOP
    J1=IABS(KXB(I))
    J2=IABS(KXE(I))
    DO 80 J=J1,J2
80  IPR(J)=0
90  CONTINUE
100 WRITE(6,110) IOUT,NXOP
110 FORMAT(26H1MAIN PROGRAM FOR METHOD A/1X,25(1H*)//
    F20H0OUTPUT CONTROL DATA/1X,19(1H-)//
    F7H0IOUT =,15,5X,6HNXOP =,15)
    IF (NXOP) 140,140,120
120 WRITE(6,130)(KXB(I),KXE(I),I=1,NXOP)
130 FORMAT(7H0 TBEG,5X,4HTEND//(15,I10))
C
C      *****
C      * READ AND PRINT FIXED DATA *
C      *****
C
C      READ AND PRINT TRUE DATA: NETWORK, GENERATOR, AND LOAD
C      CONTROL DATA
C
    IPRINT=0
140 CALL RDTNET(IPRINT,IERR)
    IF (IERR) 150,150,990
150 CALL RDMETT(IPRINT)
    CALL RDGEN(IPRINT,IERR)
    IF (IERR) 160,160,990
160 CALL RDLD(SDB,SSL,KTMX,U,IPRINT,IERR)
    IF (IERR) 170,170,990
C
C      READ AND PRINT ESTIMATOR DATA: NETWORK AND METER DATA,
C      MEASUREMENT CHOICE AND ORDER
C
```

```
170 CALL RDENET(IPRINT,IERR)
    IF (IERR) 180,180,990
180 CALL RDMETE(IPRINT)
    CALL RDMMSM(IPRINT,IERR)
    IF (IERR) 190,190,990
C
C   READ AND PRINT PARAMETERS FOR ESTIMATOR A
C
190 READ(5,200) MAXIT,JS,CRLOSS,CRLIN
200 FORMAT(2I5,2F10.5)
    WRITE(6,210) MAXIT,JS,CRLOSS,CRLIN
210 FORMAT(21HOESTIMATOR PARAMETERS/1X,20(1H-)/
    F8H0MAXIT =,I5,5X,4HJS =,I5,5X,8HCRLOSS =,F10.5,
    F5X,7HCRLIN =,F10.5)
C
C   *****
C   *   INITIALIZATION   *
C   *****
C
C   TIMEPOINT 0: COMPUTE THE TRUE VARIABLES FROM THE START DEMAND
C   (READ BY RDL0 INTO SDB), SET THE ESTIMATE EQUAL TO THE TRUE
C   STATE, COMPUTE THE MEASUREMENTS AND INITIALIZE THE ESTIMATOR
C   BY LINEARIZING AROUND THE TRUE STATE
C
    K=0
    WRITE(6,220) K
220 FORMAT(11H1TIMEPOINT ,I5/1X,15(1H*))
    DO 230 I=1,NBT
230 XT(I)=(1.0,0.0)
    CALL TRUEV(SDB,IPRINT+3,IERR)
    IF (IERR) 240,240,990
240 CALL CAWN(IPRINT+1)
    NODD=19
    CALL ALLMSM(NODD,IPRINT+1)
    CALL CAWF(IPRINT+1)
    DO 250 I=1,NBE
250 XE(I)=XT(I)
    CALL ADMA(1,2,CRLOSS,XL,CRLIN,IXIT,TIME,IPRINT+4)
    IF (IXIT) 990,990,260
260 CALL EVAL(K,IPRINT)
C
C   *****
C   *   SIMULATION   *
C   *****
C
    AEE=0.0
    AEL=0.0
    AEM=0.0
    DO 270 I=1,2
    EEMT(I)=0.0
    ELMT(I)=0.0
270 EMMT(I)=0.0
```

```
DO 280 I=1,3
EEMMT(I)=0.0
280 ELMMT(I)=0.0
C
DO 310 K=1,KTMX
IPRINT=IPR(K)
IF (IPRINT .LE. 0) WRITE(6,220) K
C
C COMPUTE THE LOAD DEMAND AND ALL TRUE VARIABLES
C
CALL CASDB(SDB,SSL,U(K),IPRINT)
CALL TRUEV(SDB,IPRINT+3,IERR)
IF (IERR) 290,290,990
C
C COMPUTE THE STANDARD DEVIATIONS FOR THE MEASUREMENT NOISE,
C ALL POSSIBLE MEASUREMENTS AND THE WEIGHTING FACTORS FOR
C THE ESTIMATOR
C
290 CALL CAWN(IPRINT+1)
CALL ALLMSM(NODD,IPRINT+1)
CALL CAWF(IPRINT+1)
C
C COMPUTE THE NEW ESTIMATE AND THE EVALUATION QUANTITIES
C
CALL ADMA(MAXIT,JS,CRLOSS,XL,CRLIN,IEXIT(K),TIME,IPRINT+4)
IF (IEXIT(K)) 990,990,300
300 CALL EVAL(K,IPRINT)
C
310 CONTINUE
C
C *****
C * PRINTOUT OF THE RESULTS *
C *****
C
WRITE(6,320) KTMX
320 FORMAT(17H1THE RESULTS FOR ,I5,11H TIMEPOINTS/1X,33(1H*)//
F12H0TIME
F,10HEST ERR
F,21HMAX ELT BUS ROI
F,11HLF ERROR
F,27HMAX ELT END LINE AOR
F,15HMSMT IND
F,5HIEXIT/)
DO 420 K=1,KTMX
K1=K+1
I=INT(EEM(K1,2))
M=MOD(I,2)
J=I/2+M
IF (M) 330,330,340
330 ROI=1HI
GO TO 350
340 ROI=1HR
350 I=INT(ELM(K1,2))
IF (I) 360,360,370
```

```
360 AOB=1HB
    GO TO 380
370 AOB=1HA
    I=IABS(I)
    M=MOD(I,2)
    J1=I/2+M
    IF (M) 380,380,390
380 AOR=1HR
    GO TO 400
390 AOR=1HA
400 WRITE(6,410) K,EE(K1),EEM(K1,1),J,ROI,EL(K1),ELM(K1,1),AOB,J1,AOR,
    1          EM(K1)
410 FORMAT(I4,5X,2F10.7,I3,3X,A1,5X,2F10.7,2X,A1,I5,3X,A1,6X,F10.7)
420 CONTINUE
    WRITE(6,430) KTMX
430 FORMAT(14H1TOTALS AFTER ,I5,11H TIMEPOINTS/1X,29(1H*))
    AEE=AEE/KTMX
    WRITE(6,440) AEE
440 FORMAT(17H0ESTIMATION ERROR/1X,16(1H-)//
    F6X,27HAVERAGE ESTIMATION ERROR  =,F10.7)
    KT=INT(EEMT(2))
    WRITE(6,450) ELMT(1),KT
450 FORMAT(6X,27HMAXIMUM ESTIMATION ERROR  =,F10.7,7H AT T =,I4)
    I=INT(EEMMT(2))
    M=MOD(I,2)
    J=I/2+M
    IF (M) 460,460,470
460 ROI=4HIMAG
    GO TO 480
470 ROI=4HREAL
480 KT=INT(EEMMT(3))
    WRITE(6,490) EEMMT(1),KT,ROI,J
490 FORMAT(6X,27HMAX ELEMENT IN ALL ERRORS =,F10.7,7H AT T =,I4,
    F4H IN /8X,A4,20H PART OF BUSVOLTAGE ,I2)
    AEL=AEL/KTMX
    WRITE(6,500) AEL
500 FORMAT(16H0LINE FLOW ERROR/1X,15(1H-)//
    F6X,27HAVERAGE LINE FLOW ERROR  =,F10.7)
    KT=INT(ELMT(2))
    WRITE(6,510) ELMT(1),KT
510 FORMAT(6X,27HMAXIMUM LINE FLOW ERROR  =,F10.7,7H AT T =,I4)
    I=INT(ELMNT(2))
    IF (I) 520,520,530
520 AOB=1HB
    GO TO 540
530 AOB=1HA
540 I=IABS(I)
    M=MOD(I,2)
    J=I/2+M
    IF (M) 550,550,560
550 AOR=6HREACT.
    GO TO 570
```

```

560 AOR=6HACTIVE
570 KT=INT(ELMMT(3))
    WRITE(6,580) ELMMT(1),KT,AOR,AOB,J
580 FORMAT(6X,27HMAX ELEMENT IN ALL ERRORS =,F10.7,7H AT T =,I4,
    F4H IN /8X,A6,9H FLOW AT ,A1,13H-END OF LINE ,I2)
    AEM=AEM/KTMX
    WRITE(6,590) AEM
590 FORMAT(20HMEASUREMENT QUALITY/1X,19(1H-)//
    F6X,27HAVERAGE MEASUREMENT INDEX =,F10.7)
    KT=INT(EMMT(2))
    WRITE(6,600) EMMT(1),KT
600 FORMAT(6X,27HMAXIMUM MEASUREMENT INDEX =,F10.7,7H AT T =,I4)
C
C *****
C * WRITE PLOTDATA TO PLOTFILE *
C *****
C
    KTMX1=KTMX+1
    WRITE(1) KTMX1,(EE(I),EL(I),EM(I),I=1,MTMX1)
C
990 STOP
    END

```

C MAINPROGRAM FOR METHOD B

C AUTHOR, TON VAN OVERBEEK 1974-02-19

C SUBROUTINE REQUIRED

C ADMB
C ALOSS
C PRRES
C CARES
C GXE
C PRENET
C PRRES
C ESTB
C JACBI
C JACLF
C JACI
C JACLF
C JACV
C UPDEP
C PRRES
C PRWF
C ALLMSM
C NODI
C CASDB
C CAWF
C CAWN
C EVAL
C RDENET
C PRENET
C RDGEN
C RDLN
C RDMETE
C RDMETT
C RDMSM
C RDTNET
C PRTNET
C TRUEV
C ELDNL
C GXT
C PRTNET
C NRLFR
C DECOM
C MPRI
C SOLVB

C PARAMETER MB=10,ML=13,MTMX=360

C PARAMETER MMB=2*MB,MML=2*ML,MM=3*(MB+MML),MTMX1=MTMX+1

C INTEGER TIME

C DIMENSION U(MTMX),IPR(MTMX),KXB(MTMX),KXE(MTMX),Q(MMB),IEXIT(MTMX)

C COMPLEX XT,YT2,YT3,YT4,YT5,YT6,YAAT,ZABT,YBBT,SDB(MB),SSL(MB),
1 XE,YE2,YE3,YE4,YE5,YE6,YAAE,ZABE,YBBE


```
C
50 M=IABS(IOUT)
   IF (IOUT .LE. 0) M=MTMX1
   DO 60 I=1,MTMX
   IPR(I)=1
   IF (I/M*M .EQ. 1) IPR(I)=0
60 CONTINUE
   IF (NXOP) 100,100,70
70 READ(5,10)(KXB(I),KXE(I),I=1,NXOP)
   DO 90 I=1,NXOP
   J1=IABS(KXB(I))
   J2=IABS(KXE(I))
   DO 80 J=J1,J2
80 IPR(J)=0
90 CONTINUE
100 WRITE(6,110) IOUT,NXOP
110 FORMAT(26H1MAIN PROGRAM FOR METHOD B/1X,25(1H*)//
   F20H0OUTPUT CONTROL DATA/1X,19(1H-)/
   F7H0IOUT =,I5,5X,6HNXOP =,I5)
   IF (NXOP) 140,140,120
120 WRITE(6,130)(KXB(I),KXE(I),I=1,NXOP)
130 FORMAT(7H0 TBEG,5X,4HTEND//(I5,I10))

C
C *****
C * READ AND PRINT FIXED DATA *
C *****
C
C READ AND PRINT TRUE DATA: NETWORK, GENERATOR, AND LOAD
C CONTROL DATA
C
   IPRINT=0
140 CALL RDTNET(IPRINT,IERR)
   IF (IERR) 150,150,990
150 CALL RDMETT(IPRINT)
   CALL RDGEN(IPRINT,IERR)
   IF (IERR) 160,160,990
160 CALL RDLD(SDB,SSL,KTMX,U,IPRINT,IERR)
   IF (IERR) 170,170,990

C
C READ AND PRINT ESTIMATOR DATA: NETWORK AND METER DATA,
C MEASUREMENT CHOICE AND ORDER
C
170 CALL RDENET(IPRINT,IERR)
   IF (IERR) 180,180,990
180 CALL RDMETE(IPRINT)
   CALL ROMSM(IPRINT,IERR)
   IF (IERR) 190,190,990

C
C READ AND PRINT PARAMETERS FOR ESTIMATOR B
C
190 READ(5,200) MAXIT,JS,J0,CRLOSS,PIN
200 FORMAT(3I5,2F10.5)
   READ(5,205)(Q(2*I-1),Q(2*I),I=1,NBT)
205 FORMAT(2E10.3)
```

```
WRITE(6,210) MAXIT,JS,JQ,CRLOSS,PIN
210 FORMAT(21H0ESTIMATOR PARAMETERS/1X,20(1H-)/
F8H0MAXIT =,15,5X,4HJS =,15,5X,4HJQ =,15,5X,8HCRLOSS =,F10.5,
F5X,5HPIN =,F10.5)
WRITE(6,215)(I,Q(2*I-1),Q(2*I),I=1,NBT)
215 FORMAT(14H0BUSNR
F,10HQREAL
F,5HQIMAG// (15,5X,2E10.3))
C
C *****
C * INITIALIZATION *
C *****
C
C TIMEPOINT 0: COMPUTE THE TRUE VARIABLES FROM THE START DEMAND
C (READ BY RDLN INTO SDB), SET THE ESTIMATE EQUAL TO THE TRUE
C STATE, COMPUTE THE MEASUREMENTS AND INITIALIZE THE ESTIMATOR
C
K=0
WRITE(6,220) K
220 FORMAT(11H1TIMEPOINT ,15/1X,15(1H*))
DO 230 I=1,NBT
230 XT(I)=(1.0,0.0)
CALL TRUEV(SDB,IPRINT+3,IERR)
IF (IERR) 240,240,990
240 CALL CAWN(IPRINT+1)
NODD=19
CALL ALLMSM(NODD,IPRINT+1)
CALL CAWF(IPRINT+1)
DO 250 I=1,NBE
250 XE(I)=XT(I)
CALL ADMB(1,1,CRLOSS,1,PIN,Q,IXIT,TIME,IPRINT+3)
IF (IXIT) 990,990,260
260 CALL EVAL(K,IPRINT)
C
C *****
C * SIMULATION *
C *****
C
AEE=0.0
AEL=0.0
AEM=0.0
DO 270 I=1,2
EEMT(I)=0.0
ELMT(I)=0.0
270 EMMT(I)=0.0
DO 280 I=1,3
EEMMT(I)=0.0
280 ELMMT(I)=0.0
C
DO 310 K=1,KTMX
IPRINT=IPR(K)
IF (IPRINT .LE. 0) WRITE(6,220) K
```

```
C
C   COMPUTE THE LOAD DEMAND AND ALL TRUE VARIABLES
C
C   CALL CASDB(SDB,SSL,U(K),IPRINT)
C   CALL TRUEV(SDB,IPRINT+3,IERR)
C   IF (IERR) 290,290,990
C
C   COMPUTE THE STANDARD DEVIATIONS FOR THE MEASUREMENT NOISE,
C   ALL POSSIBLE MEASUREMENTS AND THE WEIGHTING FACTORS FOR
C   THE ESTIMATOR
C
290 CALL CAWN(IPRINT+1)
C   CALL ALLMSM(NUDD,IPRINT+1)
C   CALL CAWF(IPRINT+1)
C
C   COMPUTE THE NEW ESTIMATE AND THE EVALUATION QUANTITIES
C
C   CALL ADMB(MAXIT,JS,CRLOSS,JQ,PIN,Q,IEXIT(K),TIME,IPRINT+3)
C   IF (IEXIT(K)) 990,990,300
300 CALL EVAL(K,IPRINT)
C
310 CONTINUE
C
C   *****
C   * PRINTOUT OF THE RESULTS *
C   *****
C
C   WRITE(6,320) KTMX
320 FORMAT(17H1THE RESULTS FOR ,15,11H TIMEPOINTS/1X,33(1H*)//
F12H0TIME
F,10HEST ERR
F,21HMAX ELT BUS ROI
F,11HLF ERROR
F,27HMAX ELT END LINE AOR
F,15HMSMT IND
F,5HIEXIT/)
DO 420 K=1,KTMX
K1=K+1
I=INT(EEM(K1,2))
M=MOD(I,2)
J=I/2+M
IF (M) 330,330,340
330 ROI=1HI
GO TO 350
340 ROI=1HR
350 I=INT(ELM(K1,2))
IF (I) 360,360,370
360 AOB=1HB
GO TO 380
370 AOB=1HA
I=IABS(I)
M=MOD(I,2)
J1=I/2+M
IF (M) 380,380,390
```

```
380 AOR=1HR
    GO TO 400
390 AOR=1HA
400 WRITE(6,410) K,EE(K1),EEM(K1,1),J,ROI,EL(K1),ELM(K1,1),AOB,J1,AOR,
    1      EM(K1),IEXIT(K)
410 FORMAT(14,5X,2F10.7,I3,3X,A1,5X,2F10.7,2X,A1,I5,3X,A1,6X,F10.7,
    F5X,I5)
420 CONTINUE
    WRITE(6,430) KTMX
430 FORMAT(14H1TOTALS AFTER ,I5,11H TIMEPOINTS/1X,29(1H*))
    AEE=AEE/KTMX
    WRITE(6,440) AEE
440 FORMAT(17H0ESTIMATION ERROR/1X,16(1H-)//
    F6X,27HAVERAGE ESTIMATION ERROR =,F10.7)
    KT=INT(EEMT(2))
    WRITE(6,450) EEMT(1),KT
450 FORMAT(6X,27HMAXIMUM ESTIMATION ERROR =,F10.7,7H AT T =,I4)
    I=INT(EEMMT(2))
    M=MOD(I,2)
    J=I/2+M
    IF (M) 460,460,470
460 ROI=4HIMAG
    GO TO 480
470 ROI=4HREAL
480 KT=INT(EEMMT(3))
    WRITE(6,490) EEMMT(1),KT,ROI,J
490 FORMAT(6X,27HMAX ELEMENT IN ALL ERRORS =,F10.7,7H AT T =,I4,
    F4H IN /8X,A4,20H PART OF BUSVOLTAGE ,I2)
    AEL=AEL/KTMX
    WRITE(6,500) AEL
500 FORMAT(16H0LINE FLOW ERROR/1X,15(1H-)//
    F6X,27HAVERAGE LINE FLOW ERROR =,F10.7)
    KT=INT(ELMT(2))
    WRITE(6,510) ELMT(1),KT
510 FORMAT(6X,27HMAXIMUM LINE FLOW ERROR =,F10.7,7H AT T =,I4)
    I=INT(ELMNT(2))
    IF (I) 520,520,530
520 AOB=1HB
    GO TO 540
530 AOB=1HA
540 I=IABS(I)
    M=MOD(I,2)
    J=I/2+M
    IF (M) 550,550,560
550 AOR=6HREACT.
    GO TO 570
560 AOR=6HACTIVE
570 KT=INT(ELMNT(3))
    WRITE(6,580) ELMNT(1),KT,AOR,AOB,J
580 FORMAT(6X,27HMAX ELEMENT IN ALL ERRORS =,F10.7,7H AT T =,I4,
    F4H IN /8X,A6,9H FLOW AT ,A1,13H-END OF LINE ,I2)
    AEM=AEM/KTMX
    WRITE(6,590) AEM
```

```
590 FORMAT(20HMEASUREMENT QUALITY/1X,19(1H-)//  
      F6X,27HAVERAGE MEASUREMENT INDEX =,F10.7)  
      KT=INT(EMMT(2))  
      WRITE(6,600) EMMT(1),KT  
600 FORMAT(6X,27HMAXIMUM MEASUREMENT INDEX =,F10.7,7H AT T =,I4)  
C  
C *****  
C * WRITE PLOTDATA TO PLOTFILE *  
C *****  
C  
      KTMX1=KTMX+1  
      IUNIT=1  
      WRITE(IUNIT) KTMX1,(EE(I),EL(I),EM(I),I=1,MTMX1)  
C  
990 STOP  
      END
```


C

```

COMMON /TRUE/ XT(MB),YT2(ML),YT3(ML),YT4(ML),YT5(ML),YT6(MB)
X      /EST/ XE(MB),YE2(ML),YE3(ML),YE4(ML),YE5(ML),YE6(MB)
X      /TNET/ NBT,NLT,LTAT(ML),LTBT(ML),YAAT(ML),ZABT(ML),YBBT(ML)
X      /ENET/ NBE,NLE,LTAE(ML),LTBE(ML),YAAE(ML),ZABE(ML),YBBE(ML)
X      /MSM/ YM1(MB),YM2(ML),YM3(ML),YM4(MML),YM5(MML),YM6(MMB)
X      /METT/ BIAS1(MB),BIAS2(ML),BIAS3(ML),BIAS4(MML),
X            BIAS5(MML),BIAS6(MMB),WN1(MB),WN2(ML),WN3(ML),
X            WN4(MML),WN5(MML),WN6(MMB),ALFT1(MB),ALFT2(ML),
X            ALFT3(ML),ALFT4(MML),ALFT5(MML),ALFT6(MMB),
X            FST1(MB),FST2(ML),FST3(ML),FST4(MML),
X            FST5(MML),FST6(MMB),BETT1(MB),BETT2(ML),BETT3(ML),
X            BETT4(MML),BETT5(MML),BETT6(MMB)
COMMON /METE/ WF1(MB),WF2(ML),WF3(ML),WF4(MML),WF5(MML),
X            WF6(MMB),ALFE1(MB),ALFE2(ML),ALFE3(ML),ALFE4(MML),
X            ALFE5(MML),ALFE6(MMB),FSE1(MB),FSE2(ML),FSE3(ML),
X            FSE4(MML),FSE5(MML),FSE6(MMB),BETE1(MB),BETE2(ML),
X            BETE3(ML),BETE4(MML),BETE5(MML),BETE6(MMB)
X      /RES/ RES1(MB),RES2(ML),RES3(ML),RES4(MML),RES5(MML),
X            RES6(MMB)
X      /MAT/ A(MB,MB),T(MB,MB)
X      /MSI/ MSK1(MB),MSK2(ML),MSK3(ML),MSK4(ML),MSK5(ML),
X            MSK6(MB),NM,NTYP(MM),NMSM(MM)
COMMON /EVL/ EE(MTMX1),EEM(MTMX1,2),AEE,EEMT(2),EEMMT(3),
X            EL(MTMX1),ELM(MTMX1,2),AEL,ELMT(2),ELMMT(3),
X            EM(MTMX1),AEM,EMMT(2)

```

C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C

```

*****
* OUTPUT CONTROL *
*****

```

```

READ AND PRINT OUTPUT CONTROL DATA
COMPUTE THE OUTPUT MASK IPR(*)

```

```

ALL DATA READ BY MAINC AND THE RESULTS ARE ALWAYS PRINTED
THE OUTPUT DURING THE SIMULATION IS DETERMINED BY THE OUTPUT
CONTROL MASK IPR(*) . IPR(*) CONTAINS THE OUTPUT PARAMETER
IPRINT FOR TIMEPOINT * . IPR(*) IS COMPUTED FROM THE FOLLOWING
OUTPUT CONTROL DATA READ BY MAINC :

```

```

IOUT POS      OUTPUT AT EVERY IOUT-TH TIMEPOINT
IOUT NEG      OUTPUT ONLY DURING THE TIMEPERIODS
                SPECIFIED BY NXOP, KXB(*) AND KXE(*)
NXOP          NUMBER OF TIMEPERIODS DURING WHICH OUTPUT
                IS PRODUCED IRRESPECTIVE OF THE VALUE OF IOUT
KXB(*)        START TIMEPOINT FOR PERIOD *
KXE(*)        END TIMEPOINT FOR PERIOD *

```

```

READ(5,10) IOUT,NXOP
10 FORMAT(2I5)
   IF (NXOP) 30,20,20
20 IF (NXOP-MTMX) 50,50,30
30 WRITE(6,40) NXOP
40 FORMAT(7HONXOP =,I5)
   GO TO 990

```



```
C
50 M=IABS(IOUT)
   IF (IOUT .LE. 0) M=MTMX1
   DO 60 I=1,MTMX
   IPR(I)=1
   IF (I/M*M .EQ. I) IPR(I)=0
60 CONTINUE
   IF (NXOP) 100,100,70
70 READ(5,10)(KXB(I),KXE(I),I=1,NXOP)
   DO 90 I=1,NXOP
   J1=IABS(KXB(I))
   J2=IABS(KXE(I))
   DO 80 J=J1,J2
80 IPR(J)=0
90 CONTINUE
100 WRITE(6,110) IOUT,NXOP
110 FORMAT(26H1MAIN PROGRAM FOR METHOD C/1X,25(1H*)//
   F20H0OUTPUT CONTROL DATA/1X,19(1H-)//
   F7H0IOUT =,15,5X,6HNXOP =,15)
   IF (NXOP) 140,140,120
120 WRITE(6,130)(KXB(I),KXE(I),I=1,NXOP)
130 FORMAT(7H0 TBEG,5X,4HTEND//(15,110))

C
C *****
C * READ AND PRINT FIXED DATA *
C *****
C
C READ AND PRINT TRUE DATA: NETWORK, GENERATOR, AND LOAD
C CONTROL DATA
C
C IPRINT=0
140 CALL ROTNET(IPRINT,IERR)
   IF (IERR) 150,150,990
150 CALL RDMETT(IPRINT)
   CALL RDGEN(IPRINT,IERR)
   IF (IERR) 160,160,990
160 CALL ROLD(SDB,SSL,KTMX,U,IPRINT,IERR)
   IF (IERR) 170,170,990

C
C READ AND PRINT ESTIMATOR DATA: NETWORK AND METER DATA,
C MEASUREMENT CHOICE AND ORDER
C
170 CALL RDENET(IPRINT,IERR)
   IF (IERR) 180,180,990
180 CALL RDMETE(IPRINT)
   CALL RDMSM(IPRINT,IERR)
   IF (IERR) 190,190,990

C
C READ AND PRINT PARAMETERS FOR ESTIMATOR C
C
190 READ(5,200) NEWA,MAXIT,CRLOSS,EPS
200 FORMAT(2I5,2F10.5)
   WRITE(6,210) NEWA,MAXIT,CRLOSS,EPS
210 FORMAT(21H0ESTIMATOR PARAMETERS/1X,20(1H-)/
```

F7H0NEWA =,I5,5X,7HMAXIT =,I5,5X,8HCRLOSS =,F10.5,
FSHEPS =,F10.5)

C
C
C
C
C
C
C
C
C
C

* INITIALIZATION *

TIMEPOINT 0: COMPUTE THE TRUE VARIABLES FROM THE START DEMAND
(READ BY RDL0 INTO SDB), SET THE ESTIMATE EQUAL TO THE TRUE
STATE, COMPUTE THE MEASUREMENTS AND INITIALIZE THE ESTIMATOR

K=0
WRITE(6,220) K
220 FORMAT(11H1TIMEPOINT ,I5/1X,15(1H*))
DO 230 I=1,NBT
230 XT(I)=(1.0,0.0)
CALL TRUEV(SDB,IPRINT+3,IERR)
IF (IERR) 240,240,990
240 CALL CAWN(IPRINT+1)
NODD=19
CALL ALLMSM(NODD,IPRINT+1)
CALL CAWF(IPRINT+1)
DO 250 I=1,NBE
250 XE(I)=XT(I)
CALL ADMC(1,1.0,MAXIT,EPS,IXIT,TIME,IPRINT+3)
IF (IXIT) 990,260,260
260 CALL EVAL(K,IPRINT)

C
C
C
C
C

* SIMULATION *

AEE=0.0
AEL=0.0
AEM=0.0
DO 270 I=1,2
EEMT(I)=0.0
ELMT(I)=0.0
270 EMMT(I)=0.0
DO 280 I=1,3
EEMMT(I)=0.0
280 ELMMT(I)=0.0

C

DO 310 K=1,KTMX
IPRINT=IPR(K)
IF (IPRINT .LE. 0) WRITE(6,220) K

C
C
C

COMPUTE THE LOAD DEMAND AND ALL TRUE VARIABLES

CALL CASDB(SDB,SSL,U(K),IPRINT)
CALL TRUEV(SDB,IPRINT+3,IERR)
IF (IERR) 290,290,990

C

```
C      COMPUTE THE STANDARD DEVIATIONS FOR THE MEASUREMENT NOISE,  
C      ALL POSSIBLE MEASUREMENTS AND THE WEIGHTING FACTORS FOR  
C      THE ESTIMATOR  
C  
290 CALL CAWN(IPRINT+1)  
    CALL ALLMSM(NODD,IPRINT+1)  
    CALL CAWF(IPRINT+1)  
C  
C      COMPUTE THE NEW ESTIMATE AND THE EVALUATION QUANTITIES  
C  
    CALL ADMC(NEWA,CRLOSS,MAXIT,EPS,IEXIT(K),TIME,IPRINT+3)  
    IF (IEXIT(K)) 990,300,300  
300 CALL EVAL(K,IPRINT)  
C  
310 CONTINUE  
C  
C      *****  
C      * PRINTOUT OF THE RESULTS *  
C      *****  
C  
    WRITE(6,320) KTMX  
320 FORMAT(17H1THE RESULTS FOR ,I5,11H TIMEPOINTS/1X,33(1H*)//  
    F12HOTIME  
    F,10HEST ERR  
    F,21HMAX ELT BUS ROI  
    F,11HLF ERROR  
    F,27HMAX ELT END LINE AOR  
    F,15HMSMT IND  
    F,5HIEXIT//  
    DO 420 K=1,KTMX  
    K1=K+1  
    I=INT(EEM(K1,2))  
    M=MOD(I,2)  
    J=I/2+M  
    IF (M) 330,330,340  
330 ROI=1HI  
    GO TO 350  
340 ROI=1HR  
350 I=INT(ELM(K1,2))  
    IF (I) 360,360,370  
360 AOB=1HB  
    GO TO 380  
370 AOB=1HA  
    I=IABS(I)  
    M=MOD(I,2)  
    J1=I/2+M  
    IF (M) 380,380,390  
380 AOR=1HR  
    GO TO 400  
390 AOR=1HA  
400 WRITE(6,410) K,EE(K1),EEM(K1,1),J,ROI,EL(K1),ELM(K1,1),AOB,J1,AOR,  
    1      EM(K1),IEXIT(K)  
410 FORMAT(I4,5X,2F10.7,I3,3X,A1,5X,2F10.7,2X,A1,I5,3X,A1,6X,F10.7,  
    F5X,I5)  
420 CONTINUE
```

```
WRITE(6,430) KTMX
430 FORMAT(14H1TOTALS AFTER ,I5,11H TIMEPOINTS/1X,29(1H*))
AEE=AEE/KTMX
WRITE(6,440) AEE
440 FORMAT(17H0ESTIMATION ERROR/1X,16(1H-)//
F6X,27HAVERAGE ESTIMATION ERROR =,F10.7)
KT=INT(EEMT(2))
WRITE(6,450) EEMT(1),KT
450 FORMAT(6X,27HMAXIMUM ESTIMATION ERROR =,F10.7,7H AT T =,I4)
I=INT(EEMT(2))
M=MOD(I,2)
J=I/2+M
IF (M) 460,460,470
460 ROI=4HIMAG
GO TO 480
470 ROI=4HREAL
480 KT=INT(EEMT(3))
WRITE(6,490) EEMT(1),KT,ROI,J
490 FORMAT(6X,27HMAX ELEMENT IN ALL ERRORS =,F10.7,7H AT T =,I4,
F4H IN /8X,A4,20H PART OF BUSVOLTAGE ,I2)
AEL=AEL/KTMX
WRITE(6,500) AEL
500 FORMAT(16H0LINE FLOW ERROR/1X,15(1H-)//
F6X,27HAVERAGE LINE FLOW ERROR =,F10.7)
KT=INT(ELMT(2))
WRITE(6,510) ELMT(1),KT
510 FORMAT(6X,27HMAXIMUM LINE FLOW ERROR =,F10.7,7H AT T =,I4)
I=INT(ELMT(2))
IF (I) 520,520,530
520 AOB=1HB
GO TO 540
530 AOB=1HA
540 I=IABS(I)
M=MOD(I,2)
J=I/2+M
IF (M) 550,550,560
550 AOR=6HREACT.
GO TO 570
560 AOR=6HACTIVE
570 KT=INT(ELMT(3))
WRITE(6,580) ELMT(1),KT,AOR,AOB,J
580 FORMAT(6X,27HMAX ELEMENT IN ALL ERRORS =,F10.7,7H AT T =,I4,
F4H IN /8X,A6,9H FLOW AT ,A1,13H-END OF LINE ,I2)
AEM=AEM/KTMX
WRITE(6,590) AEM
590 FORMAT(20H0MEASUREMENT QUALITY/1X,19(1H-)//
F6X,27HAVERAGE MEASUREMENT INDEX =,F10.7)
KT=INT(EMMT(2))
WRITE(6,600) EMMT(1),KT
600 FORMAT(6X,27HMAXIMUM MEASUREMENT INDEX =,F10.7,7H AT T =,I4)
```

C

```
C *****  
C * WRITE PLOTDATA TO PLOTFILE *  
C *****  
C  
    KTMX1=KTMX+1  
    IUNIT=1  
    WRITE(IUNIT) KTMX1,(EE(I),EL(I),EM(I),I=1,MTMX1)  
C  
990 STOP  
    END
```

```
C PLOTPROGRAM
C
C PARAMETER MTMX=360
C PARAMETER MTMX1=MTMX+1
C
C DIMENSION EE(MTMX1),EL(MTMX1),EM(MTMX1),XTEXT(4),YTEXT(3),Y(121)
C
C COMMON /RITFIG/ HTEX,HTEY,HNUX,HNUY,NWX,NWY,FMTX(3),FMTY(3),MX,MY
C
C DATA HTEX/0.24/,HTEY/0.24/,HNUX/0.24/,HNUY/0.24/,NWX/4/,NWY/-7/,
X FMTX(1)/6H(F4.0)/,FMTX(2)/6H /,FMTX(3)/6H /,
X FMTY(1)/6H(F7.5)/,FMTY(2)/6H /,FMTY(3)/6H /,
X MX/1/,MY/1/
C DATA X0/0./,DX/-1./,YMIN/0./,DY/1/,SY/4./,IX/-1/,IY/1/,NX/6/,
X IAXIS/1/,ITEXT/1/,IPLOT/0/,LINTYP/0/,XTEXT(1)/6H T/,
X XTEXT(2)/6HIME (S/,XTEXT(3)/6HAMPLES/,XTEXT(4)/6H) /
C
C READ(5,10) NPL
10 FORMAT(I5)
C
C DO 60 K=1,NPL
C CALL PLOTS(0,0,0)
C CALL PLOT(3.,0.,-3)
C
C IUNIT=K+10
C READ(IUNIT) KTMX1,(EE(I),EL(I),EM(I),I=1,MTMX1)
C Y0=0.0
C
C DO 50 I=1,KTMX1,120
C IF (I .EQ. KTMX1) GO TO 50
C XMIN=FLOAT(I-1)
C NP=MIN0(121,KTMX1-I+1)
C SX=FLOAT((NP-1)/NX)
C
C YTEXT(1)=6HESTIMA
C YTEXT(2)=6HTION E
C YTEXT(3)=6HRROR
C YMAX=1.0
C DO 20 J=1,NP
20 Y(J)=EE(I+J-1)
C CALL RITA(Y,NP,X0,Y0,XMIN,DX,YMIN,YMAX,DY,SX,SY,IX,IY,NX
C 1,IAxis,ITEXT,IPLOT,LINTYP,INTEQ,XTEXT,YTEXT)
C
C Y0=5.5
C
C YTEXT(1)=6HLINE F
C YTEXT(2)=6HLOW ER
C YTEXT(3)=6HROR
C YMAX=2.0
C DO 30 J=1,NP
30 Y(J)=EL(I+J-1)
C CALL RITA(Y,NP,X0,Y0,XMIN,DX,YMIN,YMAX,DY,SX,SY,IX,IY,NX
C 1,IAxis,ITEXT,IPLOT,LINTYP,INTEQ,XTEXT,YTEXT)
C
```

```
YTEXT(1)=6HMEASUR
YTEXT(2)=6HEMENT
YTEXT(3)=6HINDEX
YMAX=2.0
DO 40 J=1,NP
40 Y(J)=EM(I+J-1)
CALL RITA(Y,NP,X0,Y0,XMIN,DX,YMIN,YMAX,DY,SX,SY,IX,IY,NX
1,IAXIS,ITEXT,IPLOT,LINTYP,INTEQ,XTEXT,YTEXT)
C
Y0=7.0
50 CONTINUE
C
CALL PLOT(0.,0.,999)
C
60 CONTINUE
C
END
```

6. References.

1. Van Overbeek, A.J.M.; State Estimation in Power Networks I, a Literature Survey; Report 7331(©), Lund Institute of Technology, Division of Automatic Control, Lund, November 1973.
2. Van Overbeek, A.J.M.; State Estimation in Power Networks II, Comparison of Methods; Report 7403(©), Lund Institute of Technology, Division of Automatic Control, Lund, to appear.