16:05:45

OCA PAD INITIATION - PROJECT HEADER INFORMATION

06/07/90

Active

Project #: E-21-F69
Cost share #:

Center #: 10/24-6-R6955-0A0
Center shr #:

Contract#: B-6915 (EPRI TASK 1)
Mod #:

Prime #:

Subprojects ?: N
Main project #:

Project unit: EE
Unit code: 02.010.118

Project director(s):
MELIPOULOS A P EE (404)894-2926

Sponsor/division names: SOUTHERN COMPANY SERVICES /
Sponsor/division codes: 218 / 025

Award period: 890710 to 900330 (performance) 900330 (reports)

Sponsor amount
Contract value 7,992.00
Funded 7,992.00
Cost sharing amount 0.00

Does subcontracting plan apply ?: N

Title: BULK TRANSMISSION SYSTEM RELIABILITY EVALUATION FOR LARGE SCALE SYSTEM, ...

PROJECT ADMINISTRATION DATA

OCA contact: David B. Bridges 894-4820

Sponsor technical contact 
STAN SMITH (205)870-6011

SOUTHERN COMPANY SERVICES
P.O. BOX 2625
BIRMINGHAM, AL 35202

Sponsor issuing office 
GENE HORTON

(404)668-4007

SOUTHERN COMPANY SERVICES
64 PERIMETER CENTER
ATLANTA, GA 30346

Security class (U,C,S,TS) : U
Defense priority rating :
Equipment title vests with: Sponsor
NONE PROPOSED

Administrative comments -
PROJECT INITIATION, TASK 1 OF EPRI RP3159-1, DELIVERABLE DUE DATE IS 3/30
GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 02/14/91

Project No. E-21-F69
Project Director MELIPOULOS A P
Sponsor SOUTHERN COMPANY SERVICES/
Contract/Grant No. B-6915 (EPRI TASK 1)
Time Contract No.

Title BULK TRANSMISSION SYSTEM RELIABILITY EVALUATION FOR LARGE SCALE SYSTEM,

Effective Completion Date 900330 (Performance) 900330 (Reports)

<table>
<thead>
<tr>
<th>Closeout Actions Required</th>
<th>Y/N</th>
<th>Date Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Invoice or Copy of Final Invoice</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Final Report of Inventions and/or Subcontracts</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Government Property Inventory &amp; Related Certificate</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Classified Material Certificate</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Release and Assignment</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

Comments________________________

Subproject Under Main Project No. __________________

Continues Project No. __________________

Distribution Required:

<table>
<thead>
<tr>
<th>Distribution Required</th>
<th>Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Director</td>
<td>Y</td>
</tr>
<tr>
<td>Administrative Network Representative</td>
<td>Y</td>
</tr>
<tr>
<td>GTRI Accounting/Grants and Contracts</td>
<td>Y</td>
</tr>
<tr>
<td>Procurement/Supply Services</td>
<td>Y</td>
</tr>
<tr>
<td>Research Property Management</td>
<td>Y</td>
</tr>
<tr>
<td>Research Security Services</td>
<td>N</td>
</tr>
<tr>
<td>Reports Coordinator (OCA)</td>
<td>Y</td>
</tr>
<tr>
<td>GTRC</td>
<td>Y</td>
</tr>
<tr>
<td>Project File</td>
<td>Y</td>
</tr>
<tr>
<td>Other</td>
<td>N</td>
</tr>
</tbody>
</table>


MEMORANDUM

TO: Kathy Knighton
FROM: A.P. Sakis Meliopoulos
2) Production Grade Code .  
Project E21 - F68  
Sponsor: Southern Company Services

Enclosed please find the above reference report. The production code has been delivered to sponsor. Please submit through regular channels.

APSM/XYC

cc: Dr R.P.Webb, Director EE
Interim Report

BULK TRANSMISSION SYSTEM RELIABILITY EVALUATION
PRELIMINARY PROGRAM REFERENCE MANUAL

Prepared By:
Rhonda H. Cook
Kimberlie A. Hall
Murali Kumbale
Beverly B. Redwine
Telly Rusodimos
Southern Company Services, Inc.
Atlanta, Georgia 30346

A. P. Sakis Meliopoulos
Carol Cheng
King Yong Chao
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250

Murtty P. Bhavaraju
Public Service Electric & Gas Company
Newark, New Jersey 07101

Prepared for
Electric Power Research Institute
Palo Alto, California 94304

EPRI Project Manager
Mark G. Lauby

January 16, 1991
Notice

This report was prepared by the organization(s) named below as an account of work sponsored by the Electric Power Research Institute, Inc. (EPRI). Neither EPRI, members of EPRI, the organization(s) named below, nor any person acting on behalf of any of them: (a) makes any warranty, express or implied, with respect to the use of any information, apparatus, method, or process disclosed in this report or that such use may not infringe privately owned rights; or (b) assumes any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method, or process disclosed in this report.

Prepared by
Southern Company Services, Inc.
Atlanta, Georgia
CONFIDENTIALITY NOTE

The design specifications in this report have been prepared for the Electric Power Research Institute. They are to be reviewed by EPRI personnel and EPRI advisors. The report should be treated as confidential material.
EPRI PERSPECTIVE

(to be added)
ACKNOWLEDGEMENTS

(to be added)
# Table of Contents

1.1.1.1 AC Remedial Action Module - ACREMD ........................................ 4  
1.1.1.2 Classification of Power Flow Solution Submodule - PFCHCK ........ 8  
1.1.1.3 Constraint Identification Submodule - ACIDNT .......................... 11  
1.1.1.4 Constraint Coherency Analysis Submodule - ACCOHR .............. 16  
1.1.1.5 Constraint Linearization Submodule - ACLINZ ......................... 22  
1.1.1.6 Power Flow Update Submodule - ACPFUP ................................ 43  
1.1.1.7 DC Remedial Action Module - DCRAM ..................................... 50  
1.1.1.8 Constraint Identification Submodule - DCIDNT ...................... 55  
1.1.1.9 Constraint Coherency Analysis Submodule - DCCOHR ................ 60  
1.1.1.10 Constraint Linearization Submodule - DCLINZ ....................... 67  
1.1.1.11 Power Flow Update Submodule - DCPFUP ............................... 78  
1.1.1.12 Data Preparation Submodule - RAINIT ................................ 83  
1.1.1.13 Jacobian Formation Submodule - CMPJAC ............................... 96  
1.1.1.14 Linear Program Set-Up Submodule - LPMSET .......................... 98  
1.1.1.15 Linear Program Solution Submodule - LPSOLV ...................... 103  
1.1.1.16 General Network Analysis Support - DBSUPR .......................... 113  

2.0 References ........................................................................... 115  

INDEX ..................................................................................... 115
List of Illustrations
Table of Contents

Remedial Actions, Module: RAMAIN

AC Remedial Actions

AC Remedial Actions Module, ACREMD
Classification of PF Solution Submodule, PFCHCK
Constraint Identification Submodule, ACIDNT
Constraint Coherency Analysis Submodule, ACCOHR
Constraint Linearization Submodule, ACLINZ
Power Flow Update Submodule, ACPFUP

DC Remedial Actions

DC Remedial Actions, DCREMD
Constraint Identification Submodule, DCIDNT
Constraint Coherency Analysis Submodule, DCCOHR
Constraint Linearization Submodule, DCLINZ
Power Flow Update Submodule, DCPFUP

Support Submodules

Data Preparation Submodule, RAINIT
Jacobian Formation Submodule, CMPJAC
LP Setup Submodule, LPMSET
LP Solution Submodule, LPSOLV

Data base / Variable Dictionary
REMEDIAL ACTIONS MODULE: RAMAIN

The remedial action module is executed whenever a system problem still exists after the network solution. There are two remedial actions modules. The other is based on the DC Network model. One is based on the AC Network model. The two modules will be referred to as AC and DC Remedial Action Modules respectively. Each Remedial Action module consists of several submodules. Four submodules are shared by both remedial actions. A list of the submodules is illustrated in "Table 1 List of Submodules in the AC and DC Remedial" on page 3. This section provides the programmers manual for the two modules.

The remedial actions module consists of a main routine and several supporting routines. A brief description of these routines follows:

Subroutine RAMAIN

Description: This routine is the main remedial actions routine. It defines the I/O channels, default control parameters, executes the AC or DC remedial actions program and closes the report files.

Output Arrays: ACFLAG and Remedial Actions.

Routines Called: SETCNT, DCREMD, ACREMD

Subroutine SETCNT

Description: This routine defines the default values of all control variables and constants used in the remedial actions program. The control variables and constants are defined in the TRUCE/Remedial Actions Interface, Section 2.

Subroutine DIAGN1

Description: This routine supplies a warning message passed by character variable ASCR. The warning message goes to the I/O channel defined by ICHOUT.

Subroutine FATAL1

Description: This routine puts an error message into the I/O channel defined by ICHOUT and terminates the whole program.

Subroutine MESSAG

Description: This routine shows the message passed by character variable ASCR. The I/O channel is defined by ICHOUT.
Table 1 List of Submodules in the AC and DC Remedial Actions
Table 1. List of Submodules in the AC and DC Remedial Actions

<table>
<thead>
<tr>
<th>Submodules which are identical for both AC and DC Remedial Actions</th>
<th>AC Remedial Actions</th>
<th>DC Remedial Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Base Preparation Submodule RAINIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jacobian Formation Submodule CMPJAC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP Setup Submodule LPMSET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP Solution Submodule LPSOLV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Submodules which are different for AC and DC Remedial Actions

| Classification of PF Solution Submodule PFCHCK               |                   |                   |
| Constraint Identification Submodule ACIDNT                   |                   |                   |
| Constraint Coherency Analysis Submodule ACCOHR              |                   | Constraint Coherency Analysis Submodule DCCOHR |
| Constraint Linearization Submodule ACLINZ                    |                   | Constraint Linearization Submodule DCLINZ |
| Power Plow Update Submodule ACPFUP                           |                   | Power Plow Update Submodule DCPFUP |

Data Base Preparation Submodule RAINIT
Jacobian Formation Submodule CMPJAC
LP Setup Submodule LPMSET
LP Solution Submodule LPSOLV

Constraint Identification Submodule DCIDNT
Constraint Coherency Analysis Submodule DCCOHR
Constraint Linearization Submodule DCLINZ
Power Plow Update Submodule DCPFUP
1.1.1.1 AC Remedial Action Module - ACREMD

MODULE DESCRIPTION AND STRUCTURE

The AC remedial action computational module will be executed whenever a problem still exists after the network solution. Among the problems to be corrected are: circuit overloads, bus voltage violations, reactive power and area net export violations. Results of the remedial action calculation includes load curtailment at individual buses. The appropriate remedial is computed by the linear program algorithm described in Reference 6.

The list of AC remedial actions is shown in "Table 2 List of AC Remedial Actions" on page 10. The user may select the types, locations, and priorities of control actions be used in the study. A flow chart of the remedial action module is illust "Figure 3.3.31 AC Remedial Action Module - ACREMD" on page 11.

Up to ten priorities of remedial actions may be specified by the user. An option allows to have remedial actions purposely excluded from the priority list.

For remedial action purposes, interruptible load (-and critical loads, future) is specified in terms of megawatts and megavars of each load type at each bus. Firm is derived as total bus load less interruptible load. Load shedding is under constant power factor.

The AC Remedial Action Module utilizes the following submodules:

Data Base for Remedial Action Submodule - RAMINT

This submodule generates the data base required in the remedial action module. It operates on the input data to define existing remedial actions and their limits. It also allows user specified remedial actions.

Verification of the Definition of Areas - RAMDQR

This submodule checks for the consistency of the flag defining the STUDY AREA and REMEDIAL ACTION AREA

Classification of Incoming PF Solution - PFCHCK

This submodule classifies the incoming power flow solution as (a) converged (b) valid and (c) diverged. Only converged and valid power flow solutions are processed. Otherwise control is returned to main program.

Jacobian Formation Submodule - CMPJAC

This submodule forms or modifies the Jacobian matrix for a specific network condition. Three alternatives are available: (1) full Jacobian, (2) decoupled constraint Jacobian
consisting of B' and B", (3) DC network model Jacobian. Subsequently, the Jacobian matrix has been triangularized.

Constraint Identification Submodule - ACIDNT

This submodule identifies branch overloads, low or high bus voltages, bus voltage deviations above the limits, generator reactive power output violations. The violations are stored. They are also compared against tolerance to determine whether convergence has occurred.

Constraint Coherency Analysis Submodule - ACCOHR

This submodule performs the coherency analysis among failed operating constraints. It identifies the constraints which may be active in the solution, i.e., those constraints dictate the remedial actions. Only these constraints will be included in the solution process. The constraint screening procedure is performed by sensitivity analysis of selected remedial actions in the neighborhood of the constraints. If the sensitivities of two or more constraints are within a user-selected threshold, the constraints are placed within the same coherent group.

Constraint Linearization Submodule - ACLINZ

This submodule performs the linearization of the constraints to be included in the remedial action solution process. Three modes of computation will be allowed: by means of the full Jacobian matrix, the Newton-Raphson loadflow, the diagonal constant matrices from the fast-decoupled loadflow, or Jacobian matrix from DC network. The choice of the matrix to be used could greatly affect the speed of computations, and will be a user-selected option. Sensitivities below a specified threshold will be eliminated from the solution process. The bounds of validity of the linearized model will be checked.

LP Setup Submodule - LPMSET

This submodule sets up the Linear Program given the linearized constraints. In addition, it performs a model reduction.

Power Flow Update Submodule - ACPFUP

This submodule performs an update of the power flow solution after each major iteration of the remedial actions algorithm.

LP Solution Submodule - LPSOLV

This submodule performs the solution of the LP problem which consists of a mixed integer LP algorithm.

AC Remedial Action Report - ACREPT: This submodule supplies the corresponded reports for every submodule used by AC Remedial Actions. Such reports are on: the
sensitivity, violations of circuit flows, bus voltage and reactive generations, coherent violation, results from LPMSET, total computed remedial actions etc.
Table 2  List of AC Remedial Actions

<table>
<thead>
<tr>
<th>AC Capability Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MW Generation Adjustment — Slack Bus</td>
</tr>
<tr>
<td>2. MW Generation Adjustment — Slack Bus</td>
</tr>
<tr>
<td>3. MVAR Generation Adjustment</td>
</tr>
<tr>
<td>4. Generation Bus Voltage Control</td>
</tr>
<tr>
<td>5. Reserve Unit Startup (future)</td>
</tr>
<tr>
<td>6. Phase Shifter Adjustment</td>
</tr>
<tr>
<td>7. Transformer Tap Adjustment</td>
</tr>
<tr>
<td>8. Load Transfer</td>
</tr>
<tr>
<td>9. Interruptible Load Curtailment</td>
</tr>
<tr>
<td>10. Firm Load Curtailment</td>
</tr>
<tr>
<td>11. Critical Load Curtailment (future)</td>
</tr>
<tr>
<td>12. Area Interchange Adjustment</td>
</tr>
<tr>
<td>13. Shunt Reactor Switching (Discrete or Continuous)</td>
</tr>
<tr>
<td>14. Shunt Capacitor Switching (Discrete or Continuous)</td>
</tr>
</tbody>
</table>
Figure 3.3.31 AC Remedial Action Module - ACREMD

START

PREPARE REMEDIAL ACTION DATABASE: RAINIT

CLASSIFICATION OF PF SOLUTION PFCHCK FIGURE 3.3.31.1

DIVERGED?

YES

CONSTRAINT IDENTIFICATION SUBMODULE ACIDNT FIGURE 3.3.31.2

CONVERGED?

YES

STORE/RETRIEVE SOLVED CONTINGENCY SUBMODULE $SRSCON FIGURE 3.3.30.16

RETURN

NO

CONSTRAINT COHERENCY SUBMODULE ACCOHR FIGURE 3.3.31.3

CONSTRAINT LINEARIZATION SUBMODULE ACLINZ FIGURE 3.3.31.4

LP SET-UP SUBMODULE LPMSET FIGURE 3.3.33.3

LP SUBMODULE LPSOLV FIGURE NONE

POWER FLOW UPDATE SUBMODULE ACPIUP FIGURE 3.3.31.5

COMPUTE SENSITIVITIES
1.1.1.2 Classification of Power Flow Solution Submodule - PFCHCK

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule checks for the consistency of the incoming data and power flow solution. Specifically, for the incoming power flow solution, computes the real and reactive power mismatch at all buses for the specified settings of all remedial actions. In addition, it checks whether all specified settings are within the natural limits of the remedial actions. Subsequently, it returns two codes: (a) A code determining whether all settings are feasible and (b) A code classifying the power flow solution as (1) converged (2) valid or (3) diverged. A converged power flow is one for which all mismatches are below the mismatch tolerances used in the power flow solution. A valid power flow solution is one for which the maximum mismatch is above the mismatch tolerance of the power flow solution but below a user specified value (for example 20 MVA). This module is only used in AC remedial action case.

Input Array/Variables: Array of a solved contingency case

Output Arrays/Variables: CUFLAG and PFSTYP

Subroutines called: RAMDGN, CULMCH, PQMSCH

Subroutine Name: CULMCH

Description: This routine checks the limits of the control variables and determines consistency. If inconsistent, the logical variable CUFLAG is set to FALSE.

Input Arrays: Remedial action data bank

Output Arrays: CUFLAG

Subroutine Name: PQMSCH

Description: This routine computes the bus power mismatches and classifies the present solution into

(a) Converged power flow
(b) Valid power flow
(c) Diversed power flow

Input Arrays: Remedial action data bank

Output Array: PFSTYP

Routines Called: PMSMCH, QMSMCH, DCLMSM
Subroutine Name: RAMDGN

Description: This module checks consistency of the flag defining the STUDY AREA and the REMEDIAL ACTION AREA. If inconsistency happens, it issues an error message and the corresponding arrays are set to 'FALSE.' No equivalent bus or circuit is allowed to be a control in the remedial action.

Input Arrays: LBEQW, LCEQW, LGEQW

Output Arrays: RABFLG, RACFLC, SACFLG, SABFLG
CHECK FLAG FOR STUDY AND REMEDIAL ACTION ORDER

CHECK SETTING OF REMEDIAL ACTIONS FOR FEASIBILITY
- UNIT MW SETTING
- UNIT MVAR SETTING
- UNIT BUS VOLTAGE
- PHASE SHIFTER SETTING
- TRANSFORMER TAP SETTING
- ELECTRIC LOADS (INTERR., FIRM, TRANSFER)
- AREA GENERATION
- SHUNT REACTOR / CAPACITOR

LF FEASIBLE \[\Rightarrow\] CUFLAG = .T.

PERFORM CHECKS
\[
\text{MAX (BUS POWER MISMATCH)} < E_1 \Rightarrow PFSTYP = 1 \\
E_1 < \text{MAX (BUS POWER MISMATCH)} < E_2 \Rightarrow PFSTYP = 2 \\
\text{ELSE} \quad PFSTYP = 3
\]

PFSTYP = 3

Figure 3.3.31.1 Classification of Power Flow Solution Submodule - PFCHCK
Flow Chart
1.1.1.3 Constraint Identification Submodule - ACIDNT

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule identifies branch overloads, low or high bus voltages, bus voltage deviations above limit, generator reactive power output violations, and MW export violations. The violations are stored. They are also compared against tolerance values to determine whether convergence has occurred.

The flow chart for this submodule is illustrated in "Figure 3.3.31.2 Constraint Identification Submodule - ACIDNT" on page 15.

Control Variables:

EVTOLW : Tolerance to determine a voltage constraint is effective.

EFTOLW : Tolerance to determine whether a flow constraint is effective

EQTOLW : Tolerance to determine whether a reactive constraint is effective

Input Arrays/Variables:

Arrays of a solved contingency case

Output Arrays/Variables:

Arrays describing operating constraints

Subroutines Called:

STOVAC, CNUPDT, CNVCHK

Error Messages:

None

Validation Test:

This submodule has been validated manually as follows: A complete power flow report is generated as well as a report of all violated constraints. The consistency of these two reports is checked manually.

A description of the subroutines called by this submodule follows:

Subroutine Name: CNUPDT

Brief Description: This routine computes the operating constraints of the system by identifying voltage operating constraints, line flow operating constraints, generator reactive operating constraints. The constraints are packed by eliminating the identical ones. Also the convergence of remedial actions is
checked.

Control Variables:  
- EVTOLW : Tolerance to determine a voltage constraint is effective.
- EFTOLW : Tolerance to determine whether a flow constraint is effective.
- EQTOLW : Tolerance to determine whether a reactive constraint is effective.

Input Arrays/Variables:  This routine also uses a large number of arrays from the data bank.

Output Arrays/Variables:  Operating constraints as follows: IPVOC, DVVO ICVOC, ICHVOC, DTVOC, DITVOC, IPSVOC, DCVOC, IEVOC, ILVOC, NTVOC, IPFOC, ICHFOC, DPFOC, DITFOC, IPSFOC, DCFOC, IEFOC, ILFOC, NTFOC, IPQOC, DQQOC, ICQOC, ICHQOC, DTQOC, DITQOC, IPSQOC, DCQOC, IEQOC, ILQOC, NTQOC.

NVCOLD : Old number of voltage operating constraints
NPCOLD : Old number of circuit flow operating constraints
NQCOLD : Old number of reactive power operating constraints

Other Subroutines Called: ACRVOL

Methodology/Comments:  Positive deviations denote violations. Negative deviations denote constraint near rating. Validation Tests: The defined constraints should be validated by using a power flow program.

Subroutine Name: STOVAC

Brief Description:  This routine performs for all the old operating constraints the following:

1. Stores the old values
2. Computes the new values
3. Updates the total correction
4. Determines whether constraint is effective in Linear Programming Model.

Input Arrays/Variables:  When this routine is called, the following variables should have been defined:

NTVOC : Current number of voltage operating...
constraints, if not equal to zero, computes the voltage constraints.

NTFOC: Current number of circuit flow operating constraints, if not equal to zero, computes the flow constraints.

NTQOC: Current number of reactive power operating constraints, if not equal to zero, computes the reactive constraint.

DVVOC, IPVOC, NORDER, VOLT, VMAXW, ICVOC, VMINW, DTVOC, IPSVOC, SOLLGP, DPFOC, IPFOC, PLFL1, QLFL1, PLFL2, QLFL2, PBASE, TRW, IPSFOC, KTLPV, DITFOC, DQQOC, IQPQC, IGORDW, QGTLW, QGMXW, QGMNW, DTQOC, DITQOC

Output Arrays/Variables: DTVOC, DITVOC, SOL, ASOL, IEVOC, DPFOC, DTFOC, DITFOC, DQQOC, DTQOC, DITQOC, IEQOC, IEFOC

Other Subroutines Called: None

Methodology/Comments:

Validation Tests:

Subroutine Name: CNVCHK

Brief Description: This routine checks for convergence of the remedial action module. If all violations have been alleviated, NORMAL is set to 1, which informs the main RA program to stop iteration.

Input Arrays/Variables: When this routine is called, the following convergence criteria should have been specified: SMOVV, SMOFV, SMOQV, SMOXP.

Output Arrays/Variables: A report stating convergence status, and NORMAL

Other Subroutines Called: None

Methodology/Comments: This routine computes the maximum voltage violation, maximum circuit flow violation, and maximum unit reactive power violation. Subsequently, these values are compared to specified convergence criteria. If below the criteria, the remedial action computations have converged.

Validation Tests:

Subroutine Name: ACRVOL
Brief Description: This routine checks whether at a bus with a violated reactive generation, the voltage is a control variable.

Input Arrays/Variables: LTYPLF, NCVG and optimal bus number of the violated reactive generation bus.

Output Arrays/Variables: Report on the terminal.

Validation Tests:
FOR ALL THE OLD OPERATING CONSTRAINTS STORE THE OLD VALUES, COMPUTE THE NEW VALUES, UPDATE TOTAL CORRECTION AND DETERMINE IF CONSTRAINTS ARE EFFECTIVE IN LP.

IDENTIFY VOLTAGE OPERATING CONSTRAINTS, CIRCUIT FLOW OPERATING CONSTRAINTS, AND GENERATING BUS REACTIVE POWER CONSTRAINTS

PACK OPERATING CONSTRAINTS AND ELIMINATE IDENTICAL ONES.

INTERCHANGE AREAS \( >= 2 \)?

YES

ACTIVATE NET MW EXPORT CONSTRAINTS

CHECK CONVERGENCE OF REMEDIAL ACTIONS

RETURN

NO
1.1.1.4 Constraint Coherency Analysis Submodule - ACCOHR

**SUBMODULE DESCRIPTION AND STRUCTURE**

This submodule identifies the constraints which may be active in the solution, i.e., those constraints which dictate the remedial actions for the mitigation of the constraint violations. This is done by identifying groups of coherent constraints and selecting one constraint per group. If the sensitivities of two or more constraints are within a user selected threshold, the constraints are placed within the same coherent group. "Figure 3.3.31.3 Constraint Coherency Submodule - ACCOHR" on page 21 illustrates the algorithm. This algorithm is applied to circuit overload constraints, voltage constraints, and bus reactive power constraints.

Control Variables:

- **XACCOH**: Threshold value for constraint coherency analysis

Input Array: Data bank

Output Array: Results for coherency analysis of the old constraints, voltage constraints, circuit flow constraints, reactive constraints, and the logic for relinearization of constraints, are stored in the following arrays:

- KGVOLD, KGOLD, KGQOLD, ICHVOC, ICHFOC, ICHQOC, IRBUS, ILVOC, DITVOC, ILFOC, DITFOC, ILQOC, DITQOC

Subroutines Called:

- MDAC6A, MDAC6B, MDAC6C, LOGAC1, LOGAC2, LOGAC3

Error Messages:

None

Validation Tests:

The validation test consists of observing the coherency capture ratio, i.e., how many constraints have been included in the LP model and found to be binding constraints. This test should be repeated for many contingency cases whenever the reliability program is complete.

---

Subroutine Name: MDAC6A

Brief Description: This subroutine performs coherency analysis for voltage constraints. When this subroutine is called, the old number of voltage operating constraints NVCOLD, the current number of voltage operating constraints NTVOC, and the old number of voltage operating constraints included in LP KGVOLD should have been specified.

Input Array: Data bank.
Output Arrays Variables: ICHVOC and sensitivity coefficient YASCR.

Other Subroutines Called: WORST, ZOOM4, SOLVT, CSLFLW, COHGRP

Methodology/Comments: The principle of coherent network setting up and solution is the same with what mentioned in subroutine MOD4G.

Validation Tests: Same as in routine MOD4G.

-------------------------------

Subroutine Name: MDAC6B

Brief Description: This subroutine performs the coherency analysis for circuit flow constraints. When this routine is called, the old number of circuit flow operating constraints NPCOLD, the current number of circuit flow operating constraints NTFOC, and the old number of total circuit flow constraints included in LP KGFOLD, should have been specified.

Input Array: Data Bank.

Output Arrays Variables: ICHFOC and sensitivity coefficient YASCR.

Other Subroutines Called: WORST, ZOOM4, SOLVT, CSLFLW, COHGRP

Methodology/Comments: The principle of coherent network setting up and solution is the same with what mentioned in subroutine MOD4G.

Validation Tests: Same as in routine MOD4G.

-------------------------------

Subroutine Name: MDAC6C

Brief Description: This routine performs coherency analysis for the reactive constraints. When this routine is called, the old number of reactive power constraints NQCOLD, the current number of reactive power constraints NTQOC, and the old number of reactive power constraints included in LP KGQOLD, should have been specified.

Input Array: Data Bank.

Output Arrays Variables: ICHQOC, and sensitivity coefficient YASCR.

Other Subroutines Called: WORST, SOLVT, CSGENQ, COHGRP

Methodology/Comments: The principle of coherent network set-up and solution is the same with what is mentioned in subroutine MOD4C.
Validation Tests: Same as in routine MOD4C

Subroutine Name: LOGAC1

Brief Description: This routine defines the logic for the coherency of the old constraints. When this routine is called, the number of voltage operating constraints NV Cold, the number of circuit flow operating constraints NPCOLD, and the old number of reactive power constraints NQCOLD, should have been specified.

Input Array: ICHVOC, IPSVOC, DVVOC, SMOV, ICHFOC, IPSFOC, DPFOC, ICHQOC, IPSQOC, DQQOC, SMOFV, SMOQV

Output Arrays / Variables: ICHVOC, ICHFOC, ICHQOC, KGVOLD, KGFOLD, KQQOLD

Other Subroutines Called: None

Subroutine Name: LOGAC2

Brief Description: This routine defines the logic for the linearization of the old/new constraints. The current number of voltage operating constraints NTVOC, current number of circuit operating constraints NTFOC, and current number of reactive power operating constraints NTQOC, should have been specified.

Input Array: ICHVOC, KGVOLD, IEVOC, RELVOC, ICHFOC, KGFOLD, IEFOC, RELFOC, ICHQOC, KQQOLD, IEQOC, RELQOC

Output Arrays / Variables: ILVOC, DITVOC, ILFOC, DITFOC, ILQOC, DITQOC

Other Subroutines Called: None

Subroutine Name: LOGAC3

Brief Description: This routine defines the logic for the coherency of old reactive constraints which might be switched from lower to upper limit or vice versa. When this routine is used, the following control should be specified:
- If NTVOC ≠ 0, process voltage constraints
- If NTQOC ≠ 0, process reactive power constraints

Input Array: IPSVOC, IEVOC, DVVOC, ICOLP, ICVOC, IPSQOC, IEQOC, DQQOC, ICQOC

Output Arrays Variables: ILVOC, DITVOC, ILQOC, DITQOC
Subroutine Name: WORST

Brief Description: This routine computes the worst operating constraint. Specifically, given a specified violation array DOC, and corresponded undefined coherent group pointer ICHOC, it computes the KEY and K, where KEY is the number of new constraints in the undefined coherent group (pointed by ICHOC (*)) = 0, and K is the worst constraint pointer.

Input Arrays / Variables: Local variable arrays:
ICHOC, DOC, NT, ND

Output / Arrays Variables: KEY, K

Other Subroutines Called: None

Subroutine Name: COHGRP

Brief Description: This routine computes the coherent groups of the operating constraints. Specifically given the worst violation pointer KW, the group number KG, and the undefined coherent group pointer ICHOC, it defines the coherent group according to the worst pointer and assigns the less sensitive constraints as the followers to this coherent group.

Input Arrays / Variables: When this routine is called, KW, the pointer which points the worst violation and KG, the coherent group number should be defined.

Other Input Arrays and Variables: SNS, the normalized sensitivity coefficient;
XCU, tolerance for constraint coherency;
NT and ND, present and old number of constraints, respectively.

Output Arrays / Variables: ICHOC

Other Subroutines Called: None

Subroutine Name: ZOOM4

Brief Description: This routine computes the buses around circuit
IFCIR which gives the specifications for ZOOM.

Input Arrays / Variables: In addition to global variables, a local variable is IZBUS.

Output Arrays / Variables: IRBUS

Other Subroutines Called: None

Methodology/Comments: The output array IRBUS defined as follows:

IRBUS(I) = 1, report bus I
IRBUS(I) = 0, do not report bus I

where I denotes bus number in internal numbering.

All data are defined in internal numbering. Working arrays are utilized. Data are for individual circuits.

Subroutine Name: CSLFLW

Brief Description: This subroutine computes the circuit flows from the computed voltages and phase angles for the specified circuit IC (internal numbering of the circuit).

Input Arrays / Variables: IC, a dummy input variable indicating the circuit.

Output Arrays / Variables: Real and reactive power injections to the circuit from both ends. P11, QL1, PL2, QL2, the dummy output variables.

Subroutines Called: None

Subroutine Name: CSGENQ

Brief Description: This subroutine computes the generation reactive power for a specified generation bus.

Input Arrays / Variables: IG, a dummy input variable indicating the generation bus (optimal numbering)

Output Arrays / Variables: QG, the reactive power generated.

Subroutines Called: None
Identify the Set of Constraints to participate in the process by excluding those which have been linearized. Call this set A.

Is Set A empty?

- YES
  - Return

- NO
  - Select the highest violation constraint. Apply remedial actions in the vicinity of the highest violation constraint. Protect solution with a linearized model
  - Identify all constraints with a projected deviation DV greater than $a \times DV_{max}$ where $a$ is the maximum selected input and $DV_{max}$ the maximum projected deviation
  - Exclude above constraints from set A
  - Flag constraint with $DV_{max}$ as the constraint to be included in the LP model
1.1.1.5 Constraint Linearization Submodule - ACLINZ

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule performs the linearization of the constraints to be included in the remedial action solution process. Three modes of computation are allowed: by means of the full Jacobian matrix from the Newton-Raphson loadflow, the diagonal constant matrices from the fast decoupled loadflow, or a semicoupled Jacobian matrix. The choice of the matrix to be used affects the speed of computations, and is user-selected as it is described in the Jacobian formation submodule. This submodule is designed to work with any selected Jacobian. The computed sensitivity is screened as follows: Sensitivities below a user specified threshold will be eliminated.

The flow chart for this submodule is illustrated in "Figure 3.3.31.4 Constraint Linearization Submodule - ACLINZ" on page 25.

Control Variables:

NSOPT  Control Variable indicating type of constraint to be linearized
= 1 - Circuit loading (MVA flow)
  2 - Bus voltage constraint
  3 - Generator reactive output constraint
  4 - Generator real power output constraint
  5 - Transmission losses
  6 - Power balance
  7 - Circuit real power flow
  8 - Net MW export constraint

JACMOD  Control Variable indicating the availability of Jacobian matrix when computing the sensitivities.
= 3 - Form Jacobian matrix and triangularize it.
  2 - Triangularize the Jacobian
  1 - Proceed, Jacobian matrix has been already done

Input Arrays/Variables:

Data Base

Output Arrays/Variables:

SNSAUG and SJSCR both contain the sensitivities of the specified constraints with respect to all control variables.

Modified Arrays/Variables:

Same as output arrays/variables. For example, in case of no new constraints, the above arrays/variables will be simply modified.

Subroutines Called:

ACCLNZ, ELSSLP, FORMS1
Error Messages:

None

Validation Tests: Two validation tests are performed:

(a) The computed linearization coefficients are also computed by brute force using a usual power flow. The two values are manually compared. They must be approximately equal.

(b) By comparing the projected solution from the linear program submodule to an actual solution with a power flow after application of the LP computed remedial actions. The two must be approximately equal.

Methodology/Comments: Linearization of the operating constraint \( f(x,u) \) around an operating point defined with control variables \( u = u^0 \) and state variables \( x = x^0 \), yields

\[
f(x,u) = f(x^0,u^0) + \sum_i \frac{df(x,u)}{du_i} (u_i - u^0)
\]

The sensitivities \( \frac{df(x,u)}{du_i} \) are computed based on the costate method,

\[
\frac{df(x,u)}{du_i} = \frac{af(x,u) - [af(x,u)]}{au_i} \frac{ag(x,u)}{ax} \frac{ag(x,u)}{au_i}^{-1}
\]

where \( \frac{ag(x,u)}{ax} = J \) is the Jacobian matrix of the power flow equation.
Subroutine Name: ACCLNZ

Brief Description: This routine makes calls to several subroutines to set up the Jacobian and linearize eight types of constraints.

Control Variables:

NSOPT Control Variable indicating type of constraint to be linearized
   = 1 - Circuit loading (MVA flow)
   2 - Bus voltage constraints
   3 - Generator reactive output constraint
   4 - Generator real power output constraint
   5 - Transmission losses
   6 - Power balance
   7 - Circuit real power flow
   8 - Net MW export constraint

JACMOD Control Variable indicating the availability of Jacobian matrix when computing the sensitivities.

   = 3 - Form Jacobian matrix and triangularize it
   2 - Triangularize the Jacobian
   1 - Proceed, Jacobian matrix has been already done

Input Arrays/Variables:

Data Base

Output Arrays/Variables:

SJSCR, which contains the sensitivities of the specified constraints with respect to all control variables.

Subroutines Called: CMPJAC, JACFPN, MOD4A, MOD4B, MOD4C, MOD4D, MOD4E
                   MOD4F, MOD4G, MOD4H, CSNS, CSNSA, CSNSC, CSNSD, CSNSE, CSNSF, CSNSH, LUFCTO
Figure 3.3.31.4 Constraint Linearization Submodule - ACLINZ

- ENTER
- NSOPT
- = 1
  - LINEARIZE CIRCUIT LOADING CONSTRAINT
- = 2
  - LINEARIZE BUS VOLTAGE CONSTRAINT
- = 3
  - LINEARIZE GENERATOR REACTIVE OUTPUT CONSTRAINT
- = 4
  - LINEARIZE GENERATOR REAL OUTPUT CONSTRAINT
- = 5
  - LINEARIZE SYSTEM TOTAL LOSSES CONSTRAINT
- = 6
  - LINEARIZE POWER BALANCE EQUATION CONSTRAINT
- = 7
  - LINEARIZE CIRCUIT REAL POWER FLOW CONSTRAINT
- = 8
  - LINEARIZE NET MW EXPORT CONSTRAINT

MORE CONSTRAINTS TO BE LINEARIZED?

YES
- ENTER

NO
- RETURN
Subroutine Name: MOD4A

Brief Description: This routine linearizes a given circuit loading (MVA flow) constraint. When the routine is called, the pointer of circuit constraint to be linearized (internal numbering) NPOPCC, must have been specified a flow chart of the routine is illustrated in "Figure 3.3.31.4.1 Constraint Linearization Submodule Circuit" on page 28.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: The sensitivity vector of selected circuit in array SJSCR.

Other Subroutines Called: SLVTR1, SLVTR2, CSNS, CSNSA

Methodology/Comments: The sensitivities of the circuit loading with respect to all control variables are computed with the formula:

$$\frac{dT}{du} = \frac{dT}{au} - x \frac{ag}{au}$$ (1)

Where T is the circuit loading, x is the costate of the system with respect to the circuit loading, and g(x,u) is the load flow equations.

Thus, this routine computes the costate first and subsequently substitutes it into Eq. (1).

Validation Tests: Same as in routine ACLINZ.

Subroutine Name: CSNSA

Brief Description: This routine updates the sensitivity vector SJSCR of the linearized circuit loading (MVA flow) constraint if the selected circuit either contains a control tap transformer or control phase shifter or connects to a bus with generator voltage control.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: Adjusted sensitivity vector SJSCR

Other Subroutines Called: None

Methodology/Comments: The sensitivity vector is defined with the equation:
$$\frac{df}{du} = a_f - x^T \frac{ag}{au}$$

This routine computes the first part of above equation where \( f \) stands for the circuit apparent flow (MVA).

Validation Tests: None

Subroutine Name: MOD4B

Brief Description: This routine linearizes a given bus voltage constraint. When the routine is called, the optimal number of the bus voltage must be defined and, in addition, the bus should be a P-Q bus. When this subroutine is called pointer of bus voltage constraints to be linearized (optimal numbering) \( \text{NPOPCV} \), must have been specified. A flow chart of this routine is illustrated in "Figure 3.3.3.4.2 Constraint Linearization Submodule Bus" on page 29.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: The sensitivity vector of selected bus, in array SJSCR.

Other Subroutines Called: SLVTR1, SLVTR2, CSNS

Methodology/Comments: The sensitivities of the voltage magnitude operating constraint at the selected bus with respect to the control variables are computed with the following equation:

$$\frac{dV_i}{du} = \frac{aV_i}{au} - x^T \frac{ag}{au} \quad (2)$$

Where \( V_i \) is the voltage magnitude, \( x \) is the costate of the system with respect to the voltage magnitude, and \( g(x,u) \) is the load flow equations.

Thus, this routine computes the costate first and subsequently substitutes it into Eq. (2).

Validation Tests: Same as in routine ACLINZ

-----------------------------------------------------------------------
Figure 3.3.31.4.1 Constraint Linearization Submodule Circuit
Overload Constraints

ENTER

\[ K = 0 \]

\[ K = K + 1 \]

\[ K > NTFOC \]

YES

Return

NO

NO

\[ ILFOC(K) = 1 \]

YES

Compute Excitation of Adjoint Network, \( \partial f/\partial \bar{u} \) Reference 7

Compute Adjoint Network Solution, \( \hat{x} \)

* Compute Contributions
  * \( \hat{x}^T (\partial g/\partial u) \)
    for all remedial actions
  * Compute Contributions
    \( \partial f/\partial u \)
    for all remedial actions

* Eliminate Small Sensitivities
  Using the User Selected threshold
  value XACTHLD
* Store Results
Figure 3.3.31.4.2 Constraint Linearization Submodule Bus Voltage Constraints

Enter

K = 0

K = K + 1

K > NT VOC

NO

IL VOC(K) = 1

YES

Compute Excitation of Adjoint Network, $\tilde{x}$, Reference 7

Compute Adjoint Network Solution, $\tilde{x}$

* Compute Contributions
  $-\tilde{x}^T(\partial \phi / \partial u)$
  for all remedial actions
* Compute Contributions
  $\partial \phi / \partial u$
  for all remedial actions

* Eliminate Small Sensitivities
  Using the User Selected threshold value XACTHLD
* Store Results

Return
Subroutine Name: MOD4C

Brief Description: This routine linearizes a given bus reactive constraint. When the routine is called, the optimal bus number of the reactive constraint NPOPCQ, must be specified. A flow chart of this routine is illustrated in "Figure 3.3.31.4.3 Constraint Linearization Submodule Reactive" on page 32.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: The sensitivity vector of selected bus in array SJSCR.

Other Subroutines Called: SLVTR1, SLVTR2, CSNS, CSNSC

Methodology/Comments: The sensitivities of the bus reactive power with respect to all control variables are computed with the following equation:

\[
\frac{dQ}{du} = \frac{\partial Q}{\partial u} = x^T \frac{\partial g}{\partial u} \tag{3}
\]

Where Q is the bus reactive power, x is the costate of the system with respect to the bus reactive power, and g(x,u) is the load flow equations.

Thus, this routine computes the costate first and subsequently substitutes it into Eq. (3).

Validation Tests: Same as in routine ACLINZ.

Subroutine Name: CSNSC

Brief Description: This routine updates the sensitivity vector SJSCR of the linearized bus reactive constraint if the selected bus connects to controlled tap transformer, or phase shifter, or switchable capacitor or reactor or if the selected bus is PV bus or connects to any PV bus.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: Adjusted sensitivity vector SJSCR

Other Subroutines Called: None

Methodology/Comments: The sensitivity vector is defined with the
equation:
\[
\frac{df}{du} = \frac{af}{au} - x^T \frac{ag}{au}
\]

This routine computes the first part of this equation where \( f \) stands for the bus reactive constraint.

Validation Tests: None

-----------------------------------
Figure 3.3.31.4.3 Constraint Linearization Submodule Reactive Power Constraints

1. Enter
2. $K = 0$
3. $K = K + 1$
4. $K > N_QOC$
   - Yes: Return
   - No: $ILQOC(K) = 1$
8. $ILQOC(K) = 1$
   - Yes: Compute Excitation of Adjoint Network $\xi x$, Reference 7
   - No: $ILQOC(K) = 1$ (repeat)
9. Compute Adjoint Network Solution, $\hat{x}$

- Compute Contributions
  - $\hat{x}(\partial x / \partial u)$
  - for all remedial actions
- Compute Contributions
  - $\partial f / \partial u$
  - for all remedial actions

- Eliminate Small Sensitivities
  Using the User Selected threshold value $XACTHLD$
- Store Results

32
Subroutine Name: MOD4D

Brief Description: This routine linearizes a given bus real power constraint. When the routine is called, the optimal bus number of the real power constraint NPOPCP, must be specified.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: The sensitivity vector of selected bus in array SJSCR.

Other Subroutines Called: SLVTR1, SLVTR2, CSNS, CSNSD

Methodology/Comments: The sensitivities of the bus real power with respect to all control variables are computed with the following equation:

\[
\frac{dP}{du} = \frac{\partial P}{\partial u} - x^T \frac{\partial g}{\partial u}
\] (4)

Where P is the bus real power, \( \dot{x} \) is the costate of the system with respect to the bus real power, and \( g(x,u) \) is the load flow equations.

Thus, this routine computes the costate first, and subsequently substitutes it into Eq. (4).

Validation Tests: Same as in routine ACLINZ.

Subroutine Name: CSNSD

Brief Description: This routine updates the sensitivity vector SJSCR of the linearized bus real power constraint if the selected bus connects to controlled tap transformers, of phase shifters, or a voltage control generator is connected to the selected bus or to any neighboring buses.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: Adjusted sensitivity vector SJSCR

Other Subroutines Called: None

Methodology/Comments: The sensitivity vector is defined with the equation
\[
\frac{df}{du} = \frac{af}{au} - x^T \frac{ag}{au}
\]

This routine computes the first part of this equation where \( f \) stands for the bus real power.

Subroutine Name: MOD4E

Brief Description: This routine linearizes the system transmission loss constraint. When the routine is called, total number of system buses, NTOTAL, dimension of matrix B\( ^* \), KTOTAL total number of circuits, NCIR, and total number of transformers, NXFM, must be known.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: Solution of costate vector and sensitivity vector of the losses with respect to various control variables, SJSCR.

Other Subroutines Called: SLVTR1, SLVTR2, CSNS, CSNSE

Methodology/Comments: The sensitivities of the system total losses with respect to all control variables are computed with the following equation:

\[
\frac{dP_{\text{loss}}}{du} = \frac{\partial P_{\text{loss}}}{\partial u} - x^T \frac{ag}{au}
\]  \hspace{1cm} (5)

Where \( P_{\text{loss}} \) is the system total losses, \( x \) is the costate of the system with respect to the system total losses, and \( g(x,u) \) is the load flow equations.

Thus, this routine computes the costate first, and subsequently substitutes it into Eq. (5).

Validation Tests: Same as in routine ACLINZ.

Subroutine Name: CSNSE

Brief Description: This routine updates the sensitivity vector SJSCR of the linearized system transmission losses whenever controlled tap transformers or phase shifters are present.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.
Output Arrays/Variables: Adjusted sensitivity vector SJSCR

Other Subroutines Called: None

Methodology/Comments: The sensitivity vector is defined with the equation

\[
\frac{df}{du} = \frac{af}{au} : \frac{T}{au} \frac{ag}{au}
\]

This routine computes the first part of this equation where \( f \) stands for the system transmission losses.

Validation Tests: None

Subroutine Name: MOD4F

Brief Description: This routine linearizes the power balance equation. When the routine is called, total number of PV buses, total number of buses with generating real power control, total number of tap and phase control transformers, total number of buses with switchable capacitors and reactors, total number of buses with interruptible load, firm load, and transferable load should be known.

Input Arrays/Variables: In addition to the information stated above, the pointer of the order of slack bus \( KCSLC \), must have been defined when the routine is called.

This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: The sensitivity vector of power balance with respect to variable control variables, SJSCR.

Other Subroutines Called: MOD4D, CSNSF

Methodology/Comments: The sensitivities of the power balance with respect to all control variables are computed by two steps:

1. Linearization of real power constraint for slack bus

2. Adjustment of the sensitivity vector according to what kind of loads are connected to the slack bus. The computation algorithm is just like the one listed in subroutine MOD4D.

Validation Tests: Same as in routine ACLINZ.
Subroutine Name: CSNSF

Brief Description: This routine adjusts the sensitivity array SJSCR for the linearized power balance equation when these are load controls associated with the slack bus.

Input Arrays: Remedial action data bank.

Output Arrays: Adjusted sensitivity vector SJSCR

Routines Called: None

Methodology/comments: The sensitivity vector is defined with the following equation:

\[
\frac{df}{du} = \frac{af}{au} - x^T \frac{ag}{au}
\]

This routine computes the first term of the above equation where f stand for the balance equation.

Subroutine Name: MOD4G

Brief Description: This routine linearizes the real power flow (MW flow constraint of a selected circuit. When the routine is called, the circuit internal number of the specific constraint, NPOPCC, must have been assigned a value.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: The sensitivity vector of the circuit MW flow with respect to control variables, SJSCR.

Other Subroutines Called: SLVTR1, SLVTR2, CSNS

Methodology/Comments: The sensitivities of the circuit real power flow with respect to all control variables are compute with the formula:

\[
\frac{dP_c}{du} = \frac{ap_c}{au} - x^T \frac{ag}{au}
\]  

(6)

Where \(P_c\) is the circuit real power flow, \(x\) is the costate of the system with respect to the circuit real power flow, and \(g(x,u)\) is the load flow equations.

Thus, this routine computes the costate first, and subsequently substitutes it into Eq. (6).
Thus, this routine computes the costate first, and subsequently substitutes it into Eq. (6).

Validation Tests: Same as in routine ACLINZ.

----------------------------------------

Subroutine Name: MOD4H

Brief Description: This routine linearizes the net real power interchange constraint. When the routine is called, both the total number of interchange constraints (areas) NTXOC, and pointer of interchange constraint NPOPCX, to be linearized must have been specified. A flow chart of this routine is illustrated in "Figure 3.3.31.4.4 Constraint Linearization Submodule" on page 39.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: The sensitivity vector of the net real power interchange with respect to control variables, SJSCR.

Other Subroutines Called: SLVTR1, SLVTR2, CSNS, CSNSH

Methodology/Comments: The sensitivities of the net real power interchange with respect to all control variables are computed with the formulas:

\[
\frac{dP_{ex}}{du} = \frac{\partial P_{ex}}{\partial u} - \dot{x}^T \frac{\partial g}{\partial u}
\]  

(7)

Where \(P_{ex}\) is the net real power interchange, \(\dot{x}\) is the costate of the system with respect to the net real power interchange, and \(g(x,u)\) is the load flow equations.

Thus, this routine computes the costate first, and subsequently substitutes it into Eq. (7).

Validation Tests: Same as in routine ACLINZ.

----------------------------------------

Subroutine Name: CSNSH

Brief Description: This routine updates the sensitivity vector SJSCR of the linearized net real power interchange constraint with respect to voltage variables.

Input Arrays/Variables: This routine uses a large number of arrays from the data bank.

Output Arrays/Variables: Adjusted sensitivity vector SJSCR
Other Subroutines Called: GETBSP

Methodology/Comments: The sensitivity vector is defined with the equation

\[ \frac{df}{du} = \frac{af}{au} - x^T \frac{ag}{au} \]

This routine computes the first part of this equation where \( f \) stands for the net real power interchange constraint.

Validation Tests: None
Figure 3.3.31.4.4 Constraint Linearization Submodule
Area Net MW Constraints

Enter

K = 0

K = K + 1

K > NTXOC

NO

ILXOC(K) = 1

YES

Compute Excitation of Adjoint Network, $\partial f / \partial x$. Reference 7

Compute Adjoint Network Solution, $\hat{x}$

* Compute Contributions
$- \hat{x}^T (\partial g / \partial u)$
for all remedial actions
* Compute Contributions
$\partial f / \partial u$
for all remedial actions

* Eliminate Small Sensitivities
Using the User Selected threshold value XACTHLD
* Store Results

Return
Subroutine Name: CSNS

Brief Description: This routine computes a part of the sensitivity vector SJSCR. Specifically, computes the vector:

\[-x^T \frac{ag}{au} \]

The sensitivity vector SJSCR is computed with respect to all defined control variables, i.e., power mismatch control for OPF, generating unit dispatch control for OPF, real power output w.r.t. slack bus, real power output w.r.t. sink bus, generator bus voltage, bus reactive power control, transformer phase, transformer tap, switchable capacitor banks, switchable reactors, load transfer, interruptible load shedding, firm load shedding, area net MW interchange.

Input Arrays/Variables: When this routine is called, ten control codes must be specified to determine which of the controls is going to be processed. These are NVPH, NWGN, NGRP, NGSR, NCVG, NQVG, NTTC, NTPS, NSCB, NSRB, NILS, NFLS, NRLT, NIXG.

This routine uses a large number of arrays from the data bank, and also the solution of the co-state vector YASCR and the network solution YBSCR.

Output Arrays/Variables: Sensitivity vector SJSCR

Other Subroutines Called: None

Methodology/Comments: The sensitivity vector is defined with:

\[ \frac{df}{du} = \frac{af}{au} - x^T \frac{ag}{au} \]

This routine computes the second part of above equation which is independent of the specific quantity to be linearized.

Validation Tests: None

Subroutine Name: JACFPN

Source/Modifications:

Brief Description: This routine forms the pointers for the off diagonal elements of the Table of Factors for Jacobian matrix.

Input Arrays/Variables: NTOTAL, IJDPOS, IJVPOS, ITKM1, ITKT1, MAXFL
Output Arrays/Variables: IJKT1, IJKM1

Other Subroutines Called: None

Methodology/Comments: None

Validation Tests: None

Subroutine Name: LUFCTO

Brief Description: This routine computes the table of factors.

Input Arrays/Variables: NT, ITKT, ITKT1, ITKM, ITKM1, YOFFD, YD

Output Arrays/Variables: HOFFD, HD

Other Subroutines Called: None.

Methodology/Comments: None.

Validation Tests: None.

Subroutine Name: ELSSLP

Brief Description: This subroutine sets small sensitivities to zero before setting up the LP. The smallness is judged by comparing the ratio SJSCR/XCT and the threshold XACTHLD, where the XCT is the largest among SJSCR's w.r.t. one kind of control.

Input Arrays/Variables: NVPH, NWGN, NGRP, NGSR, NGVG, NQVG, NTTC, NTPS, NSCB, NSRB, NRLT, NILS, NFLS, NIXG, SJSCR

Output Arrays/Variables: The sensitivity vector SJSCR

Other Subroutines Called: None.

Methodology/Comments: Finds the maximum sensitivities and eliminates the smallest ones with respect to power injection type variables, phase shifters, voltage controlled buses, tap transformers, switchable capacitors, and reactors.

Subroutine Name: FORMS1

Brief Description: This subroutine stores the linearized equation with respect to control variables in a sparcity coded storage scheme.
Input Arrays/Variables: ISNKT, ISNKM, SJSCR, NTCNTR, MRP, MROWS

Output Arrays/Variables: SNSAUG, ISNKM, ISNKT

Other Subroutines Called: None.
1.1.1.6 Power Flow Update Submodule - ACPFUP

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule performs a power flow update within the remedial actions. Specifically, it receives the LP solution and performs an update of the following:

1. Remedial Action Variables
2. Power flow Network Arrays
3. Power flow Solution Arrays

When the reactive power output at a generation bus reaches upper or lower limits, the bus is converted (PV/PQ) and the control variables at the bus are exchanged (V/Q).

The flow chart of this submodule is illustrated in Figure 1.1.1.6.1 page 48.

Control Variables:

- NPFVPD : Maximum allowable number of iterations
- PEPUPD : Maximum allowable real power mismatch
- QEPUPD : Maximum allowable reactive power mismatch
- VQFLAG : Logical variable. If .True. perform PV/PQ logic

Input Arrays and Variables:

- SOLLP : This array contains the LP solution for remedial actions. When this routine is called, solution of the linear program should have been computed (array SOLLP).

Output Arrays: The updated control variables, load flow variables, solution of power flow, and new value of slack bus generation.

Subroutines Called:

- UPACCV, UPACLF, PFUPD, CCFLAC, CTGENQ, CGENP, PBALOC

Diagnostics:

1. When the maximum number of iterations is exceeded, a diagnostic will be issued. The information included will be the maximum real and reactive power mismatch. Program control will be unaffected.

2. The maximum real and reactive power mismatch will be monitored. If an increase of mismatches is observed, a diagnostic will be generated. Information will be printed regarding the maximum mismatches.

Error Messages:

None

43
Validation Test:

The validation test is based on the observation that the power flow update solution should very closely match the projected solution from the LP model. For example, if the LP solution projects that a certain circuit loading constraint is active in the LP solution, then the power flow solution should yield a loading for this circuit equal approximately to its rating.

The validation test consist of the following algorithm. All constraints are scanned. For the active constraints in the LP solution, the difference between the power flow solution and the constraint limits are computed. This difference is reported in a "validation test report."

Notes: None.

Subroutine Name: UPACCV

Brief Description: This routine updates the present values and the total corrected values of the control variables for the AC model.

Input Arrays Variables: This routine uses a large number of arrays from the data bank.

Output Arrays Variables: VPHANT, W1GENP, W2GENP, PVGRP, TCRGRP, PVGSR, TCRGSR, PVCVG, TCRCVG PVQVG, TCRQVG, PVTT, TCRTT, PVTPS, TCRTPS, PVSCB, TCRSCB, PVSRB, TCRSRB, PVRLT, TCRRLT, PVILS, TCRILS, PVFLS, TCRFLS, XMNRRT, XMNRRLT, TCRRLT, XMNRRT, XMNIFLS, XPVILS, XMNIFS, XMXIFLS, XPFLS, XMNXCLS, XMXXCLS, XMPCLS, PVIXG, TCRIXG, XMNIXG, XMXIXG, STPIXG

Other Subroutines Called: None

Methodology/Comments:

Subroutine Name: UPACLF

Brief Description: This routine updates the AC load flow variables.

Input Arrays Variables: This routine uses a large number of arrays and variables from the data bank.

Output Arrays Variables: Arrays that are put with the results of update variables of power mismatches at buses, bus injections for OPF, real power generated, voltage control, transformer taps, phase shifter, bus parameters, and
Subroutine Name: PFUPD

Brief Description: This routine updates the power flow solution. A Newton—Rephson approach is used including PV/PQ switching and optionally updating the Jacobian matrix. The flowchart of this routine is given in Figure 3.3.31.6.

Input Arrays Variables: NPFUPD and a large number of arrays from the data bank.

Output Arrays Variables: Power flow solutions.

Other Subroutines Called: PMSMCH, QMSMCH, FORMJAC, JACGRD, VQLOGC, CTGENQ, SOLVT

Subroutine Name: CCFLAC

Brief Description: This routine computes the circuit flows from the computed voltages and phase angles.

Input Arrays Variables: NOPTA, NTOTAL, NCIR, NXFM, LSEPW, LREPM, GLPM, BLPW, SLPW, ITRPW, TTW, TPHW, NORDP, VOLT, DELT

Output Arrays Variables: PLFL1, PLFL2, QLFL1, QLFL2

Other Subroutines Called: None

Validation Tests: The circuit flows can be validated by the power flow program.

Subroutine Name: PBALOC

Brief Description: This routine allocates a given real power to the individual units connected to the bus IGB.

Arguments: IBG, generation bus index
SOL, the amount of power to be allowed.

Input Arrays: Data bank

Output Array: PGUAPW, OVERFL, the amount of generation requirement at bus IBG not met by the on-line generating units
UNDERFL, the amount of generation decreased at bus IBG not absorbed by the on-line generating units.
Subroutine Name: VQLOGC

Brief Description: This routine performs the PV/PQ switching logic. Specifically, for those generation buses that Q limits have been violated control is defined as reactive power generation. Otherwise, if bus has reactive power margin greater than some threshold (1.0/YEQBUS), then control is bus voltage.

Input Arrays: NGENB, IORDGN, LTYPLF, QGLFA, QMAXLF, QMINLF

Output Arrays: LTYPLF, IGSTAT, VQCODE, QINJLF, QGLFA, PVQVG, TCRQVG, PVCNG, TCRCVG, JACMOD

Routines used: None.

Note: If one or more PV/PQ switching have occurred, VQCODE=1

Subroutine Name: BRINGU
Figure 3.3.31.5 Power Flow Update Submodule - ACPFUP

ENTER

ARE THERE ANY REMEDIAL ACTIONS?

YES

UPDATE REMEDIAL ACTION ARRAYS

UPDATE POWER FLOW ARRAYS

UPDATE POWER FLOW STATE VARIABLES
PERFORM PV/PQ SWITCHING LOGIC

COMPUTE CIRCUIT FLOWS
COMPUTE GENERATION BUS REACTIVE POWER (ALL GENERATION BUSES)
COMPUTE SLACK BUS REAL POWER GENERATION

ALLOCATE REACTIVE POWER TO INDIVIDUAL UNITS

RETURN

ALLOCATE REAL POWER TO SLACK BUS UNITS

NO
Figure 3.3.31.6 Flow Chart of Subroutine PFUPD
ENTER

INITIALIZE ITERPF = 0

ITERPF = 0

ANY PV/PQ SWITCHINGS?

COMPUTE/TRIANGULATE NEW JACOBIAN

COMPUTE P AND Q MISMATCHES

CONVERGED?

PERFORM PV/PQ LOGIC

ANY PV/PQ CHANGES?

SOLVE AND UPDATE FOR STATE VARIABLES

PERFORM PV/PQ LOGIC

DIVERGING?

RETURN

ITERPF = ITERPR + 1

Figure 3.3.31.6 Flow Chart of Subroutine PFUPD
Submodule Name: RADBG1

Brief Description: This module reports the relevant remedial actions in the AC model.

Input Array: Data bank

Output Array: Reports of the remedial actions

Routines Used: RPAC4, RPAC5, RPAC6, RPAC7, RPAC8, RPAC9

Subroutine Name: RPAC4

Brief Description: This routine generates a report on the sensitivity analysis, i.e., the linearized circuit flows, voltage, real/reactive generation, losses, power balance equations, interchange constraint, and other variables.

Input Array: Data bank

Output: Report on above quantities

Subroutines Called: None

Subroutine Name: RPAC5

Brief Description: This routine reports all the violations for bus voltage, circuit flow, and reactive generation.

Input Array: Data bank

Output: Report on above quantities

Subroutines Called: None

Subroutine Name: RPAC7

Brief Description: This routine reports the arrays and variables created in submodule LPMSET.

Input Array: Data bank

Output: Report on above quantities

Subroutines Called: None
Subroutine Name: RPAC8

Brief Description: This routine reports linear programming solutions for the AC model.

Input Array: Data bank

Output: Report on above quantities

Subroutines Called: None

Subroutine Name: RPAC9

Brief Description: This routine summarizes the total computed remedial actions.

Input Array: Data bank

Output: Report on above quantities

Subroutines Called: None
**1.1.1.7 DC Remedial Action Module - DCRAM**

**MODULE DESCRIPTION AND STRUCTURE**

The DC remedial action computational module is executed whenever a system problem still exists after the DC network solution. Among the problems to be corrected are: circuit overloads, and area net export violations. Results of the remedial action calculation includes load curtailment at individual buses. The remedial action is computed with a linear program based algorithm.

The list of DC remedial actions is shown in "Figure 3.3.32 DC Remedial Action Submodule - DCRAM" on page 54. The user may select the types, locations, and priorities of control actions to be used in the study. A flow chart of the remedial action module is illustrated on page 54.

Up to eight priorities of remedial actions may be specified by the user. An option allows to have remedial actions purposely excluded from the priority list.

For remedial action purposes, interruptible and critical loads are specified in terms of megawatts and megavars of each load type at each bus. Firm is derived as total bus load less interruptible load. Load shedding is under constant power factor.

The remedial action computations are controlled by user specified control parameters. These control variables are defined in the design specifications of the submodules for remedial actions.

A flag (ACFLAG) is set to .F. corresponding to the DC model used in this module. The initialization of various constants is accomplished by calling subroutine STCNTD. The subroutine GETCOM is then executed and the value of the control variable ICMDN is used to determine which of 11 options would be executed:

1. printing of on-line help message (subr. HELP)
2. preparation of remedial action interface variables (subr. RAMINT)
3. computation of Jacobian matrix (subr. CMPJAC)
4. identification of the operating constraint violations (subr. DCIDNT)
5. coherency analysis of the constraints (DCCOHR)
6. linearization of constraints (subr. DCLINZ)
7. preparation of data for linear programming module (subr. LPMSET)
8. solution of the linear programming module (subr. LPSOLV)
9. update of the power flow solution (subr. DCPFUP)
10. report on the remedial actions (subr. RPD5 and RPD9)
11. return to the main program (subr. QLEVEL)
Some of the above options are accompanied by reports for debugging purposes whose activation is marked by printing appropriate messages.

The DC Remedial Action Module uses the following submodules:

DATA BASE FOR REMEDIAL ACTION SUBMODULE - RAMINT
This module generates the data base required in the remedial action module. It operates on the input data to define existing remedial actions and their limits. It also allows user specified remedial actions.

JACOBIAN FORMATION SUBMODULE - CMPJAC
This submodule forms and triangulates the DC Jacobian according to specified options and the system state.

CONSTRAINT IDENTIFICATION SUBMODULE - DCIDNT
This submodule computes circuit flows and net MW exports for a DC Network model, identifies which constraints are effective, packs operating constraints, eliminates the identical ones, and checks convergence of remedial actions.

CONSTRAINT COHERENCY ANALYSIS SUBMODULE - DCCOHR
This submodule identifies the constraints which may be active in the solution, i.e. those constraints which dictate the remedial process. Only these constraints will be included in the solution process. The constraint screening procedure is performed by sensitivity analysis of selected remedial actions in the neighborhood of the constraints. If the sensitivities of two or more constraints are within a user selected threshold, the constraints are placed within the same coherent group.

CONSTRAINT LINEARIZATION MODULE - DCLINZ
This submodule performs the linearization of the constraints to be included in the remedial action solution process. The computation is performed by means of the constant Jacobian matrix formed from a DC Network model of power system, which greatly increases the speed of computations. Sensitivities below a user specified threshold are eliminated.

LINEAR PROGRAMMING SETUP SUBMODULE - LPMSSET
This submodule sets up the linear program. It performs a model reduction algorithm and defines the bounds of validity of the linearized model.

LINEAR PROGRAMMING SOLUTION SUBMODULE - LPSOLV
This submodule consists of a mixed integer linear programming algorithm.
POWER FLOW UPDATE SUBMODULE - DCPFUP

This submodule performs an update of the power flow solution after each major iteration of the remedial actions algorithm.

STCNTD

Various control variables are initialized in module STCNTD: flags for study and control area (SABFLG, RABFLG, SACFLG, RACFLG), variables that relate to formation of Jacobian (KTJACS, KSLACK, JACTYP, JACMOD, JACFLG) report options (KRMACn, n = 1 to 6), and various thresholds and counters. Their description can be found in the variable list.
Table 1.1.1.7.1 List of DC Remedial Actions

<table>
<thead>
<tr>
<th>DC Capability Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MW Generation Adjustment</td>
</tr>
<tr>
<td>2. Reserve Unit Startup (future)</td>
</tr>
<tr>
<td>3. Phase Shifter Adjustment</td>
</tr>
<tr>
<td>4. Load Transfer</td>
</tr>
<tr>
<td>5. Interruptible Load Curtailment</td>
</tr>
<tr>
<td>6. Firm Load Curtailment</td>
</tr>
<tr>
<td>7. Critical Load Curtailment (future)</td>
</tr>
<tr>
<td>8. Area Interchange Adjustment</td>
</tr>
</tbody>
</table>
Figure 3.3.32 DC Remedial Action Submodule - DCRAM

START

PREPARE REMEDIAL ACTION DATABASE RAMINT & RAMDGN

CONSTRAINT IDENTIFICATION SUBMODULE DCIDNT Figure 3.3.32.1

CONVERGED?

YES

STORE/RETRIEVE SOLVED CONTINGENCY SUBMODULE SRSCON Figure 3.3.30.16

RETURN

NO

JACOBIAN FORMATION SUBMODULE CMPJAC Figure 3.3.33.2

CONSTRAINT COHERENCY SUBMODULE DCCOHR Figure 3.3.32.2

CONSTRAINT LINEARIZATION SUBMODULE DCLINZ Figure 3.3.32.3

LP SET-UP SUBMODULE LPMSET Figure 3.3.33.3

LP SUBMODULE LPSOLV Figure None

POWER FLOW UPDATE SUBMODULE DCPFUP Figure 3.3.32.4

COMPUTE SENSITIVITIES
1.1.1.8 Constraint Identification Submodule - DCIDNT

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule identifies branch overloads, and area net MW export violations. The violations are stored. They are also compared against tolerance values to determine whether convergence has occurred. "Figure 3.3.32.1 Constraint Identification Submodule" on page 56 illustrates the flow chart of the submodule.

Control Variables:

EFTOLW: Tolerance to determine whether a circuit flow constraint is effective

Input Arrays/Variables:

Arrays of a solved contingency case

Output Arrays/Variables:

NTFOC Total number of circuits flow constraints
NTXOC Total number of net MW export constraints
Arrays describing operating constraints (see "Dictionary of Program Variables")

Subroutines Called:

DCIDNT, CCFLDC, STOVDC, RMCHKB

Error Messages:

None

Validation Test:

Manual validation of violated constraints using the report of this submodule and a report of power flow solution. Consistency will be checked manually.
Figure 3.3.32.1 Constraint Identification Submodule

1. Enter
2. Compute circuit flows for DC model
3. Identify line flow operating constraints
4. Pack operating constraints, eliminate identical ones
5. Compute net MW export constraints
6. Check convergence of remedial actions
7. Return
A description of subroutines used in this submodule follows:

----------------------------------------

Subroutine Name: DCIDNT

Description: This subroutine computes the operating constraints for the DC model by identifying line flow operating constraints. Positive deviations denote violations, negative deviations denote constraints near rating. The constraints are packed by eliminating the identical for old operating constraints, old values are stored, new ones are computed and total correction updated. It also determines whether constraints are effective in Linear Programming Solution.

First the call is made to the subroutine CCFLDC to calculate the new ones, update STOVDC to store the old values of constraints, calculate the new ones, update the total correction, and determine whether constraints are effective in the linear programming module.

Next, the sending end line flows (PLFL1) are checked against thresholds based on the circuit rating (TRW) and allowed tolerance (EFTOLW). If violation is detected, the pointer of the circuit (IPFOC) is set. Flow violation is stored in (DPFOC) and the pointer of the direction of the flow (ICFOC) is set. The total number of violated constraints is set in (NTFOC). A number of other pointers is initialized for further processing. In the subsequent loop, previously identified constraints are eliminated, and, finally, a pointer of interchange constraints to be relinearized (ILXOC) is set if the total number of interchange constraints (areas) is greater, or equal to 2. The convergence of remedial actions is then checked by calling the routine RMCHKB.

Control Variables: EFTOLW: Tolerance to determine whether a flow constraint is effective.

Input Variables: When this routine is called, the following variables should have been specified.

NTFOC, NCIR, EFTOLW, PBASE, NDFOC, ICHOUT, NIXG, NTXOC

Input Arrays: LCEQW, TRW, PLFL1

Output Variables: NPCOLD

Output Arrays: IPFOC, DPFOC, ICFOC, DTFOC, DITFOC, IPSFOC, DCFOC, IEFOC, ILFOC, ILXOC
Subroutines Called: CCFLDC, STOVDC, RMCHKB

Error Messages: None.

Subroutine Name: CCFLDC

Description: This routine computes the circuit flows using the DC network model.

In the first loop, circuit flows are calculated from bus phase angles (DELTW) and line susceptances (BLPW). Then, the same procedure is applied to calculate flows through transformers, taking into account their tap setting (TTW) and phase shift (TTPHW). The real flows are then calculated by multiplication with a base (PBASE) Reactive flows are declared zero.

Input Arrays/Variables: DELTW

Output Arrays/Variables: PLFL1, PLFL2

Subroutines called: None.

Error Messages: None.

Subroutine Name: STOVDC

Description: For all old operating constraints, this routine performs the following:

1. Stores the old values
2. Computes the new value
3. Updates the total correction
4. Determines whether constraint is effective in linear programming module

Total correction of the flow constraints (DPFOC) is calculated from the old (DCFOC) and new value of the violation (DPFOC). A check is made whether the constraint was effective in the solution of the linear programming (SOLLP) and the corresponding flag (IEFOC) set. Next, total power flow through interchange tie lines (pointed out by IPXOC) is calculated and total correction (and total incremental correction) is calculated (DTXOC, DITXOC) from the old values (DCXOC) and actual violations (DPXOC) are calculated from the scheduled values (PSXOC) and actual flows.

Input Variables: When this routine is called, the following variable sho
have been defined:
NTFOC: current number of circuit flow operating constra
If it is not equal to zero it computes the flow constra

NTFOC, PBASE, KTLPV, NTXOC

Input Arrays: DPFOC, IPFOC, PLFL1, PLFL2, TRW, IPSFOC,
SOLLP, NPXOC, JPXOC, IPXOC, IFXOC, PSXOC

Output Variables: None.

Output Arrays: DCFOC, DPFOC, DTFOC, DITFOC, IEFOC, DPXOC, DCXOC,
DTXOC, DITXOC

Subroutines Called: None

Error Messages: None

Subroutine Name: RMCHKB

Description: This routine checks for convergence for the remedial
actions module.

A total sum (XSUM) of flow violations (DPFOC) is calculate
and checked against a threshold SMOVFV (conv. criterion).
If satisfied, a message is printed to confirm convergence.
A similar check is then performed for export violations.

Control Variables: When this routine is called, the following convergen
criteria should have been specified : SMOVFV, SMOXP

Input Variables: NTFOC, NTXOC

Input Arrays: DPFOC, DPXOC

Output Variables: None.

Output Arrays: None

Subroutines Called: None

Error Messages: None
1.1.1.9 Constraint Coherency Analysis Submodule - DCCOHR

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule identifies the constraints which may be active in the solution, i.e. those constraints which dictate remedial actions for the initiation of the constraint violations. This is done by identifying groups of coherent constraints and selecting one constraint per group. If the sensitivities of two or more constraints are within the user selected threshold, the constraints are placed within the same coherent group. "Figure 3.3.32.2 Constraint Coherency Submodule - DCCOHR" on page 61 illustrates the flow chart of this submodule.

Control Variables:

XDCCOH: Threshold value for constraint coherency analysis (variable alpha in the flow chart).

Input Arrays/Variables:

All arrays describing the constraints (overloads).

Output Arrays/Variables:

Results for coherency analysis of the old constraints, circuits flow constraints and the logic for linearization of constraints are stored in the following arrays: KGFOLD, ICHFOC, IRBUS, ILFOC, DITFOC

Subroutines Used:

DCCOHR, LOGDC1, LOGDC2, WRSTDC, ZOOM3, SOLVT, CHGRDC, RPDC5, RPDC6

Error Messages:

None.
Enter

Identify the Set of Constraints to participate in the process by excluding those which have been linearized. Call this set A.

Is Set A empty?

YES → Return

NO → Select the highest violation constraint. Apply remedial actions in the vicinity of the highest violation constraint. Project solution with a linearized model.

- Identify all constraints with a projected deviation DV greater than DVmax where DVmax is the maximum projected deviation.
- Exclude above constraints from set A.

Flag constraint with DVmax as the constraint to be included in the LP model.
A description of subroutines used in this submodule follows:

Subroutine name: **DCCOHR**

**DESCRIPTION:** This subroutine is the main routine of the submodule. It computes the coherent groups of the operating constraints for the DC model.

Coherency analysis of old constraints is performed by calling subroutine **LOGDC1**. A message is printed if the old and new number of constraints are equal. The subsequent processing is done only if there are new constraint violations. The pointer of the worst flow violation (KW) is determined by calling subroutine **WRSTDC**. The tier of buses around the buses connecting the circuit with the worst violation of flow is then calculated in **ZOOM3**. It is reported in array **IRBUS**. The driving vector dimension is then approximately increased and network solved using subroutine **SOLVT**. The sensitivity vector **XASCR** is then computed using the results of the previous calculation and the base case quantities and sensitivity coefficients normalized for all flow violations (DPFOC) greater than 0.001 pu by dividing their value by **DPFOC**. Coherent group is then computed by calling the subroutine **CHGRDC**. After repeating this procedure for all coherent groups subroutine **LOGDC2** is called to determine constraints to be relinearized.

**INPUT VARIABLES/ARRAYS:** When this routine is called, the following should have been known: old number of flow operating constraints **NPCOLD**, new number of flow operating constraints and the old number of flow operating constraints included in Linear Programming Mode, **KGFOLD**. It uses many arrays and variables from the data bank.

**OUTPUT VARIABLES:** **IFCIR,IZBUS**

**OUTPUT ARRAYS:** **ICHFOC,XASCR**

**SUBROUTINES CALLED:** **LOGDC1,WRSTDC,ZOOM3,SOLVT,CHGRDC,LOGDC2,RPDC5,RPDC5**

**ERROR MESSAGES:** None.

Subroutine name: **LOGDC1**

**DESCRIPTION:** This routine defines the logic for the coherency of the old constraints for the DC model.

If the old number of cat. flow constraints (**NPCOLD**) is
non-zero, the circuit flow violations (DPFOC) are checked against the tolerance (SMOFV) and the logic for linearization of constraints (ICHFOC, ILFOC, DITFOC) is activated if violations larger than threshold are detected.

INPUT VARIABLES: The old number of flow operating constraints, NPCOLD should have been known when this routine is called. NPCOLD,SMOFV

INPUT ARRAYS: ICHFOC,IPSFOC,DPFOC

OUTPUT VARIABLES: KGFOLD

OUTPUT ARRAYS: ILFOC,DITFOC

SUBROUTINES CALLED: None.

ERROR MESSAGES: None.

Subroutine name: LOGDC2

DESCRIPTION: This routine defines the logic for the linearization of old/new constraints for the DC model.

If the new number of flow violations (NTFOC), the pointer of constraint to be linearized (ILFOC) is set for all new constraints (those in positions behind the first KGFOLD constraints), then ????????????????????????????????????????????

INPUT VARIABLES: Current number of circuit flow operating constraints, NTFOC Should have been specified, as well as KGFOLD.

INPUT ARRAYS: ICHFOC

OUTPUT VARIABLES: None.

OUTPUT ARRAYS: ILFOC,DITFOC

SUBROUTINES CALLED: None.

ERROR MESSAGES: None.

Subroutine name: WRSTDC

DESCRIPTION: This routine identifies the worst operating constraint defined as the one with the maximum violation.

For all flow violations (DPFOC) which were not processed
previously (identified by a pointer of flow violation ICHFOC being zero), the search is made and the pointer K is set to point to the worst one.

INPUT VARIABLES: NT
INPUT ARRAYS: DOC, ICHOC
OUTPUT VARIABLES: KEY
OUTPUT ARRAYS: NONE.
SUBROUTINES CALLED: None.
ERROR MESSAGES: None.

Subroutine name: CHGRDC
DESCRIPTION: This subroutine calculates coherent groups of operating constraints
For all violations (NTFOC of them) the ratio of sensitivity of the worst violation (SNS(I) / SNS(KW)) is then checked against the tolerance (XCUT, or XDCCOH in the calling program). If the ratio is greater than the tolerance, ICHFOC for those constraints is set to the negative value of the group number (KG), indicating that those constraints are not active in LP because of their coherency with the worst constraint.

INPUT VARIABLES: KW, KG, NT (KW is the pointer which points the worst violation and KG is the coherent group numbers)
INPUT ARRAYS: SNS
OUTPUT VARIABLE: SNSMX, XCUT
OUTPUT ARRAYS: ICHOC
SUBROUTINES CALLED: None.
ERROR MESSAGES: None.

Subroutine name: ZOOM3
DESCRIPTION: This subroutine computes the buses around the circuit IFCIR given the zoom specifications. It returns an array IRBUS which is defined as follows:
IRBUS(I) = 1, Include bus in the set.
IRBUS(I) = 0, Do not include bus in the set.

I is bus number in internal numbering. All data are defined in internal numbering. Working arrays are utilized. Data are for individual circuits.

This subroutine computes IZBUS tiers (IZBUS = 1 in the calling program) of buses around the buses LSEBUS (IFCIF) and LREBUS (IFCIR) connected to the circuit with the worst flow violation. The output is the integer array IRBUS which, when set to 1, indicates that the corresponding bus is a neighbor to the above mentioned buses.

INPUT VARIABLES: IFCIR, NTOTAL, IZBUS, ROW

INPUT ARRAYS: LSEBUS, LREBUS

OUTPUT VARIABLES: None.

OUTPUT ARRAYS: IRBUS, IASCR

SUBROUTINES CALLED: None.

ERROR MESSAGES: None.

Subroutine RPDC5

DESCRIPTION: This subroutine creates the summary report on operating constraints. It is used for debugging purposes.

INPUT VARIABLES: GTITLE, ICHREP, NTFOC, IYES, INO, PBASE, NTXOC

INPUT ARRAYS: IPFOC, LSEPWL, LREPWL, LBSMW, PLFL1, PLFL2, TRW, ILFOC, IPSFOC, IEFOC, DCFOC, DTFOC, DPFOC, ILXOC, IPSXOC, PSXOC, DPXOC, DCXOC, DTXOC, DITXOC

OUTPUT VARIABLES: IC, IS, IR, ISS, ISR, SLN1, SLN2, XF, XM, IDUM, IDUM1, IDUM2, XOLD, XCOR, DP1, I, AMSR1, PS1, DP1, DC1, DT1, DIT1

OUTPUT ARRAYS: None

SUBROUTINES CALLED: MESSAG

ERROR MESSAGES: None.

Subroutine RPDC6
DESCRIPTION: This subroutine creates the summary report of the model reduction module. It is used for debugging purposes.

INPUT VARIABLES: GTITLE, ICHREP, NTFOC, PBASE

INPUT ARRAYS: IPFOC, LSEPW, LREPW, LBSMW, PLFL1, TRW, ICHFOC, DPFOC

OUTPUT VARIABLES: IYES, INO, IC, IS, IR, ISS, ISR, XF, XM, IC1, IDUM, DP1

OUTPUT ARRAYS: None

SUBROUTINES CALLED: None

ERROR MESSAGES: None
1.1.1.10 Constraint Linearization Submodule - DCLINZ

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule performs the linearization of the constraints to be included in the formation of the LP model. Computation is based on the resistive companion adjoint network. The computed sensitivity is screened as follows: sensitivities below a user specified threshold are eliminated. "Figure 3.3.32.3 Constraint Linearization Submodule - DCLI" on page 70 illustrates the flow chart of the submodule.

Control Variables:

NSOPT - (Indicates type of constraint: 1 - circuit loading (MVA flow) 2 - power balance equation. 3 - net Mw export constraint)

KTDCS - Control variable indicating the availability of Jacobian matrix when computing sensitivities
1 - Matrix B' must be triangulated
2 - Matrix B' is triangulated.

Input Arrays/Variables:

Data Base

Output Arrays/Variables:

Enumerated in subroutine descriptions.

Subroutines called:

DCCLNZ, MDDC4A, MDDC4D, MDDC4H, CSCSDC, LUPNTD, LUFCTD, SLVTR1, SLVTR2, RPDC4.

Error Messages:

None.

Validation Tests:

Two validation tests are performed:

(a) The computed linearization coefficients are also computed by brute force using a usual power flow. The two values are manually compared. They must be approximately equal.

(b) By comparing the projected solution from the linear program submodule to an actual solution with a power flow after application of the LP computed remedial actions. The two must be approximately equal.

Methodology/Comments: Linearization of the operating constraint f(x,u)
around an operating point defined with control variables \( u = u^0 \) and state variables \( x = x^0 \), yields

\[
f(x, u) = f(x^0, u^0) + \sum_i \frac{df(x, u)}{du_i} (u_i - u^0)
\]

The sensitivities \( \frac{df(x, u)}{du_i} \) are computed based on the costate method,

\[
\frac{df(x, u)}{du_i} = \frac{af(x, u)}{au_i} - [\frac{af(x, u)}{ax}] [\frac{ag(x, u)}{ax}]^{-1} [\frac{ag(x, u)}{au_i}]
\]

where \( \frac{ag(x, u)}{ax} \) is the Jacobian matrix of the power flow equation.

Description of subroutines follows.

Subroutine name: DCCLNZ

DESCRIPTION: This module computes the linearized operating constraints using the resistive companion adjoint network. If triangulates the DC model Jacobian Matrix, selects operating constraint (circuit loading, generator real output, or net MW export constraint).

The total number of control variables (NTCNTR) is calculated as a sum of the total number of buses with active generation (NGRP), transformers with phase control (NTPS), buses with interruptible load (NILS), buses with firm load (NRLT), and the total number of interconnected systems minus one (NIXG - 1). If the number of control variables is zero, then an appropriate message is printed. Following is the initialization of various variables relating to sensitivity matrix SNSAUG: total number of columns expressed per row of SNSAUG (ISNKT), total number of rows (MROWS), and flags for various types of constraints which indicate whether they should be relinearized (NPOPCC, NPOPCP, NPOPCL, NPOPCB). If the total number of equality constraints is zero, the real power balance equation is linearized by calling subroutine FORMS1 after calculating sensitivities in DCCLNZ and neglecting the small ones in ELSSLP. Also set are the appropriate elements of vector of weights of constraints (WGLCP), b value for equality constraint (BLCONT, equal to zero in the case of real power balance equation), a code for type of violation (ICOP), pointer of position of equality constraint in the set of constraints (ILCONT), and the total number of equality constraints (NLCONT, set to one). If the total number of interchange constraints is non-zero, a loop is set to check for those that need to be relinearized (pointed by ILXOC). After setting pointer of constraint to be relinearized (pointed by ILXOC). After setting
pointer of constraint to be relinearized (NPOPCX) checking whether it is a new constraint and increase the number of rows (MROWS) in sensitivity matrix, if necessary, setting the flag for interchange constraints in DCCLNZ (variable NSOPT = 3) and weight for constraints (WGLCP), linearization is performed by calling DCCLNZ, small sensitivities set to zero by ELSSLP and new equation set into SNASUG by calling FORMS1. The counter of equality constraints is incremented (NLCONT), if necessary, and b value of the constraint set. A similar loop is set for circuit constraints which need to be relinearized (pointed by ILFOC) and appropriate constraints relinearized by calling the same subroutines (DCCLNZ, ELSSLP, and FORMS1), if circuit ratings are not zero (such condition is declared by checking ratings and setting a message, if necessary).

The flags are set not to print Jacobian and debug reports and to print sensitivity reports (KRSMN1, KRSMN2, KRSMN3). Value of flag (KTDCS) is checked in order to determine whether the triangulation of B' matrix is needed and triangulation is performed, if necessary, by calling subroutines LUPNTO and LUFCTO. Value of the flag NSOPT is checked to determine the type of constraint to be linearized, and call responding subroutines called (MDDC4A for loading constraints, MDDC4D for generator real output constraints, and MDDC4H for net MW export constraint). Sensitivity reports are printed after execution of those subroutines.

INPUT VARIABLES: KTDCS,ICHOUT,MD3,NTOTAL,NSOPT,NPOPCC,NPOPCP,NPOPCX

INPUT ARRAYS: Large number

OUTPUT VARIABLES: ADIAG,NTOTAL,KRSMN1,KRSMN2,KRSMN3,MAXFL,NT

OUTPUT ARRAYS: AMESG

SUBROUTINES CALLED: LUPNTO,LUFCTO,MDDC4A,MDDC4D,MDDC4H,RPDC4

ERROR MESSAGES: Generated when values of the variables KTDCS, NSOPT, NPOPCC, or NPOPCP are invalid.
Figure 3.3.32.3 Constraint Linearization Submodule - DCLinz

ENTER

NSOPT

= 1
LINEARIZE CIRCUIT LOADING CONSTRAINT

= 2
LINEARIZE THE POWER BALANCE EQUATION

= 3
LINEARIZE NET MW EXPORT CONSTRAINT

RETURN
Figure 3.3.32.3.1 Constraint Linearization Submodule

Enter

Call Jacobian Formation Submodule

K = 0

K = K + 1

K > NTFOC

YES

Return

NO

IFOC(K) = 1

YES

NO

Compute Excitation of Adjoint Network, 3 \( \frac{\partial g}{\partial x} \) Reference 7

Compute Adjoint Network Solution, 2

• Compute Contributions
  \( z^T \frac{\partial g}{\partial u} \)
  for all remedial actions

• Compute Contributions
  \( g \frac{\partial u}{\partial u} \)
  for all remedial actions

• Eliminate Small Sensitivities
  Using the User Selected threshold value XACTHLD
• Store Results
Subroutine name: MDDC4A

DESCRIPTION: This subroutine sets and solves the adjoint network for linearizing a flow constraint. When this subroutine is called, the pointer of net constraint to linearized, NPOPCC, must have been specified. The flow chart of this subroutine is illustrated in "Figure 3.32.3.1 Constraint Linearization Submodule" on page 71. The sensitivities of net loading vs. all control variables are computed with formula

\[
\frac{dT}{du} = \frac{\partial T}{\partial u} - \mathbf{x}^T \frac{\partial g}{\partial u}
\]

Where \( T \) is the circuit loading, \( x \) is costate of the system with respect to circuit loading and \( g(x,u) \) are power flow equations (DC model).

This subroutine sets and solves the adjoint network for linearizing a flow constraint. After forming driving vector XBSCR by initializing parameters that relate to the constraint, adjoint network is solved by calling subroutines SLVTR1 and SLVTR2 and sensitivity vector (SJSCR) is computed by calling CSNSDC.

INPUT VARIABLES: NPOPC

INPUT ARRAYS: LSEPW, LREPW, NORDRP, BLPW, IPTRW, TTW

OUTPUT VARIABLES: NTOTAL, KTOTAL

OUTPUT ARRAYS: XASCR

SUBROUTINES CALLED: SLVTR1, SLVTR2, CSNSDC

ERROR MESSAGES: None

Subroutine name: MDDC4D

DESCRIPTION: This subroutine performs the linearization of the real power balance equation. When this subroutine is called, the optimal bus number of the real power constraint, NPOPCP, must be specified. The sensitivities of the bus real power vs. all control variables are computed with the following equation

\[
\frac{dP}{du} = \frac{\partial P}{\partial u} - \mathbf{x}^T \frac{\partial g}{\partial u}
\]

Where \( P \) is the bus real power, \( x \) is costate of the system with respect to bus real power and \( g(x,u) \) are real power balance equations (DC model).
This routine sets and solves the adjoint network for linearizing a real power injection constraint. After forming a driving vector XBSCR by initializing parameters that relate to the constraint, adjoint network is solved by calling subroutines SLVTR1 and SLVTR2 and sensitivity vector (SJSCR) is computed by calling subroutine CSNSDC.

INPUT VARIABLES: NPOPCP

INPUT ARRAYS: PARAM,ITKT,ITKM,NLMT,BCLFA

OUTPUT VARIABLES: NTOTAL

OUTPUT ARRAYS: XASCR

SUBROUTINES CALLED: SLVTR1,SLVTR2,CSNSDC

ERROR MESSAGES: None.

Subroutine name: MDDC4H

DESCRIPTION: This routine performs the linearization of the interchange constraints. The flow chart of the routine is illustrated in "Figure 3.3.32.3.2 Constraint Linearization Submodule," on page 75. When this routine is called, both the total number of interchange constraints NTXOC and pointer of interchange constraints NPOPCX to be linearized, must be specified. The sensitivities of the net real power interchange vs. all control vari are computed with the formula

\[
\frac{dP_{ex}}{du} = \frac{\partial P_{ex}}{\partial u} - x^T \frac{\partial g}{\partial u}
\]

Where \(P_{ex}\) is the net real power interchange, \(x\) is costate of the system vs. net real power interchange and \(g(x,u)\) are real power balance equations (DC model).

This routine sets and solves the adjoint network for an interchange constraint. A driving vector XASCR is initialized after initializing parameters that define the lines for linearization. Adjoint network is then solved by calling SLVTR1 and SLVTR2 and sensitivity vector (SJSCR) is formed by calling SSNSDC.

INPUT VARIABLES: NPOPCX,NTXOC,ICHOUT,IFTIEL

INPUT ARRAYS: PARAM,JPXOC,IPXOC,LSEP,W,LREP,W,NORDRP,BLPW,IPTRW,TTW,LCOW,IFXOC

OUTPUT VARIABLES: ADIAG,NTOTAL,NW,ISTIEL,ITIE,ICP,IC,I1,I2,K1,K2,BC,TT,IT,XBC,K,X1

OUTPUT ARRAYS: AMESG,XASCR
SUBROUTINES CALLED: SLVTR1, SLVTR2, CSNSDC

ERROR MESSAGES: Diagnostics generated when the value of pointer for MW export linearization (NPOPCX) or pointer for tie line MW flow linearization (LCDW(IC)) are invalid.
Figure 3.3.32.3.2 Constraint Linearization Submodule, Area MW Constraints, DC Remedial Actions, Flow Chart

1. Enter

2. Call Jacobian Formation Submodule

3. $K = 0$

4. $K = K + 1$

5. $K > NTXOC$?
   - Yes: Return
   - No: $ILFOC(K) = 1$?
     - Yes: Compute Evolution of Adjoint Network, $\frac{\partial^2 f}{\partial x^2}$, Reference 7
     - No: Compute Adjoint Network Solution, $\xi$

6. *Compute Contributions $\xi T(\partial g/\partial u)$ for all remedial actions

7. *Compute Contributions $\partial f/\partial u$ for all remedial actions

8. *Eliminate Small Sensitivities Using the User Selected threshold value XACTHLD

9. *Store Results

Note: The flowchart includes logic for iterating and making decisions based on conditions such as $K$, $NTXOC$, and $ILFOC(K)$.
Subroutine name: CSNSDC

DESCRIPTION: This routine computes the sensitivity vector SJSCR. With respect to all defined control variables: generator real power output, transformer phase shift, interruptible load shedding, load transfer out area net generation interchange.

The sensitivity vector SJSCR is formed by concatenating parts of the vector XASCR which correspond to: generator real output, phase shifting transformers, interruptible load, firm load, real load transfer, and area net generation adjustments. The formation of SJSCR is accomplished in subsequent DO loops labeled as above.

INPUT VARIABLES: NGRP, NTPS, NILS, NFLS, NRLT, NIXG
(six control codes for classes of control)

INPUT ARRAYS: PARAM, IPGRP, XASCR, IPTPS, ITRPW, LSEPW, LREPW, NORDRP, T TW, BLPW, IPILS, IPFLS, IP1RLT, IP2RLT, JPIXG, JBPIXG, PRTIXG

OUTPUT VARIABLES: NTOTAL, NTCNTR

OUTPUT ARRAYS: SJSCR

SUBROUTINES CALLED: None

ERROR MESSAGES: None

Subroutine name: RPDC4

DESCRIPTION: This subroutine creates the summary sensitivity report of the DC model. It is used for debugging purposes.

INPUT VARIABLES: ICHREP, NSOPT, NPOPC, PBASE, NGRP, NTPS, NILS, NFLS, NRLT, NIXG

INPUT ARRAYS: LSEPW, LREPW, LBSMW, PLFL1, NORDER, IGORDW, PGTLW, IPGRP, XMXGRP, XMNGRP, PVGRP, IPTPS, ITRPW, XMXTPS, XNTPS, PVTPS, SJSCR, IPILS, XMXILS, XMNILS, PVILS, IPFLS, XMFLS, XMNFLS, PVFLS, IP1RLT, IP2RLT, XMRLT, XMNRLT, PVRLT, NPIXG, PJXG

OUTPUT VARIABLES: I1, I2, IS1, S2, XF, L, L1, AMSR1, K, ID, II, IS, XM, XMX, XMN, XPV, IT, IC, NUNITS, PV, S1A

OUTPUT ARRAYS: None

SUBROUTINES CALLED: None
ERROR MESSAGES: None
1.1.1.11 Power Flow Update Submodule - DCPFUP

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule performs a power flow update using the DC Network model. Specifically, it receives the LP solution and performs an update of the following:

1. Remedial Action Variables 2. Load flow Network Arrays 3. Load flow Solution Arrays

The flow chart of this routine is illustrated in "Figure 3.3.32.4 Power Flow Update Submodule - DCPFUP" on page 79.

Control Variables:

Input Variables/Arrays:

SOLLP - Solution of 1p Model.

Output Arrays:

Shown in subroutine description. (updated control variables, load flow variables, solution of power flow new value of slack bus generation by the subroutines called).

Existing subroutines:

DCPFUP, UPDCCV, UPDCLF, DCPMS, SOLVT, GLFREP

Diagnostics:

Error Messages:

None.

Validation Test:

Load flow update should closely match the projected solution from the LP model. Algorithms: all constraints in the LP solution will be scanned. For all the active constraints, the difference between the load flow solution and the constraint limit will be computed. This difference will be reported in "validation test report".
Figure 3.3.32.4 Power Flow Update Submodule - DCPFUP

1. Enter
2. Update Remedial Actions Arrays
3. Update Power Flow Network Arrays
4. Solve DC Power Flow
5. Return
A description of subroutines called by the DC power flow update submodule follows:

Subroutine name: DCPFUP

DESCRIPTION: This subroutine is the main routine of the DC power flow update submodule. It updates all the control and power flow variables using the linear programming solution, and calculates the new powerflow solution. It creates the report of computed DC remedial actions.

A sum of absolute values of the LP solution vector (SOLLP) is checked against a small threshold to check whether any remedial actions were computed, and an appropriate message is printed. The control variables are then updated by calling UPDCCV, DC load flow variables are updated by calling UPDCLF and array of mismatched formed in DCPMS. DC load flow equation is solved in SOLVT and report generated by calling GLFREP.

INPUT VARIABLES: KTLPV, NDEBG, ICHOUT, NTOTAL, ICHREP

INPUT ARRAYS: SOLLP

OUTPUT VARIABLES: XT, KODE, ICHW

OUTPUT ARRAYS: IRBUS

SUBROUTINES CALLED: RPDC8, UPDCCV, UPDCLF, DCPMS, SOLVT, GLFREP

ERROR MESSAGES: None

Subroutine name: UPDCCV

DESCRIPTION: This routine updates the present values of the control variables for the DC model.

If the real power generation variables counter (NGRP) is nonzero, the solution of the LP (SOLLP) is accessed using the pointer of control variables included in LP (IPLPV1), present bus generating real power (PVGRP), and total change of real power generation at the bus (TCRGRP) adjusted. Th other control actions which are the result of LP are accounted for in a similar fashion.

INPUT VARIABLES: NGRP, NTPS, NILS, NFLS, NRLT

INPUT ARRAYS: IPLPV1, SOLLP, PVGRP, TCRGRP, PVTPS, PCRTPS, PVILS, TCRILS, PVFLS, TCRFLS, PVRLT, TCRRLT

OUTPUT VARIABLES: None.
Subroutine name: UPDCLF

DESCRIPTION: This routine updates the DC load flow variables.

The net real power bus injection (PINJLF) and total bus real generation (PGTLW) are adjusted by using the LP solution vector (SOLLP) and pointer array of buses with controlled real injection (IPGRP). In a similar fashion, adjustments are made to account for the effect of phase shifters, interruptible load, firm load shedding, and load transfer.

INPUT VARIABLES: NGRP, NTPS, NILS, NFLS, NRLT, NDEBG, ICHOUT

INPUT ARRAYS: IGPLPV1, SOLLP, IPGRP, NORDER, IPTPS, IPTPSW, XPHASH, IPHASH, ITRPW, ICRON, TTW, IPILS, IPFLS, IPRLT, NDEBG, NRLT, NFLS, NILS, NTPS, NGRP

OUTPUT VARIABLES: None.

OUTPUT ARRAYS: PINDLF, PGTLW, TTPHW, XPHASH, GSLFA, BSLFA, PLDW.

SUBROUTINES CALLED: GETBSP

ERROR MESSAGES: None
Subroutine name: DCPMS

DESCRIPTION: This subroutine computes mismatches for the DC power flow solution.

This routine computes mismatches by equating the driving vector SASCR with the vector of net real power bus injections (PINJLF) and adjusts for injections from phase shifters and constant impedance loads.

INPUT ARRAYS / VARIABLES: XASCR, PINJLF, NTOTAL, NTPSHS, IPHASH, ICILW, NORDRP, GCILW

OUTPUT ARRAYS / VARIABLES: XASCR

SUBROUTINES CALLED: None

ERROR MESSAGES: None

---

Subroutine name: SOLVT

DESCRIPTION: This routine performs the forward and back substitution on a given vector \( *B* \) with a given set of factors.

INPUT VARIABLES: None.

INPUT ARRAYS: B, HO, IKT1, ITKM1, HOFFD, HD

OUTPUT VARIABLES: None.

OUTPUT ARRAYS: X

SUBROUTINES CALLED: None

ERROR MESSAGES: None

---
1.1.1.12 Data Preparation Submodule - RAINIT

SUBMODULE DESCRIPTION AND STRUCTURE

This routine builds up the database only for remedial action purposes. It gets the minimum amount of data from the TRUCE program and consists of three parts. The first part initializes all the remedial action variables. The second part modifies the variables for a specified contingency. The third part restores the variables for a specified contingency. The task to be performed by this module is determined by a control variable. "Figure 3.3.33.1 Data Preparation Submodule AC or DC" on page 85 illustrates the flow chart of this submodule. "Table 3 Remedial Action Data Base Preparation Submodule" on page 86 lists the defined/modifies arrays for the various options of the submodule.

The following control variables are defined:

1. MW generation adjustments - slack bus
2. MW generation adjustments - sink bus
3. Generation voltage adjustment
4. MVAR generation adjustment
5. Phase shifter adjustments
6. Transformer tap adjustments
7. Shunt reactor switching
8. Shunt capacitor switching
9. Load transfer
10. Interruptable load curtailment
11. Firm load curtailment
12. Critical load curtailment (future)
13. Area interchange control

Control Variable: RAMSET = 1, initialize entire remedial action data base
                   = 2, modify remedial action database for specified contingency
                   = 3, restore remedial action database for specified contingency

CONPAR, defines the contingency to be considered.

ACFLAG = .T., perform remedial actions based on AC mode
         .F., perform remedial actions based on DC mode
Input Array: Database

Output Array: Database for the remedial actions.

Routine used: DBINIT, RAMI1, DBMDFY, DBRSTR, RAMI3.

Desirable Diagnostics:

For each control variable, if the present value exceeds the upper or lower bound, a diagnostic is issued.

Error Messages: None

Validation Test: A report routine is included which optionally reports all the control variables, their settings, and their bounds. This routine is used for validation purposes.
Figure 3.3.33.1 Data Preparation Submodule AC or DC Remedial Actions Flow Chart - RAINIT
<table>
<thead>
<tr>
<th>Remedial Action Module Based on AC Network Model</th>
<th>Remedial Action Module Based on DC NETWORK Model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialize</strong></td>
<td><strong>Modify for Contingency</strong></td>
</tr>
<tr>
<td>RAMSET = 1</td>
<td>RAMSET = 2</td>
</tr>
<tr>
<td>Gen Bus Voltage</td>
<td>Gen Bus Voltage</td>
</tr>
<tr>
<td>Gen BUs Real Power</td>
<td>Gen Bus Real Power</td>
</tr>
<tr>
<td>Phase Shift</td>
<td>Phase Shift</td>
</tr>
<tr>
<td>Switchable Capacitor</td>
<td>Switchable Reactor</td>
</tr>
<tr>
<td>Switchable Reactor</td>
<td>Inter/Firm Load</td>
</tr>
<tr>
<td>Inter/Firm Load</td>
<td>Load Transfer</td>
</tr>
<tr>
<td>Load Transfer</td>
<td>Interchange Control</td>
</tr>
<tr>
<td>Interchange Control</td>
<td>Interchange Control</td>
</tr>
</tbody>
</table>
Subroutine name: RAMI1

DESCRIPTION: This subroutine sets all the remedial action variables passed from the TRUCE program. It operates on the input data to define existing remedial actions and their limits.

INPUT ARRAY: TRUCE, Data bank, and ACFLAG

OUTPUT ARRAY: Data base for the remedial actions.

ROUTINE USED: SLCTVC, SLCTQC, SLCTRP, SLCTSR, SLTTTC, SLTPSC, SLTSCC, SLTSIC, SLTILC, SLTCLC, SLTFLC, FMHLD, DEFMWE

COMMENTS: This routine uses the data base to generate the remedial action arrays. In addition a complementary data file can be defined to specify control variables for specific devices which are different from the existing data in the data base. The complementary data file has the name SCNAME.DT3 and it is described in Appendix A.

Subroutine name: SLCTVC

DESCRIPTION: This subroutine defines the voltage control variables.

INPUT ARRAY: IGORDW, RABFLG, VBSPW, VMAXW, VMINW, NORDRP

OUTPUT ARRAY: IPCVG, XMVCVG, XMCVG, STPCVG, TCRCVG, NCVG

ROUTINE USED: None

Subroutine name: SLCTRP
DESCRIPTION: This subroutine forms the generation bus real power control variables with respect to system slack bus.

INPUT ARRAY: NGENB, IGORDW, NORDRP, PGTLW, PGMXW, PGMNW, RABFLG

OUTPUT ARRAY: KCSLC, NGRP, XMXGRP, XMNGRP, STPGRP, IPGRP, PVGRP, TCRGRP

ROUTINE USED: None

Subroutine name: SLTTTC

DESCRIPTION: This subroutine defines the transformer tap control variables.

INPUT ARRAY: NXFM, NPARW, IPTRW, LCDW, TTW, TTHW, TTLW, RACFLG

OUTPUT ARRAY: IPTTC, PVTTTC, XMXTTC, XMNTTC, STPTTC, TCRTTC, NTTC

ROUTINE USED: None

Subroutine name: SLTPSC

DESCRIPTION: This subroutine defines the transformer phase shifter control variables.

INPUT ARRAY: NXFM, ITRPW, ITPSW, LCDW, TTPHW, PHMXW, PHMNW, RACFLG

OUTPUT ARRAY: IPTPS, PVTPS, XMXTPS, XMNTPS, STPTPS, TCRTPS, NTPS

ROUTINE USED: None

Subroutine name: SLTSCC

DESCRIPTION: This subroutine defines the switched capacitor control variables.

INPUT ARRAY: NCAPW, NORDRP, XCAPW, ICAPW, CAPMNW, CAPSTW, CAPMXW, RABFLG

OUTPUT ARRAY: IPSCB, PVSCB, XMXSCB, XMNSCB, STPSCB, TCRSCB, NSCB

ROUTINE USED: None

Subroutine name: SLTSIC

DESCRIPTION: This subroutine defines the switched inductor
control variables.

INPUT ARRAY: NIDUW, NORDRP, XIDUW, IIDUW, XIDSTW, XIDMNW, XIDMXW RABFLG

OUTPUT ARRAY: IPSRB, PVSRB, XMKSRB, XMNSRB, STPSRB, TCRSRB, NSRB

ROUTINE USED: None

Subroutine name: SLTILC

DESCRIPTION: This subroutine defines the interruptible load control variables.

INPUT ARRAY: NLBW, NORDRP, ILBW, RABFLG and BSINTP, BSINTQ (both from TRUCE)

OUTPUT ARRAY: IPILS, PVILS, XMXILS, XMNILS, STPILS, TCRILS, NILS

ROUTINE USED: None

Subroutine SLTFLC

DESCRIPTION: This subroutine defines the firm load control.

INPUT ARRAY: NLBW, NORDRP, ILBW, RABFLG, and BSFRMP, BSFRMQ (both from TRUCE)

OUTPUT ARRAY: IPFLS, PVFLS, XMXFLS, XMNFLS, STPFLS, TCRFLS, NFLS

ROUTINE USED: None

Subroutine name: FIRMLD

DESCRIPTION: This subroutine defines the firm load shedding arrays. The firm load is defined as the difference of the total bus load minus the interruptible load.

INPUT ARRAY: NLBW, ILBW, NILS, PLDW, QLDW, IPILS, XMXILS, XPFILS, NORDRP, RABFLG

OUTPUT ARRAY: PVFLS, XMNFLS, XMXFLS, XPFFLS, STPFLS, TCRFLS

ROUTINE USED: None
Subroutine name: DEFMWE

DESCRIPTION: This subroutine defines and interprets the MW interchange data. It also collects and stores the data of controlling units.

INPUT ARRAY: LSEPWL, PREPW, LBSMW, IPXOC, NPARAW, NXOC, RACFLG, RABFLG AND DATA BANK

OUTPUT ARRAY: IPSXOC, DPXOC, DXOCPSXOC, DTXOC, DITXOC, NPXOC, JPIXG, JPIXG, TXRXG, OPXOC, IFXOC, JBPIXG, PRTIXG, XMIXG, XMNIXG, PVIXG, NIXG, NTXOC

ROUTINE USED: GETBSP, GETOPB, GETINC

Subroutine name: SLCRSR

DESCRIPTION: This routine forms generation bus real power controls with respect to area sink buses.

INPUT ARRAY: NGENB, data bank

OUTPUT ARRAY: NGSR, XMXGSR, XMNGSR, STPGSR, IPGSR, PVGSR, TCVGSR

ROUTINE USED: None

Subroutine name: SLCRQC

DESCRIPTION: This routine defines the reactive power controls.

INPUT ARRAY: Data bank

OUTPUT ARRAY: IPQVG, XMXQVG, XMNQVG, STPQVG, NQVG

ROUTINE USED: None

Subroutine name: DBINIT

DESCRIPTION: This routine builds the database for RA program based on the information from the TRUCE software. The arrays and variables defined in DBINIT are documented in the interface documentation. During the passing of the variables, different inferential assumptions to each side have been recognized and resolved. Finally, the resulting database matches all the conventions built into the RA program.

INPUT ARRAY: TRUCE database
OUTPUT ARRAY: RA database

ROUTINE USED: SRTBUS, FRMLNK, SNGLST, ISLND, SYSTYP, FDLFC1, FDLFC2
FDLFC3, FDLFD3, CMPJAC, CIRCFL, YEQUIB, ELJUPM

COMMENTS: For those not available from TRUCE, they are computed here.

Subroutine name: SNGLST
DESCRIPTION: This routine computes the single link list in internal order given the list of all branches (only for committed circuits). The list of branches should be ordered in ascending order.

INPUT ARRAY: NCIR, LSEPW, LREPW
OUTPUT ARRAY: IHKT, IHKM
ROUTINE USED: None

Subroutine name: CIRCFL
DESCRIPTION: This routine computes the circuit flows using the AC network model.

INPUT ARRAY: Database
OUTPUT ARRAY: PLFL1, PLFL2, QLFL1, QLFL2
ROUTINE USED: None

Subroutine name: SYSTYP
DESCRIPTION: This routine determines if the present operation state is a normal state by checking all the voltage and circuit flow constraints.

INPUT ARRAY: Database
OUTPUT ARRAY: NORMAL
ROUTINE USED: None

Subroutine name: FDLFC1
DESCRIPTION: This routine forms the Right-of-Way arrays.
INPUT ARRAY: NCIR, NPARW

OUTPUT ARRAY: LSELFA, LRELFA, ICROW, NROW

ROUTINE USED: FDLFC4

Subroutine name: FDLFC2

DESCRIPTION: This routine forms the Right-of-Way arrays, and makes them in optimal order.

INPUT ARRAY: NROW, LSELFA, LRELFA, NORDRP, NCIR

OUTPUT ARRAY: IAROW, IBROW, ICROW

ROUTINE USED: None

Subroutine name: FDLFC3

DESCRIPTION: This routine computes all the data necessary for the phase shifters and stores the results in two arrays.

INPUT ARRAY: Database

OUTPUT ARRAY: NTPHSH, XPHASH, IPHASH

ROUTINE USED: None

Subroutine name: FDLFD3

DESCRIPTION: This routine determines two elements in IPHASH as pointers needed during power flow calculation related to the phase shifters.

INPUT ARRAY: NTPHSH, IPHASH

OUTPUT ARRAY: IPHASH (6,*) AND IPHASH (7,*)

ROUTINE USED: GETIDX

Subroutine name: FDLFC4

DESCRIPTION: This routine performs the sorting of the R.O.W.s in the ascending order such that LRELFA > LSELFA.

INPUT ARRAY: LRELFA, LSELFA, NROW, NTOTAL
OUTPUT ARRAY: KASCR, the adjustment order between the before and after rearranging of LRELFA and LSELF.

ROUTINE USED: None

Subroutine name: YEQUIB
DESCRIPTION: This routine computes the bus equivalent admittance using an approximate method (sum of admittances).

INPUT ARRAY: Database
OUTPUT ARRAY: YEQBUS
ROUTINE USED: None

Subroutine name: ISLND
DESCRIPTION: This routine identifies all the islands which may exist in the system.

INPUT ARRAY: IHKT, IHKM, NTOTAL
OUTPUT ARRAY: Island related arrays and variables.
ROUTINE USED: None

Subroutine name: FRMLNK
DESCRIPTION: This routine generates pointer arrays (double link list) ICIRKT, ICIRKM, AND ICIRPN from the pointer arrays LSEPW AND LREP. Only committed circuits are considered.

INPUT ARRAY: NCIR, NTOTAL, LSEPW, LREP, CIRCOM
OUTPUT ARRAY: ICIRKT, ICIRKM, ICIRPN
ROUTINE USED: None

Subroutine name: SRTBUS
DESCRIPTION: This routine sorts the circuit arrays such that the sending and receiving bus numbers are in ascending order by major.

INPUT ARRAY: Database
OUTPUT ARRAY: All work arrays related to circuits and transformers.

ROUTINE USED: FDLFA1

Subroutine name: FDLFA1
DESCRIPTION: This routine performs the sorting of the circuits in ascending order such that \( LREP > LSEP \).

INPUT ARRAY: NTOTAL, NCIR, LCIRN, LSEP, LREP

OUTPUT ARRAY: ICIRTT

ROUTINE USED: None

Subroutine name: ELJUMP
DESCRIPTION: This routine gets rid of the jumpers in the mathematical model of power system. This jumper is defined as the branch whose series susceptance is greater than \( 10^4 \) (in p.u.). The outcomes of the subroutines are the reordered pointer of optimal bus number, and number of optimal buses. The internal bus number remains unchanged.

INPUT ARRAY: LSEPW, LREPW, BLPW, BUSO2I, BUSI20, NBUSES

OUTPUT ARRAY: NORDER, NTOTAL

ROUTINE USED: None

Subroutine name: FIXUNT
DESCRIPTION: This routine fixes data inconsistencies in the unit data. Specifically, the following fixes are performed.

1. If unit output is out of min-max range, the min-max values are adjusted.

2. Bus reactive power is allocated to individual units on the average.

INPUT ARRAY: IGFRST, IGNEXT, PGUAPW, UNTMXP, UNTMNP, QGTLW

OUTPUT ARRAY: UNTMXP, UNTMXP, QGUAP, QGUAPW

ROUTINE USED: None
Subroutine name: DBMDFY

DESCRIPTION: This routine modifies the database for RA programs for each contingency. This information is passed from TRUCE to RA through the CONPAR array. It only does the smallest amount of changes as possible in order to enhance the overall TRUCE execution efficiency.

INPUT ARRAY: CONPAR and data from TRUCE

OUTPUT ARRAY: Data base of RA

ROUTINE USED: FDLFC3, FDLFD3, CMPJAC, YEQUIB.

Subroutine name: DBRSTR

DESCRIPTION: This module restores the RA database to the before contingency state.

INPUT ARRAY:

OUTPUT ARRAY:

ROUTINE USED:

Subroutine name: DBDBG

DESCRIPTION: This routine reports all the arrays and variables used by the remedial actions. Most of the arrays are assumed to be passed from program TRUCE.

INPUT ARRAY: Database for RA

OUTPUT ARRAY: Formated and easy read reports.

ROUTINE USED: DBRP1, DBRP2, DBRP3, DBRP4, DBRP5, DBRP6, DBRP7, DBRP8, DBRP9, DBRP10, DBRP11, DBRP12, NEWMIS, TRDATA
1.1.1.13 Jacobian Formation Submodule - CMPJAC

SUBMODULE DESCRIPTION AND STRUCTURE:

This submodule forms, modifies or triangulates the Jacobian matrix for a specific network condition. Four alternative matrices can be used as follows: (1) full Jacobian, (2) semicoupled Jacobian (future) (3) decoupled constant Jacobian consisting of the B' and B", (4) DC network model Jacobian. The flow chart of the submodule is illustrated in "Figure 3.3.33.2 Jacobian Formation Submodule - CMPJAC" on page 97.

Control Variables:

JACFLG: A flag,
    when equals to .FALSE., Jacobian pointers have not been computed.
JACMOD: A code,
    = 2, perform factor update due to PQ to PV changes
    = 3, compute Jacobian matrix and table of factors
JACTYP: A code,
    = 1, Full jacobian
    = 2, Semicoupled jacobian (future)
    = 3, Decoupled constant jacobian
    = 4, DC Network model jacobian
JACSIZ: A code,
    = 1, exclude slack bus and PV-bus
    = 2, include all equations

Input Array: Data bank, LTYPLF

Output Arrays: NTJACB, IJKT, IJKM, IJKT1, IJKM1, YJOFD, YJD, HJOFD, HJD

Routine used: JACMPN, JACFPN, JACGRD, LUPNTO, LUFCTO, FRMJAC, IKTMA, BPDGRD, BPMTRX, BDTRX.

Validation Test:

This submodule is validated with a three step procedure:

1. A printout of the formed Jacobian is generated and the entries are manually verified.
2. A program was written which verified that the computed triangular factors are correct. Specifically, this program computes the product of the Jacobian matrix inverse using the factors and the Jacobian matrix. The result must be the identity matrix.
3. During the linearization process, if the linearization is correct, then the Jacobian matrix is correct.
Figure 3.3.33.2 Jacobian Formation Submodule - CMPJAC

ENTER

(FULL JACOBIAN) 1

(JACTYP = ?)

3 (B',B")

(JACFLG = ?)

FALSE

FORM THE POINTER OF THE JACOBIAN MATRIX & TABLE OF FACTOR OF THE POINTER

JACMOD = ?

= 2

= 3

FORM JACOBIAN

PERFORM FACTOR UPDATE, COMPUTE POINTERS OF THE TOF AND TRIANGULATE JACOBIAN MATRIX

PERFORM FACTOR UPDATE, TRIANGULATE THE DECOUPLED JACOBIAN MATRCIES

UPDATE FACTORS, TRIANGULATE THE DC NETWORK JACOBIAN MATRIX

RETURN
1.1.1.14 Linear Program Set-Up Submodule - LPMSET

SUBMODULE DESCRIPTION AND STRUCTURE

This submodule defines all the variables required by the LP solution module. Specifically, it requires the following information:

1. number of constraints
2. the linearization coefficients of the constraints
3. the total violation of the constraints
4. the bounds of the control variables
5. the type of control variables (continuous/discrete)

Based on this information, it builds the LP variables, the constraint matrix A, the vector of violations B, the vector of upper bounds U, a code indicating whether a variable is continuous or discrete, and an indicator for priority of each control variable.

The flow chart for this submodule is illustrated in "Figure 3.3.33.3 Linear Program Set-Up Submodule - LPMSET" on page 100.

Input Array: Data bank.

Output Arrays: UBVLP for vector of U (upper bound); SVLP & CVLP for vector b & c respectively; IAKT, IAKM, YOFA for matrix A; IPLPV1, IPLPV2, ILPDIS, IPRIOR, IBASE, JBASE, IBND, LPCNTP

Routine Used: SUBVLP, STBVLP, SETLP1, SETPL2, STCVLP

Diagnostics:

1. All upper bounds should be nonnegative. Otherwise, a diagnostic is issued.

2. At least one constraint should have a nonzero violation. Otherwise, a diagnostic is issued.

Error Messages:

None.

Validation Test:

The validation test will consist of manual construction of the LP model and comparing it to the computer generated LP model.

Methodology/Comments: The submodule sets the linear program in the following canonical form:
Minimize: $C^T x$

Subject to: $Ax < b$
$b > x > 0$

where: $b$ is an $m \times 1$
$x$ is an $n \times 1$ vector of unknown variables
$A$ is an $m \times n$ matrix
$C$ is an $n \times 1$ vector
$B$ is an upper bound vector of $n \times 1$
Figure 3.3.3 Linear Program Set-Up Submodule - LPMSET

Enter

Construct an LP model

Identify and flag discrete remedial actions, build priority list

Define the allowed excursion for LP variables

Check LP set and issue diagnostics if inconsistent

Return
Subroutine Name: SUBVLP
Brief Description: This subroutine forms the upper bound vector of the linear program: UBVLP, and priority list for corresponding LP variables.
Input Array: Data bank.
Output Arrays Variables: Upper bound vectors UBVLP, and IPRIOR
Routines Used: None

Subroutine Name: STBVLP
Brief Description: This subroutine forms the b vector of the linear BVLF.
Input Array: Data bank.
Output Arrays Variables: B, the vector b of the linear programming which relates to the voltage constraints, circuit flow constraints, and reactive power constraints.
Other Subroutines Called: None

Subroutine Name: STCLVP
Brief Description: This subroutine forms the cost vector of the linear CVLP.
Input Array: Data bank.
Output Arrays Variables: The vector C of the linear program, which is also called cost vector of control variables.
Routines Used: None

Subroutine Name: SETLP1
Brief Description: This subroutine sets the A matrix of the LP from scratch, and adjusts U vector in LP.
Input Array: Data bank.
Output Arrays Variables: IAKT, IAKM, YOFA, the matrix A in linear
programming.

IPLPV1, pointer of control variables included in the LP formulation
IPLPV2, inverse pointer of nonnegative variables in the LP formulation
UBVLP, upper bound vector of the linear programming

Routines Used: None

Subroutine Name: SETLP2

Brief Description: This subroutine packs the vector UBVLP, CVLP, define priorities and discrete steps for variables, initializes the array of IBASE, JBASE, IBND, LPCNTP, and sets the slack variables as basic variables.

Input Array: Data bank.

Output Arrays Variables: UBVLP, CVLP, IPRIOR, ILPDIS, JBASE, IBND, LPCNTP

Routines Used: None
1.1.1.15 Linear Program Solution Submodule - LPSOLV

SUBMODULE DESCRIPTION AND STRUCTURE

This module solves a linear programming problem with the simplex method upper bounds or with the dual method on the primal tableau - upper bounds. The problem is defined as follows:

\[
\begin{align*}
\text{min. } & \quad C^T x \\
\text{s.t. } & \quad A x = b \\
& \quad 0 \leq x \leq h
\end{align*}
\]

Input Arrays/Variables:

- KTLPV: total number of LP variables
- KTLPC: total number of constraints in LP formulation
- IACK: pointer of first nonzero element per column in Matrix A
- IACM: pointer of zero for nonzero elements in Matrix A
- CVLP: the cost vector C of LP
- BVLP: b vector of the LP
- UBVLP: upper bound vector of the LP
- IBASE: pointer array of basic variables in the extended basis
- JBASE: a code = 1, variable is in the basis = 0, variable is not in the basis
- WGLCP: vector of weights of constraints
- ILPDIS: a code = n, variable is discrete (n is the number of discrete steps between 0 and the upper bound.
- LPCNTP: a code = "I", inequality constraint = "E", equality constraint
- IPRIOR: a code indicating the priority of the LP variables

Output Arrays/Variables:

Subroutines used: LPMOD, DREBPNN, LPDVA1, (or LPDVA2, LPDVA3), LPUPAB, LPSOL, LPCADD, LPPOST

Error Messages: None
Validation Test: None

Subroutine Name: LPSOLV

Brief Description: This routine solves the linear programming problem by the following steps:

1. Sets equal priority for all variables.
2. Solves LP problem and computes penalties.
3. Scans the basic variables.
4. Replaces basic variable with a nonbasic to observe singularities.
5. Sets discrete variable to upper and lower limit
6. Updates the basis matrix.

Input Arrays Variables: This routine uses a large number of arrays from the data bank.

Output Arrays Variables: Basic matrix YWD, YWOFD, IWKT, IWKM

Other Subroutines Called: LPMOD, ROWB, SLVT1, CLMNB, RPBSB, MSVST, LPRP

Methodology/Comments: Linear Programming Module - The problem is define as follows:

\[
\begin{align*}
\text{Minimize:} & \quad CX \\
\text{Subject to:} & \quad Ax = B \\
& \quad 0 < x < H
\end{align*}
\]

The following arrays must be defined prior to the module being called:

IBND : A code defining the status of variable I

IBASE : A code defining the status of basic or nonbasic variables and NB, NV, IAKT, IAKM, YOFA, C, B, UB.

Subroutine Name: LPMOD

Brief Description: This routine solves an LP problem. The initial solution can be selected by controlling the code LPCODO.
Input Arrays/Variables: Control code LPCODO

= 1, start from scratch
= 2, start with primal feasible solution
= 3, start with dual feasible solution
= 4, start with a solution of unknown classification

And data bank.

Output Arrays/Variables:

SUBROUTINES CALLED: LPIO2A, LPIO3A, DFEAS, LPIO1A, LPA, LPB

Methodology/Comments: There are four ways used to solve LP problems here which are:

- dual feasible solution
- big M method
- primal simplex method
- dual simplex method

Subroutine Name: LPIO1A

Brief Description: This routine defines an initial solution with all artificial variables, and forms the basis matrix.

Input Arrays/Variables: IAKT, IAKM, LTLPV, KTLPC

Output Arrays/Variables: JBASE, YND, YWOFD, IWKT, IWKM, LPCOD6

Subroutine Name: LPIO2A

Brief Description: This routine initializes the basic matrix and working arrays for B and C to incorporate the slack and artificial variables introduced as starting feasible solution.

Input Arrays/Variables: IAKT, IAKM, YOFA, KTLPC, KTLPV, KTLPV2, LPCNTP, BVLP, CVLP

Output Arrays/Variables: BVLP, YOFA, IAKM, IAKT, IBND, JBASE, CVLP, LPACTV, ILPDIS, UBVLP, IBASE, ILPSEG

Methodology/Comments: First, initializes B matrix into the unit matrix, then replaces each column of the B matrix with the corresponding column of the A matrix.

Subroutine Name: LPIO3A
Brief Description: This routine checks the feasibility of a linear programming problem which is defined as minimizing $CX$ subject to $AX = B$ and $0 < x < H$ by returning the code LPCODF. It also defines the code LPCODE.

Input Arrays/Variables: LPCOD1, IBASE, JBASE, IBND, ITKT, ITKM, IAKT, IAKM, YD, YOFFD, YOFA, B, C, UB, NB, NV, ITKT1, ITKM1, HD, HOFFD, AK, AI, ALK, S, IAK, IAI, ISCR, JSCR, KSCR

Output Arrays/Variables: LPCODF, a code defined as follows:

- $= 0$, primal feasible and dual feasible
- $= 1$, primal infeasible and dual feasible
- $= 2$, primal feasible and dual infeasible
- $= 3$, primal infeasible and dual infeasible

LPCOD6, a code $= 0$, solution feasible
- $= 1$, solution infeasible

Other Subroutines Called: MSVST, RSDGN, MOSTN, ROWB, GTRSL1, GTRSL2, LPREP3

Subroutine Name: DFEAS

Brief Description: This routine computes an initial dual feasible solution to the upper bounded linear programming problem. Variables with negative reduced cost coefficients are initialized to their opposite bound thus, an initial dual feasible solution is obtained. It assumes that SNSLP2 contains the reduced cost coefficients.

Input Arrays Variables: SNSLP2, KTLPVW, IBND, UBVLP, IAKT, IAKM, BVLPW, YOFA, CVLPW

Output Arrays Variables: BVLDW, CVLPW, IBND

Subroutine Name: LPA

Brief Description: This routine solves the LP problem by primal simplex method with upper bounds. Refer to the flow chart for more detail.

Input Arrays Variables: IBASE, JBASE, IBND, IAKT, IAKM, YOFA, BVLPW, CVLPW, IWKT1, IWKM1, HWS, HWFD

Output Arrays Variables: IWKM, IWKT, YWD, YWOFD, BVLPW, CVLPW, IBASE
JBASE, IBND

Other Subroutines Called: MSVST, LPDBG1, LPDBG2, LPDBG3, GTRSL1, GTRSL2 UPDATE

Methodology/Comments: The following steps are involved with this method

1. Solve for basic variables
2. Resolve degeneracies
3. Determine entering variable L
4. Determine leaving variable M

Subroutine Name: MOSTN

Brief Description: This routine determines the negative entry of the present LP solution vector SNSLP1 with the absolute maximum negative value.

Input Arrays/Variables: KTLPC, SNSLP1

Output Arrays/Variables: The pointer of the most negative variable, M (= 0 if all variables are positive).

Other Subroutines Called: None

Subroutine Name: RSDGN

Brief Description: This routine resolves existing degeneracies of the primal problem. Basic variables which are at a bound are set to a value which is away from the bounds. Subsequently, the vector B is adjusted for consistency.

Input Arrays/Variables: IBASE, IWKT, IWKM, YD, YOFFD, SNSLP1, UBVLP, YWD, YWOFD, KTLPC

Output Arrays/Variables: BVLPW

Other Subroutines Called: None

Methodology/Comments: This problem is solved by two steps:

1. Identifying the existence of a degenerate solution
2. Updating vector B.

Subroutine Name: CLMNB
Brief Description: This routine makes some operation on the column of a matrix.

Input Arrays/Variables: A matrix in sparse form with N, ITKT, ITKM, YD, and YOFFD.

Output Arrays/Variables: The K-th column of matrix YK.

Other Subroutines Called: None

Subroutine Name: ROWB

Brief Description: This routine makes some operation on the row of a matrix.

Input Arrays/Variables: A matrix in sparse form with N2, ITKT, ITKM, and YOFFD.

Output Arrays/Variables: The K-th row of matrix YK.

Other Subroutines Called: None

Subroutine Name: MSVST

Brief Description: This routine solves a set of linear equations in sparse form:

\[ AX = B \]

The matrix A may be any general matrix which is stored in IWKT, IWKM, YWD, or YNOFD. The table of factors is generated and stored in the array IWKT1, IWKM1, HWD, and HOWFD. Triangularization is performed with pivoting on the maximum element to avoid zero diagonal elements and numerical errors. Thus, the column of matrix A can be interchanged.

The array IP1ROW provides information about this transformation. The driving vector is assumed to be BVLPW. The solution is returned to IP1ROW and SNSLP1.

Input Arrays/Variables: KTLP, YWD, YWOFD

Output Arrays/Variables: IWKT1, IWKM1, HWD, HWFD, IP1ROW, SNSLP1

Other Subroutines Called: GMLU, CLMNB, RPBSB, SLVT1
Subroutine Name: GMLU

Brief Description: This routine triangulates a given unsingular matrix. Numerical error is minimized by pivoting on the absolute largest element.

Input Arrays Variables: NB, ITKT, ITKM, YD, YOFFD

Output Arrays Variables: ROW, YM, YI, IYM, IYI, YD, YOFFD

Other Subroutines Called: CLMNB, RPBSB

Subroutine Name: SLVT1

Brief Description: This subroutine performs the forward and back substitution of a driving vector given the triangular factor of matrix A.

Input Arrays Variables: B, N, ITKT1, ITKM1, HD, HOFFD

Output Arrays Variables: Solution X

Other Subroutines Called: None

Subroutine Name: GTRSL1

Brief Description: This routine performs the dual forward substitution with the transpose matrix.

Input Arrays Variables: NT, B, ITKT1, ITKM1, HD, HOFFD

Output Arrays Variables: Vector X

Other Subroutines Called: None

Subroutine Name: GTRSL2

Brief Description: This routine performs the dual back substitution with the transpose matrix.

Input Arrays Variables: NT, B, ITKT1, ITKM1, HD, HOFFD

Output Arrays Variables: Vector X

Other Subroutines Called: None
Subroutine Name: LPSOL

Brief Description: This routine computes the complete solution of the linear program.

Input Arrays/Variables: IBASE, IBND, JBASE, KTLPC, KTLPV, KTLPI, SNSLP1, UBVLP, ILPSEG

Output Arrays/Variables: LPEFFC, SOLLSP, LPCODE

Other Subroutines Called: None

Subroutine Name: DREBPN

Brief Description: This routine computes the Driebeeks penalties at the present solution. These penalty factors are for discrete variables only.

Input Arrays Variables: This routine uses a large number of arrays from the data bank.

Output Arrays Variables: PNLTYO, PNLTY1

Other Subroutines Called: GTRSL1, GTRSL2

Subroutine Name: LPB

Brief Description: This routine solves the LP problem with the dual simplex method with upper bounds operated on the primal tableau.

Input and Output Arrays/Variables: IBASE, JBASE, IBND, SNSLP1, UBVLP, BVPWP, IWKM, IAKT, IAKM, CVLPW, YWD, YWOFD, IWKT1, IWKM1, YOFA, LPACTV, IP1ROW

Output Arrays/Variables: YWD, YWFFD, IWKT, IWKM, CVLPW, BVPWP, IBASE, JBASE, IBND, and output from routine MSVST

Other Subroutines Called: MSVST, LPDBG1, LPDBG2, LPDBG3, MOSTN, GTRSL1, GTRSL2, UPDATE, MOSTN

Methodology/Comments: The following steps are involved with this method

1. Solving for basic variables
2. Correcting upper bounds
3. Getting the most negative variable
4. Determining the leaving and entering variables.
Subroutine Name: RPBSB

Brief Description: This routine substitutes one column vector of a matrix with another. The matrix is stored in sparse form.

Input Arrays/Variables: ITKT, ITKM, YD, YOFFD, AJ, NB, AK, J

Other Subroutines Called: None

Subroutine Name: UPDATE

Brief Description: This subroutine modifies the LP basic matrix to reflect a change in the basic. Specifically, variable L enters the basic, variable M leaves and row M of the basic matrix is reflected by row L of the LP matrix.

Input Arrays/Variables: M which is the leaving variable, L which is the entering variable, IWKT, IWKM, IAKT, IAKM, YOFA, YWOFD

Output Arrays/Variables: IWKT, IWKM, YWD, YWOFD

Other Subroutines Called: None

Subroutine Name: LPUPAB

Brief Description: This routine updates the entries up the columns of the columns of the A matrix in LP that correspond to artificial variables after the assignment of some discrete variables to their upper bounds.

Input Arrays/Variables: BVLP, IAKT, IAKM, YOFA, UBVL, KT LPC, KT LPV, KTLPV1, EPTHNN, IBND, LPACTV, BVLPW

Output Arrays/Variables: YOFA

Other Subroutines Called: None

Subroutine Name: LPCADD

Brief Description: This routine verifies that a pair of LP variables resulting from a control variable are both nonzero.
Input Arrays/Variables: KTLPV, KTLPC, SOLLP, EPTHNN, IAKM, IAKT, YOFA

Output Arrays/Variables: None

Other Subroutines Called: ROWB

Subroutine Name: LPPOST

Brief Description: This routine performs postoptimality analysis. It computes the sensitivity of optimal value of the objective function with respect to the binding constraints of the LP solution. Two types of constraints are analyzed:
   a) Functional constraints
   b) Upper bounding constraints

Input Arrays/Variables: KTLPC, KTLPV, KTLPV2, IBASE, JBASE, IBND, CVLP, IAKT, IAKM, YOFA, IWKT1, IWKM1, HWD, HWFD, IP1ROW, LPCNTP

Output Arrays/Variables: SNSLP1, SNSLP2

Other Subroutines Called: SLVT1, ROWB

Subroutine Name: LPDVA1

Brief Description: This routine assigns discrete values to the discrete type variables that were given non-discrete values during LP solution (basic). The basic variables that leave the basic of LP solution are replaced by eligible slack variables.

Input Arrays/Variables: KTLPC, KTLPV, ILPDIS, KTLPV1, KTLPV2, IBASE, JBASE, IBND, IP1ROW, IAKT, IAKM, YOFA, CVLP, UBVLp, EPTHNN, ILPSEG, BVLPW, CVLPW, PNLTYO, PNLTY1

Output Arrays/Variables: BVLPW, CVLPW, IBASE, JBASE, YWD, YWOFD, IWKT, IWKM

Other Subroutines Called: GTRSL1, GTRSL2, UPDATE, MSVST

112
INITIALIZATION OF WORKING ARRAYS AND VARIABLES

SOLVE THE LP PROBLEM BY LPMOD

ASSIGN DISCRETE VALUES TO THE DISCRETE VARIABLES ACCORDING TO CONTROL CODE KODDVA BY LPDVA1, LPDVA2, OR LPDVA3

UPDATE THE B-VECTOR AND A-MATRIX BY LPUPAB AND SOLVE THE LP PROBLEM BY LPMOD

ISSUE THE SOLUTION TYPE TYPE ONTO THE SCREEN ACCORDING TO CONTROL CODE LPCODG

PERFORM POSTOPTIONALITY ANALYSIS BY LPPOST
SOLVE LP PROBLEM WITH PRIMAL SIMPLEX BY LPA

LPCOD0 = ?

(100) LPCODF = ?

(400) SELECT PRIMAL OR DUAL METHOD

COMPUTE A DUAL FEASIBLE SOLUTION BY DFEAS

SOLVE LP PROBLEM WITH THE DUAL SIMPLEX BY LPB

FEASIBLE LPCOD6 = 0 ?

DETERMINE EXISTANCE OF ARTIFICIAL VARIABLE IN SOLUTION BY ASSIGNING LPCOD6, 0 OR 1

NO LPCOD6 = 1 ?

NO LPCOD6 = 17?

TRY NEXT PRIORITY

RETURN

KEY = 1 ?

FLOW CHART OF SUBROUTINE LPMOD
CHECK FOR PRIMAL FEASIBILITY BY GETTING THE MOST NEGATIVE VARIABLE IN THE SOLUTION USING MOSTN

M = 0 ?

NO

LPCODG = 1

YES

CHECK FOR DUAL FEASIBILITY, BUT NO ALTERING PRESENT VARIABLES BY EVALUATING ALL Z - C, TO DETERMINE L

l = 0 ?

NO

LPCODG = 1

YES

COMPUTE LPCODF ACCORDING TO M & L

GENERATE REPORT OPTIONALLY

RETURN

FLOW CHART OF SUBROUTINE LPI03A
For the column \( L \), compute \( \delta_t + (U_t - \delta_t) \) to determine basic variable \( M \) leaves the basic variable \( L \) enters basis, and variable \( M \) leaves (actually \( M_2 \)).

- If \( L = 0 \), return.
- If \( \text{EPS1} > \text{EPS2} \), variable \( L \) enters basis, and variable \( M \) leaves (actually \( M_2 \)).
- If non-basic variable \( L \) hits opposite limit, change bound indicator for variables, adjust BVLPW, CVLPW and SART another iteration.
- If variable \( M_2 \) hits opposite limit, adjust BVLPW, CVLPW accordingly.

Update the basis matrix \( YWD \), \( YWOFD \), \( IWKT \), \( IWKM \), replace row \( M \) of basis matrix with row \( L \) by update.
DETERMINE THE ENTERING VARIABLE \( L \) BY EVALUATING \( (Z_j - C_j) / Y_{r_j} \) FOR ALL \( Y_{r_j} < 0 \).

IF \( Y_{r_j} > 0 \) FOR ALL \( j \), STOP; THE DUAL IS UNBOUNDED AND THE PRIMAL IS INFEASIBLE, OTHERWISE.

VARIABLE \( L \) ENTERS BASIS AND \( M \) LEAVES (ACTUALLY \( M_2 \))
UPDATE IBASE, JBASE, AND USING UPDATE TO MODIFY YWD, YWOFD, IWKT, IWKM.

FLOW CHART OF SUBROUTINE LPB
2.0 REFERENCES


References 113
INDEX

S

<table>
<thead>
<tr>
<th>Submodules</th>
<th>Subroutines</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADBG1</td>
<td>48</td>
</tr>
<tr>
<td>RPAC4</td>
<td>48</td>
</tr>
<tr>
<td>RPAC5</td>
<td>48</td>
</tr>
<tr>
<td>RPAC7</td>
<td>48</td>
</tr>
<tr>
<td>RPAC8</td>
<td>49</td>
</tr>
<tr>
<td>RPAC9</td>
<td>49</td>
</tr>
<tr>
<td>ACRVOL</td>
<td>13</td>
</tr>
<tr>
<td>CCFLAC</td>
<td>45</td>
</tr>
<tr>
<td>CCFLD</td>
<td>58</td>
</tr>
<tr>
<td>CHGRDC</td>
<td>64</td>
</tr>
<tr>
<td>CIRCFL</td>
<td>91</td>
</tr>
<tr>
<td>CLMN</td>
<td>107</td>
</tr>
<tr>
<td>CNVCA</td>
<td>13</td>
</tr>
<tr>
<td>CNVPDT</td>
<td>11</td>
</tr>
<tr>
<td>CSHGRP</td>
<td>19</td>
</tr>
<tr>
<td>CSGENQ</td>
<td>20</td>
</tr>
<tr>
<td>CSLNW</td>
<td>20</td>
</tr>
<tr>
<td>CSNSA</td>
<td>26</td>
</tr>
<tr>
<td>CSNSC</td>
<td>30</td>
</tr>
<tr>
<td>CSNSD</td>
<td>33</td>
</tr>
<tr>
<td>CSNSDC</td>
<td>76</td>
</tr>
<tr>
<td>CSNSE</td>
<td>34</td>
</tr>
<tr>
<td>CSNSF</td>
<td>36</td>
</tr>
<tr>
<td>CSNSH</td>
<td>37</td>
</tr>
<tr>
<td>DBDBG</td>
<td>95</td>
</tr>
<tr>
<td>DBINIT</td>
<td>90</td>
</tr>
<tr>
<td>DBMDFY</td>
<td>95</td>
</tr>
<tr>
<td>DBRSTR</td>
<td>95</td>
</tr>
<tr>
<td>DCCLNZ</td>
<td>68</td>
</tr>
<tr>
<td>DCOHR</td>
<td>62</td>
</tr>
<tr>
<td>DCIDNT</td>
<td>57</td>
</tr>
<tr>
<td>DCPFUP</td>
<td>80</td>
</tr>
<tr>
<td>DCPMS</td>
<td>82</td>
</tr>
<tr>
<td>DEFMWE</td>
<td>90</td>
</tr>
<tr>
<td>DFEAS</td>
<td>106</td>
</tr>
<tr>
<td>DISCRA</td>
<td>104</td>
</tr>
<tr>
<td>DREBP</td>
<td>110</td>
</tr>
<tr>
<td>ELJUMP</td>
<td>94</td>
</tr>
<tr>
<td>ELSSL</td>
<td>41</td>
</tr>
<tr>
<td>FDLFA1</td>
<td>94</td>
</tr>
<tr>
<td>FDLFC1</td>
<td>91</td>
</tr>
<tr>
<td>FDLFC</td>
<td>92</td>
</tr>
<tr>
<td>FDLFC2</td>
<td>92</td>
</tr>
<tr>
<td>FDLFC3</td>
<td>92</td>
</tr>
<tr>
<td>FDLFC4</td>
<td>92</td>
</tr>
<tr>
<td>FDLFD3</td>
<td>92</td>
</tr>
<tr>
<td>FIRMLD</td>
<td>89</td>
</tr>
<tr>
<td>FIXUNT</td>
<td>94</td>
</tr>
<tr>
<td>FORMS1</td>
<td>41</td>
</tr>
<tr>
<td>FRMLNK</td>
<td>93</td>
</tr>
<tr>
<td>GMLU</td>
<td>109</td>
</tr>
<tr>
<td>GTRSL1</td>
<td>109</td>
</tr>
<tr>
<td>GTRSL2</td>
<td>109</td>
</tr>
<tr>
<td>ISLND</td>
<td>93</td>
</tr>
<tr>
<td>JACFPN</td>
<td>40</td>
</tr>
<tr>
<td>LGAC1</td>
<td>18</td>
</tr>
<tr>
<td>LGAC2</td>
<td>18</td>
</tr>
<tr>
<td>LGAC3</td>
<td>18</td>
</tr>
<tr>
<td>LGDC1</td>
<td>62</td>
</tr>
<tr>
<td>LGDC2</td>
<td>63</td>
</tr>
<tr>
<td>LPA</td>
<td>106</td>
</tr>
<tr>
<td>LPB</td>
<td>110</td>
</tr>
<tr>
<td>LPCADD</td>
<td>111</td>
</tr>
<tr>
<td>LPDVA1</td>
<td>112</td>
</tr>
<tr>
<td>LPI01A</td>
<td>105</td>
</tr>
<tr>
<td>LPI02A</td>
<td>105</td>
</tr>
<tr>
<td>LPI03A</td>
<td>105</td>
</tr>
<tr>
<td>LPMOD</td>
<td>104</td>
</tr>
<tr>
<td>LPPOST</td>
<td>112</td>
</tr>
<tr>
<td>LPSOL</td>
<td>110</td>
</tr>
<tr>
<td>LPUPAB</td>
<td>111</td>
</tr>
<tr>
<td>LUFCTO</td>
<td>41</td>
</tr>
<tr>
<td>MDAC6A</td>
<td>16</td>
</tr>
<tr>
<td>MDAC6B</td>
<td>17</td>
</tr>
<tr>
<td>MDAC6C</td>
<td>17</td>
</tr>
<tr>
<td>MDC4A</td>
<td>72</td>
</tr>
<tr>
<td>MDC4D</td>
<td>72</td>
</tr>
<tr>
<td>MDC4H</td>
<td>73</td>
</tr>
<tr>
<td>MOD4A</td>
<td>26</td>
</tr>
<tr>
<td>MOD4B</td>
<td>27</td>
</tr>
<tr>
<td>MOD4C</td>
<td>30</td>
</tr>
<tr>
<td>MOD4D</td>
<td>33</td>
</tr>
<tr>
<td>MOD4E</td>
<td>34</td>
</tr>
<tr>
<td>MOD4F</td>
<td>35</td>
</tr>
<tr>
<td>MOD4G</td>
<td>36</td>
</tr>
<tr>
<td>MOD4H</td>
<td>37</td>
</tr>
<tr>
<td>MOSTN</td>
<td>107</td>
</tr>
<tr>
<td>MSVST</td>
<td>108</td>
</tr>
<tr>
<td>PBAOC</td>
<td>45</td>
</tr>
<tr>
<td>PFUPD</td>
<td>45</td>
</tr>
<tr>
<td>RAMI1</td>
<td>87</td>
</tr>
<tr>
<td>RAMI3</td>
<td>87</td>
</tr>
<tr>
<td>RMCHKB</td>
<td>59</td>
</tr>
<tr>
<td>ROWB</td>
<td>108</td>
</tr>
<tr>
<td>RPBSB</td>
<td>111</td>
</tr>
<tr>
<td>RPD</td>
<td>76</td>
</tr>
<tr>
<td>RPD5</td>
<td>65</td>
</tr>
<tr>
<td>Code</td>
<td>Value</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>RPDC6</td>
<td>65</td>
</tr>
<tr>
<td>RPDC8</td>
<td>81</td>
</tr>
<tr>
<td>RSDGN</td>
<td>107</td>
</tr>
<tr>
<td>SETLP1</td>
<td>101</td>
</tr>
<tr>
<td>SETLP2</td>
<td>102</td>
</tr>
<tr>
<td>SLCTQC</td>
<td>90</td>
</tr>
<tr>
<td>SLCTRP</td>
<td>87</td>
</tr>
<tr>
<td>SLCTSR</td>
<td>90</td>
</tr>
<tr>
<td>SLCTVC</td>
<td>87</td>
</tr>
<tr>
<td>SLTFLC</td>
<td>89</td>
</tr>
<tr>
<td>SLTILC</td>
<td>89</td>
</tr>
<tr>
<td>SLTPSC</td>
<td>88</td>
</tr>
<tr>
<td>SLTSCC</td>
<td>88</td>
</tr>
<tr>
<td>SLTSIC</td>
<td>88</td>
</tr>
<tr>
<td>SLTTTC</td>
<td>88</td>
</tr>
<tr>
<td>SLVT1</td>
<td>109</td>
</tr>
<tr>
<td>SNGLST</td>
<td>91</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRTBUS</td>
<td>93</td>
</tr>
<tr>
<td>STBVLP</td>
<td>101</td>
</tr>
<tr>
<td>STCVLP</td>
<td>101</td>
</tr>
<tr>
<td>STOVAC</td>
<td>12</td>
</tr>
<tr>
<td>STOVDC</td>
<td>58</td>
</tr>
<tr>
<td>SUBVLP</td>
<td>101</td>
</tr>
<tr>
<td>SYSTYP</td>
<td>91</td>
</tr>
<tr>
<td>UPACCV</td>
<td>44</td>
</tr>
<tr>
<td>UPACLFLF</td>
<td>44</td>
</tr>
<tr>
<td>UPDATE</td>
<td>111</td>
</tr>
<tr>
<td>UPDCCV</td>
<td>80</td>
</tr>
<tr>
<td>UPDCLF</td>
<td>81</td>
</tr>
<tr>
<td>WORST</td>
<td>19</td>
</tr>
<tr>
<td>WRSTDC</td>
<td>63</td>
</tr>
<tr>
<td>YEQUIB</td>
<td>93</td>
</tr>
<tr>
<td>ZOOM3</td>
<td>64</td>
</tr>
<tr>
<td>ZOOM4</td>
<td>19</td>
</tr>
<tr>
<td>ID</td>
<td>File</td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>Tab1</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>acremd</td>
<td>?</td>
</tr>
<tr>
<td>actab</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>acram</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>pfck</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>acidnt</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>accohr</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>aclinz</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>acl1</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>acl2</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>acl3</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>acl4</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>acpfup</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>dcram</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>dcidnt</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>dcohr</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>dclinz</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>dclin1</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>dclin2</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>dcpfup</td>
<td>3159SAK1</td>
</tr>
<tr>
<td>Module</td>
<td>Page</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>ramint</td>
<td>85</td>
</tr>
<tr>
<td>dctab</td>
<td>86</td>
</tr>
<tr>
<td>cmpjac</td>
<td>97</td>
</tr>
<tr>
<td>ipmset</td>
<td>100</td>
</tr>
</tbody>
</table>

**Imbed Trace**
Debugging Submodule

MODULE DESCRIPTION AND STRUCTURE

This submodule contains subroutines which generate a report of specific arrays or groups of arrays from the database. They are used in the debugging process. Each subroutine is independent of each other. It is also non-intrusive (i.e. it does not modify the database).

Remedial Action Report Module (AC Model) (RADBG1)

Subroutine name RPAC4: This routine gives the sensitivity report.

Subroutine name RPAC5: This routine reports operating constraints and model reduction during the operation of the power system. The constraints include bus voltage, circuit flow and reactive generation.

Subroutine name RPAC7: This routine reports the arrays and variables created in submodule LPMSET. Basically, it reports the coefficient matrices in the LP form:

\[
\begin{align*}
\text{Min.} & \quad c^T x \\
\text{s.t.} & \quad A x = b \\
& \quad x < h
\end{align*}
\]

Subroutine name RPAC8: This routine reports the linear programming solutions for the AC model.

Subroutine name RPAC9: This routine summarizes the total computed remedial actions.

Subroutine name SQMTRX: This routine generates a report of a square matrix stored in sparsity form with double precision.

Debug Report Routines for Remedial Action Data Setup (RADBG2)

Subroutine name RDBG01: This routine generates a report of the database for remedial actions which is listed in the Remedial Action documentation

Subroutine name RDBG03: This routine generates a report of all pertinent arrays for the computation of the Jacobian matrix. It also reports the B' matrix.

Subroutine name RDBG04: This routine generates a report of bus data, circuit data and data of power apparatus.

Subroutine name OPFDB1: This routine generates a report of the data related to the OPF.
Module RADBG3 reports all the arrays and variables used by the remedial actions. Most of the arrays are assumed to be passed by program TRUCE.

Subroutine name RADBG3: This routine generates a report of all system count numbers of interested quantities such as:

- Number of generation units in system NGUNP
- Total buses in system NTOTAL
- Total number of circuits NCIR
- Total number of transformers NXFM
- Total number of phase shifters NTPHSH
- Total number of generation buses NGENB
- Total number of areas in system NTAREA
- Total number of constant power load NLBW
- Total number of impedance load NCILW
- Total number of shunt capacitors NCAPW
- Total number of voltage regulators NREGW
- Total number of DC links DCLINK
- Total number of Right of Ways NROW

Subroutine name DBRP1: This routine reports the generating units and their link list in coordination with TRUCE program.

Subroutine name DBRP2: Generate report of all bus Working arrays.

Subroutine name DBRP2: Generate a report of circuit arrays.

Subroutine name DBRP4: This routine generates a report of linklists for circuits.

Subroutine name DBRP5: This routine generates a report of the working arrays for the generation buses.

Subroutine name DBRP6: This routine reports the bus equivalent impedances and circuit flows.

Subroutine name DBRP7: This routine reports the formation of islands.

Subroutine name DBRP8: This routine computes the area generation, load losses and generates a report.

Subroutine name DBRP9: This routine generates a report of all the load flow arrays in conjunction with TRUCE program.
Subroutine name DBRP11: This routine reports the Jacobian matrix and the table of factors.

Subroutine name DBRP12: This routine reports the fast decoupled power flow matrices, B' and B" by calling two subroutines: PRSM01, PSMTRX.

Subroutine name PHMTRX: This routine generates a report of a matrix pointers stored in half sparsity form.

Subroutine name PSMTRX: This routine generates a report of a square matrix stored in sparsity form.

Subroutine name PRSM01: This routine generates a report of a square matrix stored in sparsity form.

Remedial Action Database Supporting Report (RADBG4)

Subroutine name DBG004: This routine generates a report of all the flow arrays which are:

- Maximum number of iterations
- Real power mismatch criterion
- Reactive power mismatch criterion
- DC link equation mismatch
- AC/DC power flow solution code
- Load flow convergence code
- A load flow solution code
- A load flow solution code
- A load flow solution code
- A load flow solution code
- A load flow solution code
- A load flow solution code
- A load flow solution code
- # of increase mismatch for div.
- # of succ viol. for PV to PQ switch
- IAROW, IBROW, ICROW, IWRST, NORDER, NORDR, LBSMW, NBPBOP, LTYPLF, IJDPOS, IJVPOS, GSLFA, BSLFA, PINJLF, QINJLF, VOLTW, DELTW, YEQBUS, LSELF, RELFA, GLFA, BCLFA, IORDGN, VLCFA, PLLFA, QLLFA, GLLFA, QMAXLF, QMINLF, XPHASH, IPHASH, DCFROM, DCTO, DCCIRP, DCPARM, DCCNTR, DCCONS

Subroutine name DBG007: This routine generates a report of the generation
units and their link list:
IDGUN, ISGUN, ISGUNW, UNSTAT, NGBUSS, NGBUSI, KPGUN, KVGUN,
IGFRST, IGNEXT, UNTMXP, UNTMNP, PGUAP, QGUAP, SQUMXP,
PGUAPW, PFGUPW, QFGUPW

Subroutine name PFDBG2: This routine reports the maximum real power mismatch.

Subroutine name PFDBG3: This routine reports the maximum reactive mismatch.

Subroutine name PFDBG4: This routine reports the maximum mismatch of the DC links.

Subroutine name GLFREP: This routine reports the power flow results in a 'PICO' alike format.

KODE = 1 CRT display — full or zoomed system
2 Hardcopy to file defined with channel

Subroutine name DBG032: This routine reports the Jacobian matrix and the table of factors by calling two subroutine names: PSMTRX, PSMTRX.

SUBMODULE DESCRIPTION AND STRUCTURE
This submodule contains subroutines which are used as general support routines for various network analysis problems.

Subroutine name: STEPAC
Description: This routine defines the valid bounds of the constraints in the linearized model.
Input arrays: NCVG, NORDER, IGORDW, QGTLW, QGMNW, YD, PVCVG, NGRP,
IPGRP, NTTC, IPTTC, TIRPW, LSEPW, LREPW, NORDP, NBPDBP, YVD, BLPW, all
from the remedial action data bank.
Output arrays: STPCVG, STPGRP, STPTTC, STPTS, STPSCB, STPSRB, STPFLS,
STPRLT, STPIXG, STPILS
Subroutines called: None

Subroutine name: CSGENP
Description: This routine computes the total real power generation at slack buses.

Input arrays: KISLND, SBISLD, PLLFA, ICIRKT, ICIRPN, PLFL1, PLFL2, DCFLOW, and data bank.

Output arrays: PGTLW

Subroutines called: GETBSP, GSTCFL, CSLFLW

Subroutine name: CTGENQ

Description: This routine computes the generated reactive power for all generating buses.

Input arrays: When this routine is called, bus type(optimal order), LTYPLF should have been defined. Also, for the total number of generation buses NGENB, and pointer of circuits as well as circuit flows. PARAM, IGORDW, NORDRP, VOLT, NLBW, ILBW, QLDW, NCILW, ICILW, BCILW, NCAPW, XCAPW, NIDUM, IIDUM, XIDUM, NLPN, ILPN, LSEPW, LREPW, PBASE

Output arrays: QGTLW

Subroutines called: None

Subroutine name: DCPMS

Description: This routine computes the mismatches for the DC power flow solution.

Input arrays: PINJFL, NTPHSH, IPHASH, XPHASH, ICILW, NORDRP, GCILW

Output arrays: XASCR

Subroutines called: None

Subroutine name: CCFLDC

Description: This routine computes the circuit real power flow using the DC network model.

Input arrays: LSEPW, LREPW, NORDPR, BLPW, LCDW, DELTW, IPTRW, TTPHW, TTW

Output arrays: PLFL1, PLFL2, QLFL1, QLFL2

Subroutines called: None
Subroutine name: CCFLAC
Description: This routine computes the circuit flows using the AC network model.
Input arrays: LSEPW, LREPW, NORDRP, GLPW, BLPW, SLPW, VOLTW, LCDW, DELTW
Output arrays: PLFL1, PLFL2, QLFL1, QLFL2
Subroutines called: None

Subroutine name: LUPNTO
Description: This routine simulates the Gaussian elimination to determine the number and location of fill-ins. It is customized for network type matrices which is expected to be topologically symmetric.
Input arrays: NT, ITKT, ITKM
Output arrays: ITKT1, ITKM1
Subroutines called: None

Subroutine name: LUFCTO
Description: This routine computes the table of factors. It is customized for network type matrices.
Input arrays: ITKT, ITKM, ITKT1, ITKM1, YOFFD, YD
Output arrays: HOFFD, HD
Subroutines called: None

Subroutine name: SLVTR1
Description: This routine performs the dual forward substitution with the transposed matrix.
Input arrays: NT, NS, B, ITKT1, ITKM1, HD, HOFFD
Output arrays: X
Subroutines called: None
Subroutine name: SLVTR2
Description: This routine performs the dual back substitution with the transposed matrix.
Input arrays: NT, B, ITKT1, ITKM1, HD, HOFFD
Output arrays: X
Subroutines called: None

Subroutine name: SOLVT
Description: This routine performs the forward and back substitution on a given vector \( b \) with a given set of factors.
Input arrays: NT, B, ITKT1, ITKM1
Output arrays: X
Subroutines called: None

Subroutine name: GETINC
Description: This routine defines the internal circuit number given the terminal bus symbolic number and circuit number.
Input arrays: NCIR, NPARW, LSEPW, LREPW, IBS1, IBS2, ICKT
Output arrays: ICI
Subroutines called: None

Subroutine name: GETIBN
Description: This routine determines the internal bus number of a given symbolic bus number.
Input arrays: NTOTAL, ISBN
Output arrays: IIBN
Subroutines called: None
Subroutine name: GETBSP

Description: This routine determines the order of:

- Generation bus
- Constant power factor load bus
- Shunt capacitor bus
- Shunt reactor bus.

Input arrays:
- K1: bus internal number
- KP: order of bus K1, <> 0, pointer of bus in array IBPNTR
  = 0, bus K1 is not included in array IBPNTR
- NT: total number of entries in IBPNTR

Output arrays: KP

Subroutines called: None

-----------------------------

Subroutine name: GETCFL

Description: This routine determines the circuits which are connected to bus K1.

Input arrays:
- NCIR, LSEPW, LREPW

Output arrays:
- NLPN, number of circuits connected to bus K1
- IASCR, pointer array of above circuits (points arrays LSEPW, LREPW)

Subroutines called: None

-----------------------------

Subroutine name: ITKTMA

Description: This routine computes the network topology pointers ITKT and ITKM. It also defines the array DTOSRW which points the ROW from the double link list.

Input arrays:
- NTOTAL, LSELF, LRELFA

Output arrays:
- ITKT, ITKM, DTOSRW

Subroutines called: None

-----------------------------

Subroutine name: JACMPN

Description: This routine forms the pointers for the off diagonal elements of the full Jacobian matrix.
Input arrays: ITKM, ITKT, IJDPOS, IJVPOS

Output arrays: IJKT, IJKM, IJDDPS, IJVVPS, JACPDP, JACPVP, JACQDP, JACQVP

Subroutines called: None

Subroutine name: JACFPN

Description: This routine forms the pointers for the off diagonal elements of the Table of Factors of the Jacobian matrix.

Input arrays: ITKT1, ITKM1, IJDPOS, IJVPOS

Output arrays: IJKT1, IJKM1

Subroutines called: None

Subroutine name: FRMJCA

Description: This routine computes the entries of the full Jacobian matrix. The form of the Jacobian is in bus block. The following arrays must have been defined as input variables:

IJKT, IJKM, IJDPOS, IJVPOS, IJDDPS, IJVVPS, JACPDP, JACPQDP, JACQVP

Input arrays: In addition to the above arrays, also the remedial action data bank.

Output arrays: YJD, YJOFD

Subroutines called: None

Subroutine name: JACGRD

Description: This routine grounds the Jacobian modes which correspond to slack bus by adding 50000.0 to them.

Argument: KODE = 1, ground appropriate modes
KODE = 2, unground appropriate modes

Input arrays: KISLND, SBISLD, LBTPW, NORDER, NORDRP, IJDPOS, YJD, IJVPOS

Output arrays: YJD

Subroutines called: None
Subroutine name: BPMTRX

description: This routine forms the arrays of the \( B' \) matrix in sparse form.

Input arrays: ITKT, ITKM, DTOSRW, BCLFA, NTPHSH, IPHASH, XPHASH

Output arrays: YOFFD, YD

Subroutines called: None

Subroutine name: BDMTRX

description: This routine computes \( b'' \) matrix in sparse form. It assumes (1) \( B' \) matrix has been formed (2) Array LTYPLF has been defined.

Input arrays: NTOTAL, KTOTAL, ITKT, ITKM, DTOSRW, GCLFA, BCLFA, IPHASH, XPHASH, NTPHSH

Output arrays: YVOFD, YVD, IVKM, IVKT

Subroutines called: None

Subroutine name: BPDGRD

description: This routine grounds or ungrounds the matrices \( B' \) and \( b'' \) of the fast decoupled power flow.

Argument KODE1 = 1, ground the matrices
= 2, unground the matrices

KODE2 = 1, operate on both \( B' \) and \( b'' \)
=-1, operate on \( B'' \) only

Input arrays: KISLND, SBISLD, NORDER, NORDRP, YD, YVD, NBPBDP, 
IBFRST, IBNEXT

Output arrays: YD, YVD

Subroutines called: None

Subroutine name: QMSMCH

description: This routine computes the reactive power mismatches.

Argument KODEW = 1, divide mismatch with bus voltages
= 0, do not modify mismatches

DQMAX, maximum recorded reactive power mismatch
KMAX, bus optimal number with max mismatch
KB = 0, convergence criterion is satisfied
   = 1, convergence criterion fails

Input arrays:  KTOTAL, NVORDR, ITKT, ITKM, etc., remedial action
data bank.

Output arrays:  KODEW, DQMAX, KMAX, KB, XBSCR

Subroutines called:  None

Subroutine name:  PMSMCH

Description:  This routine computes the real power mismatches
Arguments:  KODEW = 1, divide mismatches with bus voltage
            = 0, do not modify mismatches
            DPMAX, maximum recorded real power mismatch
            KMAX, bus optimal order with max mismatch
            KA = 0, convergence criterion is satisfied
                 = 1, convergence criterion fails

Input arrays:  NTOTAL, ITKT, TIKM, DTOSRW, etc., remedial action data
bank.

Output arrays:  KODEW, DPMAX, KMAX, KA, XASCn

Subroutines called:  None

Subroutine name:  GETIDX

Description:  This routine returns the index of the array IARRAY
which matches the specified number ISN
Arguments:  ISN, specified number, if positive return a zero in case
            of an unsuccessful search; if negative, generate a
            fatal error in case of an unsuccessful search.

            IIDX, returned array index (zero if not found)

Input arrays:  ISN

Output arrays:  IARRAY, IIDX

Subroutines called:  None
<table>
<thead>
<tr>
<th>REMACT</th>
<th>LBTPH</th>
<th>MD1</th>
<th>I</th>
<th>A Code Defining Type of Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 1 for Slack Bus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 0 for PQ Bus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 2 for PV Bus</td>
</tr>
<tr>
<td>REMACT</td>
<td>LBEQM</td>
<td>MD1</td>
<td>I</td>
<td>A Code = -1 Bus not to be Considered in Security Analysis When Different Than -1, Equals The Area Number</td>
</tr>
<tr>
<td>REMACT</td>
<td>LBSNM</td>
<td>MD1</td>
<td>I</td>
<td>Bus Symbolic Number</td>
</tr>
<tr>
<td>REMACT</td>
<td>VKVM</td>
<td>MD1</td>
<td>R</td>
<td>Bus Nominal Voltage (in KV)</td>
</tr>
<tr>
<td>REMACT</td>
<td>VMINH</td>
<td>MD1</td>
<td>R</td>
<td>Bus Low Voltage Limit (in pu)</td>
</tr>
<tr>
<td>REMACT</td>
<td>VMAXH</td>
<td>MD1</td>
<td>R</td>
<td>Bus High Voltage Limit (in pu)</td>
</tr>
<tr>
<td>REMACT</td>
<td>LCDM</td>
<td>MD2</td>
<td>I</td>
<td>Circuit Type Identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 1 for Transmission Lines</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 2 for Transformers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 3 for a DC Link</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 4 for jumbers (jumbers are identified with susceptance less than -9999)</td>
</tr>
<tr>
<td>REMACT</td>
<td>XKVLM</td>
<td>MD2</td>
<td>R</td>
<td>Circuit Nominal Voltage (in KV)</td>
</tr>
<tr>
<td>REMACT</td>
<td>LCEQM</td>
<td>MD2</td>
<td>I</td>
<td>A Code: Equals the Area Number (na) to Which Circuit Belongs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= na Circuit Resulting from Equivalent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= na Actual Circuit</td>
</tr>
<tr>
<td>REMACT</td>
<td>LSEPIM</td>
<td>MD2</td>
<td>I</td>
<td>Circuit Sending Bus Number</td>
</tr>
<tr>
<td>REMACT</td>
<td>LREPIM</td>
<td>MD2</td>
<td>I</td>
<td>Circuit Receiving Bus Number</td>
</tr>
<tr>
<td>REMACT</td>
<td>LMETH</td>
<td>MD2</td>
<td>I</td>
<td>Meter Side Code</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 1 meter at sending end,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= -1 meter at receiving end</td>
</tr>
<tr>
<td>REMACT</td>
<td>GLPM</td>
<td>MD2</td>
<td>R</td>
<td>Circuit Conductance (in pu)</td>
</tr>
<tr>
<td>REMACT</td>
<td>BLPM</td>
<td>MD2</td>
<td>R</td>
<td>Circuit susceptance (in pu)</td>
</tr>
<tr>
<td>REMACT</td>
<td>SLPM</td>
<td>MD2</td>
<td>R</td>
<td>Circuit pi-Equivalent Shunt Susceptance (in pu) (note half of line)</td>
</tr>
<tr>
<td>REMACT</td>
<td>TRM</td>
<td>MD2</td>
<td>R</td>
<td>Working Circuit Rating (in pu)</td>
</tr>
</tbody>
</table>
REMACT TRAM MD2 R Circuit Normal Rating (in mva)
REMACT TRBM MD2 R Circuit 1 Hour Rating (in mva)
REMACT TRCM MD2 R Circuit 15 Minute Rating (in mva)
REMACT NPARH MD2 I Count Number of Parallel Circuits
REMACT IPTRM MD2 I Pointer Array of Transformers, IPTRM(I) = J
  I : Internal Numbering of Circuits
  J : Internal Numbering of Transformers
  J = 0 if LCDH(I) is different than 2
REMACT ITRPH MD11 I Inverse Pointer Array for Transformers, ITRPH(J) = I
  J : Internal Numbering of Transformer
  I : Internal Numbering of Circuit
REMACT ITPSM MD11 I Pointer of Phase Shifter
  = 0 Points Count Number of Phase Shifter
  = 0 Nonphase Shifter
REMACT TTLM MD11 R Transformer Low Tap Setting
REMACT TTHM MD11 R Transformer High Tap Setting
REMACT TTM MD11 R Transformer Tap Setting
REMACT PWHMN MD11 R Transformer Low Phase Shift Setting (in radians)
REMACT PHDMN MD11 R Transformer High Phase Shift Setting (in radians)
REMACT TTPHM MD11 R Transformer Phase Shift (in radians)
REMACT IGORDM MD7 I Inverse Pointer Array of Generation Buses, IGORDM(I) = J
  J : Generating Bus Internal Number
  I : Internal Number of Generation Buses
REMACT LGEDM MD7 I A Code = -1 Generation Bus not to be Considered in Security Analysis
  = 0 Generation Bus to be Considered
REMACT VBSPM MD7 R Specified Voltage of Bus, in p.u.
REMACT PGTLM MD7 R Total Bus Real Generation, in p.u.
REMACT QGTLN M07 R Total Bus Reactive Generation, in p.u.
REMACT PGM00 M07 R Maximum MM Capability, in p.u.
REMACT PGMM M07 R Minimum MM Capability, in p.u.
REMACT QGMM M07 R Maximum MVAR Capability, in p.u.
REMACT QGMM M07 R Minimum MVAR Capability, in p.u.
REMACT ILBN M010 I Pointer Array of Constant P.F. Load Buses
REMACT PLOM M010 R Load Real Power, in p.u.
REMACT QLOM M010 R Load Reactive Power, in p.u.
REMACT ICILN M010 I Pointer Array of Constant Impedance Load Buses (internal numbering)
REMACT GCILN M010 R Conductance of Constant Impedance Load Buses, in p.u.
REMACT BCILN M010 R Susceptance of Constant Impedance Load Buses, in p.u.
REMACT ICAPN M010 I Pointer Array of Buses at which Capacitors are Connected
REMACT XCAPN M010 R Capacitive Admittance Connected to Bus, in p.u. (positive value)
REMACT CAPSTN M010 R Discrete Step Size of Capacitor, in p.u.
REMACT IIDUN M010 I Pointer Array of Buses at which Reactors are Connected
REMACT XIDUN M010 R Inductive Admittance Connected to Bus, in p.u. (Positive Value)
REMACT XIOMNN M010 R Minimum-Maximum Value of Inductive Admittance, in p.u. (Positive Value)
REMACT XIOSTN M010 R Discrete Step Size of Inductive Admittance, in p.u. (Positive Value)
REMACT MITER # I Maximum number of iterations allowed in the loadflow (working)
REMACT PEPSLN # R Real power mismatch criterion in the load flow (working)
REMACT QEPSLN # R Reactive power mismatch criterion in the load flow (working)
REMACT DCEPSL # R DC link equation mismatch criterion in the loadflow (working)
REMACT KODESL # I A code = 1 AC Power Flow Solution
              = 2 DC Power Flow Solution
REMACT KLFDIV I A code = 0 Loadflow Has Converged
= 1 Loadflow Has Diverged Because Mismatches Started Increasing
= 2 Loadflow Exceeded Maximum Allowable Number of Iterations

REMACT KFDLF1 I A code = 1 Start from Scratch
= 2 B',B' are Formed
= 3 B',B' are Formed and Triangulated

REMACT KFDLF2 I A code = 1 Use Flat Start
= 2 Use Existing DELTA, VOLTG Values
= 3 Use Existing DELTP, VOLT Values

REMACT KFDLF3 I A code = 1 Normal System State Solution
= 2 Single Circuit Outage Solution
= 3 Double Circuit Outage Solution

REMACT KFDLF4 I A code = 1 Matrices B' and B'' are Formed for the Base Case
= 2 Matrices B' and B'' are Formed for the Outage

REMACT KODEQL I A code = 0 Activate VAR Limits Observation
= 1 Deactivate VAR Limits Observation

REMACT KNTDV1 I Number of iterations that mismatches are allowed to increase before a case is declared as divergence

REMACT MITERB I Maximum number of iterations allowed in the loadflow for base case

REMACT MITERC I Maximum number of iterations allowed in the loadflow for contingency case

REMACT PEPSB R Real power mismatch criterion in the base case

REMACT QEPSB R Reactive power mismatch criterion in the base case

REMACT DCEPSB R DC link equations mismatch in the base case

REMACT PEPS C R Real power mismatch criterion in the contingency case

REMACT QEPS C R Reactive power mismatch criterion in the contingency case

REMACT DCEPS C R DC link equations mismatch in a contingency case

REMACT CONPAR 90 R It contains information about the outage
REMACT CONPAR (cont.)

as follows:

1: A code = 0 base case, neglect entries 2 through 15
   = 1 circuit outage
   = 2 generating unit outage

2: Circuit Outage: Circuit pointer (internal numbering)
   Unit Outage: Bus number (optimal number)

3: Circuit Outage: Circuit series conductance
   Unit Outage: Unit real power output

4: Circuit Outage: Circuit series susceptance
   Unit Outage: Unit reactive power output

5: Circuit Outage: Circuit shunt conductance, sending end
   Unit Outage:
      A Code = 1 bus remains PV bus
      = 2 bus converts to PQ bus

6: Circuit Outage: Circuit shunt susceptance, sending end
   Unit Outage:

7: Circuit Outage: Circuit shunt conductance, receiving end
   Unit Outage:

8: Circuit Outage: Circuit shunt susceptance, receiving end
   Unit Outage:

9: Circuit Outage: Circuit sending and bus number (optimal numbering)

10: Circuit Outage: Circuit receiving and bus number (optimal numbering)

11: Circuit Outage: For phase shifters only: Re(Yo)

12: Circuit Outage: For phase shifters only: Im(Yo)

13: Circuit Outage: Right of way pointer (optimal numbering)

14: Circuit Outage: Pointer of phase shifter, if shifter
It contains the system parameters as follows:
1: NTOTAL - Total number of buses
2: NGTL - Total number of PV buses
3: KTOTAL - Total number of PQ buses plus 1
4: NCIR - Total number of circuits
5: NRON - Total number of right of ways (R.O.W.)
6: NPHSH - Total number of phase shifters

Number of iterations in which a reactive power is violated before bus is changed to PQ
Working number of PV buses plus slack bus
Branch Optimal Ordering Array; IAROM(I) = J
I: Optimal ROM Number
J: Internal ROM Number
Inverse Branch Optimal Ordering Array; IBROM(J) = I
I: Optimal ROM Number
J: Internal ROM Number
Array Giving the Optimal ROM of Each Circuit; ICROM(IC) = I
IC: Internal Circuit Number
I: Optimal ROM Number
Points the Maximum Capacity Circuit on the Right of Way; IMRST(I) = J
I: Optimal ROM
J: Circuit Number (internal numbering)
Bus Optimal Ordering Array; NORDER(I) = J
I: Optimal Bus Number
J: Internal Bus Number
Inverse Optimal Ordering Array; NORDRP(J) = I
I: Optimal Bus Number
J: Internal Bus Number
Ordering Array for PQ Buses; NVORDR(J) = I
I: Optimal Bus Number
J: Position in B" Matrix
Array Giving Position of a Bus in B" Matrix; NBPBDP(I) = J
I: Optimal Bus Number
J: Position in B" Matrix
Bus Type (Optimal Order) = 1 for Slack Bus
= 0 for PQ Bus
= 2 for PV Bus

REMACT VOLTD MD1 R Array of Bus Voltage Magnitudes (Optimal Order)
REMACT DELTAD MD1 R Array of Voltage Phases (Optimal Order)
REMACT GSLFA MD1 R Total Bus Conductance (Optimal Order)
REMACT BSLFA MD1 R Total Bus Susceptance (Optimal Order)
REMACT PINJLF MD1 R Net Real Power Bus Injection (PGi - PLi) (Optimal Order)
REMACT QINJLF MD1 R Net Reactive Power Bus Injection (QLi) (Optimal Order)
REMACT LSELF FA MD2 I Sending Bus (Optimal) of Branches (Optimal Order)
REMACT LRELFA MD2 I Receiving Bus (Optimal) of Branches (Optimal Order)
REMACT GCLFA MD2 R Branch Conductance; in p.u. (Optimal Order)
REMACT BCLFA MD2 R Branch Susceptance; in p.u. (Optimal Order)
REMACT VBLFA MD7 R PV Bus Specified Voltage; in p.u. (Optimal Order)
REMACT QGLFA MD7 R Generated VAR of PV Bus (units only); in p.u.
REMACT QLLFA MD7 R Constant Power Electric Load MVAR of PV Bus; in p.u.
REMACT MINLF MD7 R Minimum MVAR Bus Capability for PV Buses; in p.u.
REMACT MMAXLF MD7 R Maximum MVAR Bus Capability for PV Buses; in p.u.
REMACT IORDGN MD7 I Pointer Array of Generating Buses; IORDGN(IG) = I
Phasor Phase Shifter Model

\[ Y = g + \frac{ja}{E} \]
\[ n = \frac{\sqrt{2}}{E} \]
Y = jb
S S

Y : Admittance Matrix of Phase Shifter

\[
Y = \begin{bmatrix}
Y & Y \\
11 & 12
\end{bmatrix}
\]

\[
Y = \begin{bmatrix}
Y & Y \\
21 & 22
\end{bmatrix}
\]

Y = Y - \frac{2(1-n)}{Y}
\frac{0}{12}
\frac{21}{Z}
\frac{1+1}{1+1}

Description of Array XPHASH

I : Phase Shifter Number

XPHASH (1,1) = Re(Y) = \frac{2}{Z}
\frac{21}{1+1}
\frac{2b}{2}
\frac{1+1}{1+1}
\frac{2t}{Z}

XPHASH (2,1) = Im(Y) = \frac{2b}{Z}
\frac{1+1}{1+1}
\frac{2t}{Z}
\[ XPHASH(3,I) = \text{Re}(Y) = -\frac{1}{1+t} \left( g\cos t + b\sin t \right) \]

\[ XPHASH(4,I) = \text{Im}(Y) = -\frac{2t}{1+t} \left( g\sin t - b\cos t \right) \]

\[ XPHASH(5,I) = \text{Re}(Y) = -\frac{2t}{1+t} \left( g\cos t - b\sin t \right) \]

\[ XPHASH(6,I) = \text{Im}(Y) = -\frac{2t}{1+t} \left( g\sin t + b\cos t \right) \]

\[ XPHASH(7,I) = \text{Re}(Y) = \frac{2}{1+t} g \]

\[ XPHASH(8,I) = \text{Im}(Y) = \frac{2}{1+t} \left( b + \frac{a}{2} \right) \]

\[ XPHASH(9,I) = t \]

\[ XPHASH(10,I) = a \text{ (radians)} \]

\[ XPHASH(11,I) = g \]

\[ XPHASH(12,I) = b \]

\[ XPHASH(13,I) = 0.0 \]

\[ XPHASH(14,I) = b \]

\[ XPHASH(15,I) = \text{Re}(Y) = -\frac{4bt}{1+t} \sin t \]

\[ XPHASH(16,I) = \text{Im}(Y) = \frac{4gt}{1+t} \sin t \]
Description of Array IPHASH

I : Phase Shifter Number

IPHASH (1,I) : Internal Circuit Number
IPHASH (2,I) : Optimal ROM Number
IPHASH (3,I) : Optimal Number of Sending Bus
IPHASH (4,I) : Optimal Number of Receiving Bus
IPHASH (5,I) : A code = 1 Phase Shifter in Service
              = 0 Phase Shifter is in Outage

IPHASH (6,I) : A Pointer: Points the order of the sending bus in the PV bus array in optimal ordering. If sending bus is not a PV bus, then zero.

REMACT IJKT MDJ1 I Pointer of First Nonzero Off Diagonal Elements in a Row of the Jacobian Matrix
REMACT IJKM MDJ2 I Pointer of Column Number for Nonzero Off Diagonal Elements of the Jacobian Matrix
REMACT YJD MDJ1 R Array of Diagonal Elements of Jacobian Matrix
REMACT YJDFD MDJ2 R Array of Off Diagonal Elements of Jacobian Matrix
REMACT IJKT1 MDJ1 I Pointer of First Nonzero Off Diagonal Elements in a Row of the LU Factors of the Jacobian Matrix
REMACT IJKM1 MDJ3 I Pointer of Column Number for Nonzero Off Diagonal Elements of the LU Factors of the Jacobian Matrix
REMACT JACPKN MDJ2 I Pointer of Elements in the YJDFD Array (which is the Array of Non-Diagonal Elements of Jacobian Matrix)
REMACT HJD MDJ1 R Array of Diagonal Elements of the LU Factors of the Jacobian Matrix
REMACT HJDFD MDJ3 R Array of Off Diagonal Elements of the LU Factors of the Jacobian Matrix
REMACT IJDPOS MAXBUS I Delta Variable Position in Jacobian Matrix Array; IJDPOS(I) = J
          I : Bus Number in Optimal Ordering
          J : Position of w_i in Jacobian
REMACT IJVPOS MAXBUS I Voltage Variable Position in Jacobian Matrix Array; IJVPOS(I) = J
          I : Bus Number in Optimal Ordering
J : Position of Vi in Jacobian matrix; J = 0 when Bus is PV

REMACT IJDOPS MAXBUS I Pointer of Location of the Term MQ/MW
in the Off Diagonal Terms of the Jacobian Matrix; IJDOPS(I) = J
I : Bus Number (optimal)
J : Entry in the Off Diagonal Array

REMACT IJVVP MAXBUS I Pointer of Location of the Term MP/MV
in the Off Diagonal Terms of the Jacobian Matrix; IJVVP(I) = J
I : Bus Number (optimal)
J : Entry in the Off Diagonal Array of the Jacobian (zero if it does not exist)

REMACT JACPDP MD14 I Pointer of Location of the Off Diagonal Terms MP/MW in the Jacobian Matrix; JACPDP(J) = L
J : Pointer of Off Diagonal Term in the Bus Topology Matrix (Array ITKH)
L : Pointer of Off Diagonal Term in Jacobian Matrix

REMACT JACPVP MD14 I Same as for JACPDP for the Terms MP/MV
REMACT JACQDP MD14 I Same as for JACPDP for the Terms MQ/MJ
REMACT JACQVP MD14 I Same as for JACPDP for the Terms MQ/MV
REMACT ITKT MD1 I Pointer of First Off-Diagonal Entry of Matrix B'
REMACT YD MAXBUS R Diagonal Elements of Matrix B'
REMACT ITKT1 MAXBUS I Pointer of First Off-Diagonal Entry in the Table of Factors of Matrix B''
REMACT YD-HD MAXBUS R Diagonal Elements of the Table of Factors of Matrix B'
REMACT IVKT MAXBUS I Pointer of First Off-Diagonal Entry of
REMACT YVD MAXBUS R Diagonal Elements of Matrix B''
REMACT IVKT1 MAXBUS I Pointer of First Off-Diagonal Entry in the Table of Factors of Matrix B''
REMACT YVD MAXBUS R Diagonal Elements of the Table of Factors of Matrix B''
| REMACT | ITKM | MD14 | I | Pointer of Off-Diagonal Elements of Matrix $B'$ |
| REMACT | YOFFD | MD14 | R | Off-Diagonal Elements of Matrix $B'$ |
| REMACT | IVKM | MD14 | I | Pointer of Off-Diagonal Elements of Matrix $B''$ |
| REMACT | YVOFD | MD14 | R | Off-Diagonal Elements of Matrix $B''$ |
| REMACT | DТОСRM | MD14 | I | Pointer of Right of Ways |
| REMACT | ITKM1 | MD3 | I | Pointer of Off-Diagonal Elements in the Table of Factors of Matrix $B'$ |
| REMACT | IVKM1 | MD3 | I | Pointer of Off-Diagonal Elements in the Table of Factors of Matrix $B''$ |
| REMACT | NOFD | MD3 | R | Off-Diagonal Elements of the Table of Factors of $B'$ |
| REMACT | HVOFD | MD3 | R | Off-Diagonal Elements of the Table of Factors of $B''$ |
| REMACT | SABFLG | MAXBUS | L | Flag for defining buses belonging to the study area. SABFLG(IBI) = .T. bus IBI belongs to study area. IBI is internal numbering |
| REMACT | SACFLG | MAXCIR | L | Flag for defining circuits belonging to the study area SACFLG(ICI) = .T. circuit ICI belongs to study area. |
| REMACT | RABFLG | MAXBUS | L | Flag for defining buses belonging to the remedial actions area RABFLG(IBI) = .T. Bus IBI belongs to the remedial actions area. |
| REMACT | RACFLG | MAXCIR | L | Flag for defining circuits belonging to the remedial actions area RACFLG(ICI) = .T. Circuit ICI belongs to remedial actions area. |
| REMACT | MVCOLD | N | I | Old Number of Voltage Operating Constraints |
| REMACT | NTVOC | N | I | Current Number of Voltage Operating Constraints |
| REMACT | KQVOLD | N | I | Old Number of Voltage Operating Constraints included in LP |
| REMACT | NPCOLD | N | I | Old Number of Circuit Flow Operating Constraints |
| REMACT | NTFOC | N | I | Current Number of Circuit Flow Operating Constraints |
| REMACT | KOFOLD | N | I | Old Number of Circuit Flow Operating Constraints included in LP |
| REMACT | NQCOLD | N | I | Old Number of Reactive Power Operating Constraints |
| REMACT | NTQOC | N | I | Current Number of Reactive Power Operating Constraints = Total Number of reactive power violations |
| REMACT | KQQOLD | N | I | Old Number of Reactive Power Constraints included in LP |
| REMACT | IPVOC | NDVOC | I | Pointer Array of Buses Violating the Voltage Limits (Optimal Ordering) |
REMACT  DVOC  NOVOC  R  Voltage violations in p.u.
  = + Actual Violation
  = - Near Violation

REMACT  ICVOC  NOVOC  I  A Code = 1 Upper Limit is Violated
  = -1 Lower Limit is Violated

REMACT  ICHVOC  NOVOC  I  Pointer Array of Coherent Voltage Violations
  = + Leading Violation
  = - Follower Violation

REMACT  IPSVOC  NOVOC  I  Pointer of Voltage Constraint Indicating the Row in Matrix SNSAUG to Store Linearized Constraint
  If = 0, no voltage constraint is included in the LP model.

FILECT  IEVOC  NOVOC  I  A Code = 1 Voltage Constraint is Effective in L.P.
  = 0 Voltage Constraint is Not Effective in L.P.

REMACT  DCVOC  NOVOC  R  Old Value of Voltage Violations in p.u.

REMACT  ILVOC  NOVOC  I  A Code = 1 Voltage Constraint Needs to be Linearized
  = 0 Otherwise

REMACT  DTVOC  NOVOC  R  Total Correction of Voltage Constraints

REMACT  DITVOC  NOVOC  R  Total Incremental Correction of Voltage Constraints Between Linearizations

REMACT  IPFOC  MAXFCN  I  Pointer Array of Circuits Violating the MVA Flow Limits (Internal Ordering)

REMACT  ICFOC  NOFOC  I  A Code = 1 Flow is Positive (DC Model)
  = -1 Flow is Negative (DC Model)

REMACT  DPFOC  NOFOC  R  Flow Violations in p.u.
  = + Actual Violation
  = - Near Violation

REMACT  ICHFDC  NOFOC  I  Pointer Array of Coherent Flow Violations
  = + Leading Violation
  = - Follower Violation

REMACT  IPSFOC  NOFOC  I  Pointer Array of Flow Constraints Indicating Row of Matrix SNSAUG

REMACT  IEFOC  NOFOC  I  A Code = 1 Flow Constraint is Effective in L.P.
  = 0 Otherwise

REMACT  DCFOC  NOFOC  R  Old Value of Flow Violations in p.u.

REMACT  ILFOC  NOFOC  I  A Code = 1 Flow Constraint Needs to be Re-linearized
  = 0 Otherwise
REMACT DTFOC NDFOC R Total Correction of Flow Constraints
REMACT DIDFOC NDFOC R Total Incremental Normalized (w.r.t. the Rated Flow) Correction of Flow Constraints Between Linearizations
REMACT NTQOC # I Total Number of Reactive Power Violations
REMACT IPQOC NDQOC I Pointer Array of Generation Buses Violating the Reactive Generating Limits (Optimal Numbering)
REMACT DQQOC NDQOC R Reactive Generation Violations in p.u.
  = + Actual Violation
  = - Near Violation
REMACT ICQOC NDQOC I A Code = 1 Upper Limit is Violated
  = - Lower Limit is Violated
REMACT ICHQOC NDQOC I Pointer Array of Coherent Reactive Generating Violations
  = + Leading Violation
  = - Follower Violation
REMACT IPSQOC NDQOC I Pointer Array of Reactive Constraints Indicating Row of Matrix SNSAUG
REMACT IEQOC NDQOC I A Code = 1 Reactive Constraint is Effective in L.P.
  = 0 Otherwise
REMACT DCQOC NDQOC R Old Value of Reactive Violations in p.u.
REMACT ILQOC NDQOC I A Code = 1 Reactive Constraint Needs to be Relinearized
  = 0 Otherwise
REMACT DTQOC NDQOC R Total Correction of Reactive Constraints
REMACT DIDQOC NDQOC R Total Incremental Correction of Reactive Constraints Between Linearizations
REMACT NTXOC # I Total Number of Interchange Constraints (areas) = NIXG
REMACT PSXOC NDXOC R Scheduled Net MN Export in p.u.
REMACT DPXOC NDXOC R Net MN Export Deviation in p.u.
REMACT ICHXOC NDXOC I Idle
REMACT IPSXOC NDXOC I Pointer Array of Interchange Constraint Indicating Row of Matrix SNSAUG
REMACT IEXOC NDXOC I A Code = 1 Interchange Constraint is Effective in L.P.
  = 0 Otherwise
REMACT DCXOC NDXOC R Old Value of MN Export Deviation in p.u.
REMACT ILXOC NDXOC I A code = 1 Constraints Need to be
Relinearized
= 0 Otherwise
REMACT DTXOC NDXOC R Total Correction of Net MW Exports in p.u.
REMACT DITXOC NDXOC R Total Incremental Correction of Net MW Export Between Linearizations in p.u.
REMACT NPXOC NDXOC I Number of Tie Lines in Each Interchange Constraint
REMACT JPOXC NDXOC I Array Indicating the Position of the Last Tie Line for the Areas
REMACT IPXOC NCXOC1 I Array Indicating the Circuit Number (internal numbering) of the Tie Lines
+ meter is on the sending end
- meter is on the receiving end
REMACT IFXOC NCXOC1 I Array Indicating the Direction of Flow
+1 Power Flow is Outwards
-1 Otherwise
REMACT NPOPCC # I Pointer of Circuit Constraint to be Linearized (Internal Numbering)
REMACT NPOPCV # I Pointer of Bus Voltage Constraint to be Linearized (Optimal Numbering)
REMACT NPOPCQ # I Pointer of Bus Reactive Constraint to be Linearized (Optimal Numbering)
REMACT NPOPCP # I Pointer of Bus Real Constraint to be Linearized (Optimal Numbering)
REMACT NPOPCX # I Pointer of Interchange Constraint to be Linearized
REMACT NPOPCL # I A Code
= 1 Linearize-Bussc losses
= 0 Do Not Linearize
REMACT NPOPCB # I A Code
= 1 Linearize Power Balance
Equation
= 0 Do Not Linearize
REMACT NSOPT # I A Code
= 1 Linearize Circuit Apparent
Flow (MVA)
= 2 Linearize Voltage
= 3 Linearize Reactive Generation
= 4 Linearize Real Generation
= 5 Linearize Losses
= 6 Linearize Power Balance
Equation
= 7 Linearize Circuit Real Flow
(MW)
= 8 Linearize Interchange
Constraint
REMACT JTOTAL # I Number of Q-V Equations in the Jacobian Matrix for Sensitivity Analysis (Slack Bus
| REMACT | NVPH   | *   | I   | Total number of power mismatch control variables in optimal power flow formulation. At present, there is only one which is \( V \).
| REMACT | VPHANT | *   | R   | Power mismatch control variable \( V \) in OPF formulation. Zero value of \( V \) means power flow has converged.
| REMACT | PPHANT | MAXBUS | R | Real power mismatches at each iteration of OPF solution, which are also used as the distribution factors of \( V \).
| REMACT | QPHANT | MAXBUS | R | Reactive power mismatches at each iteration of OPF solution, which are also used as the distribution factors.
| REMACT | STPMIS | MAXBUS | R | Allowable step in each iteration of changing the mismatch control variable \( V \).
| REMACT | NWGN   | *   | I   | Total number of incremental real generation control variables in optimal power flow formulation. At the time being, there are two of such controls, \( W_1 \) and \( W_2 \) for increasing and decreasing respectively.
| REMACT | W1GENP | *   | R   | Total real generation increased in one iteration in OPF.
| REMACT | W1PART | MAXGEN | R | Economic participation factor for each generation bus, working with \( W_1 \) to increase generation output.
| REMACT | W2GENP | *   | R   | Total real generation decreased in one iteration in OPF.
| REMACT | W2PART | MAXGEN | R | Distribution factor for each generation bus to download.
<table>
<thead>
<tr>
<th>REJACT</th>
<th>PRTFLG</th>
<th>*</th>
<th>LOGICAL</th>
<th>A flag when equal to 'TRUE', the economic participation factors for generating units are normalized in a meaningful way. If equal to 'FALSE', system slack bus has been changed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>REJACT</td>
<td>SLCFLG</td>
<td>*</td>
<td>LOGICAL</td>
<td>A flag = 'TRUE', system slack bus has been changed. = 'FALSE', system slack remains the same.</td>
</tr>
<tr>
<td>REJACT</td>
<td>OPFFLG</td>
<td>*</td>
<td>I</td>
<td>A code, = 1, program in OPF option. = 2, program in conventional power flow with load specified outcomes.</td>
</tr>
<tr>
<td>REJACT</td>
<td>WAPART</td>
<td>MAXUNT</td>
<td>R</td>
<td>Economic participation factors of units allowable to increase output.</td>
</tr>
<tr>
<td>REJACT</td>
<td>WBPART</td>
<td>MAXUNT</td>
<td>R</td>
<td>Economic participation factors of units allowable to decrease output.</td>
</tr>
<tr>
<td>REJACT</td>
<td>PFUGRP</td>
<td>MAXUNT</td>
<td>R</td>
<td>Unit participation factors for remedial action purpose only.</td>
</tr>
<tr>
<td>REJACT</td>
<td>PVERP</td>
<td>MAXGBS</td>
<td>R</td>
<td>Present bus real power output generated by economically participated units at one bus.</td>
</tr>
<tr>
<td>REJACT</td>
<td>XMWERP</td>
<td>MAXGBS</td>
<td>R</td>
<td>Bus maximum real output by economically participated units.</td>
</tr>
<tr>
<td>REJACT</td>
<td>XMNERP</td>
<td>MAXGBS</td>
<td>R</td>
<td>Bus minimum real output by economically participated units.</td>
</tr>
<tr>
<td>REJACT</td>
<td>STPERP</td>
<td>MAXGBS</td>
<td>R</td>
<td>Allows to step in each iteration of changing the economically dispatched real generation at a bus.</td>
</tr>
</tbody>
</table>
Always Included:

REMACT KSLACK = 1 I A Code = 1 If Not Sensitivity wrt, Slack Bus Voltage is Required = 0 Otherwise

REMACT NTCNTR = I Total Number of Controls
REMACT MXLPCU = P Declared Dimension of Control Variables
REMACT MXLPCN = P Declared Dimension of Constraints
REMACT VQFLAG = L A logical variable for activation of the PV/PQ Switching Logic = .true. logic is ON.

REMACT VQCODE = I A working code for determining changes of PV/PQ assignments.

REMACT NGRP --- I Total Number of Buses with Generating Real Power Control
REMACT IPGRP MAXGBS I Pointer Array of Buses at which Generating Real Power Output is Controlled (Optimal Ordering), IPGRP(IB) = IBO.
REMACT IGPRGB MAXGBS I Pointer of generation control index form generating bus index, IGPRGB(IGB) = IG
REMACT IGBGRP MAXGBS I Pointer of generation bus from Generation Control Bus. IGBGRP (IG) = IGB
REMACT PVGRP MAXGBS R Present Bus Generating Real Power Values; in p.u.
REMACT XMXGRP MAXGBS R Max P generation
REMACT XMNGRP MAXGBS R Min P generation
REMACT STPGRP MAXGBS R Allowable Step in Each Iteration of Changing the Generating Real Power; in p.u.
REMACT TCRGRP MAXGBS R Total Change of Real Power Generation at the Bus; in p.u.
REMACT KCSLC --- I Points Slack Bus Order in Above Arrays
REMACT NGSR --- I Total number of buses with pg control
REMACT IPGSR MAXGBS I Pointer of optimal bus number IBO
REMACT IGSRINK MAXGBS I Pointer of sink bus number (optimal numbering)
REMACT IGBGSR MAXGBS I Pointer of generating bus index IGB
REMACT PVGSR MAXGBS R Present bus real generation
REMACT XMGSR MAXGBS R Max P generation
REMACT XMNGSR MAXGBS R Min P generation
REMACT STPGSR MAXGBS R Allowable step in each iteration
| REMACT   | TCRGSR   | MAXGBS | R | Total change of P generation |
| REMACT   | SLACKP   | ---    | I | Index of slack bus          |
| REMACT   | NCVG     | ---    | I | Total Number of Buses with Voltage Control |
| REMACT   | IPCVG    | MAXGBS | I | Pointer Array of Buses at which Voltage is Controlled (Optimal Ordering) |
| REMACT   | IGBCVG   | MAXGBS | I | Pointer of generation bus from voltage control bus. IGBCVG!IVU) = IGB |
| REMACT   | PVCVG    | MAXGBS | R | Present Bus Voltage Values; in p.u. |
| REMACT   | XMXCVG   | MAXGBS | R | Bus Maximum Voltage Limit; in p.u. |
| REMACT   | XMNCVG   | MAXGBS | R | Bus Minimum Voltage Limit; in p.u. |
| REMACT   | STPCVG   | MAXGBS | R | Allowable Step in Each Iteration of Changing the Voltage at this Bus; in p.u. |
| REMACT   | TCRCVG   | MAXGBS | R | Total Correction of Voltage at this Bus; in p.u. |
| REMACT   | NCVG     | ---    | I | Total number of buses with control |
| REMACT   | IPQVO    | MAXGBS | I | Same as IPCVG |
| REMACT   | IGSTAT   | MAXGBS | I | A code = 1 control variable is V, =-1 control variable is Q at this bus. |
| REMACT   | PVQVG    | MAXGBS | R | Present bus Q generation (in pu) |
| REMACT   | XMXQVG   | MAXGBS | R | Max Q limit (in pu) |
| REMACT   | XMNQVG   | MAXGBS | R | Min Q limit (in pu) |
| REMACT   | STPQVG   | MAXGBS | R | Allowable Q step (in pu) |
| REMACT   | TCRQVG   | MAXGBS | R | Total Q correction (in pu) |
| REMACT   | NTTC     | ---    | I | Total Number of Transformers with Tap Control |
| REMACT   | IPTTC    | MAXTAP | I | Pointer of Transformers of which the Tap is Controlled |
| REMACT   | PVTTC    | MAXTAP | R | Present Transformer Tap Values |
| REMACT   | XHOTTTC  | MAXTAP | R | Transformer High Tap Setting |
| REMACT   | XMNTTC   | MAXTAP | R | Transformer Low Tap Setting |
| REMACT   | STPTTC   | MAXTAP | R | Allowable Step in Each Iteration of Changing the Transformer Tap |
| REMACT   | TCRTTTC  | MAXTAP | R | Total Correction of Transformer Tap |
| SYRLE    | NTTPS    | ---    | I | Total Number of Transformers with Phase Control |
REMACT IPTPS MAXPHS I Pointer of Transformer of which the Phase Shift is Controlled
REMACT PVTPS MAXPHS R Present Transformer Phase Shift Values; in radians
REMACT XMHTPS MAXPHS R Transformer High Phase Shift Settings; in radians
REMACT XMNTPS MAXPHS R Transformer Low Phase Shift Setting; in radians
REMACT STPTPS MAXPHS R Allowable Step in Each Iteration of Changing the Transformer Phase Shift; in radians
REMACT TCRTPS MAXPHS R Total Correction of Transformer Phase Shift; in radians
REMACT NSCB --- I Total Number of Buses with Switchable Capacitors
REMACT IPSCB MAXCAP I Pointer Array of Buses at which Capacitor Banks can be Switched (Optimal Ordering)
REMACT IPAICB MAXCAP I Pointer of switchable capacitor banks. IPDSCB(IUC) = ICAP
REMACT PVSCB MAXCAP R Present Bus Capacitive Susceptance Values; in p.u. (Positive Number)
REMACT XMSCB MAXCAP R Bus Maximum Capacitive Susceptance; in p.u.
REMACT XMNSCE) MAXCAP R Bus Minimum Capacitive Susceptance; in p.u.
REMACT STPSCB MAXCAP R Allowable Step in Each Iteration of Changing the Capacitive Susceptance; in p.u.
REMACT TCRSCB MAXCAP R Total Switched Capacitive Susceptance; in p.u.
REMACT NSRB --- I Total Number of Buses with Switchable Reactors
REMACT IPSRB MAXIND I Pointer Array of Buses at which Reactor Banks can be Switched (Optimal Ordering)
REMACT IPDSRB MAXIND I Pointer of switchable inductor banks IPDSCB(IUC) = ICAP
REMACT PVSRB MAXIND R Present Bus Inductive Susceptance values; in p.u. (Positive Number)
REMACT XMMSRB MAXIND R Bus Maximus Inductive Susceptance; in p.u.
REMACT XMNSRB MAXIND R Bus minimum inductive susceptance; in p.u.
REMACT STPSRB MAXIND R Allowable Step in Each Iteration of Changing the Inductive Susceptances in p.u.
REMACT TCRSRB MAXIND R Total Switched Inductive Susceptance; in p.u.
REMACT NRLT --- I Total Number of Buses with Transferrable Load
REMACT IP1RLT MAXLDX I Pointer Array of "From Bus" that Load can be Transferred (Optimal Ordering)
REMACT IPD1LT MAXLDX I Pointer of transferred load. Points the index at the 'From bus'.
REMACT IP2RLT MAXLDX I Pointer Array of "To Bus" for the Load Transfer (Optimal Ordering)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPD2LT</td>
<td>I</td>
<td>Pointer of transferred load. Points the load index at the 'To bus'.</td>
</tr>
<tr>
<td>PVRLT</td>
<td>MAXLDX</td>
<td>Present Bus Real Load Transfer Values; in p.u.</td>
</tr>
<tr>
<td>XNRLT</td>
<td>MAXLDX</td>
<td>Bus Maximum Real Transferable Load; in p.u.</td>
</tr>
<tr>
<td>XNRLT</td>
<td>MAXLDX</td>
<td>Bus Minimum Real Transferable Load; in p.u. (Presently Set to Zero)</td>
</tr>
<tr>
<td>STPRLT</td>
<td>MAXLDX</td>
<td>Allowable Step in Each Iteration of Changing the Real Load Transfer; in p.u.</td>
</tr>
<tr>
<td>XPFRIL</td>
<td>MAXLDX</td>
<td>Power Factor of Transferable Load (MVAR/MM)</td>
</tr>
<tr>
<td>XMXRLT</td>
<td>MAXLDX</td>
<td>Bus Maximum Real Transferable Load; in p.u.</td>
</tr>
<tr>
<td>XMNRLT</td>
<td>MAXLDX</td>
<td>Bus Minimum Real Transferable Load; in p.u.</td>
</tr>
<tr>
<td>STPRLT</td>
<td>MAXLDX</td>
<td>Allowable Step in Each Iteration of Changing the Real Load Transfer; in p.u.</td>
</tr>
<tr>
<td>XPFRLT</td>
<td>MAXLDX</td>
<td>Power Factor of Transferable Load (MVAR/MM)</td>
</tr>
<tr>
<td>TCRRLT</td>
<td>MAXLDX</td>
<td>Total Load Transferred, in p.u.</td>
</tr>
<tr>
<td>NILS</td>
<td>---</td>
<td>Total Number of Buses with Interruptible Load</td>
</tr>
<tr>
<td>IPILS</td>
<td>MAXLOD</td>
<td>Pointer Array of Buses at which Interruptible Load can be Shedded (Optimal Ordering)</td>
</tr>
<tr>
<td>IPDLILS</td>
<td>MAXLOD</td>
<td>Pointer of interruptible load from load buses. IPDLILS (ILL) = IILS</td>
</tr>
<tr>
<td>PVILS</td>
<td>MAXLOD</td>
<td>Present Value of Bus Interruptible Load; in p.u.</td>
</tr>
<tr>
<td>XNILS</td>
<td>MAXLOD</td>
<td>Bus Maximum Interruptible Load; in p.u.</td>
</tr>
<tr>
<td>XNIFL</td>
<td>MAXLOD</td>
<td>Bus Minimum Interruptible Load; in p.u.</td>
</tr>
<tr>
<td>STPIILS</td>
<td>MAXLOD</td>
<td>Allowable Step in Each Iteration of Changing the Interruptible Load; in p.u.</td>
</tr>
<tr>
<td>XPFIILS</td>
<td>MAXLOD</td>
<td>Power Factor of Interruptible Load (MVAR/MM)</td>
</tr>
<tr>
<td>TCRFLS</td>
<td>MAXLOD</td>
<td>Total Load Interrupted; in p.u.</td>
</tr>
<tr>
<td>NFLS</td>
<td>---</td>
<td>Total Number of Buses with Firm Load</td>
</tr>
<tr>
<td>IPFLS</td>
<td>MAXLOD</td>
<td>Pointer Array of Buses at which Firm Load (Constant P.F.) can be Shedded (Optimal Ordering)</td>
</tr>
<tr>
<td>IPDFLS</td>
<td>MAXLOD</td>
<td>Pointer of firm load buses. IPDFLS (IFU) = IFLS</td>
</tr>
<tr>
<td>PVFLS</td>
<td>MAXLOD</td>
<td>Present Value of Firm Load; in p.u.</td>
</tr>
<tr>
<td>XNDFLS</td>
<td>MAXLOD</td>
<td>Bus Maximum Firm Load; in p.u.</td>
</tr>
<tr>
<td>XNIFLS</td>
<td>MAXLOD</td>
<td>Bus Minimum Firm Load Value; in p.u.</td>
</tr>
<tr>
<td>XPFFLS</td>
<td>MAXLOD</td>
<td>Power Factor of Firm Load (MVAR/MM)</td>
</tr>
<tr>
<td>STPFILS</td>
<td>MAXLOD</td>
<td>Allowable Step in Each Iteration of Changing the Firm Load; in p.u.</td>
</tr>
<tr>
<td>TCRFLS</td>
<td>MAXLOD</td>
<td>Total Firm Load Shed; in p.u.</td>
</tr>
<tr>
<td>REMACT</td>
<td>NCLS</td>
<td>I</td>
</tr>
<tr>
<td>REMACT</td>
<td>IPCLS</td>
<td>MAXLOD</td>
</tr>
<tr>
<td>REMACT</td>
<td>IPDCLS</td>
<td>MAXLOD</td>
</tr>
<tr>
<td>REMACT</td>
<td>PVCLS</td>
<td>MAXLOD</td>
</tr>
<tr>
<td>REMACT</td>
<td>XODCLS</td>
<td>MAXLOD</td>
</tr>
<tr>
<td>REMACT</td>
<td>XPNCLS</td>
<td>MAXLOD</td>
</tr>
<tr>
<td>REMACT</td>
<td>XPFCLS</td>
<td>MAXLOD</td>
</tr>
<tr>
<td>REMACT</td>
<td>STPCLS</td>
<td>MAXLOD</td>
</tr>
<tr>
<td>REMACT</td>
<td>TCRCLS</td>
<td>MAXLOD</td>
</tr>
<tr>
<td>REMACT</td>
<td>NIXG</td>
<td>I</td>
</tr>
<tr>
<td>REMACT</td>
<td>NPIXG</td>
<td>MAXARE</td>
</tr>
<tr>
<td>REMACT</td>
<td>JPIXG</td>
<td>MAXARE</td>
</tr>
<tr>
<td>REMACT</td>
<td>PVIXG</td>
<td>MAXARE</td>
</tr>
<tr>
<td>REMACT</td>
<td>STPIXG</td>
<td>MAXARE</td>
</tr>
<tr>
<td>REMACT</td>
<td>TCRIXG</td>
<td>MAXARE</td>
</tr>
<tr>
<td>REMACT</td>
<td>J8P/XO</td>
<td>MAXARE</td>
</tr>
<tr>
<td>REMACT</td>
<td>PWIXG</td>
<td>MAX10A</td>
</tr>
<tr>
<td>REMACT</td>
<td>PRTIXG</td>
<td>MAX10A</td>
</tr>
<tr>
<td>REMACT</td>
<td>XM1XG</td>
<td>MAX10A</td>
</tr>
<tr>
<td>REMACT</td>
<td>XM2XG</td>
<td>MAX10A</td>
</tr>
<tr>
<td>REMACT</td>
<td>LPC900</td>
<td>=</td>
</tr>
<tr>
<td>REMACT</td>
<td>LPC903</td>
<td>=</td>
</tr>
</tbody>
</table>
A Code = 0, indicates problem is feasible
   = 1, indicates problem is infeasible

A Code determines the LP problem when
   = 1, primal feasible and dual feasible
   = 2, primal infeasible and dual feasible
   = 3, primal feasible and dual infeasible
   = 4, primal infeasible and dual infeasible

= KTLPV + KTLPC, i.e. the total number of both primal and dual variables
= KTLPV + 2 * KTLPC
Working variable for vector b in LP
Working array for cost vector C in LP
Working array for the upper bound vector in LP
The length of one segment for the discrete type variables (normalization factor)
Working variable defining the priority
Defines the debugging level
   = 1, first level, algorithm check only
   = 2, second level, algorithm check and listing
   = 0, no debug report from LP SOLVER
= 10^5
= 10^6
Number of Constraints in LP formulation
Number of Variables in LP formulation (excluding slack and artificial variables)
Declared Dimension of Variables in LP
Declared Dimension for the Off Diagonal Terms in the Constraints Matrix A
Declared Dimension of the Size of the Basic Matrix
Declared Dimension of the Off Diagonal Terms in the Basic Matrix or Its Table of Factors
REMACT MCLMS # I Total Number of Columns in Sensitivity Matrix (SNSAUG) Expressed in Terms of Control Variables

REMACT MRP # I Working pointer of constraints (new or old) to be inserted in the sensitivity matrix SNSAUG

REMACT MROHS # I Total Number of Rows in Sensitivity Matrix SNSAUG

REMACT MDSN1 # P Declared Dimension of Rows in Sensitivity Matrix SNSAUG

REMACT MDSN2 # P Declared Dimension of Nonzero Entries in Sensitivity Matrix SNSAUG

REMACT ISAKT MDSN1 I Pointer of First Nonzero Elements (Per Row) in Matrix SNSAUG

REMACT ISAKM MDSN2 I Pointer of Column for Nonzero Elements in Matrix SNSAUG

REMACT SNSAUG MDSN2 R Array Containing the Nonzero Elements of the Sensitivity Matrix Expressed in Terms of Control Variables (" SNSAUG"

REMACT ICOLP MDSN1 I A Code = 1 if Row in Matrix SNSAUG Violates Upper Limit

REMACT IAKT MDLP1 I Pointer of First Nonzero Element at each row in Constraint Matrix A^T

REMACT IAKM MDLP2 I Pointer of Column for Nonzero Elements in Constraint Matrix A^T

REMACT YOFA MDLP2 R Array Containing the Nonzero Elements of the Constraint Matrix A (Ax = b)

REMACT BVLP MDLPB1 R b Vector of the Linear Program

REMACT NDCNP MDLPB1 R Vector of Weights of Constraints (used to Compute Partial Solution)

REMACT LPCNTP MDLPB1 C#1 A Code = 'I' Inequality Constraint

REMACT CVLP MDLP1 R The Cost Vector c of the Linear Program
REMACT UBVLP MDLP1 R Upper Bound Vector of the Linear Program
REMACT IBASE MDSN1 I Pointer Array of Basic Variables in the Extended Basis (Basic Array)
REMACT JBASE MDLP1 I A Code = 1 Variable is in the Basis
= 0 Variable is not in the Basis
REMACT IBND MDLP1 I A Code = 1 Variable is in the Lower Bound
= -1 Variable is in the Upper Bound
REMACT SOLLP MDLP1 R Solution of the Linear Program
REMACT SNSLP2 MDLP1 R Sensitivity of the Linear Program w.r.t. to the Vector x
REMACT SNSLP1 MDLPB1 R Sensitivity of the Linear Program w.r.t. to the Vector b
REMACT LPEFFC MDLPB1 I A Code = 1 Constraint is Effective in the Solution
= 0 Constraint is Not Effective in the Solution
REMACT INKT MDLPB1 I Pointer of First Nonzero Off Diagonal Elements (Per Column) of the Basis Matrix
REMACT INKM MDLPB2 I Pointer of Row for Nonzero Off Diagonal Elements of the Basis Matrix
REMACT INKT1 MDLPB1 I Pointer of First non-zero Off Diagonal Elements (Per Column) of the LU Factors of the Basis Matrix
REMACT INKM1 MDLPB2 I Pointer of Row for Nonzero Off Diagonal Elements of the LU Factors of the Basis Matrix
REMACT ILPSCR MDLPB1 I Scratch Array
REMACT JLPSCR MDLPB1 I Scratch Array
REMACT KLPSCR MDLPB1 I Scratch Array
REMACT LLPSCR MDLPB1 I Scratch Array
REMACT YLPSCR MDLPB1 R Scratch Array
REMACT ZLPSCR MDLPB1 R Scratch Array
REMACT TLPSCR MDLPB1 R Scratch Array
REMACT HMD MDLPB1 R Array Containing the Diagonal Elements of the LU Factors of the Basis Matrix
REMACT HMFD MDLPB2 R Array Containing the Nonzero Off Diagonal Elements of the LU Factors of the Basis Matrix
REMACT VMD MDLPB1 R Array Containing the Diagonal Elements of the Basis Matrix
REMACT VMFD MDLPB2 R Array Containing the Nonzero Off Diagonal Elements of the Basis Matrix
| REMACT | NLCONT  | I   | Total Number of Equality Constraints |
| REMACT | BLCONT  | NDCONT | b Value of Equality Constraints |
| REMACT | ILCONT  | NDCONT | Pointer of Position of Equality Constraint in the Set of Constraints |
|        |         |        | + Artificial Variables is Zero |
|        |         |        | - Artificial Variables is Large = zero |
| REMACT | IPLPV1  | MDLPI | Pointer of Control Variables Included in the LP Formulation: IPLPV1(I) = J |
|        |         |        | I : Index of Control Variable |
|        |         |        | J : Index of Nonnegative Variable in LP, first of two (equals zero if control variable I not included in LP) |
| REMACT | IPLPV2  | MDLPI | Inverse Pointer of Nonnegative Variables in the LP Formulation: IPLPV2(I) = J |
|        |         |        | I : Index of Nonnegative Variable in LP |
|        |         |        | J : Index of Control Variables |
| REMACT | ILPDIS  | MDLPI | A code = n Variable is Discrete (n is the number of discrete steps between zero and upper bound |
|        |         |        | = 0 Variable is Continuous |
| REMACT | ILPSEG  | MDLPI | |
| REMACT | VOLTM   | MAXBUS | VOLTM(I) = base case voltage magnitude at bus I |
|        |         |        | I : bus number (optimal ordering) |
| REMACT | DELTM   | MAXBUS | DELTM(I) = base case voltage angle at bus I |
|        |         |        | I : bus number (optimal ordering) |
| REMACT | SJSCR   | 78    | Sensitivity vector of selected constraint with respect to the existing control variables: |
|        |         |        | MGBP with respect to real power generation |
|        |         |        | NCVG with respect to generator bus voltage |
|        |         |        | NTPS with respect to phase shifter angle |
|        |         |        | NTTC with respect to tap ratio |
|        |         |        | NSCB with respect to switchable capacitor |
|        |         |        | NSRB with respect to switchable reactor |
|        |         |        | NILS with respect to interruptible load |
|        |         |        | NFLS with respect to firm load |
|        |         |        | NCLS with respect to critical load |
| REMACT | YASCR   | MDJ1  | Solution of adjoint network |
| REMACT | YBSCR   | MDJ1  | Used as: |
|        |         |        | 1 - driving vector |
|        |         |        | 2 - network solution |
| REMACT | PLFL1   | MAXLIN | PLFL1(I) = real power circuit flow from sending bus side |
|        |         |        | I : circuit number (internal) |
| REACT | QLFL1 | MAXLIN | R | QLFL1(I) = reactive power circuit flow from sending bus side  
|       |      |        |   | I : circuit number (internal numbering)  
| REACT | PLFL2 | MAXLIN | R | PLFL2(I) = real power circuit flow from receiving bus side  
|       |      |        |   | I : circuit number (internal numbering)  
| REACT | QLFL2 | MAXLIN | R | QLFL2(I) = reactive power circuit flow from receiving bus side  
|       |      |        |   | I : circuit number (internal numbering)  
| REACT | PBASE |        | R | Base case MVA  
| REACT | MAXFL |        | P | Maximum number of fill-ups in Gaussian elimination  
| REACT | ILPN  |        | I | Pointer array of circuits connected to bus kl (internal numbering)  
| REACT | NLPN  |        | I | Number of circuits connected to bus number kl (internal numbering)  
| REACT | IRBUS | MAXBUS | I | Array indicating buses around specified circuit. It is defined as follows:  
|       |      |        |   | IRBUS(I) = 1 report bus  
|       |      |        |   | IRBUS(I) = 0 do not report bus  
|       |      |        |   | I : bus number in internal numbering  
| REACT | LPACTV| MDLP1  | I | A code  
|       |      |        |   | = 1 Variable Participates in the LP Solution  
|       |      |        |   | = 0 Variable Does Not Participate in the LP Solution  
| REACT | IPPRIOR| MDLP1   | I | Priority level of LP variables. It assumes values 1,2,3, etc. A value of 99 indicates that the variable is excluded from priority lists  
| REACT | PNLTY0| MDLP1   | R | Zero Level Penalty Factor (discrete variables only)  
| REACT | PNLTY1| MDLP2   | R | One Level Penalty Factor (discrete variables only)  
| REACT | IP1RON| MDLPB1  | I | Pointer of Constraint Order in the Table of Factors of the Basis Matrix IP1RON(I) = J  
|       |      |        |   | I : Index of Constraint in the Table of Factors of the Basis Matrix  
|       |      |        |   | J : Index of Constraint in the Original LP matrix  
| REACT | IP2RON| MDLPB1  | I | Pointer of Constraint Order in the Table of Factors of the Basis Matrix IP2RON(IJ) = J  
|       |      |        |   | I : Index of Constraint in the Table of Factors of the Basis Matrix  
|       |      |        |   | J : Index of Constraint in the Original LP matrix  
| REACT | XCDCOH|         |    | Tolerance of Constraint Coherency  
| REACT | XCDTHLD|        |    | Threshold for Small Sensitivity Elimination  
| REACT | NTFDC |         |    | Total Number of Circuit Flow Constraints  
| REACT | KGFOLE |         |    | Old Number of Total Circuit Flow Constraints  
| REACT | XCDGRP|         |    | Normalized Linearization Bound for Real Power Generation Control Variable |
REMACT XCDTPS Normalized Linearization Bound for Transformer Phase Shift Control Variable
REMACT XCDILS Normalized Linearization Bound for Interruptible Load Control Variable
REMACT XCDFLS Normalized Linearization Bound for Firm Load Control Variable
REMACT XCDRTL Normalized Linearization Bound for Load Transfer Control Variable
REMACT EVTOLN Tolerance to Determine whether a Voltage Constraint is Effective
REMACT EFTOLN Tolerance to Determine whether a Flow Constraint is Effective
REMACT EQTOLN Tolerance to Determine whether a Reactive Constraint is Effective
REMACT SHOYV Convergence Criterion for Voltage
REMACT SMOFV Convergence Criterion for Flow Constraints
REMACT SNOQV Convergence Criterion for Reactive Constraints
REMACT SWOP Convergence Criterion for Net MN Export Constraints
REMACT XACCON Tolerance For Constraint Coherency (Voltage, Flow, or Reactive)
REMACT XACTHLD Threshold Value for Elimination of Small Sensitivities
REMACT RELVOC This Variable Controls the Relinearization of Voltage Constraints
REMACT RELFOC This Variable Controls the Relinearization of Flow Constraints
REMACT RELQOC This Variable Controls the Relinearization of Reactive Constraints
REMACT XCDCGV Normalized Linearization Bound for Generator Voltage Control Variables
REMACT XCDGRP Normalized Linearization Bound for Real Power Generation Control Variables
REMACT XCDTTC Normalized Linearization Bound for Transformer Tap Control Variables
REMACT XCDTPS Normalized Linearization Bound for Transformer Phase Shift Control Variables
REMACT XCDSCB Normalized Linearization Bound for Capacitive Reactance Control
REMACT XCDSRB Normalized Linearization for Inductive Reactance Control Variables
REMACT XCDILS Normalized Linearization Bound for Interruptible Load Control Variables
REMACT XCDFLS Normalized Linearization Bound for Firm Load Control Variables
REMACT XCDRTL Normalized Linearization Bound for Transfer Load Control Variables
REMACT NTFOC Total Number of Circuit Flow Constraints
Total Number of Bus Voltage Constraints
Total Number of unit Reactive Power Constraints
Old Number of Total Circuit Flow Constraints
Old Number of Total Bus Voltage Constraints
Old Number of Total Unit Reactive Power Constraints
Total Number in the Voltage Profile
Pointer of Bus Voltage to be Included in the Voltage Profile (optimal ordering)

Remedial Actions Control Variables:

- **REMAct RAMSET** # I A code to indicate which part of the remedial action program is to be run
  - 1. initializes the remedial action variables
  - 2. modifies the variables for a specified contingency
- **REMAct CKRATE** # I A code specifying the circuit rating to be selected
  - 1. select circuit rating A
  - 2. select circuit rating B
  - 3. select circuit rating C
- **REMAct ACFLAG** # LOGIC A code specifying which model of remedial action has been read.
  - .T. based on AC model
  - .F. based on DC model
- **REMAct CUFLAG** # LOGIC A flag determining whether all remedial actions are within physical limits.
- **REMAct KTDCS** # I A code
  - 1. Matrix B' must be Triangulated
  - 2. Matrix B' is Triangulated
- **REMAct KTJACS** # I An Internal Code to Control Handling of the Jacobian matrix
  - 1. From Jacobian Matrix
  - 2. Triangulate Jacobian Matrix
  - 3. Matrix is already Triangulated
- **REMAct JACTYP** # I A code
  - 1. Full Jacobian
  - 2. Semicoupled Jacobian (Future)
  - 3. Decoupled constant Jacobian
  - 4. Dc Network model Jacobian
- **REMAct JACFLG** # LOGIC A flag when equals to . FALSE., Jacobian pointers have not been computed and control returns to the calling routine
- **REMAct JACHOD** # I A code
  - 2. perform factor update (due to PQ to PV changes)
  - 3. compute Jacobian matrix and table of factors
  - 1. do nothing for Jacobian
REMACPFSTYP

A code

- 1, converged power flow solution
- 2, valid power flow solution
- 3, diverged power flow

JACSIZ

A code

- 1, exclude slack bus and Q-equations for PV-bus
- 2, include all equations