เอกสารอ้างอิง

1. Edward E. Hogwood Jr., David E. Rice, "The Electrical Aspects of Cogeneration System Design", IEEE Trans.Ind.Appl., Vol.IA-23, No.4, Jul./Aug. 1987.
2. L.E. Goff, G.J. Potochney, L.J Powell, and M.J. Rook, "Application and protective relays on a large industrial-utility tie with industrial co-generation", IEEE Trans.Power App. Syst., Vol.Pas 100, No.6, pp. 2804-2812, June 1981.
3. James M. Daley, "Design Considerations for Operating On-Site Generators in Parallel with Utility Service", IEEE Trans.Ind. Appl., Vol.IA-21, No.1, Jan./Feb. 1985.
4. Louie J. Powell, "An Industrial View of Utility Cogeneration Protection Requirements", IEEE Trans.Ind.Appl., Vol.24, No.1, Jan/Feb. 1988.
5. Somyot Ruchirawat, "CASE STUDY II : Energy Management in Pulp & Paper Industry by System Analysis, "ENERGY CONSERVATION IN INDUSTRY, ERTC., Bangkok, Thailand, Mar. 1986.
6. วัชรา ทนิษฐบุตร และดร. วิวัฒน์ ตันตะพานิชกุล, "การหาเงื่อนไขการทำงานที่ลดค่าใช้จ่ายด้านพลังงานให้ต่ำสุดในโรงงานที่ผลิตกระแสไฟฟ้าและไอน้ำใช้เอง", วิศวกรรมสาร, 1, 81-86, 2530
7. P.M. Anderson, A.A. Fouad : Power System Control and Stability, The Iowa State University Press, 1st edition, 1977.
8. Paul C.Krause : Analysis of Electric Machinery , McGraw-Hil, 1st printing, 1987.
9. Wei-Jen Lee, Mo-Shing Chen and Lucas B. Williams, "Load Model for

Stability Studies", IEEE Trans.Ind.Appl., Vol.IA-23, No.1, Jan./Feb. 1987.

10. Stagg ,G.W. and A.H. El Abiad : Computer Methods in Power System Analysis ,McGraw-Hill ,11th printing ,1985.

11. Philip A. Nobile,"Power System Studies For Cogeneration : What's Really Needed ?", IEEE Trans.Ind.Appl., Vol.IA-23, No.5, Sep./Oct. 1987.

12. M.A. Pai: Computer Techniques in Power System Analysis,McGraw-Hill, TATA, 1980.

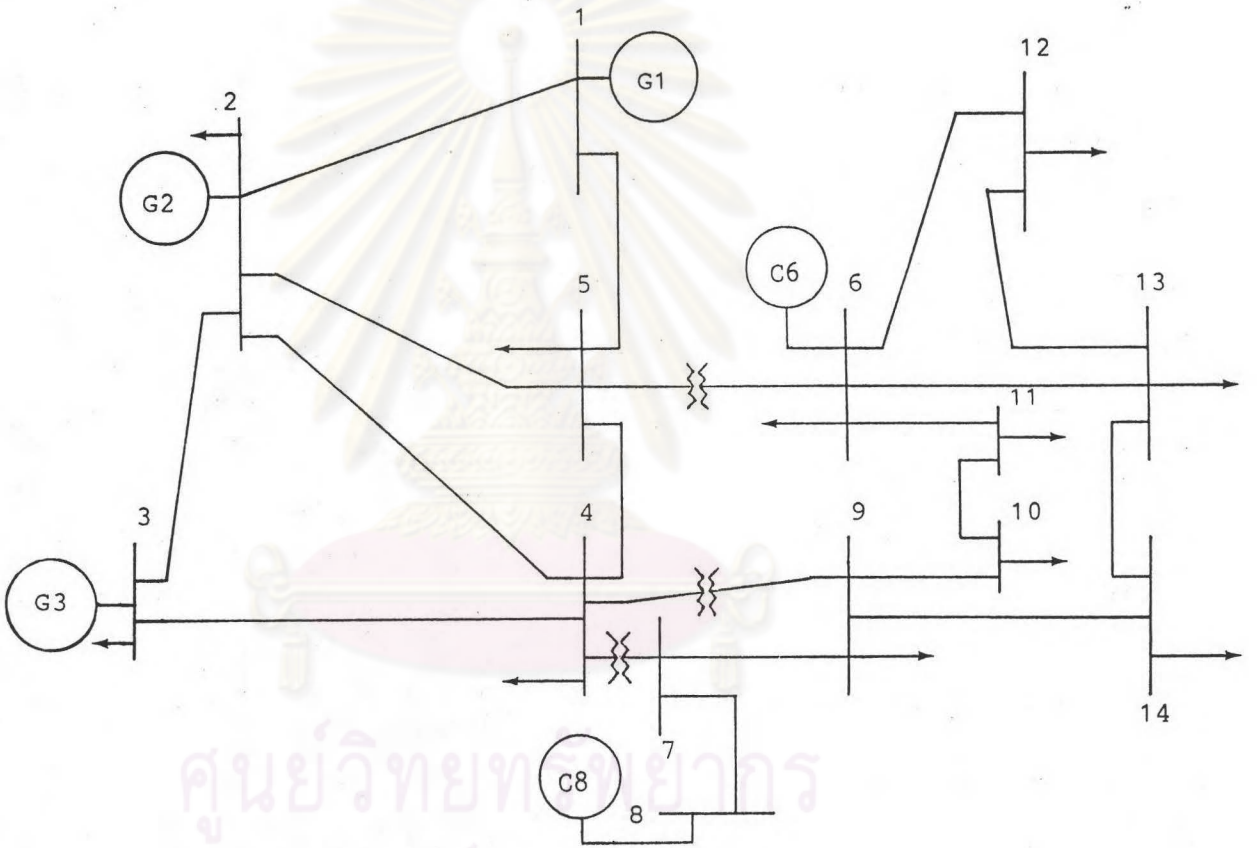
13. P.M. Rane and D.L. Shenoy, "Transient Stability Analysis of Multi-machine Power System on Digital Computer.," Proceeding of the Symposium on Computer Applications in Large Scale Power Systems, Vol.2, p.333-334, New Delhi, India, Aug. 1979.

14. Hamdy S. Khalil and Ahmed M.El-Serafi, "Transient Stability Studies of AC/DC Systems ", Proceeding of the Symposium on Computer Applications in Large Scale Power Systems, Vol.2, p.187-194, New Delhi, India, Aug. 1979.

15. Michael Patrick Toy, "Potential Industrial Cogeneration Based on System Optimization at Plant Level", Thesis of Drexel University, Jun. 1984.

ภาคผนวก ก

ข้อมูลของระบบไฟฟ้ามาตราฐาน 14 บัสของ IEEE



รูปที่ ก-1 ระบบไฟฟ้ามาตรฐาน 14 บัส ของ IEEE

ตารางที่ ก-1 Line data

Line No.	Between buses	Line impedance		Half line charging susceptance per unit
		R per unit	X per unit	
1	1-2	0.01938	0.05917	0.02640
2	2-3	0.04699	0.19797	0.02190
3	2-4	0.05811	0.17632	0.01870
4	2-5	0.05695	0.17388	0.01700
5	1-5	0.05403	0.22304	0.02460
6	3-4	0.06701	0.17103	0.01760
7	4-5	0.01335	0.04211	0.00640
8	5-6	0.00000	0.25202	0.00000
9	4-7	0.00000	0.20912	0.00000
10	7-8	0.00000	0.17615	0.00000
11	4-9	0.00000	0.55618	0.00000
12	7-9	0.00000	0.11001	0.00000
13	9-10	0.03181	0.08450	0.00000
14	6-11	0.09498	0.19890	0.00000
15	6-12	0.12291	0.25581	0.00000
16	6-13	0.06615	0.13027	0.00000
17	9-14	0.12711	0.27038	0.00000
18	10-11	0.08205	0.19207	0.00000
19	12-13	1.22092	0.19988	0.00000
20	13-14	0.17093	0.34802	0.00000

ตารางที่ ก-2

Transformer data

Transformer	Between buses	Tap setting
1	4-7	0.978
2	4-9	0.969
3	5-6	0.932

ตารางที่ ก-3

Shut capacitor data

Bus Number	Suceptance per unit
9	0.190

ตารางที่ ก-4

Regulated bus data (P-V buses)

Bus No.	Voltage magnitude per unit	Reactive power limits	
		Minimum Mvar	Maximum Mvar
2	1.045	-40.0	50.0
3	1.010	0.0	40.0
6	1.070	-6.0	24.0
8	1.090	-6.0	24.0

ภาคผนวก ข

ข้อมูลรายละเอียดของโหลดที่เป็นมอเตอร์

ตารางที่ ข-1

Load (Motor) Data (per unit)

Motor Size (kW)	Stator Resistance (Rs)	Rotor Resistance (Rr)	Mutual Reactance (Xm)	Stator Reactance (Xs)	Inertia Constant (H)
.25	0.22	0.08	1.022	0.125	0.017
.25	0.254	0.062	0.837	0.167	0.014
.25	0.273	0.024	0.681	0.143	0.017
.25	0.287	0.028	0.826	0.111	0.041
.25	0.221	0.028	0.804	0.125	0.021
.37	0.241	0.053	1.105	0.125	0.027
.37	0.214	0.057	1.421	0.102	0.03
.37	0.19	0.068	1.18	0.116	0.014
.37	0.201	0.063	1.449	0.1	0.024
.37	0.178	0.087	1.085	0.132	0.011
.75	0.199	0.046	1.324	0.1	0.036
.75	0.184	0.06	1.142	0.111	0.019
.75	0.214	0.03	1.373	0.083	0.075
.75	0.212	0.03	1.335	0.1	0.043
.75	0.194	0.046	1.146	0.125	0.029
1.1	0.159	0.048	1.494	0.091	0.032
1.1	0.166	0.061	1.258	0.111	0.061
1.1	0.188	0.03	1.655	0.083	0.03
1.1	0.216	0.048	1.326	0.111	0.048
1.1	0.203	0.042	1.354	0.111	0.042
1.5	0.147	0.045	1.624	0.083	0.043
1.5	0.156	0.054	1.309	0.1	0.022
1.5	0.192	0.029	1.804	0.077	0.09
1.5	0.218	0.028	1.656	0.083	0.176
1.5	0.179	0.044	1.45	0.111	0.036
2.2	0.124	0.043	1.738	0.077	0.038
2.2	0.145	0.048	1.364	0.1	0.024
2.2	0.172	0.044	1.513	0.111	0.03
2.2	0.155	0.031	1.899	0.083	0.065
2.2	0.152	0.04	1.386	0.091	0.029
3	0.122	0.036	1.814	0.071	0.05
3	0.13	0.048	1.431	0.091	0.022
3	0.173	0.031	1.512	0.111	0.201
3	0.143	0.064	1.695	0.1	0.034
3	0.148	0.057	1.683	0.083	0.034
4	0.113	0.039	2.007	0.071	0.066
4	0.102	0.056	1.572	0.091	0.026
4	0.132	0.045	1.789	0.083	0.055
4	0.078	0.046	1.782	0.076	0.065
4	0.118	0.038	2.098	0.075	0.059
5.5	0.109	0.033	2.017	0.072	0.096
5.5	0.088	0.053	1.649	0.089	0.029
5.5	0.11	0.031	2.014	0.07	0.099
5.5	0.091	0.033	2.171	0.088	0.3
5.5	0.087	0.026	1.485	0.088	0.097
7.5	0.098	0.036	2.153	0.07	0.027
7.5	0.102	0.04	1.942	0.079	0.093
7.5	0.08	0.047	1.666	0.071	0.034
7.5	0.07	0.03	1.598	0.086	0.091

ตารางที่ ข-1 (ต่อ)

Load (Motor) Data (cont.) (per unit)

Motor Size (kW)	Stator Resistance (Rs)	Rotor Resistance (Rr)	Mutual Reactance (Xm)	Stator Reactance (Xs)	Inertia Constant (H)
7.5	0.097	0.032	1.801	0.074	0.071
11	0.084	0.03	2.081	0.068	0.197
11	0.089	0.039	1.601	0.068	0.053
11	0.095	0.03	1.877	0.083	0.123
11	0.092	0.041	1.41	0.111	0.098
11	0.093	0.025	1.883	0.072	0.13
15	0.083	0.029	1.907	0.078	0.107
15	0.063	0.043	1.817	0.085	0.048
15	0.084	0.03	2.081	0.081	0.142
15	0.087	0.032	1.653	0.063	0.053
15	0.086	0.025	2.075	0.072	0.116
18.5	0.084	0.03	2.081	0.071	0.106
18.5	0.061	0.022	1.82	0.077	0.059
18.5	0.075	0.03	2.213	0.085	0.166
18.5	0.069	0.022	1.693	0.072	0.101
18.5	0.074	0.025	2.106	0.072	0.118
22	0.081	0.019	1.839	0.077	0.145
22	0.065	0.025	1.951	0.077	0.071
22	0.075	0.03	2.213	0.082	0.238
22	0.063	0.018	1.757	0.071	0.105
22	0.079	0.017	1.916	0.077	0.147
30	0.074	0.02	2.215	0.098	0.25
30	0.063	0.018	1.757	0.074	0.111
30	0.058	0.021	1.966	0.074	0.069
30	0.076	0.013	2.34	0.071	0.218
30	0.081	0.013	2.327	0.074	0.218
37	0.074	0.02	2.215	0.067	0.25
37	0.055	0.029	1.775	0.093	0.09
37	0.064	0.015	2.131	0.071	0.188
37	0.051	0.022	1.73	0.079	0.053
37	0.059	0.016	2.143	0.074	0.188
45	0.053	0.026	1.727	0.091	0.092
45	0.082	0.033	2.323	0.074	0.161
45	0.053	0.019	1.677	0.079	0.067
45	0.055	0.021	2.154	0.076	0.154
45	0.062	0.012	2.136	0.077	0.208
55	0.056	0.011	1.831	0.071	0.293
55	0.052	0.015	1.728	0.071	0.122
55	0.066	0.03	2.367	0.071	0.429
55	0.053	0.028	1.979	0.078	0.126
55	0.037	0.018	2.101	0.079	0.398
75	0.054	0.007	1.777	0.071	0.354
75	0.051	0.011	1.731	0.071	0.158
75	0.061	0.027	2.74	0.075	0.652
75	0.05	0.025	1.911	0.068	0.161
75	0.039	0.014	2.195	0.078	0.493
90	0.055	0.007	1.835	0.071	0.353
90	0.049	0.011	1.789	0.071	0.154
90	0.063	0.025	2.989	0.067	0.648

ตารางที่ ข-1 (ต่อ)

Load (Motor) Data (cont.) (per unit)

Motor Size (kW)	Stator Resistance (Rs)	Rotor Resistance (Rr)	Mutual Reactance (Xm)	Stator Reactance (Xs)	Inertia Constant (H)
90	0.048	0.025	2.173	0.068	0.159
90	0.038	0.012	2.198	0.074	0.492
110	0.056	0.007	1.971	0.071	0.634
110	0.051	0.014	2.067	0.083	0.259
110	0.061	0.027	2.544	0.091	0.809
110	0.049	0.022	1.914	0.075	0.18
110	0.037	0.012	2.201	0.083	0.646
132	0.065	0.007	1.95	0.071	0.593
132	0.04	0.011	1.936	0.077	0.241
132	0.053	0.025	2.766	0.077	0.813
132	0.044	0.022	1.925	0.071	0.182
132	0.032	0.012	2.212	0.074	0.66
160	0.045	0.024	2.791	0.079	0.816
160	0.042	0.018	1.863	0.067	0.181
160	0.044	0.005	2	0.071	0.598
160	0.043	0.007	2.086	0.071	0.232
160	0.045	0.011	2.178	0.079	0.514
200	0.044	0.007	2.183	0.071	0.655
200	0.041	0.011	2.19	0.077	0.291
200	0.047	0.02	2.582	0.093	0.771
200	0.043	0.018	1.928	0.083	0.232
200	0.036	0.007	2.453	0.071	0.609
250	0.039	0.007	2.194	0.071	0.608
250	0.036	0.608	2.201	0.071	0.286
250	0.057	0.024	2.26	0.089	0.55
250	0.044	0.022	1.925	0.088	0.18
250	0.035	0.007	2.319	0.071	0.567
280	0.048	0.01	3.041	0.061	0.888
280	0.04	0.022	1.935	0.089	0.172
280	0.039	0.009	2.444	0.071	0.732
280	0.03	0.014	2.47	0.071	0.311
280	0.032	0.007	2.216	0.071	0.517
315	0.037	0.007	2.199	0.071	0.592
315	0.032	0.011	2.213	0.071	0.254
315	0.039	0.018	2.097	0.081	0.231
315	0.031	0.007	2.629	0.071	0.602
315	0.032	0.016	2.115	0.075	0.279
355	0.035	0.007	2.319	0.071	0.637
355	0.032	0.011	2.213	0.074	0.461
355	0.031	0.011	2.031	0.074	0.584
355	0.035	0.007	2.319	0.071	0.637
355	0.032	0.016	2.213	0.075	0.281
400	0.032	0.005	2.327	0.071	0.631
400	0.032	0.011	2.213	0.072	0.451
400	0.031	0.011	2.116	0.071	0.575
400	0.032	0.018	1.953	0.083	0.159
400	0.045	0.012	2.294	0.083	0.328

From : Manufacturer

ตารางที่ ข-2

Logarithmic Least Squares Fit for Stator Resistance (Rs)
y = a ln(bx)

Coefficients in least squares approximation:
Coefficient 0 (a) : -2.65507823863800E-0002
Coefficient 1 (b) : 1.54911808173530E-0003

Standard Deviation : 2.1896E-0002

ตารางที่ ข-3

Logarithmic Least Squares Fit for Rotor Resistance (Rr)
y = a ln(bx)

Coefficients in least squares approximation:
Coefficient 0 (a) : -6.2783787041E-03
Coefficient 1 (b) : 6.3339277850E-04

Standard Deviation : 9.545669E-03

ตารางที่ ข-4

Logarithmic Least Squares Fit for Mutual Reactance (Xm)
y = a ln(bx)

Coefficients in least squares approximation:
Coefficient 0 (a) : 1.5670009658E-01
Coefficient 1 (b) : 8.7246018260E+03

Standard Deviation : 2.562040E-01

ตารางที่ ข-5

Logarithmic Least Squares Fit for Stator Reactance (Xs)
y = a ln(bx)

Coefficients in least squares approximation:
Coefficient 0 (a) : -5.3812754207E-03
Coefficient 1 (b) : 1.0002554369E-08

Standard Deviation : 1.240232E-02

ตารางที่ ข-6

Logarithmic Least Squares Fit for Inertia Constant (H)
y = a4.x^4 + a3.x^3 + a2.x^2 + a1.x + a0

Coefficients in least squares approximation:
Coefficient 0 (a0) : 3.8835752799E-02
Coefficient 1 (a1) : 4.7870426021E-03
Coefficient 2 (a2) : -1.0318613043E-05
Coefficient 3 (a3) : -1.9599581747E-08
Coefficient 4 (a4) : 5.5098263277E-11

Standard Deviation : 1.425753E-01

ตารางที่ ข-7

Load (Motor) Data by Least Square Fitting

(per unit)

Motor Size (kW)	Stator Resistance (Rs)		Rotor Resistance (Rr)		Mutual Reactance (Xm)		Stator Reactance (Xs)		Inertia Constant (H)	
	Fitted value	Residual value	Fitted value	Residual value	Fitted value	Residual value	Fitted value	Residual value	Fitted value	Residual value
0.25	0.208592	-0.01140	0.054940	-0.02505	1.204648	0.182648	0.106585	-0.01841	0.040031	0.023031
0.37	0.198183	-0.04281	0.052478	-0.00052	1.266081	0.161081	0.104475	-0.02052	0.040605	0.013605
0.75	0.179423	-0.01457	0.048042	0.002042	1.376801	0.230801	0.100673	-0.02432	0.042420	0.013420
1.1	0.169254	0.010254	0.045638	-0.00236	1.436816	-0.05718	0.098612	0.007612	0.044088	0.012088
1.5	0.161019	-0.01798	0.043690	-0.00030	1.485417	0.035417	0.096943	-0.01405	0.045993	0.009993
2.2	0.150851	-0.00114	0.041286	0.001286	1.545432	0.159432	0.094882	0.003882	0.049317	0.020317
3	0.142616	0.020616	0.039339	0.003339	1.594033	-0.21996	0.093213	0.022213	0.053103	0.003103
4	0.134978	0.021978	0.037532	-0.00146	1.639113	-0.36788	0.091665	0.020665	0.057817	-0.00818
5.5	0.126523	0.017523	0.035533	0.002533	1.689015	-0.32798	0.089951	0.017951	0.064849	-0.03115
7.5	0.118288	0.020288	0.033586	-0.00241	1.737616	-0.41538	0.088282	0.018282	0.074150	0.002150
11	0.108119	0.015119	0.031181	0.006181	1.797631	-0.08536	0.086221	0.014221	0.090219	-0.03978
15	0.099884	0.013884	0.029234	0.004234	1.846233	-0.22876	0.084552	0.012552	0.108256	-0.00774
18.5	0.094316	0.020316	0.027917	0.002917	1.879096	-0.22690	0.083424	0.011424	0.123746	0.005746
22	0.089715	0.010715	0.026829	0.009829	1.906247	-0.00975	0.082491	0.005491	0.138960	-0.00803
30	0.081480	0.000480	0.024882	0.011882	1.954849	-0.37215	0.080822	0.006822	0.172675	-0.04532
37	0.075912	0.016912	0.023565	0.007565	1.987712	-0.15528	0.079694	0.005694	0.200940	0.012940
45	0.070715	0.017715	0.022336	-0.00366	2.018385	0.291385	0.078640	-0.01235	0.231797	0.139797
55	0.065387	0.009387	0.021077	0.010077	2.049830	0.218830	0.077560	0.006560	0.268152	-0.02484
75	0.057152	0.003152	0.019129	0.012129	2.098432	0.321432	0.075891	0.004891	0.333296	-0.02070
90	0.052311	-0.00268	0.017985	0.010985	2.127001	0.292001	0.074910	0.003910	0.375415	0.022415
110	0.046983	-0.00901	0.016725	0.009725	2.158447	0.187447	0.073830	0.002830	0.422535	-0.21146
132	0.042143	-0.02285	0.015580	0.008580	2.187016	0.237016	0.072849	0.001849	0.462583	-0.13041
160	0.037035	-0.00796	0.014372	-0.00962	2.217161	-0.57383	0.071814	-0.00718	0.496435	-0.31956
200	0.031110	-0.01288	0.012971	0.005971	2.252128	0.069128	0.070613	-0.00038	0.514860	-0.14013
250	0.025186	-0.01381	0.011570	0.004570	2.287094	0.093094	0.069412	-0.00158	0.499667	-0.10833
280	0.022177	-0.02582	0.010859	0.000859	2.304853	-0.73614	0.068803	0.007803	0.478643	-0.40935
315	0.019050	-0.01794	0.010119	0.003119	2.323310	0.124310	0.068169	-0.00283	0.452763	-0.13923
355	0.015876	-0.01912	0.009369	0.002369	2.342042	0.023042	0.067525	-0.00347	0.436056	-0.20094
400	0.012707	-0.01929	0.008619	0.003619	2.360744	0.033744	0.066883	-0.00411	0.458817	-0.17218

ภาคผนวก ค

โปรแกรมที่ใช้งาน

```
PROGRAM  DATAENT; (DATA ENTRY PROGRAM)
USES
  DOS,CRT,COMMON,TOOLS;

TYPE  GEN       = RECORD
                    BUS_NO : INTEGER;
                    H      : REAL;
                    XD     : REAL;
                  END;
      BUS       = RECORD
                    BUS_NO : INTEGER;
                    VOLT   : COMPLEX;
                    SGEN   : COMPLEX;
                    SLOAD  : COMPLEX;
                  END;
      LINE      = RECORD
                    BUSI     : INTEGER;
                    BUSJ     : INTEGER;
                    LINEIMP  : COMPLEX;
                    LINECHRG : COMPLEX;
                  END;
      TRFM      = RECORD
                    BUSI     : INTEGER;
                    BUSJ     : INTEGER;
                    TAPI_J   : REAL;
                    TRIMP    : COMPLEX;
                  END;

VAR   GEN_FIL,BUS_FIL,LOAD_FIL,SHUNT_FIL,
      LINE_FIL,TRAN_FIL        : TEXT;
      GENREC                   : GEN;
      BUSREC                   : BUS;
      LINEREC                  : LINE;
      GG,BB,LL,TT,LD,SH        : STRING;

FUNCTION RTRIM(VAR ST : STRING):STRING;
VAR  LL,ENDCHK,CK : INTEGER;
     TMP          : STRING;
BEGIN
  LL := LENGTH(ST);
  TMP := ST;
  ENDCHK := POS(' ',ST);
  WHILE POS(' ',ST) > 0 DO
  BEGIN
    CK := POS(' ',ST);
    ST[CK] := '*';
    IF POS(' ',ST)-CK < 0 THEN
      IF LL <> CK THEN
        ENDCHK := LL+1;
    IF POS(' ',ST)-CK > 1 THEN
      ENDCHK := POS(' ',ST);
  END;
  IF ENDCHK <> 0 THEN
    DELETE(TMP,ENDCHK,LL-ENDCHK+1);
  RTRIM := TMP;
END;
```

```
PROCEDURE READINT(VAR VALUE : INTEGER);
VAR     STRVALUE : STRING[10];
        STRCHK   : INTEGER;
BEGIN
  READLN(STRVALUE);
  VAL(STRVALUE,VALUE,STRCHK);
  WHILE STRCHK <> 0 DO
    BEGIN
      WRITE('DATA TYPE MISMATCH! ...PLEASE TRY AGAIN : ');
      READLN(STRVALUE);
      VAL(STRVALUE,VALUE,STRCHK);
    END;
END;  {READINT}

PROCEDURE READREAL(VAR VALUE : REAL);
VAR     STRVALUE : STRING;
        STRCHK   : INTEGER;
BEGIN
  READLN(STRVALUE);
  IF RTRIM(STRVALUE) = '*' THEN
    STRVALUE := '99999.0';
  VAL(STRVALUE,VALUE,STRCHK);
  WHILE STRCHK <> 0 DO
    BEGIN
      WRITE('DATA TYPE MISMATCH! ...PLEASE TRY AGAIN : ');
      READLN(STRVALUE);
      VAL(STRVALUE,VALUE,STRCHK);
    END;
END;  {READREAL}

PROCEDURE  CHKFILE(VAR FIL : TEXT;VAR FILENAME : STRING);
VAR  CH : CHAR;
BEGIN
  REPEAT
    CH := 'Y';
    ASSIGN(FIL,FILENAME);
{$I-} RESET(FIL); {$I+}
    IF IORESULT = 0 THEN  { The file already exists. }
      BEGIN
        CLOSE(FIL);
        WRITELN;
        WRITE('Old data file (',FILENAME,') exists. ');
        WRITE('Write over it (Y/N)? ');
        CH := UPCASE(READKEY);
        WRITELN(CH);
      END;
    IF CH = 'Y' THEN
      BEGIN
        REWRITE(FIL);
        IOCHECK;
      END;
    IF CH = 'N' THEN
      BEGIN
        WRITELN;
        WRITE('Enter new data file name ');
```

```
        READLN(FILENAME);
      END;
  UNTIL((CH = 'Y') AND NOT(IOERR));
END;  {CHKFILE}

PROCEDURE  ENTER1(VAR GNFIL : TEXT;
                  VAR BSFIL : TEXT;
                  VAR LNFIL : TEXT;
                  VAR TRFIL : TEXT;
                  VAR LDFIL : TEXT;
                  VAR SHFIL : TEXT);
VAR  VB,VF,HH,XXD,SGE,SGFMAX,SGFMIN,SLR,SLF,LIMPE,STC,
     DYN,YSR,LIMPF,LCHRF,TIMPF,TAPIJ,VB,MVAB     : REAL;
     XD0,XQ0,XD1,XQ1,XD2,XQ2,TD1,TQ1,TD2,TQ2,XLS : REAL;
     BSNO,BSI,BSJ,BSCHK,MVA,TYP,TYPL,TYPG        : INTEGER;

BEGIN
  CLRSCR;
  WRITELN;
  WRITELN('SYSTEM-BASE :');
  WRITELN;
  WRITE('MVA-BASE : ');
  READINT(MVA);
  WRITELN(GNFIL,MVA);
  WRITELN;
  WRITELN('SWING-BUS DATA :'); {TYPE 0}
  WRITELN('( Bus-No., V(Re) , V-bus(base-kV) )');
  WRITELN;
  WRITE('Bus-No. : ');
  READINT(BSNO);
  BSCHK := BSNO;
  WRITE('V(Re)   : ');
  READREAL(VB);
  VF := 0;
  WRITE('V-bus(base-kV)   : ');
  READREAL(VB);
  TYP := 0;  {0 = SLACK BUS}
  XYTOCOMPLEX(VB,VF,QUAN[1]);
  WRITE(BSFIL,TYP:4);WRITE(BSFIL,VB:6:2);
  WRITE(BSFIL,BSNO:4);WRITE(BSFIL,QUAN[1].RE:18:9);
  WRITE(BSFIL,QUAN[1].IM:18:9);WRITE(BSFIL,QUAN[1].MAG:18:9);
  WRITE(BSFIL,QUAN[1].ANG:18:9);WRITE(BSFIL,0.0:18:9);
  WRITE(BSFIL,0.0:18:9);WRITE(BSFIL,0.0:18:9);WRITE(BSFIL,0.0:18:9);
  WRITE(BSFIL,0.0:18:9);WRITE(BSFIL,0.0:18:9);WRITELN(BSFIL,0.0:18:9);
  WRITELN;
  WRITELN('GENERATOR-BUS DATA : Quit by enter Bus-No. with "-1"');
  WRITELN('( Bus-No., H, Xd, V-bus(base-kV), V(Re), V(Im), Pg, Qmax, Qmin, Pl, Ql)');
  WRITELN;
  REPEAT
    WRITE('Bus-No. : ');
    READINT(BSNO);
    IF BSNO <> -1 THEN
      BEGIN
        WRITELN('Generator Model : 1) Classical Model ,2) Detailed Model');
        WRITE('Generator Model :');
        REPEAT
```

```
  READINT(TYPG);
UNTIL (TYPG = 1) OR (TYPG = 2);
IF TYPG = 1 THEN
BEGIN
  WRITE('MVA-bus(base)  : ');
  READREAL(MVAB);
  WRITE('H-const.: ');
  READREAL(HH);
  WRITE('Xd      : ');
  READREAL(XXD);
END
ELSE
BEGIN
  WRITE('MVA-bus(base)  : ');
  READREAL(MVAB);
  WRITE('H-const.: ');
  READREAL(HH);
  WRITELN('Xd0,Xq0 = STEADY-STATE REACTANCES');
  WRITE('Xd0     : ');
  READREAL(XD0);
  WRITE('Xq0     : ');
  READREAL(XQ0);
  WRITELN('Xd1,Xq1 = TRANSIENT REACTANCES');
  WRITE('Xd1     : ');
  READREAL(XD1);
  WRITE('Xq1     : ');
  READREAL(XQ1);
  WRITELN('Xd2,Xq2 = SUBTRANSIENT REACTANCES');
  WRITE('Xd2     : ');
  READREAL(XD2);
  WRITE('Xq2     : ');
  READREAL(XQ2);
  WRITELN('Xls = LEAKAGE REACTANCES');
  WRITE('Xls     : ');
  READREAL(XLS);
  WRITELN('Td01,Tq01 = TRANSIENT OPEN-CIRCUIT TIME CONSTANTS');
  WRITE('Td01    : ');
  READREAL(TD1);
  WRITE('Tq01    : ');
  READREAL(TQ1);
  WRITELN('Td02,Tq02 = SUBTRANSIENT OPEN-CIRCUIT TIME CONSTANTS');
  WRITE('Td02    : ');
  READREAL(TD2);
  WRITE('Tq02    : ');
  READREAL(TQ2);
END;
IF BSNO <> BSCHK THEN
  BEGIN
    WRITE('V-bus(base-kV)   : ');
    READREAL(VB);
    WRITE('V(Re)   : ');
    READREAL(VR);
    WRITE('V(Im)   : ');
    READREAL(VF);
    WRITE('P(gen.) : ');
    READREAL(SGE);
```

```
            WRITE('Qmax(gen.) : ');
            READREAL(SGFMAX);
            WRITE('Qmin(gen.) : ');
            READREAL(SGFMIN);
            IF SGFMIN = 99999.0 THEN
              SGFMIN := -99999.0;
            WRITE('P(load) : ');
            READREAL(SLE);
            WRITE('Q(load) : ');
            READREAL(SLF);
            WRITELN('Type of Load : 1) Constant Power ,2) Composite Load');
            WRITE('Type of Load : ');
            REPEAT
              READINT(TYPL);
            UNTIL (TYPL = 1) OR (TYPL = 2);
            WRITE(LDFIL,BSNO:4);WRITE(LDFIL,TYPL:2);
            IF TYPL = 2 THEN
            BEGIN
              WRITELN('Composite Load : 1) Static Load ,2) Dynamic Load');
              WRITE('Percent of Static Load (0 - 100%) : ');
              REPEAT
                READREAL(STC);
              UNTIL (STC >= 0) AND (STC <= 100);
              DYN := 100 - STC;
              WRITE('Percent of Dynamic Load (0 - 100%) : ',DYN);
              WRITE(LDFIL,STC/100:8:5);
              WRITE(LDFIL,DYN/100:8:5);
            END;
            WRITELN(LDFIL);
            XYTOCOMPLEX(VB,VF,QUAN[1]);
            TYP := 1;  {1 = P-V BUS}
            WRITE(BSFIL,TYP:4);WRITE(BSFIL,VB:6:2);WRITE(BSFIL,BSNO:4);
            WRITE(BSFIL,QUAN[1].RE:18:9);WRITE(BSFIL,QUAN[1].IM:18:9);
            WRITE(BSFIL,QUAN[1].MAG:18:9);WRITE(BSFIL,QUAN[1].ANG:18:9);
            WRITE(BSFIL,SGE:18:9);WRITE(BSFIL,SGFMAX:18:9);
            WRITE(BSFIL,SGFMIN:18:9);
            XYTOCOMPLEX(SLE,SLF,QUAN[1]);
            WRITE(BSFIL,QUAN[1].RE:18:9);WRITE(BSFIL,QUAN[1].IM:18:9);
            WRITE(BSFIL,QUAN[1].MAG:18:9);WRITELN(BSFIL,QUAN[1].ANG:18:9);
          END;
        IF TYPG = 1 THEN
        BEGIN
          WRITE(GNFIL,TYPG);WRITE(GNFIL,MVAB:6:2);
          WRITE(GNFIL,BSNO:4);WRITE(GNFIL,HH);WRITELN(GNFIL,XXD);
        END
        ELSE
        BEGIN
          WRITE(GNFIL,TYPG);WRITE(GNFIL,MVAB:6:2);
          WRITE(GNFIL,BSNO:4);WRITE(GNFIL,HH);WRITE(GNFIL,XD0);
          WRITE(GNFIL,XQ0);WRITE(GNFIL,XD1);WRITE(GNFIL,XQ1);
          WRITE(GNFIL,XD2);WRITE(GNFIL,XQ2);WRITE(GNFIL,XLS);
          WRITE(GNFIL,TD1);WRITE(GNFIL,TQ1);WRITE(GNFIL,TD2);
          WRITELN(GNFIL,TQ2);
        END;
      END;
    WRITELN;
```

```
UNTIL BSNO = -1;
WRITELN;
WRITELN('LOAD-BUS DATA : Quit by enter Bus-No. with "-1"');
WRITELN('( Bus-No., V-bus(base-kV) , P(load) , Q(load) )');
WRITELN;
REPEAT
  WRITE('Bus-No. : ');
  READINT(BSNO);
  IF BSNO <> -1 THEN
    BEGIN
      IF BSNO <> BSCHK THEN
      BEGIN
        WRITE('V-bus(base-kV)   : ');
        READREAL(VB);
        WRITE('P(load) : ');
        READREAL(SLE);
        WRITE('Q(load) : ');
        READREAL(SLF);
        WRITELN('Type of Load : 1) Constant Power ,2) Composite Load');
        WRITE('Type of Load : ');
        REPEAT
          READINT(TYPL);
        UNTIL (TYPL = 1) OR (TYPL = 2);
        WRITE(LDFIL,BSNO:4);WRITE(LDFIL,TYPL:2);
        IF TYPL = 2 THEN
        BEGIN
          WRITELN('Composite Load : 1) Static Load ,2) Dynamic Load');
          WRITE('Percent of Static Load (0 - 100%) : ');
          REPEAT
            READREAL(STC);
          UNTIL (STC >= 0) AND (STC <= 100);
          DYN := 100 - STC;
          WRITE('Percent of Dynamic Load (0 - 100%) : ',DYN);
          WRITE(LDFIL,STC/100:8:5);
          WRITE(LDFIL,DYN/100:8:5);
        END;
        WRITELN(LDFIL);
        XYTOCOMPLEX(SLE,SLF,QUAN[1]);
        TYP := 2;  {2 = P-Q BUS}
        WRITE(BSFIL,TYP:4);WRITE(BSFIL,VB:6:2);WRITE(BSFIL,BSNO:4);
        WRITE(BSFIL,1.0:18:9);WRITE(BSFIL,0.0:18:9);WRITE(BSFIL,1.0:18:9);
        WRITE(BSFIL,0.0:18:9);WRITE(BSFIL,0.0:18:9);WRITE(BSFIL,0.0:18:9);
        WRITE(BSFIL,0.0:18:9);WRITE(BSFIL,QUAN[1].RE:18:9);
        WRITE(BSFIL,QUAN[1].IM:18:9);WRITE(BSFIL,QUAN[1].MAG:18:9);
        WRITELN(BSFIL,QUAN[1].ANG:18:9);
      END
      ELSE
        WRITELN('This is Swing Bus !.. Please Enter The Next One');
    END;
  WRITELN;
UNTIL BSNO = -1;
WRITE(BSFIL,-1:4);WRITE(GNFIL,-1:4);WRITE(LDFIL,-1:4);
CLOSE(LDFIL);
CLOSE(BSFIL);
CLOSE(GNFIL);
WRITELN;
```

```
WRITELN('LINE DATA : Quit by enter Bus(p) with "-1"');
WRITELN('( Bus(p) , Bus(q) , Z(Re) , Z(Im) , y(pq)/2 )');
WRITELN;
REPEAT
  WRITE('Bus(p)  : ');
  READINT(BSI);
  IF BSI <> -1 THEN
    BEGIN
      WRITE('Bus(q)  : ');
      READINT(BSJ);
      WRITE('Z(Re)   : ');
      READREAL(LIMPE);
      WRITE('Z(Im)   : ');
      READREAL(LIMPF);
      WRITE('y(pq)/2 : ');
      READREAL(LCHRF);
      XYTOCOMPLEX(LIMPE,LIMPF,QUAN[1]);
      WRITE(LNFIL,BSI:4);WRITE(LNFIL,BSJ:4);
      WRITE(LNFIL,QUAN[1].RE:18:9);WRITE(LNFIL,QUAN[1].IM:18:9);
      WRITE(LNFIL,QUAN[1].MAG:18:9);WRITE(LNFIL,QUAN[1].ANG:18:9);
      XYTOCOMPLEX(0.0,LCHRF,QUAN[1]);
      WRITE(LNFIL,QUAN[1].RE:18:9);WRITE(LNFIL,QUAN[1].IM:18:9);
      WRITE(LNFIL,QUAN[1].MAG:18:9);WRITELN(LNFIL,QUAN[1].ANG:18:9);
    END;
  WRITELN;
UNTIL BSI = -1;
WRITE(LNFIL,-1:4);
CLOSE(LNFIL);
WRITELN;
WRITELN('TRANSFORMER DATA : Quit by enter Bus(p) with "-1"');
WRITELN('( Bus(p) , Bus(q) , Tap(p->q) , Ztr(Im) )');
WRITELN;
REPEAT
  WRITE('Bus(p)  : ');
  READINT(BSI);
  IF BSI <> -1 THEN
    BEGIN
      WRITE('Bus(q)  : ');
      READINT(BSJ);
      WRITE('Tap(p-q): ');
      READREAL(TAPIJ);
      WRITE('Ztr(Im) : ');
      READREAL(TIMPF);
      XYTOCOMPLEX(0.0,TIMPF,QUAN[1]);
      WRITE(TRFIL,BSI:4);WRITE(TRFIL,BSJ:4);WRITE(TRFIL,TAPIJ:18:9);
      WRITE(TRFIL,QUAN[1].RE:18:9);WRITE(TRFIL,QUAN[1].IM:18:9);
      WRITE(TRFIL,QUAN[1].MAG:18:9);WRITELN(TRFIL,QUAN[1].ANG:18:9);
    END;
  WRITELN;
UNTIL BSI = -1;
WRITE(TRFIL,-1:4);
CLOSE(TRFIL);
WRITELN;
WRITELN('SHUNT CAPACITOR DATA : Quit by enter Bus(p) with "-1"');
WRITELN('( Bus(p) , Ysh(Im) )');
WRITELN;
```

```
  REPEAT
    WRITE('Bus(p)  : ');
    READINT(BSI);
    IF BSI <> -1 THEN
      BEGIN
        WRITE('Ysh(Im) : ');
        READREAL(YSH);
        XYTOCOMPLEX(0.0,YSH,QUAN[1]);
        WRITE(SHFIL,BSI:4);
        WRITE(SHFIL,QUAN[1].RE:18:9);WRITE(SHFIL,QUAN[1].IM:18:9);
        WRITE(SHFIL,QUAN[1].MAG:18:9);WRITELN(SHFIL,QUAN[1].ANG:18:9);
      END;
    WRITELN;
  UNTIL BSI = -1;
  WRITE(SHFIL,-1:4);
  CLOSE(SHFIL);
END;  {ENTER1}

BEGIN  {MAIN PROGRAM}
  GG := 'GEN14.DAT';
  BB := 'BUS14.DAT';
  LL := 'LINE14.DAT';
  TT := 'TRAN14.DAT';
  LD := 'LOAD14.DAT';
  SH := 'SHUNT14.DAT';
  CLRSCR;
  CHKFILE(GEN_FIL,GG);
  CHKFILE(BUS_FIL,BB);
  CHKFILE(LINE_FIL,LL);
  CHKFILE(TRAN_FIL,TT);
  CHKFILE(LOAD_FIL,LD);
  CHKFILE(SHUNT_FIL,SH);
  ENTER1(GEN_FIL,BUS_FIL,LINE_FIL,TRAN_FIL,LOAD_FIL,SHUNT_FIL);
END.  {MAIN PROGRAM}
```

```
PROGRAM  LDNWTON; (LOAD FLOW PROGRAM BY NEWTON-RAPHSON METHOD)
{$R+}                  { Enable range checking }
{$I-}                  { Disable I/O checking }
{$M 65520,0,655360}

USES
  TOOLS,MATRIX,DOS,CRT,COMMON;
TYPE  CMPXMATRX  = ARRAY [1..30,1..30] OF COMPLEX;
      CMPXVECTOR = ARRAY [1..30] OF COMPLEX;
      GEN        = RECORD
                     BUS_NO   : INTEGER;
                     H        : REAL;
                     XD       : REAL;
                   END;
      BUS        = RECORD
                     BUS_NO   : INTEGER;
                     VOLT     : COMPLEX;
                     SGEN     : COMPLEX;
                     SLOAD    : COMPLEX;
                   END;
      LINE       = RECORD
                     BUSI     : INTEGER;
                     BUSJ     : INTEGER;
                     LINEIMP  : COMPLEX;
                     LINECHRG : COMPLEX;
                   END;
      TRFM       = RECORD
                     BUSI     : INTEGER;
                     BUSJ     : INTEGER;
                     TAPI_J   : REAL;
                     TRIMP    : COMPLEX;
                   END;
      QCAL       = RECORD
                     BS       : INTEGER;
                     QCL      : REAL;
                   END;
      JACOBIAN   = ARRAY [1..60,1..60] OF REAL;
      VECTOR     = ARRAY [1..60] OF REAL;
      INTVECTOR  = ARRAY [1..60] OF INTEGER;
      QCLVECTOR  = ARRAY [1..10] OF QCAL;
 VAR  YBUS                                  : CMPXMATRX;
      VTRAN,SGTRAN,SLTRAN,ADD               : CMPXVECTOR;
      JACOB                                 : JACOBIAN;
      V_VEC,S_VEC,SCON,SGQMAX,SGQMIN,
      SLQ,VMAG,VBS                          : VECTOR;
      INDX,TYP,STAT                         : INTVECTOR;
      BFILE,LFILE,TFILE,LST,GFILE,YB,SFILE : TEXT;
      GNREC                                 : GEN;
      BSREC                                 : BUS;
      LNREC                                 : LINE;
      TRREC                                 : TRFM;
      S,II,JJ,MVA,SWING,BSN,BI,BJ,INN,CT    : INTEGER;
      SS,SGP,MAX,TAP,XXX,VB                 : REAL;
      UNT                                   : COMPLEX;
      QC                                    : QCLVECTOR;
```

```
PROCEDURE  BUSNUMBER(VAR BSFILE : TEXT;VAR SIZ : INTEGER);
VAR  CHK  : REAL;
BEGIN
  SIZ := 0;
  REPEAT
    READLN(BSFILE,CHK);
    IF CHK <> -1 THEN
      SIZ := SIZ+1;
  UNTIL CHK = -1;
END;

PROCEDURE  FORMYBUS(VAR LNFILE : TEXT;
                    VAR TRFILE : TEXT;
                    VAR YBS    : CMPXMATRX;
                        SZ     : INTEGER;
                    VAR SFILE  : TEXT);
VAR  LREC    : LINE;
     TREC    : TRFM;
     CON     : REAL;
     I,J,BS  : INTEGER;
     UNI     : COMPLEX;
BEGIN
  XYTOCOMPLEX(1,0,UNI);
  REPEAT
    READ(LNFILE,LREC.BUSI);
    IF LREC.BUSI <> -1 THEN
      BEGIN
        READ(LNFILE,LREC.BUSJ);
        READ(LNFILE,LREC.LINEIMP.RE);READ(LNFILE,LREC.LINEIMP.IM);
        READ(LNFILE,LREC.LINEIMP.MAG);READ(LNFILE,LREC.LINEIMP.ANG);
        READ(LNFILE,LREC.LINECHRG.RE);READ(LNFILE,LREC.LINECHRG.IM);
        READ(LNFILE,LREC.LINECHRG.MAG);READLN(LNFILE,LREC.LINECHRG.ANG);
        DEVIDE(UNI,LREC.LINEIMP,QUAN[1]);
        MULTICMPX(-1,QUAN[1],QUAN[1]);
        SUM(YBS[LREC.BUSI,LREC.BUSJ],QUAN[1],YBS[LREC.BUSI,LREC.BUSJ]);
        YBS[LREC.BUSJ,LREC.BUSI] := YBS[LREC.BUSI,LREC.BUSJ];
        SUM(YBS[LREC.BUSI,LREC.BUSI],LREC.LINECHRG,YBS[LREC.BUSI,LREC.BUSI]);
        SUM(YBS[LREC.BUSJ,LREC.BUSJ],LREC.LINECHRG,YBS[LREC.BUSJ,LREC.BUSJ]);
      END;
  UNTIL LREC.BUSI = -1;
  REPEAT
    READ(SFILE,BS);
    IF BS <> -1 THEN
    BEGIN
      READ(SFILE,QUAN[1].RE);READ(SFILE,QUAN[1].IM);
      READ(SFILE,QUAN[1].MAG);READLN(SFILE,QUAN[1].ANG);
      SUM(YBS[BS,BS],QUAN[1],YBS[BS,BS]);
    END;
  UNTIL BS = -1;
  REPEAT
    READ(TRFILE,TREC.BUSI);
    IF TREC.BUSI <> -1 THEN
      BEGIN
        READ(TRFILE,TREC.BUSJ);READ(TRFILE,TREC.TAPI_J);
        READ(TRFILE,TREC.TRIMP.RE);READ(TRFILE,TREC.TRIMP.IM);
        READ(TRFILE,TREC.TRIMP.MAG);READ(TRFILE,TREC.TRIMP.ANG);
```

```
        QUAN[1] := TREC.TRIMP;
        CON := -1/TREC.TAPI_J;
        DEVIDE(UNI,QUAN[1],QUAN[2]);  {QUAN[2] = Ypq}
        MULTICMPX(CON,QUAN[2],YBS[TREC.BUSI,TREC.BUSJ]);
        YBS[TREC.BUSJ,TREC.BUSI] := YBS[TREC.BUSI,TREC.BUSJ];
        CON := 1/TREC.TAPI_J*(1/TREC.TAPI_J-1);
        MULTICMPX(CON,QUAN[2],QUAN[1]);  {QUAN[1] = (B)}
        SUM(YBS[TREC.BUSI,TREC.BUSI],QUAN[1],YBS[TREC.BUSI,TREC.BUSI]);
        CON := 1-1/TREC.TAPI_J;
        MULTICMPX(CON,QUAN[2],QUAN[1]);  {QUAN[1] = (C)}
        SUM(YBS[TREC.BUSJ,TREC.BUSJ],QUAN[1],YBS[TREC.BUSJ,TREC.BUSJ]);
      END;
  UNTIL TREC.BUSI = -1;
  FOR I := 1 TO SZ DO
    BEGIN
      FOR J := 1 TO SZ DO
        BEGIN
          IF (YBS[I,J].MAG <> 0) AND (J <> I) THEN
            SUB(YBS[I,I],YBS[I,J],YBS[I,I]);
        END;
    END;
END;  {FORM YBUS[I,J] = G[I,J] - JB[I,J]}

PROCEDURE  CALCULATEdP(VAR SVEC          : VECTOR;
                       VAR SCON          : VECTOR;
                           VVEC          : VECTOR;
                           YBS           : CMPXMATRX;
                           SWNO          : INTEGER;
                           S             : INTEGER;
                           TYP           : INTVECTOR;
                           SGQMAX,SGQMIN : VECTOR;
                       VAR STAT          : INTVECTOR;
                           SLQ           : VECTOR;
                           MVA           : INTEGER;
                       VAR CT            : INTEGER;
                       VAR QC            : QCLVECTOR);
VAR  I,J       : INTEGER;
BEGIN
  CT := 0;
  FOR I := 1 TO S DO
    BEGIN
      IF I <> SWNO THEN
        BEGIN
          SVEC[I] := 0;SVEC[I+S] := 0;
          FOR J := 1 TO S DO
          BEGIN
      SVEC[I] := SVEC[I]+VVEC[I]*(VVEC[J]*YBS[I,J].RE+VVEC[J+S]*-YBS[I,J].IM)
                 +VVEC[I+S]*(VVEC[J+S]*YBS[I,J].RE-VVEC[J]*-YBS[I,J].IM);
SVEC[I+S] := SVEC[I+S]+VVEC[I+S]*(VVEC[J]*YBS[I,J].RE+VVEC[J+S]*-YBS[I,J].IM)
                  -VVEC[I]*(VVEC[J+S]*YBS[I,J].RE-VVEC[J]*-YBS[I,J].IM);
          END;
          STAT[I] := 0; {0 = P-Q BUS}
          IF TYP[I] = 1 THEN
          BEGIN
            CT := CT+1;
            QC[CT].BS := I;
```

```
          IF (SVEC[I+S] <= SGQMAX[I]/MVA) AND
             (SVEC[I+S] >= SGQMIN[I]/MVA) THEN
          BEGIN
            SCON[I+S] := SQR(VMAG[I]);
            STAT[I] := 1; {1 = VOLTAGE CONTROLLED BUS}
            SVEC[I+S] := SQR(VVEC[I])+SQR(VVEC[I+S]);
          END
          ELSE
          BEGIN
            IF SVEC[I+S] > SGQMAX[I]/MVA THEN
            BEGIN
              SCON[I+S] := (SGQMAX[I]-SLQ[I])/MVA;
            END
            ELSE
            BEGIN
              SCON[I+S] := (SGQMIN[I]-SLQ[I])/MVA;
            END;
          END;
          SVEC[I] := SCON[I]-SVEC[I];
          SVEC[I+S] := SCON[I+S]-SVEC[I+S];
        END
      ELSE
        BEGIN
          SVEC[I] := 0;
          SVEC[I+S] := 0;
        END;
    END;
END;

PROCEDURE  CALCJACB(VAR JACB  : JACOBIAN;
                        VVEC  : VECTOR;
                        YBS   : CMPXMATRX;
                        SWNO  : INTEGER;
                        S     : INTEGER;
                        STAT  : INTVECTOR);
VAR  I,J,K,N    : INTEGER;
     TT         : REAL;
BEGIN
  FOR I := 1 TO S DO  {FOR J1}
  BEGIN
    IF I <> SWNO THEN
    BEGIN
      FOR J := 1 TO S DO
      BEGIN
        IF J <> SWNO THEN
        BEGIN
          IF I = J THEN  {DIAGONAL ELEMENT OF J1}
          BEGIN
            TT := 0;
            FOR K := 1 TO S DO
            BEGIN
              IF I <> K THEN
                TT := TT+VVEC[K]*YBS[I,K].RE+VVEC[K+S]*-YBS[I,K].IM;
            END;
            JACB[I,J] := 2*VVEC[I]*YBS[I,I].RE+TT;
```

```
        END
        ELSE  {OFF-DIAGONAL ELEMENT OF J1}
          JACB[I,J] := VVEC[I]*YBS[I,J].RE-VVEC[I+S]*-YBS[I,J].IM;
      END
      ELSE
        JACB[I,J] := 0;
    END;
  END
  ELSE
  BEGIN
    FOR J := 1 TO S DO
      JACB[I,J] := 0;
  END;
END;
FOR I := 1 TO S DO  {FOR J2}
BEGIN
  IF I <> SWNO THEN
  BEGIN
    FOR J := S+1 TO 2*S DO
    BEGIN
      IF J <> SWNO+S THEN
      BEGIN
        IF I+S = J THEN  {DIAGONAL ELEMENT OF J2}
        BEGIN
          TT := 0;
          FOR K := S+1 TO 2*S DO
          BEGIN
            IF I+S <> K THEN
              TT := TT+VVEC[K]*YBS[I,K-S].RE-VVEC[K-S]*-YBS[I,K-S].IM;
          END;
          JACB[I,J] := 2*VVEC[I+S]*YBS[I,I].RE+TT;
        END
        ELSE  {OFF-DIAGONAL ELEMENT OF J2}
          JACB[I,J] := VVEC[I]*-YBS[I,J-S].IM+VVEC[I+S]*YBS[I,J-S].RE;
      END
      ELSE
        JACB[I,J] := 0;
    END;
  END
  ELSE
  BEGIN
    FOR J := S+1 TO 2*S DO
      JACB[I,J] := 0;
  END;
END;
FOR I := S+1 TO 2*S DO  {J3}
BEGIN
  N := I-S;
  IF I <> SWNO+S THEN
  BEGIN
    FOR J := 1 TO S DO
    BEGIN
      IF J <> SWNO THEN
      BEGIN
        IF STAT[N] = 0 THEN
        BEGIN
```

```
          IF I-S = J THEN  {DIAGONAL ELEMENT OF J3}
          BEGIN
            TT := 0;
            FOR K := 1 TO S DO
            BEGIN
              IF I-S <> K THEN
                TT := TT+VVEC[K+S]*YBS[I-S,K].RE-VVEC[K]*-YBS[I-S,K].IM;
            END;
            JACB[I,J] := 2*VVEC[I-S]*-YBS[I-S,I-S].IM-TT;
          END
          ELSE  {OFF-DIAGONAL ELEMENT OF J3}
            JACB[I,J] := VVEC[I-S]*-YBS[I-S,J].IM+VVEC[I]*YBS[I-S,J].RE;
        END
        ELSE
        BEGIN
          IF I-S = J THEN  {DIAGONAL ELEMENT OF J5 AND P-V BUS}
            JACB[I,J] := 2*VVEC[I-S]
          ELSE  {OFF-DIAGONAL ELEMENT OF J5 AND P-V BUS}
            JACB[I,J] := 0;
        END;
      END
      ELSE
        JACB[I,J] := 0;
    END;
  END
  ELSE
  BEGIN
    FOR J := 1 TO S DO
      JACB[I,J] := 0;
  END;
END;
FOR I := S+1 TO 2*S DO  {J4}
BEGIN
  N := I-S;
  IF I <> SWNO+S THEN
  BEGIN
    FOR J := S+1 TO 2*S DO
    BEGIN
      IF J <> SWNO+S THEN
      BEGIN
        IF STAT[N] = 0 THEN
        BEGIN
          IF I = J THEN  {DIAGONAL ELEMENT OF J4}
          BEGIN
            TT := 0;
            FOR K := S+1 TO 2*S DO
            BEGIN
              IF I <> K THEN
                TT := TT+VVEC[K-S]*YBS[I-S,K-S].RE+VVEC[K]*-YBS[I-S,K-S].IM;
            END;
            JACB[I,J] := 2*VVEC[I]*-YBS[I-S,I-S].IM+TT;
          END
          ELSE  {OFF-DIAGONAL ELEMENT OF J4}
            JACB[I,J] := -VVEC[I-S]*YBS[I-S,J-S].RE+VVEC[I]*-YBS[I-S,J-S].IM;
        END
        ELSE
```

```
        BEGIN
          IF I = J THEN  {DIAGONAL ELEMENT OF J6 AND P-V BUS}
            JACB[I,J] := 2*VVEC[I]
          ELSE  {OFF-DIAGONAL ELEMENT OF J6 AND P-V BUS}
            JACB[I,J] := 0;
        END;
      END
      ELSE
        JACB[I,J] := 0;
    END;
  END
  ELSE
  BEGIN
    FOR J := S+1 TO 2*S DO
      JACB[I,J] := 0;
  END;
 END;
END;

PROCEDURE  ARRAYINGFILE1(VAR BFL          : TEXT;
                             YBS          : CMPXMATRX;
                         VAR JACB         : JACOBIAN;
                VAR SVEC         : VECTOR;
                         VAR VVEC         : VECTOR;
                    S            : INTEGER;
                VAR SCON         : VECTOR;
                    SWNO         : INTEGER;
                             TYP          : INTVECTOR;
                             SGQMAX,SGQMIN : VECTOR;
                         VAR STAT         : INTVECTOR;
                             SLQ          : VECTOR;
                             MVA          : INTEGER);
BEGIN
  CALCULATEdP(SVEC,SCON,VVEC,YBS,SWNO,S,TYP,SGQMAX,SGQMIN,STAT,SLQ,MVA,CT,QC);
  CALCJACB(JACB,VVEC,YBS,SWNO,S,STAT);
END;

procedure Abort;
  begin
    Window(1, 1, 80, 25);
    NormVideo;
    ClrEol;
    GotoXY(1, 25);
    Write('Program terminated by user.');
    Halt;
  end; { Abort }

PROCEDURE  SOLVE(VAR V_VEC : VECTOR);
var
  Dimen : integer;          { Dimen of the square matrix }
  Coefficients : TNmatrix;  { The matrix }
  Constants : TNvector;     { Constant terms in the equations }
  Solution : TNvector;      { Solution to the set of equations }
  Error : byte;             { Flags if something went wrong }

  procedure Initial(var Dimen       : integer;
```

```
                var Coefficients : TNmatrix;
                var Constants    : TNvector);

begin
  Dimen := 0;
  FillChar(Coefficients, SizeOf(Coefficients), 0);
  FillChar(Constants, SizeOf(Constants), 0);
end; { procedure Initial }

procedure GetDataFromFile(var Dimen        : integer;
                          var Coefficients : TNmatrix;
                          var Constants    : TNvector);

var
  Row1,Column1,SWCHK1,SWCHK2,ROW2,COLUMN2,I,J : integer;

begin
  SWCHK1 := SWING;
  Row1 := 0;ROW2 := 0;
  while Row1 < 2*Dimen-2 do
  begin
    Row1 := Succ(Row1);ROW2 := SUCC(ROW2);
    IF ROW2 >= SWCHK1 THEN
    BEGIN
      SWCHK1 := SWCHK1+S;
      ROW2 := SUCC(ROW2);
    END;
    SWCHK2 := SWING;
    Column1 := 0;COLUMN2 := 0;
    while Column1 < 2*Dimen-2 do
    begin
      Column1 := Succ(Column1);Column2 := Succ(Column2);
      IF COLUMN2 >= SWCHK2 THEN
      BEGIN
        SWCHK2 := SWCHK2+S;
        COLUMN2 := SUCC(COLUMN2);
      END;
      Coefficients[Row1, Column1] := JACOB[ROW2,COLUMN2];
    end;
  end;
  SWCHK1 := SWING;
  Row1 := 0;Row2 := 0;
  while Row1 < 2*Dimen-2 do
  begin
    Row1 := Succ(Row1);Row2 := Succ(Row2);
    IF ROW2 >= SWCHK1 THEN
    BEGIN
      ROW2 := SUCC(ROW2);
      SWCHK1 := SWCHK1+S;
    END;
    Constants[Row1] := S_VEC[ROW2];
  end;
end; { procedure GetDataFromFile }

procedure Results(Dimen        : integer;
            var Coefficients : TNmatrix;
```

```
                var Constants    : TNvector;
                var Solution     : TNvector;
                    Error        : byte;
                VAR V_VEC        : VECTOR);

  var
    ROW2,Row1,SWCHK1 : integer;

  begin
    case Error of
      0 : begin
            SWCHK1 := SWING;
            Row1 := 0;Row2 := 0;
            while Row1 < 2*Dimen do
            begin
              Row1 := Succ(Row1);Row2 := Succ(Row2);
              IF ROW1 = SWCHK1 THEN
              BEGIN
                ROW1 := SUCC(ROW1);
                SWCHK1 := SWCHK1+S;
              END;
              V_VEC[ROW1] := V_VEC[ROW1]+SOLUTION[ROW2];
            END;
          END;

      1 : BEGIN
            Writeln('The dimension of the matrix must be greater than 1'
                    ,' in LDNWTON.PAS');
            READLN;
            ABORT;
          END;

      2 : BEGIN
            Writeln('There is no solution to this set of equations'
                    ,' in LDNWTON.PAS');
            READLN;
            ABORT;
          END;

    end; { case }
  end; { procedure Results }

begin { program Partial_Pivoting }
  ClrScr;
  Initial(Dimen, Coefficients, Constants);
  DIMEN := S;
  GetDataFromFile(DIMEN, Coefficients, Constants);
  Partial_Pivoting(DIMEN*DIMEN-2, Coefficients, Constants, Solution, Error);
  Results(Dimen, Coefficients, Constants, Solution, Error,V_VEC);
end; { program Partial_Pivoting }

BEGIN  {MAIN}
  ASSIGN(BFILE,'D:\tp5\BUS14.DAT');
  RESET(BFILE);
  BUSNUMBER(BFILE,S);
  RESET(BFILE);
```

```
ASSIGN(LFILE,'D:\tp5\LINE14.DAT');
RESET(LFILE);
ASSIGN(GFILE,'D:\tp5\GEN14.DAT');
RESET(GFILE);
READLN(GFILE,MVA);
ASSIGN(TFILE,'D:\tp5\TRAN14.DAT');
RESET(TFILE);
ASSIGN(SFILE,'D:\tp5\SHUNT14.DAT');
RESET(SFILE);
FOR II := 1 TO S DO
  BEGIN
    FOR JJ := 1 TO S DO
      BEGIN
        YBUS[II,JJ].RE := 0;
        YBUS[II,JJ].IM := 0;
        YBUS[II,JJ].MAG := 0;
        YBUS[II,JJ].ANG := 0;
      END;
  END;
FORMYBUS(LFILE,TFILE,YBUS,S,SFILE);
ASSIGN(YB,'D:\TP5\YBUS14.DAT');REWRITE(YB);
WRITELN(YB,S);
FOR II := 1 TO S DO
  BEGIN
    FOR JJ := 1 TO S DO
      BEGIN
        IF (YBUS[II,JJ].RE <> 0) OR (YBUS[II,JJ].IM <> 0) OR
           (YBUS[II,JJ].MAG <> 0) OR (YBUS[II,JJ].ANG <> 0) THEN
        BEGIN
          WRITE(YB,II:3);WRITE(YB,JJ:3);WRITE(YB,' ');
          WRITE(YB,YBUS[II,JJ].RE);WRITE(YB,' ');WRITE(YB,YBUS[II,JJ].IM);
          WRITE(YB,' ');WRITE(YB,YBUS[II,JJ].MAG);WRITE(YB,' ');
          WRITELN(YB,YBUS[II,JJ].ANG);
        END;
      END;
  END;
WRITELN(YB,-1);
CLOSE(YB);
RESET(LFILE);RESET(TFILE);
FOR II := 1 TO 2*S DO
  BEGIN
    FOR JJ := 1 TO 2*S DO
BEGIN
      JACOB[II,JJ] := 0;
    END;
  V_VEC[II] := 0;
  END;
FOR II := 1 TO S DO
BEGIN
  READ(BFILE,INN);
  READ(BFILE,XXX);
  READ(BFILE,BSN);
  TYP[BSN] := INN;
  READ(BFILE,V_VEC[BSN]);READ(BFILE,V_VEC[BSN+S]);READ(BFILE,VMAG[BSN]);
  READ(BFILE,SS);READ(BFILE,SGP);READ(BFILE,SGQMAX[BSN]);
  READ(BFILE,SGQMIN[BSN]);READ(BFILE,SCON[BSN]);READ(BFILE,SLQ[BSN]);
```

```
  READLN(BFILE);
  SCON[BSN] := (SGP-SCON[BSN])/MVA;
  IF INN <> 1 THEN
    SCON[BSN+S] := -SLQ[BSN]/MVA;
  IF II = 1 THEN
  BEGIN
    SWING := BSN;
    SCON[BSN] := 0;
    SCON[BSN+S] := 0;
  END;
END;
AREAYINGFILE1(BFILE,YBUS,JACOB,S_VEC,V_VEC,S,SCON,SWING,TYP,SGQMAX,
              SGQMIN,STAT,SLQ,MVA);
SOLVE(V_VEC);
JJ := 0;
REPEAT
  JJ := JJ+1;
  IF JJ > 100 THEN        {MAXIMUM 100 ITERATIONS}
  BEGIN
    WRITE('THIS DATA DOES NOT CONVERGE ON LDNWTON.PAS');
    READLN;
    ABORT;
  END;
  CALCULATEdP(S_VEC,SCON,V_VEC,YBUS,SWING,S,TYP,SGQMAX,SGQMIN,STAT,SLQ,
              MVA,CT,QC);
  MAX := ABS(S_VEC[1]);
  FOR II := 1 TO 2*S DO
  BEGIN
    IF ABS(S_VEC[II]) > MAX THEN
    BEGIN {}
      BI := II; {}
MAX := ABS(S_VEC[II]);
    END; {}
  END;
  IF MAX > 0.0001 THEN
  BEGIN
    CALCJACB(JACOB,V_VEC,YBUS,SWING,S,STAT);
    SOLVE(V_VEC);
  END;
UNTIL MAX <= 0.0001;
FOR II := 1 TO S DO
 BEGIN
   IF (II = SWING) OR (TYP[II] = 1) THEN
     BEGIN
       S_VEC[II] := 0;S_VEC[II+S] := 0;
       FOR JJ := 1 TO S DO
       BEGIN
           S_VEC[II] := S_VEC[II]+V_VEC[II]*(V_VEC[JJ]*YBUS[II,JJ].RE+
                        V_VEC[JJ+S]*-YBUS[II,JJ].IM)+V_VEC[II+S]*
                     (V_VEC[JJ+S]*YBUS[II,JJ].RE-V_VEC[JJ]*-YBUS[II,JJ].IM);
           S_VEC[II+S] := S_VEC[II+S]+V_VEC[II+S]*(V_VEC[JJ]*YBUS[II,JJ].RE+
                          V_VEC[JJ+S]*-YBUS[II,JJ].IM)-V_VEC[II]*
                       (V_VEC[JJ+S]*YBUS[II,JJ].RE-V_VEC[JJ]*-YBUS[II,JJ].IM);
       END;
       XYTOCOMPLEX(S_VEC[II],S_VEC[II+S],ADD[II]);
     END;
```

```
  END;
 RESET(BFILE);
 II := 0;
 REPEAT
   READ(BFILE,SS);
   IF SS <> -1 THEN
   BEGIN
     READ(BFILE,VB);
     READ(BFILE,BI);
     II := II+1;
     INDX[II] := BI;
     VBS[II] := VB;
     READ(BFILE,VTRAN[II].RE);READ(BFILE,VTRAN[II].IM);
     READ(BFILE,VTRAN[II].MAG);READ(BFILE,VTRAN[II].ANG);
     READ(BFILE,SGTRAN[II].RE);READ(BFILE,SGTRAN[II].IM); {Pg = SGTRAN.RE}
     READ(BFILE,SGTRAN[II].MAG);READ(BFILE,SLTRAN[II].RE); {Qmax = SGTRAN.IM}
     READ(BFILE,SLTRAN[II].IM);READ(BFILE,SLTRAN[II].MAG); {Qmin = SGTRAN.MAG}
     READLN(BFILE,SLTRAN[II].ANG);
   END;
 UNTIL SS = -1;
 ASSIGN(BFILE,'D:\TP5\BSTRN14.DAT');REWRITE(BFILE);
 FOR II := 1 TO S DO
 BEGIN
   JJ := 0;
   REPEAT
     JJ := JJ+1;
   UNTIL (INDX[II] = QC[JJ].BS) OR (JJ > CT);
   WRITE(BFILE,TYP[INDX[II]]:4);WRITE(BFILE,' ');
   WRITE(BFILE,VBS[II]:6:2);WRITE(BFILE,INDX[II]:4);WRITE(BFILE,' ');
   IF INDX[II] = SWING THEN
     MULTICMPX(MVA,ADD[SWING],SGTRAN[II]); {ADD = P(slack) - jQ(slack)}
   XYTOCOMPLEX(V_VEC[INDX[II]],V_VEC[INDX[II]+S],VTRAN[II]);
   WRITE(BFILE,VTRAN[II].RE);WRITE(BFILE,' ');WRITE(BFILE,VTRAN[II].IM);
   WRITE(BFILE,' ');WRITE(BFILE,VTRAN[II].MAG);WRITE(BFILE,' ');
   WRITE(BFILE,VTRAN[II].ANG);WRITE(BFILE,' ');
   WRITE(BFILE,SGTRAN[II].RE/MVA);
   IF JJ > CT THEN
   BEGIN
     WRITE(BFILE,' ');WRITE(BFILE,SGTRAN[II].IM/MVA);WRITE(BFILE,' ');
     WRITE(BFILE,SGTRAN[II].IM/MVA);WRITE(BFILE,' ');
   END
   ELSE
   BEGIN
     MULTICMPX(MVA,ADD[INDX[II]],ADD[INDX[II]]);
     SUM(ADD[INDX[II]],SLTRAN[II],QUAN[1]);
     WRITE(BFILE,' ');WRITE(BFILE,QUAN[1].IM/MVA);WRITE(BFILE,' ');
     WRITE(BFILE,QUAN[1].IM/MVA);WRITE(BFILE,' ');
   END;
   WRITE(BFILE,SLTRAN[II].RE/MVA);
   WRITE(BFILE,' ');WRITE(BFILE,SLTRAN[II].IM/MVA);WRITE(BFILE,' ');
   WRITE(BFILE,SLTRAN[II].MAG/MVA);WRITE(BFILE,' ');
   WRITELN(BFILE,SLTRAN[II].ANG);
 END;
 WRITELN(BFILE,-1:4);
 CLOSE(BFILE);CLOSE(LFILE);CLOSE(GFILE);CLOSE(TFILE);
 WRITE('SUCCESSFUL');READLN;END.
```

```
PROGRAM  STAB: (TRANSIENT STABILITY PROGRAM)
{$N+}
USES
  TOOLS,MATRIX,COMMON,CRT;
{$R+}                  { Enable range checking }
{$I-}                  { Disable I/O checking }

TYPE  CMPXMATRX  = ARRAY [1..30,1..30] OF COMPLEX;
      CMPXVECTOR = ARRAY [1..40] OF COMPLEX;
      GEN1       = RECORD
                     BUS_NO   : INTEGER;
                     MVAB     : REAL;
                     H        : REAL;
                     XD       : REAL;
                   END;
      GEN2       = RECORD
                     BUS_NO   : INTEGER;
                     MVAB     : REAL;
                     H        : REAL;
                     XD0      : REAL;
                     XQ0      : REAL;
                     XD1      : REAL;
                     XQ1      : REAL;
                     XD2      : REAL;
                     XQ2      : REAL;
                     XLS      : REAL;
                     TD1      : REAL;
                     TQ1      : REAL;
                     TD2      : REAL;
                     TQ2      : REAL;
                   END;
      BUS        = RECORD
                     BUS_NO   : INTEGER;
                     VOLT     : COMPLEX;
                     SGEN     : COMPLEX;
                     SLOAD    : COMPLEX;
                   END;
      LINE       = RECORD
                     BUSI     : INTEGER;
                     BUSJ     : INTEGER;
                     LINEIMP  : COMPLEX;
                     LINECHRG : COMPLEX;
                   END;
      TRFM       = RECORD
                     BUSI     : INTEGER;
                     BUSJ     : INTEGER;
                     TAPI_J   : REAL;
                     TRIMP    : COMPLEX;
                   END;
      LOADADM    = RECORD
                     BUSI     : INTEGER;
                     LDADM    : COMPLEX;
                   END;
      LINEPARA   = RECORD
                     BI       : INTEGER;
                     BJ       : INTEGER;
```

```
                    YLPQ     : COMPLEX;
                  END;
    LDVECTOR   = ARRAY [1..25] OF LOADADM;
    GNVECTOR1  = ARRAY [1..10] OF GEN1;
    GNVECTOR2  = ARRAY [1..10] OF GEN2;
    VECTOR     = ARRAY [1..30] OF REAL;
    INTVECTOR  = ARRAY [1..20] OF INTEGER;
    LNVECTOR   = ARRAY [1..60] OF LINEPARA;
    SWVECTOR   = ARRAY [1..10] OF REAL;
    COEFMATRX  = ARRAY [1..30,1..5] OF REAL;
VAR YBUS                                        : CMPXMATRX;
    YL                                          : LNVECTOR;
    VTRAN,SGTRAN,SLTRAN,SLTRNOLD                : CMPXVECTOR;
    V_VEC,S_VEC,SCON,SGQMAX,SGQMIN,SLQ,STC      : VECTOR;
    BS,RR,XM,XS,H,PDS,PQS,PDR,PQR               : VECTOR;
    FDS,FQS,FK1,FK2,FKD,FFD,EXFD,TI,TE          : SWVECTOR;
    TF,TA1,KA,SKF,VREF,V1,DELTA,VASMAG          : SWVECTOR;
    TL,WR,DV,PLCON,QLCON,VBUS                   : VECTOR;
    A1,A2,A3,A4,A5,A6,A7,A8,A9                  : SWVECTOR;
    B1,B2,B3,B4,B5,B6,B7,B8,B9                  : SWVECTOR;
    RSS,RKQ1,RKQ2,RKD,RFD,XMD,XMQ,ZR,WRG       : SWVECTOR;
    BFILE,LFILE,TFILE,LDFILE,PFILE,QFILE        : TEXT;
    GFILE,YBS,ANGFILE,WRFILE,VFILE,LST          : TEXT;
    S,T2,T,TT,I,J,LL,KK,BSF,K,CNT,MVA           : INTEGER;
    CHKONE,FREQ,BF,CHKLD,TYPL,CHKGN             : INTEGER;
    GNARR1                                      : GNVECTOR1;
    GNARR2                                      : GNVECTOR2;
    IND,BSL                                     : INTVECTOR;
    ANGFL,WRFL,VFL,PFL,QFL                      : STRING;
    TIM,DT,TIM_END,TIMCLR                       : REAL;
    W,ANG,DANG                                  : SWVECTOR;
    COEF                                        : COEFMATRX;
    LDAD                                        : LDVECTOR;

FUNCTION  GENNUMBER(VAR GNFILE : TEXT) : INTEGER;
VAR  CHK  : REAL;
     XX   : INTEGER;
BEGIN
  XX := 0;
  REPEAT
    READLN(GNFILE,CHK);
    IF CHK <> -1 THEN
      XX := XX+1;
  UNTIL CHK = -1;
  GENNUMBER := XX;
END;

PROCEDURE  CALyp01(VAR LDAD   : LDVECTOR;
                   VAR KK     : INTEGER); {KK = # OF y(p0)}
VAR  I  : INTEGER;
BEGIN
  KK := 0;
  FOR I := 1 TO S DO   {CALCULATE y(p0)}
  BEGIN
     KK := KK+1;
     LDAD[KK].BUSI := I;
```

```
        CONJUGATE(SLTRAN[I],QUAN[1]);
        MULTICMPX(1/(SQR(VTRAN[I].MAG)),QUAN[1],LDAD[KK].LDADM);
    END;
END;

PROCEDURE  CALyp02(VAR LDAD   : LDVECTOR);
VAR  I    : INTEGER;
BEGIN
  FOR I := 1 TO S DO   {CALCULATE y(p0)}
  BEGIN
    IF ABS(SLTRNOLD[I].MAG) < 1E-15 THEN
    BEGIN
      XYTOCOMPLEX(0,0,LDAD[I].LDADM);
    END
    ELSE
    BEGIN
      DEVIDE(LDAD[I].LDADM,SLTRNOLD[I],LDAD[I].LDADM);
      PRODUCT(LDAD[I].LDADM,SLTRAN[I],LDAD[I].LDADM);
    END;
  END;
END;

PROCEDURE READINT(VAR VALUE : INTEGER);
VAR     STRVALUE : STRING[10];
        STRCHK   : INTEGER;
BEGIN
  READLN(STRVALUE);
  VAL(STRVALUE,VALUE,STRCHK);
  WHILE STRCHK <> 0 DO
    BEGIN
      WRITE('DATA TYPE MISMATCH! ...PLEASE TRY AGAIN : ');
      READLN(STRVALUE);
      VAL(STRVALUE,VALUE,STRCHK);
    END;
END;  {READINT}

PROCEDURE  CHKFILE(VAR FIL : TEXT;VAR FILENAME : STRING);
VAR  CH : CHAR;
BEGIN
  REPEAT
    CH := 'Y';
    ASSIGN(FIL,FILENAME);
{$I-} RESET(FIL); {$I+}
    IF IORESULT = 0 THEN  { The file already exists. }
      BEGIN
        CLOSE(FIL);
        WRITELN;
        WRITE('Old data file (',FILENAME,') exists. ');
        WRITE('Write over it (Y/N)? ');
        CH := UPCASE(READKEY);
        WRITELN(CH);
      END;
    IF CH = 'Y' THEN
      BEGIN
        REWRITE(FIL);
        IOCHECK;
```

```
      END;
    IF CH = 'N' THEN
      BEGIN
        WRITELN;
        WRITE('Enter new data file name ');
        READLN(FILENAME);
      END;
  UNTIL((CH = 'Y') AND NOT(IOERR));
END;  {CHKFILE}

PROCEDURE  CALYLpq(VAR YL     : LNVECTOR;
                   VAR K      : INTEGER);  {K = # OF YL-ELEMENTS}
VAR  IX,JY,RR   : INTEGER;
BEGIN
  K := 0;
  FOR IX := 1 TO S DO   {YL(pq) = Y(pp)}
  BEGIN
    FOR JY := 1 TO TT DO
    BEGIN
      IF (YBUS[IX,JY].MAG <> 0) AND (IX <> JY) THEN
      BEGIN
        K := K+1;
        YL[K].BI := IX;YL[K].BJ := JY;
        YL[K].YLPQ := YBUS[IX,IX];
      END;
    END;
  END;
  FOR IX := 1 TO TT-S DO  {YL(pq) = Y(pp) + y(pq)}
  BEGIN
    RR := GNAME1[IX].BUS_NO;
    FOR JY := 1 TO K DO  {K = DIMENSION OF YL[I]}
    BEGIN
      IF YL[JY].BI = RR THEN
      BEGIN
        XYTOCOMPLEX(0,GNAME1[IX].YD,QUAN[1]);
        SUM(YL[JY].YLPQ,QUAN[1],YL[JY].YLPQ);
      END;
    END;
  END;
  FOR IX := 1 TO KK DO   {YL(pq) = Y(pp) + y(pq) +y(p0)}
  BEGIN                          {KK = NUMBER OF BUS CONNECTED TO LOAD}
    RR := LDAD[IX].BUSI;
    JY := 1;
    REPEAT
      IF YL[JY].BI = RR THEN
      BEGIN
         SUM(YL[JY].YLPQ,LDAD[IX].LDADM,YL[JY].YLPQ);
      END;
      JY := JY+1;
    UNTIL JY > K;
  END;
  FOR IX := 1 TO K DO    {YL(pq) = Y(pq)/(Y(pp) + y(pq) + y(p0))}
  BEGIN
     DEVIDE(YBUS[YL[IX].BI,YL[IX].BJ],YL[IX].YLPQ,YL[IX].YLPQ);
  END;
END;
```

```
PROCEDURE  MODIYBUS(VAR YBS     : TEXT; (MODIFY NETWORK DATA FOR NEW REPRESENTATION)
                    VAR GFILE   : TEXT;
                    VAR BFILE   : TEXT;
                    VAR YBUS    : CMPXMATRX;
                    VAR TT      : INTEGER;
                    VAR YL      : LNVECTOR;
                    VAR S       : INTEGER;
                    VAR VTRAN   : CMPXVECTOR;
                    VAR SGTRAN  : CMPXVECTOR;
                    VAR SLTRAN  : CMPXVECTOR;
                    VAR GNARR1  : GNVECTOR1;
                    VAR GNARR2  : GNVECTOR2;
                    VAR IND     : INTVECTOR;
                    VAR KK      : INTEGER;
                    VAR K       : INTEGER;
                    VAR MVA     : INTEGER;
                    VAR T2      : INTEGER;
                    VAR CHKGN   : INTEGER;
                    VAR VBUS    : VECTOR1;

VAR  I,J,II,JJ,XX,T,BI      : INTEGER;
     UNI                    : COMPLEX;
     YY,VB                  : REAL;

BEGIN
  READLN(YBS,S);     {READ 'YBUS.DTA'}
  FOR I := 1 TO S DO
  BEGIN
    FOR J := 1 TO S DO
    BEGIN
      YBUS[I,J].RE := 0;YBUS[I,J].IM := 0;
      YBUS[I,J].MAG := 0;YBUS[I,J].ANG := 0;
    END;
  END;
  REPEAT
    READ(YBS,I);
    IF I <> -1 THEN
    BEGIN
      READ(YBS,J);READ(YBS,YBUS[I,J].RE);READ(YBS,YBUS[I,J].IM);
      READ(YBS,YBUS[I,J].MAG);READLN(YBS,YBUS[I,J].ANG);
    END;
  UNTIL I = -1;
  XYTOCOMPLEX(1,0,UNI);
  T := 0;T2 := 0;
  READ(GFILE,MVA);
  REPEAT      {READ 'GEN.DTA'}
    READ(GFILE,XX);
    IF XX <> -1 THEN
    BEGIN
      IF XX = 1 THEN
      BEGIN
        T := T+1;
        READ(GFILE,GNARR1[T].MVAB);
        READ(GFILE,GNARR1[T].BUS_NO);
        IND[T] := S+T;
```

```
      READ(GFILE,GNARR1[T].H);READLN(GFILE,YY);
      GNARR1[T].H := GNARR1[T].H*GNARR1[T].MVAB/MVA;
      YY := YY/GNARR1[T].MVAB*MVA;
      GNARR1[T].XD := -1/YY;  {1/Xd'}
    END
    ELSE
    BEGIN
      T2 := T2+1;
      READ(GFILE,GNARR2[T2].MVAB);READ(GFILE,GNARR2[T2].BUS_NO);
      READ(GFILE,GNARR2[T2].H);READ(GFILE,GNARR2[T2].XD0);
      READ(GFILE,GNARR2[T2].XQ0);READ(GFILE,GNARR2[T2].XD1);
      READ(GFILE,GNARR2[T2].XQ1);READ(GFILE,GNARR2[T2].XD2);
      READ(GFILE,GNARR2[T2].XQ2);READ(GFILE,GNARR2[T2].XLS);
      READ(GFILE,GNARR2[T2].TD1);READ(GFILE,GNARR2[T2].TQ1);
      READ(GFILE,GNARR2[T2].TD2);READLN(GFILE,GNARR2[T2].TQ2);
      GNARR2[T2].H := GNARR2[T2].H*GNARR2[T2].MVAB/MVA;
      GNARR2[T2].XD0 := GNARR2[T2].XD0/GNARR2[T2].MVAB*MVA;
      GNARR2[T2].XQ0 := GNARR2[T2].XQ0/GNARR2[T2].MVAB*MVA;
      GNARR2[T2].XD1 := GNARR2[T2].XD1/GNARR2[T2].MVAB*MVA;
      GNARR2[T2].XQ1 := GNARR2[T2].XQ1/GNARR2[T2].MVAB*MVA;
      GNARR2[T2].XD2 := GNARR2[T2].XD2/GNARR2[T2].MVAB*MVA;
      GNARR2[T2].XQ2 := GNARR2[T2].XQ2/GNARR2[T2].MVAB*MVA;
      GNARR2[T2].XLS := GNARR2[T2].XLS/GNARR2[T2].MVAB*MVA;
    END;
  END;
UNTIL XX = -1;
CHEGN := 1;
IF T2 <> 0 THEN CHEGN := 2;
TT := S+T;
FOR I := 1 TO T DO   {AUGMENTED YBUS}
BEGIN
  II := GNARR1[I].BUS_NO;JJ := IND[I];
  FOR J := 1 TO S DO
  BEGIN
    IF J = II THEN
      XYTOCOMPLEX(0,-GNARR1[I].XD,YBUS[J,JJ])
    ELSE
      XYTOCOMPLEX(0,0,YBUS[J,JJ]);
  END;
END;
REPEAT    {READ 'BUS.DTA'}
  READ(BFILE,XX);
  IF XX <> -1 THEN
  BEGIN
    READ(BFILE,VB);READ(BFILE,BI);
    VBUS[BI] := VB;
    READ(BFILE,VTRAN[BI].RE);READ(BFILE,VTRAN[BI].IM);
    READ(BFILE,VTRAN[BI].MAG);READ(BFILE,VTRAN[BI].ANG);
    READ(BFILE,SGTRAN[BI].RE);READ(BFILE,SGTRAN[BI].IM); {Pg = SGTRAN.RE}
    READ(BFILE,SGTRAN[BI].MAG);READ(BFILE,SLTRAN[BI].RE); {Qmax = SGTRAN.IM}
    READ(BFILE,SLTRAN[BI].IM);READ(BFILE,SLTRAN[BI].MAG); {Qmin = SGTRAN.MAG}
    READLN(BFILE,SLTRAN[BI].ANG);
  END;
UNTIL XX = -1;
CALyp01(LDAD,KK);  {LDAD(p) = y(p0)}
CALYLpq(YL,K);
```

```
END;

PROCEDURE  CALE(VAR VTRAN    : CMPXVECTOR);  (SOLVE NETWORK PERFORMANCE EQUATIONS)

VAR  I,J,CC,IT,PASS,II    : INTEGER;
     MAX1,MAX2,MAGE,MAGF  : VECTOR;
     MAX                  : REAL;
     ADT                  : COMPLEX;
BEGIN
  IF CHKONE = 0 THEN
  BEGIN
    FOR I := S+1 TO TT DO   {calculate voltage behind Xd'}
    BEGIN
      CC := GNAME1[I-S].BUS_NO;
      XYTOCOMPLEX(SGTRAN[CC].RE,-SGTRAN[CC].IM,QUAN[1]);
      CONJUGATE(VTRAN[CC],QUAN[2]);
      DEVIDE(QUAN[1],QUAN[2],QUAN[1]);
      XYTOCOMPLEX(0,-1/GNAME1[I-S].XD,QUAN[2]);
      PRODUCT(QUAN[2],QUAN[1],QUAN[1]);
      SUM(QUAN[1],VTRAN[CC],VTRAN[I]);
    END;
  END;
  IT := 0;
  IF BSF <> -1 THEN
  BEGIN
    XYTOCOMPLEX(0,0,VTRAN[BSF]);
    MAGE[BSF] := 0;MAGF[BSF] := 0;
  END;
  FOR I := 1 TO S DO
  BEGIN
    MAGE[I] := VTRAN[I].RE+1;MAGF[I] := VTRAN[I].IM+1;
  END;
  REPEAT
    IT := IT+1;
    CC := 0;
    MAX := 0;
    FOR I := 1 TO S DO
    BEGIN
        MAX1[I] := ABS(VTRAN[I].RE-MAGE[I]);
        MAX2[I] := ABS(VTRAN[I].IM-MAGF[I]);
      IF MAX1[I] > MAX THEN
        MAX := MAX1[I];
      IF MAX2[I] > MAX THEN
        MAX := MAX2[I];
    END;
    IF MAX > 0.0001 THEN
    BEGIN
      FOR I := 1 TO S DO
      BEGIN
        PASS := -1;
        MAGE[I] := VTRAN[I].RE;MAGF[I] := VTRAN[I].IM;
        IF (I <> BSF) AND (PASS = -1) THEN
        BEGIN
          XYTOCOMPLEX(0,0,ADT);
          FOR CC := 1 TO K DO
          BEGIN
```

```
            IF YL[CC].BI = I THEN
            BEGIN
              PRODUCT(YL[CC].YLPQ,VTRAN[YL[CC].BJ].QUAN[1]);
              SUM(ADT,QUAN[1],ADT);
            END;
          END;
          MULTICMPX(-1,ADT,VTRAN[I]);
        END;
      END;
    END;
  UNTIL MAX <= 0.0001;
END;

PROCEDURE  INITIALMODEL(VAR RS      : VECTOR;
                        VAR RR      : VECTOR;
                        VAR XM      : VECTOR;
                        VAR XS      : VECTOR;
                        VAR H       : VECTOR);

VAR I                               : INTEGER;
    SIZE,P1,P2,TM1,TM2,MM,PF,NUM    : REAL;
BEGIN
  SIZE := 400; {SUBSTITUTE DYNAMIC LOAD WITH MOTOR 400 kW MODEL}
  PF := 0.9;   {POWER FACTOR = 0.9}
  FOR I := 1 TO CNT DO
  BEGIN
    IF STC[BSL[I]] <> -1 THEN
    BEGIN
      IF STC[BSL[I]] <> 1 THEN
      BEGIN
        MM := (1-STC[BSL[I]])*SLTRAN[BSL[I]].MAG*PF;
        NUM := MM*MVA*1E3/SIZE;
        TM1 := -2.655078238638E-2*LN(1.5491180817353E-3*SIZE);
        RS[I] := (TM1/NUM)*(MVA*1E3/SIZE);
        TM1 := -6.2783787041E-3*LN(6.333927785E-4*SIZE);
        RR[I] := (TM1/NUM)*(MVA*1E3/SIZE);
        TM1 := 1.5670009658E-1*LN(8.724601826E+3*SIZE);
        XM[I] := (TM1/NUM)*(MVA*1E3/SIZE);
        TM1 := -5.3812754207E-3*LN(1.0002554369E-8*SIZE);
        XS[I] := (TM1/NUM)*(MVA*1E3/SIZE);
        TM1 :=  5.5098263277E-11*SIZE*SIZE*SIZE*SIZE;
        TM1 :=  TM1-1.9599581747E-8*SIZE*SIZE*SIZE;
        TM1 :=  TM1-1.0318613043E-5*SIZE*SIZE;
        TM1 :=  TM1+4.7870426021E-3*SIZE;
        TM1 :=  TM1+3.8835752799E-2;
        H[I] := (TM1*NUM)/(MVA*1E3/SIZE);
      END
      ELSE
      BEGIN
        RS[I] := -1;
        RR[I] := -1;
        XM[I] := -1;
        XS[I] := -1;
        H[I] := -1;
      END;
    END
```

```
    ELSE
    BEGIN
      RS[I] := -1;
      RR[I] := -1;
      XM[I] := -1;
      XS[I] := -1;
      H[I] := -1;
    END;
  END;
END;

PROCEDURE  INITIALPARAMETER(VAR PDS          : VECTOR; (CALCULATE INITIAL VALUE OF LOAD-
                            VAR PQS          : VECTOR;  PARAMETERS)
                            VAR PDR          : VECTOR;
                            VAR PQR          : VECTOR;
                            VAR TL           : VECTOR;
                            VAR WR           : VECTOR;
                            VAR COEF         : COEFMATRX;
                            VAR PLCON        : VECTOR;
                            VAR QLCON        : VECTOR);

VAR  I                       : INTEGER;
     IQS,IDS,IDR,IQR,D,XSS,XRR : REAL;
     WB,WB,PL,QL               : REAL;
BEGIN
  WB := 2*PI*FREQ;
  WE := WB;
  FOR I := 1 TO CNT DO
  BEGIN
    IF RS[I] <> -1 THEN
    BEGIN
      XSS := XM[I]+XS[I];
      XRR := XSS;
      D := XSS*XRR-SQR(XM[I]);
      COEF[I,1] := RS[I]*XRR/D;
      COEF[I,2] := RS[I]*XM[I]/D;
      COEF[I,3] := RR[I]*XM[I]/D;
      COEF[I,4] := RR[I]*XSS/D;
      COEF[I,5] := XM[I]/D;
      IQS := SLTRAN[BSL[I]].RE*(1-STC[BSL[I]])/(VTRAN[BSL[I]].MAG);
      IDS := SLTRAN[BSL[I]].IM*(1-STC[BSL[I]])/(VTRAN[BSL[I]].MAG);
      PDS[I] := VTRAN[BSL[I]].MAG-RS[I]*IQS;
      PQS[I] := RS[I]*IDS;
      IDR := (PDS[I]-IDS*(XS[I]+XM[I]))/XM[I];
      IQR := (PQS[I]-IQS*(XS[I]+XM[I]))/XM[I];
      PQR[I] := XM[I]*IQS+IQR*(XM[I]+XS[I]);
      PDR[I] := XM[I]*IDS+IDR*(XM[I]+XS[I]);
      TL[I] := SQRT(3)/2*(XM[I]/D*(PQS[I]*PDR[I]-PQR[I]*PDS[I]));
      WR[I] := WE-WB*RR[I]*(IQR-IDR)/(PQR[I]-PDR[I]);
      PL := VTRAN[BSL[I]].MAG*(VTRAN[BSL[I]].MAG
            -WE/WB*PDS[I])/RS[I];
      QL := VTRAN[BSL[I]].MAG*WE/WB/RS[I]*PQS[I];
      PLCON[BSL[I]] := SLTRAN[BSL[I]].RE*(1-STC[BSL[I]])-PL;
      QLCON[BSL[I]] := SLTRAN[BSL[I]].IM*(1-STC[BSL[I]])-QL;
    END
    ELSE
```

```
    BEGIN
      COEF[I,1] := -1;
      COEF[I,2] := -1;
      COEF[I,3] := -1;
      COEF[I,4] := -1;
      COEF[I,5] := -1;
      PDS[I] := -1;
      PQS[I] := -1;
      PDR[I] := -1;
      PQR[I] := -1;
      TL[I] := -1;
      WR[I] := -1;
    END;
  END;
END;

PROCEDURE  CALPQLOAD(VAR SLTRAN   : CMPXVECTOR; (PART LOAD)
                     VAR PDS      : VECTOR;
                     VAR PQS      : VECTOR;
                     VAR PDR      : VECTOR;
                     VAR PQR      : VECTOR;
                     VAR WR       : VECTOR;
                     VAR SLTRNOLD : CMPXVECTOR);

VAR  I                              : INTEGER;
     B,C,BB,CC,DWR0,WR0,DWR1        : REAL;
     PDS0,PQS0,PDR0,PQR0,MAX        : REAL;
     DF1,DF2,DF3,DF4,WR,WB,PL,QL    : REAL;
     DPDR0,DPQR0,DF5                : REAL;
BEGIN
  WB := 2*PI*FREQ;
  WR := WB;
  BB := 0.3;B := 0.3;  {BB,B = PROPORTION OF CONST CURRENT LOAD FOR P,Q}
  CC := 0.2;C := 0.2;  {CC,C = PROPORTION OF CONST IMPEDANCE LOAD FOR P,Q}
  FOR I := 1 TO CNT DO
  BEGIN
    SLTRNOLD[BSL[I]] := SLTRAN[BSL[I]];
    IF STC[BSL[I]] <> -1 THEN
    BEGIN
      IF STC[BSL[I]] <> 1 THEN
      BEGIN
        PDS0 := PDS[I]+1;PQS0 := PQS[I]+1;
        PDR0 := PDR[I]+1;PQR0 := PQR[I]+1;WR0 := WR[I]+1;
        REPEAT
          DF1 := ABS(PDS[I]-PDS0);
          DF2 := ABS(PQS[I]-PQS0);
          DF3 := ABS(PDR[I]-PDR0);
          DF4 := ABS(PQR[I]-PQR0);
          MAX := DF1;
          IF DF2 > MAX THEN MAX := DF2;
          IF DF3 > MAX THEN MAX := DF3;
          IF DF4 > MAX THEN MAX := DF4;
          IF MAX > 0.005 THEN
          BEGIN
            PDS0 := PDS[I];PQS0 := PQS[I];
            PDR0 := PDR[I];PQR0 := PQR[I];WR0 :=WR[I];
```

```
                DPQR0 := WB*(COEF[I,3]*PQS[I]-COEF[I,4]*PQR[I]-(WE-WR[I])/
                         WB*PDR[I]);
                DPDR0 := WB*(COEF[I,3]*PDS[I]-COEF[I,4]*PDR[I]+(WE-WR[I])/
                         WB*PQR[I]);
                DWR0  := WB/(2*2*H[I])*(SQRT(3)/2*COEF[I,5]*(PQS[I]*PDR[I]-
                         PQR[I]*PDS[I])-TL[I]);
                PDS[I] := WB/WE*(VTRAN[BSL[I]].MAG-COEF[I,1]*
                          PQS[I]+COEF[I,2]*PQR[I]);
                PQS[I] := WB/WE*(COEF[I,1]*PDS[I]-COEF[I,2]*PDR[I]);
                PDR[I] := PDR[I]+DT/2*(WB*(COEF[I,3]*PDS[I]-COEF[I,4]*PDR[I]+
                          (WE-WR[I])/WB*PQR[I])+DPDR0);
                PQR[I] := PQR[I]+DT/2*(WB*(COEF[I,3]*PQS[I]-COEF[I,4]*PQR[I]-
                          (WE-WR[I])/WB*PDR[I])+DPQR0);
                WR[I]  := WR[I]+DT/2*(WB/(2*2*H[I])*(SQRT(3)/2*COEF[I,5]*(PQS[I]*
                          PDR[I]-PQR[I]*PDS[I])-TL[I])+DWR0);
              END;
            UNTIL MAX <= 0.005;
            PL := VTRAN[BSL[I]].MAG*(VTRAN[BSL[I]].MAG
                  -WE/WB*PDS[I])/RS[I]+PLCON[BSL[I]];
            QL := VTRAN[BSL[I]].MAG*WE/WB/RS[I]*PQS[I]+
                  QLCON[BSL[I]];
            XYTOCOMPLEX(PL,QL,QUAN[1]);
          END
          ELSE
            XYTOCOMPLEX(0,0,QUAN[1]);
          PL := STC[BSL[I]]*SLTRAN[BSL[I]].RE+DV[BSL[I]]*(BB+
                2*CC*(VTRAN[BSL[I]].MAG/SQRT(2)-DV[BSL[I]]));
          QL := STC[BSL[I]]*SLTRAN[BSL[I]].IM+DV[BSL[I]]*(B+
                2*C*(VTRAN[BSL[I]].MAG/SQRT(2)-DV[BSL[I]]));
          XYTOCOMPLEX(PL,QL,QUAN[2]);
          SUM(QUAN[1],QUAN[2],SLTRAN[BSL[I]]);
        END;
      END;
    END;

PROCEDURE  INITGNCOEF(VAR A1    : SWVECTOR;
                      VAR A2    : SWVECTOR;
                      VAR A3    : SWVECTOR;
                      VAR A4    : SWVECTOR;
                      VAR A5    : SWVECTOR;
                      VAR A6    : SWVECTOR;
                      VAR A7    : SWVECTOR;
                      VAR A8    : SWVECTOR;
                      VAR A9    : SWVECTOR;
                      VAR B1    : SWVECTOR;
                      VAR B2    : SWVECTOR;
                      VAR B3    : SWVECTOR;
                      VAR B4    : SWVECTOR;
                      VAR B5    : SWVECTOR;
                      VAR B6    : SWVECTOR;
                      VAR B7    : SWVECTOR;
                      VAR B8    : SWVECTOR;
                      VAR B9    : SWVECTOR;
                      VAR RSS   : SWVECTOR;
                      VAR REQ1  : SWVECTOR;
                      VAR REQ2  : SWVECTOR;
```

```
                    VAR RKD   : SWVECTOR;
                    VAR RFD   : SWVECTOR;
                    VAR XMD   : SWVECTOR;
                    VAR XMQ   : SWVECTOR);

VAR  XLK1,XLK2,XLKD,XLFD,XK1,XK2,XKD,XFD,RK10,RK20,RKD0,RFD0,
     DD,DQ,WB,MAX,DF1,DF2,DF3,DF4,CCC  : REAL;
     I                                 : INTEGER;
BEGIN
  WB := 2*PI*FREQ;
  FOR I := 1 TO T2 DO
  BEGIN
    RSS[I] := 0;
    XMQ[I] := GNARR2[I].XQ0-GNARR2[I].XLS;
    XMD[I] := GNARR2[I].XD0-GNARR2[I].XLS;
    XLK1 := XMQ[I]*(GNARR2[I].XLS-GNARR2[I].XQ1)/(GNARR2[I].XQ1-
            GNARR2[I].XLS-XMQ[I]);
    XLK2 := XLK1*XMQ[I]*(GNARR2[I].XLS-GNARR2[I].XQ2)/(GNARR2[I].XQ2*XMQ[I]
            +GNARR2[I].XQ2*XLK1-GNARR2[I].XLS*XMQ[I]-GNARR2[I].XLS*XLK1
            -XMQ[I]*XLK1);
    XLFD := XMD[I]*(GNARR2[I].XLS-GNARR2[I].XD1)/(GNARR2[I].XD1-
            GNARR2[I].XLS-XMD[I]);
    XLKD := XLFD*XMD[I]*(GNARR2[I].XLS-GNARR2[I].XD2)/(GNARR2[I].XD2*XMD[I]
            +GNARR2[I].XD2*XLFD-GNARR2[I].XLS*XMD[I]-GNARR2[I].XLS*XLFD
            -XMD[I]*XLFD);
    XK1 := XLK1+XMQ[I];
    XK2 := XLK2+XMQ[I];
    XFD := XLFD+XMD[I];
    XKD := XLKD+XMD[I];
    DD := SQR(XMD[I])*(GNARR2[I].XD0-2*XMD[I]+XFD+XKD)
          -GNARR2[I].XD0*XFD*XKD;
    DQ := SQR(XMQ[I])*(GNARR2[I].XQ0-2*XMQ[I]+XK1+XK2)
          -GNARR2[I].XQ0*XK1*XK2;
    A1[I] := (XK1*XK2-SQR(XMQ[I]))/DQ;
    A2[I] := (-XMQ[I]*XK2+SQR(XMQ[I]))/DQ;
    A3[I] := (-XMQ[I]*XK1+SQR(XMQ[I]))/DQ;
    A4[I] := (XMQ[I]*XK2-SQR(XMQ[I]))/DQ;
    A5[I] := (-GNARR2[I].XQ0*XK2+SQR(XMQ[I]))/DQ;
    A6[I] := (GNARR2[I].XQ0*XMQ[I]-SQR(XMQ[I]))/DQ;
    A7[I] := (XMQ[I]*XK1-SQR(XMQ[I]))/DQ;
    A8[I] := (GNARR2[I].XQ0*XMQ[I]-SQR(XMQ[I]))/DQ;
    A9[I] := (-GNARR2[I].XQ0*XK1+SQR(XMQ[I]))/DQ;
    B1[I] := (XFD*XKD-SQR(XMD[I]))/DD;
    B2[I] := (-XMD[I]*XKD+SQR(XMD[I]))/DD;
    B3[I] := (-XMD[I]*XFD+SQR(XMD[I]))/DD;
    B4[I] := (XMD[I]*XKD-SQR(XMD[I]))/DD;
    B5[I] := (-GNARR2[I].XD0*XKD+SQR(XMD[I]))/DD;
    B6[I] := (GNARR2[I].XD0*XMD[I]-SQR(XMD[I]))/DD;
    B7[I] := (XMD[I]*XFD-SQR(XMD[I]))/DD;
    B8[I] := (GNARR2[I].XD0*XMD[I]-SQR(XMD[I]))/DD;
    B9[I] := (-GNARR2[I].XD0*XFD+SQR(XMD[I]))/DD;
    RKQ1[I] := 1;RKQ2[I] := 1;RKD[I] := 1;RFD[I] := 1;
    RK10 := 2;RK20 := 2;RKD0 := 2;RFD0 := 2;
    REPEAT
      DF1 := ABS(RKQ1[I]-RK10);
      DF2 := ABS(RKQ2[I]-RK20);
```

```
      DF3 := ABS(EKD[I]-EKD0);
      DF4 := ABS(EFD[I]-EFD0);
      MAX := DF1;
      IF DF2 > MAX THEN MAX := DF2;
      IF DF3 > MAX THEN MAX := DF3;
      IF DF4 > MAX THEN MAX := DF4;
      IF MAX > 0.01 THEN
      BEGIN
        EK10 := EKQ1[I];EK20 := EKQ2[I];EKD0 := EKD[I];EFD0 := EFD[I];
        EKQ2[I] := (XLK2+XMQ[I])/WB/(GNAME2[I].TQ1-(XLK1+XMQ[I])/WB/EKQ1[I]);
        EKD[I] := (XLKD+XMD[I])/WB/(GNAME2[I].TD1-(XLFD+XMD[I])/WB/EFD[I]);
        CCC := ((XLK2+XMQ[I]*XLK1/(XLK1+XMQ[I]))/(WB*EKQ2[I]))/GNAME2[I].TQ2-1;
        EKQ1[I] := (XLK1+XMQ[I])/(((XLK2+XMQ[I])/(WB*EKQ2[I]))*WB)*CCC;
        CCC := ((XLKD+XMD[I]*XLFD/(XLFD+XMD[I]))/(WB*EKD[I]))/GNAME2[I].TD2-1;
        EFD[I] := (XLFD+XMD[I])/(((XLKD+XMD[I])/(WB*EKD[I]))*WB)*CCC;
      END;
    UNTIL MAX <= 0.01;
    XMQ[I] := XMQ[I]/SQRT(3);
    XMD[I] := XMD[I]/SQRT(3);
    EKQ1[I] := EKQ1[I]/SQRT(3);
    EKQ2[I] := EKQ2[I]/SQRT(3);
    EKD[I] := EKD[I]/SQRT(3);
    EFD[I] := EFD[I]/SQRT(3);}
  END;
END;

PROCEDURE  INITPARAGN(VAR FDS          : SWVECTOR; (CALCULATE INITIAL VALUE OF GENERATOR
                      VAR FQS          : SWVECTOR;  PARAMETERS)
                      VAR FK1          : SWVECTOR;
                      VAR FK2          : SWVECTOR;
                      VAR FKD          : SWVECTOR;
                      VAR FFD          : SWVECTOR;
                      VAR EXFD         : SWVECTOR;
                      VAR ZR           : SWVECTOR;
                      VAR DELTA        : SWVECTOR;
                      VAR TI           : SWVECTOR);

VAR  I                                 : INTEGER;
     IFD,IQS,IDS,SN,TN,VD0,VQ0,TMP1,TMP2 : REAL;
     TMP3,XLFD,XFD                     : REAL;
     EA,IAS                            : COMPLEX;

BEGIN
  FOR I := 1 TO T2 DO
  BEGIN
    XYTOCOMPLEX((SGTRAN[GNAME2[I].BUS_NO].RE-SLTRAN[GNAME2[I].BUS_NO].RE),
    -(SGTRAN[GNAME2[I].BUS_NO].IM-SLTRAN[GNAME2[I].BUS_NO].IM),QUAN[1]);
    CONJUGATE(VTRAN[GNAME2[I].BUS_NO],QUAN[2]);
    MULTICMPX(1/SQRT(2),QUAN[2],QUAN[2]);
    DEVIDE(QUAN[1],QUAN[2],QUAN[1]);  {P(B) = SQRT(3)*VBph(rms)*IB(rms)}
    MULTICMPX(1,QUAN[1],IAS);
    XYTOCOMPLEX(RSS[I],GNAME2[I].XQ0,QUAN[3]);
    PRODUCT(QUAN[3],IAS,QUAN[3]);
    CONJUGATE(QUAN[2],QUAN[2]);
    SUM(QUAN[2],QUAN[3],EA);
    DELTA[I] := EA.ANG;
```

```
    ZR[I] := DELTA[I]+(VTRAN[GNARR2[I].BUS_NO].ANG);
    IDS := -SQRT(2)*IAS.MAG*SIN(IAS.ANG-(VTRAN[GNARR2[I].BUS_NO].ANG)-
           DELTA[I]);
    IQS := SQRT(2)*IAS.MAG*COS(IAS.ANG-(VTRAN[GNARR2[I].BUS_NO].ANG)-
           DELTA[I]);
    EXFD[I] := SQRT(2)*EA.MAG+(GNARR2[I].XD0-GNARR2[I].XQ0)*IDS;
    TMP1 := EXFD[I]*VTRAN[GNARR2[I].BUS_NO].MAG/GNARR2[I].XD0*SIN(DELTA[I]);
    TI[I] := SQRT(3)/2*(TMP1+1/2*(1/GNARR2[I].XQ0-1/GNARR2[I].XD0)*
             SIN(2*DELTA[I])*SQR(VTRAN[GNARR2[I].BUS_NO].MAG));
    IFD := EXFD[I]/XMD[I];
    XLFD := XMD[I]*(GNARR2[I].XLS-GNARR2[I].XD1)/(GNARR2[I].XD1-
            GNARR2[I].XLS-XMD[I]);
    XFD := XLFD+XMD[I];
    FQS[I] := -GNARR2[I].XQ0*IQS;
    FDS[I] := -GNARR2[I].XD0*IDS+XMD[I]*IFD;
    FK1[I] := -XMQ[I]*IQS;
    FK2[I] := -XMQ[I]*IQS;
    FFD[I] := XFD*IFD-XMD[I]*IDS;
    FKD[I] := XMD[I]*IFD-XMD[I]*IDS;
  END;
END;

PROCEDURE  INITPARAEXC(VAR VREF          : SWVECTOR;
                       VAR V1            : SWVECTOR);

CONST  TF   = 1; {TF = TA}
       TA1  = 1.098901099E-1;
       KA   = 10.98901099;
       SKF  = -8.099999999E-2;
VAR I   : INTEGER;
BEGIN
  FOR I := 1 TO T2 DO
  BEGIN
    V1[I] := SKF*EXFD[I];
    VREF[I] := EXFD[I]/KA+VTRAN[GNARR2[I].BUS_NO].MAG+V1[I];
  END;
END;

PROCEDURE  CALEXFD(VAR EXFD  : SWVECTOR;  (PART EXCITER)
                   VAR V1    : SWVECTOR);

CONST  TF   = 1; {TF = TA}
       TA1  = 1.098901099E-1;
       KA   = 10.98901099;
       SKF  = -8.099999999E-2;
       EMX  = 3;
       EMN  = 0;

VAR II                 : INTEGER;
   EXFD0,V10,VE0,VE,DV1,DEXFD,DF1,DF2,DF3,MAX2,V1II,EXFDII : REAL;
BEGIN
  FOR II := 1 TO T2 DO
  BEGIN
    EXFD[II] := EXFD[II]*KA;
    VE := VREF[II]-VTRAN[GNARR2[II].BUS_NO].MAG-V1[II];
    EXFD0 := EXFD[II]+1;V10 := V1[II]+1;VE0 := VE+1;
```

```
    REPEAT
      DF1 := ABS(EXFD[II]-EXFD0);
      DF2 := ABS(V1[II]-V10);
      MAX2 := DF1;
      IF DF2 > MAX2 THEN MAX2 := DF2;
      IF MAX2 > 0.005 THEN
      BEGIN
        EXFD0 := EXFD[II];V10 := V1[II];VE0 := VE;
        DV1 := SKF/TF*EXFD[II]-V1[II]/TF;
        V1II := V1[II]+DT*DV1;
        VE := VREF[II]-VTRAN[GNARR2[II].BUS_NO].MAG-V1II;
        DEXFD := KA/TA1*VE-EXFD[II]/TA1;
        EXFDII := EXFD[II]+DT*DEXFD;
        V1[II] := V1[II]+DT/2*(SKF/TF*EXFDII-V1[II]/TF+DV1);
        VE := VREF[II]-VTRAN[GNARR2[II].BUS_NO].MAG-V1[II];
        EXFD[II] := EXFD[II]+DT/2*(KA/TA1*VE-EXFD[II]/TA1+DEXFD);
        IF (EXFD[II]/KA > EMX) OR (EXFD[II]/KA < EMN) THEN
        BEGIN
          IF EXFD[II]/KA > EMX THEN
            EXFD[II] := EMX*KA
          ELSE
            EXFD[II] := EMN*KA;
        END;
      END;
    UNTIL MAX2 <= 0.005;
  END;
END;

PROCEDURE  CALGNVOLT(VAR FDS          : SWVECTOR;  (PART GENERATOR)
                     VAR FQS          : SWVECTOR;
                     VAR FK1          : SWVECTOR;
                     VAR FK2          : SWVECTOR;
                     VAR FKD          : SWVECTOR;
                     VAR FFD          : SWVECTOR;
                     VAR TE           : SWVECTOR;
                     VAR VASMAG       : SWVECTOR;
                         DELTA        : SWVECTOR;
                         ZR           : SWVECTOR;
                         WR           : SWVECTOR);

CONST  KA   = 10.98901099;
VAR  MAX1,FDS0,FQS0,FK10,FK20,FKD0,FFD0,DF1,DF2,
     DF3,DF4,DF5,DF6,WB,WB,VAS,VBS,VCS,VQS,VDS,
     IQS,IDS,IAS,DFK1,DFK2,EXFDI,
     DFFD,DFKD              : REAL;
     I                      : INTEGER;
BEGIN
  WB := 2*PI*FREQ;
  WR := WB;
  FOR I := 1 TO T2 DO
  BEGIN
    EXFDI := EXFD[I]/KA;
    FDS0 := FDS[I]+1;FQS0 := FQS[I]+1;FK10 := FK1[I]+1;
    FK20 := FK2[I]+1;FKD0 := FKD[I]+1;FFD0 := FFD[I]+1;
    VQS := VTRAN[GNARR2[I].BUS_NO].MAG*COS(DELTA[I]);
    VDS := VTRAN[GNARR2[I].BUS_NO].MAG*SIN(DELTA[I]);
```

```
      REPEAT
        DF1 := ABS(FDS[I]-FDS0);
        DF2 := ABS(FQS[I]-FQS0);
        DF3 := ABS(FK1[I]-FK10);
        DF4 := ABS(FK2[I]-FK20);
        DF5 := ABS(FKD[I]-FKD0);
        DF6 := ABS(FFD[I]-FFD0);
        MAX1 := DF1;
        IF DF2 > MAX1 THEN MAX1 := DF2;
        IF DF3 > MAX1 THEN MAX1 := DF3;
        IF DF4 > MAX1 THEN MAX1 := DF4;
        IF DF5 > MAX1 THEN MAX1 := DF5;
        IF DF6 > MAX1 THEN MAX1 := DF6;
        IF MAX1 > 0.005 THEN
        BEGIN
          FDS0 := FDS[I];FQS0 := FQS[I];FK10 := FK1[I];
          FK20 := FK2[I];FKD0 := FKD[I];FFD0 := FFD[I];
          DFK1 := WB*(-RKQ1[I]*A4[I]*FQS[I]-RKQ1[I]*A5[I]*FK1[I]
                  -RKQ1[I]*A6[I]*FK2[I]);
          DFK2 := WB*(-RKQ2[I]*A7[I]*FQS[I]-RKQ2[I]*A8[I]*FK1[I]
                  -RKQ2[I]*A9[I]*FK2[I]);
          DFFD := WB*RFD[I]/XMD[I]*(EXFDI-XMD[I]*B4[I]*FDS[I]
                  -XMD[I]*B5[I]*FFD[I]-XMD[I]*B6[I]*FKD[I]);
          DFKD := WB*(-RKD[I]*B7[I]*FDS[I]-RKD[I]*B8[I]*FFD[I]
                  -RKD[I]*B9[I]*FKD[I]);
          FDS[I] := VQS+RSS[I]*A1[I]*FQS[I]+RSS[I]*A2[I]*FK1[I]
                    +RSS[I]*A3[I]*FK2[I];
          FQS[I] := -VDS+RSS[I]*B1[I]*FDS[I]-RSS[I]*B2[I]*FFD[I]
                    -RSS[I]*B3[I]*FKD[I];
          FK1[I] := FK1[I]+DT/2*(WB*(-RKQ1[I]*A4[I]*FQS[I]-RKQ1[I]*A5[I]*FK1[I]
                    -RKQ1[I]*A6[I]*FK2[I])+DFK1);
          FK2[I] := FK2[I]+DT/2*(WB*(-RKQ2[I]*A7[I]*FQS[I]-RKQ2[I]*A8[I]*FK1[I]
                    -RKQ2[I]*A9[I]*FK2[I])+DFK2);
          FFD[I] := FFD[I]+DT/2*(WB*RFD[I]/XMD[I]*(EXFDI-XMD[I]*B4[I]*FDS[I]
                    -XMD[I]*B5[I]*FFD[I]-XMD[I]*B6[I]*FKD[I])+DFFD);
          FKD[I] := FKD[I]+DT/2*(WB*(-RKD[I]*B7[I]*FDS[I]-RKD[I]*B8[I]*FFD[I]
                    -RKD[I]*B9[I]*FKD[I])+DFKD);
        END;
      UNTIL MAX1 <= 0.005;
      TE[I] := SQRT(3)/2*((A1[I]-B1[I])*FQS[I]*FDS[I]+FDS[I]*(A2[I]*FK1[I]+
               A3[I]*FK2[I])-FQS[I]*(B2[I]*FFD[I]+B3[I]*FKD[I]));
    END;
  END;

  PROCEDURE  CALSWING2(VAR ANGFILE      : TEXT; (SOLVE DIFFERENTIAL EQUATIONS FOR ELECTRICAL
                       VAR WRFILE       : TEXT;  ANGLE OF THE GENERATORS)
                       VAR VTRAN        : CMPXVECTOR;
                       VAR W            : SWVECTOR;
                       VAR ANG          : SWVECTOR;
                       VAR DANG         : SWVECTOR;
                       VAR PDS          : VECTOR;
                       VAR PQS          : VECTOR;
                       VAR PDR          : VECTOR;
                       VAR PQR          : VECTOR;
                       VAR WR           : VECTOR;
                       VAR VFILE        : TEXT;
```

```
                    VAR SLTRAN          : CMPXVECTOR;
                    VAR LDAD            : LDVECTOR;
                    VAR YL              : LNVECTOR;
                    VAR SLTRNOLD        : CMPXVECTOR;
                    VAR FDS             : SWVECTOR;
                    VAR FQS             : SWVECTOR;
                    VAR FK1             : SWVECTOR;
                    VAR FK2             : SWVECTOR;
                    VAR FKD             : SWVECTOR;
                    VAR FFD             : SWVECTOR;
                    VAR EXFD            : SWVECTOR;
                    VAR ZR              : SWVECTOR;
                    VAR TR              : SWVECTOR;
                    VAR V1              : SWVECTOR;
                    VAR DELTA           : SWVECTOR;
                    VAR WRG             : SWVECTOR);

VAR  PE,W0,DW0,DW1,ANG0,DANG0,DANG1,DZR0,ZR0,DZR1,
     WRG0,DWRG0,DWRG1     : SWVECTOR;
     I,J                  : INTEGER;
     CURR                 : CMPXVECTOR;
     TMP1,TMP2,ZAS        : REAL;
BEGIN
  FOR I := 1 TO TT-S DO  {FIND Pe(i)}
  BEGIN
    IF GNARR1[I].BUS_NO <> BSF THEN
    BEGIN
      SUB(VTRAN[S+I],VTRAN[GNARR1[I].BUS_NO],QUAN[1]);
      XYTOCOMPLEX(0,GNARR1[I].XD,QUAN[2]);
      PRODUCT(QUAN[1],QUAN[2],CURR[I]);
      CONJUGATE(VTRAN[S+I],QUAN[1]);
      PRODUCT(QUAN[1],CURR[I],QUAN[1]);
      PE[I] := QUAN[1].RE;
    END
    ELSE
      PE[I] := 0;
  END;
  FOR I := 1 TO TT-S DO
  BEGIN
    DANG0[I] := W[I]-2*PI*FREQ;
    ANG0[I] := ANG[I]+DANG0[I]*DT;
    DW0[I] := PI*FREQ*(SGTRAN[GNARR1[I].BUS_NO].RE-PE[I])/GNARR1[I].H;
    W0[I] := W[I]+DW0[I]*DT;
    POLARTOCOMPLEX(VTRAN[S+I].MAG,ANG0[I],VTRAN[S+I]);
  END;
  IF CHKLD = 2 THEN
  BEGIN
    FOR I := 1 TO S DO
      DV[I] := VTRAN[I].MAG/SQRT(2);
  END;
  IF CHKGN = 2 THEN
  BEGIN
    CALEXFD(EXFD,V1);
    CALGNVOLT(FDS,FQS,FK1,FK2,FKD,FFD,TR,VASMAG,DELTA,ZR,WRG);
    FOR I := 1 TO T2 DO
    BEGIN
```

```
    DZR0[I] := WRG[I]-2*PI*FREQ;
    ZR0[I] := ZR[I]+DZR0[I]*DT;
    ZAS := VTRAN[GNARR2[I].BUS_NO].ANG+DZR0[I]*DT;
    DWRG0[I] := -PI*FREQ*(TE[I]-TI[I])/GNARR2[I].H;
    WRG0[I] := WRG[I]+DWRG0[I]*DT;
  END;
END;
CALE(VTRAN);
IF CHKGN = 2 THEN
BEGIN
  FOR I := 1 TO T2 DO
    DELTA[I] := ZR0[I]-VTRAN[GNARR2[I].BUS_NO].ANG;
END;
IF CHKLD = 2 THEN
BEGIN
  FOR I := 1 TO S DO
    DV[I] := VTRAN[I].MAG/SQRT(2)-DV[I];
  CALPQLOAD(SLTRAN,PDS,PQS,PDR,PQR,WR,SLTRNOLD);
  CALyp02(LDAD);  {LDAD(p) = y(p0)}
  CALYLpq(YL,K);
END;
FOR I := 1 TO TT-S DO  {FIND Pe(i)}
BEGIN
  IF GNARR1[I].BUS_NO <> BSF THEN  {}
  BEGIN                            {}
    SUB(VTRAN[S+I],VTRAN[GNARR1[I].BUS_NO],QUAN[1]);
    XYTOCOMPLEX(0,GNARR1[I].XD,QUAN[2]);
    PRODUCT(QUAN[1],QUAN[2],CURR[I]);
    CONJUGATE(VTRAN[S+I],QUAN[1]);
    PRODUCT(QUAN[1],CURR[I],QUAN[1]);
    PE[I] := QUAN[1].RE;
  END                              {}
  ELSE                             {}
    PE[I] := 0;                    {}
END;                               {}
FOR I := 1 TO TT-S DO
BEGIN
  DW1[I] := PI*FREQ*(SGTRAN[GNARR1[I].BUS_NO].RE-PE[I])/GNARR1[I].H;
  W[I] := W[I]+(DW0[I]+DW1[I])*DT/2;
  DANG1[I] := W[I]-2*PI*FREQ;
  ANG[I] := ANG[I]+(DANG0[I]+DANG1[I])*DT/2;
  POLARTOCOMPLEX(VTRAN[S+I].MAG,ANG[I],VTRAN[S+I]);
END;
IF CHKLD = 2 THEN
BEGIN
  FOR I := 1 TO S DO
    DV[I] := VTRAN[I].MAG/SQRT(2);
END;
IF CHKGN = 2 THEN
BEGIN
  CALEXFD(EXFD,V1);
  CALGNVOLT(FDS,FQS,FK1,FK2,FKD,FFD,TE,VASMAG,DELTA,ZR0,WRG0);
  FOR I := 1 TO T2 DO
  BEGIN
    DWRG1[I] := -PI*FREQ*(TE[I]-TI[I])/GNARR2[I].H;
    WRG[I] := WRG[I]+(DWRG0[I]+DWRG1[I])*DT/2;
```

```
      DZR1[I] := WRG[I]-2*PI*FREQ;
      ZR[I] := ZR[I]+(DZR0[I]+DZR1[I])*DT/2;
      ZAS := VTRAN[GNARR2[I].BUS_NO].ANG+(DZR0[I]+DZR1[I])*DT/2;
    END;
  END;
  CALE(VTRAN);
  IF CHKGN = 2 THEN
  BEGIN
    FOR I := 1 TO T2 DO
      DELTA[I] := ZR[I]-VTRAN[GNARR2[I].BUS_NO].ANG;
  END;
  IF CHKLD = 2 THEN
  BEGIN
    FOR I := 1 TO S DO
      DV[I] := VTRAN[I].MAG/SQRT(2)-DV[I];
    CALPQLOAD(SLTRAN,PDS,PQS,PDR,PQR,WR,SLTRNOLD);
    CALyp02(LDAD);  {LDAD(p) = y(p0)}
    CALYLpq(YL,K);
  END;
  WRITE(ANGFILE,TIM);WRITE(ANGFILE,'  ');
  WRITE(WRFILE,TIM);WRITE(WRFILE,'  ');
  FOR I := 1 TO TT-S DO
  BEGIN
    WRITE(ANGFILE,ANG[I]*180/PI);WRITE(ANGFILE,'  ');
    WRITE(WRFILE,W[I]/2/PI/FREQ);WRITE(WRFILE,'  ');
    WRITELn('ANG[',GNARR1[I].BUS_NO,'] = ',ANG[I]*180/PI);
  END;
  IF CHKGN = 2 THEN
  BEGIN
    FOR I := 1 TO T2 DO
    BEGIN
      WRITE(ANGFILE,ZR[I]*180/PI);WRITE(ANGFILE,'  ');
      WRITE(WRFILE,WRG[I]/2/PI/FREQ);WRITE(WRFILE,'  ');
      WRITELn('ANG[',GNARR2[I].BUS_NO,'] = ',ZR[I]*180/PI);
    END;
  END;
  WRITELN(ANGFILE);
  WRITELN(WRFILE);
END;

BEGIN  {MAIN}
  FREQ := 60;  {FREQUENCY = 60 Hz}
  TIM := 0;
  TIM_END := 1;
  TIMCLR := 0.03;
  DT := 0.001;
  WRITE('Fault(3-ph) on bus-No. : ');
  READINT(BF);BSF := BF;
  ASSIGN(YBS,'D:\TP5\YBUS14.DAT');RESET(YBS);
  ASSIGN(GFILE,'D:\TP5\GEN14.DAT');RESET(GFILE);
  ASSIGN(BFILE,'D:\TP5\BUSTRN14.DAT');RESET(BFILE);
  ASSIGN(LDFILE,'D:\TP5\LOAD14.DAT');RESET(LDFILE);  {}
  CHKLD := 1;                                        {}
  CNT := 0;
  REPEAT                                             {}
    READ(LDFILE,LL);                                 {LL = BUS-No.}
```

```
  IF LL <> -1 THEN                                    {}
  BEGIN                                               {}
    CNT := CNT+1;                                     {}
    BSL[CNT] := LL;                                   {}
    READ(LDFILE,TYPL);                                {}
    IF TYPL = 2 THEN                                  {}
    BEGIN                                             {}
      CHKLD := 2;                                     {}
      READ(LDFILE,STC[LL]);                           {}
    END                                               {}
    ELSE                                              {}
      STC[LL] := -1;                                  {CONST.LOAD = -1}
    READLN(LDFILE);                                   {}
  END;                                                {}
UNTIL LL = -1;                                        {}
ANGFL := 'D:\TP5\ANG142.DAT';
WRFL := 'D:\TP5\Wr142.DAT';
CHKFILE(ANGFILE,ANGFL);
CHKFILE(WRFILE,WRFL);
MODIYBUS(YBS,GFILE,BFILE,YBUS,TT,YL,S,VTRAN,SGTRAN,SLTRAN,GNARR1,GNARR2,
        IND,KK,K,MVA,T2,CHKGN,VBUS);
IF CHKLD = 2 THEN
BEGIN
  INITIALMODEL(RS,RR,XM,XS,H);
  INITIALPARAMETER(PDS,PQS,PDR,PQR,TL,WR,COEF,PLCON,QLCON);
END;
IF CHKGN = 2 THEN
BEGIN
  INITGNCOEF(A1,A2,A3,A4,A5,A6,A7,A8,A9,B1,B2,B3,B4,B5,B6,B7,B8,B9,
             ESS,EKQ1,EKQ2,EKD,EFD,EMD,EMQ);
  INITPARAGN(FDS,FQS,FK1,FK2,FKD,FFD,EXFD,ZR,DELTA,TI);
  INITPARAEXC(VREF,V1);
  FOR I := 1 TO T2 DO
    WRG[I] := 2*PI*FREQ;
END;
CHKONE := 0;
CALE(VTRAN);
FOR I := 1 TO TT-S DO
BEGIN
  WRITE(ANGFILE,GNARR1[I].BUS_NO);WRITE(ANGFILE,'  ');
  WRITE(WRFILE,GNARR1[I].BUS_NO);WRITE(WRFILE,'  ');
END;
IF CHKGN = 2 THEN
BEGIN
  FOR I := 1 TO T2 DO
  BEGIN
    WRITE(ANGFILE,GNARR2[I].BUS_NO);WRITE(ANGFILE,'  ');
    WRITE(WRFILE,GNARR2[I].BUS_NO);WRITE(WRFILE,'  ');
  END;
END;
WRITELN(ANGFILE);
WRITELN(WRFILE);
WRITE(ANGFILE,TIM);WRITE(ANGFILE,'  ');
WRITE(WRFILE,TIM);WRITE(WRFILE,'  ');
CHKONE := 1;
WRITELN('---------------- TIME = ',TIM,' ----------------');
```

```
  FOR I := 1 TO TT-S DO
  BEGIN
    ANG[I] := VTRAN[S+I].ANG;   {INITIAL VALUE}
    W[I] := 2*PI*FREQ;          {INITIAL VALUE}
    WRITELn('ANG[',GNAME1[I].BUS_NO,'] = ',ANG[I]*180/PI);
    WRITE(ANGFILE,ANG[I]*180/PI);WRITE(ANGFILE,' ');
    WRITE(WRFILE,W[I]/2/PI/FREQ);WRITE(WRFILE,' ');
  END;
  IF CHKGN = 2 THEN
  BEGIN
    FOR I := 1 TO T2 DO
    BEGIN
      WRITE(ANGFILE,ZR[I]*180/PI);WRITE(ANGFILE,' ');
      WRITE(WRFILE,WRG[I]/2/PI/FREQ);WRITE(WRFILE,' ');
      WRITELn('ANG[',GNAME2[I].BUS_NO,'] = ',ZR[I]*180/PI);
    END;
  END;
  WRITELN(ANGFILE);
  WRITELN(WRFILE);
  REPEAT
    TIM := TIM+DT;
    WRITELN('---------------- TIME = ',TIM,' ----------------');
    IF TIM >= TIMCLR THEN
      BSF := -1;
    CALSWING2(ANGFILE,WRFILE,VTRAN,W,ANG,DANG,PDS,PQS,PDR,PQR,WR,
              VFILE,SLTRAN,LDAD,YL,SLTRNOLD,FDS,FQS,FK1,FK2,FKD,
              FFD,BXFD,ZR,TE,V1,DELTA,WRG);
  UNTIL TIM > TIM_END;
  CLOSE(ANGFILE);
  CLOSE(WRFILE);
  WRITE('SUCCESSFUL!');READLN;
END.
```

ประวัติผู้เขียน

นายพรชัย ปฏิภาณปรีชาวุฒิ เกิดวันที่ 2 สิงหาคม พ.ศ. 2508 ที่จังหวัดกรุงเทพมหานคร สำเร็จปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากจุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2530 หลังจากนั้นได้เข้าศึกษาต่อปริญญาโทในภาควิชาวิศวกรรมไฟฟ้า สาขาพลังงานไฟฟ้า ที่จุฬาลงกรณ์มหาวิทยาลัย ระหว่างปีการศึกษา 2530 ถึง 2531 ได้ทำหน้าที่เป็นผู้ช่วยวิจัย ประจำศูนย์วิจัยและอบรมพลังงาน จุฬาลงกรณ์มหาวิทยาลัย