เอกสารอ้างอิง

1. Kirchmayer, L.K., "Economic Operation of Power Systems", New York, Wiley and Sons, 1958.
2. Wood, A.J. and Wollenberg, B.F., "Power Generation Operation and Control", Wiley and Sons, New York, 1984. 1841-853, May/June 1968.
3. Elgerd, O.L., "Electric Energy Systems Theory : An Introduction", McGraw-Hill, 1983.
4. Happ, H.H, "Optimal Power Dispatch", IEEE Trans. Power. App. and System, Vol. PAS-93, pp.820-830, No.3 1974.
5. Happ, H.H, "Optimal Power Dispatch - A comprehensive Survey", IEEE Trans. Power. App. and System, Vol. PAS-96, pp.
6. Dopazo, J.F.; Klitin, O.A.; Stagg, G.W and Watson, M., "An Optimal Technique for Real and Reactive Power Allocation", Proceedings of IEEE, Vol. 55, pp 1877-1885, Nov. 1967.
7. Stevenson, D. William JR., "Element of Power System Analysis", 4th ed, McGraw-Hill, 1982
8. ทรงศักดิ์ คงน้อย, "การใช้คอมพิวเตอร์วางแผนขยายสายส่งของระบบไฟฟ้ากำลัง", วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2528.
9. Stagg, G.W and El-abaid, "Computor Methods in Power System Analysis", New York, McGraw-Hill, 1968
10. Arrillaga, J.; Arnord, C.P. and Harker, B.J., "Computer Modelling of Electrical Power Systems", Wiley and Sons, 1983
11. Dommel, H.W. and Tinney, W.F., "Optimal Power Flow Solution", IEEE Trans. Power. App. and System, Vol. PAS-87, pp. 1861-1876, Oct. 1968.
12. Burchett, R.C.; Happ, H.H.; Vierath, D.R. and Wrigan, K.A. "Developments in Optimal Power Flow", IEEE Trans. Power. App. and System, Vol. PAS-101, pp.406-414, Feb. 1968.

13. Shoults, R.R and Sun, D.T., "Optimal Power Flow Base upon P-Q Decomposition", IEEE Trans. Power. App. and System, Vol. PAS-101, pp.397-405, Feb. 1982.

14. Bala, J.L. and Thanikachalam, A., "An Improved Second Order Method for Optimal Load Flow", IEEE Trans. Power. App. and System, Vol. PAS-97, pp.1239-1244, Jul/Aug. 1978.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

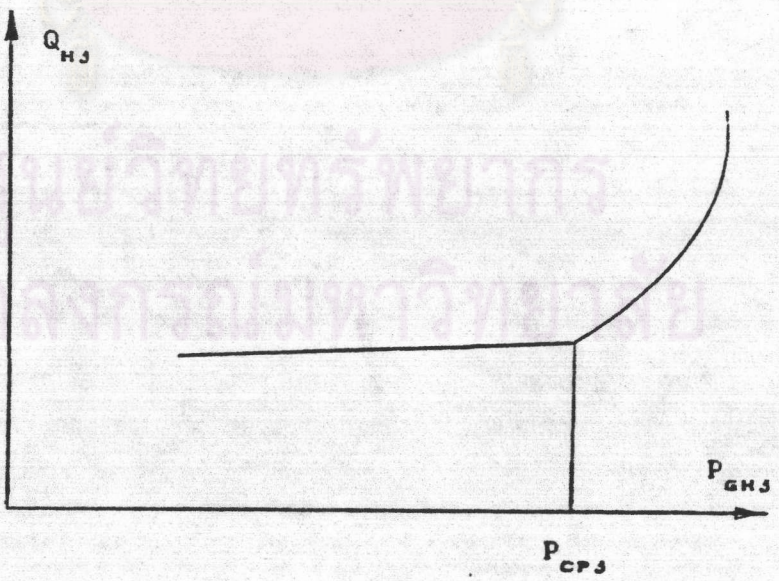
ภาคผนวก ก

การจ่ายโหลดอย่างประหยัดในกรณีที่มีเครื่องกำเนิดไฟฟ้าพลังน้ำ

ในระบบไฟฟ้ากำลังที่ประกอบด้วยเครื่องกำเนิดไฟฟ้าพลังน้ำ(hydro-generator) และเครื่องกำเนิดไฟฟ้าพลังความร้อน(thermal-generator) การจ่ายโหลดอย่างประหยัดจะต้องคำนึงถึงเงื่อนไขบังคับของเครื่องกำเนิดไฟฟ้าพลังน้ำ เช่น ความต้องการใช้น้ำทางด้านท้ายเขื่อน อัตราการไหลของน้ำเข้าอ่างเก็บน้ำ ฯลฯ

อัตราการใช้น้ำของเครื่องกำเนิดไฟฟ้าพลังน้ำ(Q_{Hj}) สามารถถือได้ว่าเป็นฟังก์ชันของกำลังไฟฟ้าที่ผลิต(P_{HGj}) โดยทั่วไปจะประมาณว่าอัตราการใช้น้ำเป็นฟังก์ชันโพลีโนเมียล 2 ส่วนต่อกัน ดังรูปที่ ก.1 หรือเขียนเป็นสมการได้ดังนี้[11,15]

$$Q_{Hj} = \begin{cases} a_j + b_jP_{GHj} + c_jP_{GHj}^2 + \ldots & \text{เมื่อ } P_{GHj}<P_{CPj} \\ d_j + e_jP_{GHj} + f_jP_{GHj}^2 + \ldots & \text{เมื่อ } P_{GHj}>P_{CPj} \end{cases} \qquad (ก.1)$$



รูปที่ ก.1 อัตราการใช้น้ำของเครื่องกำเนิดไฟฟ้าพลังน้ำ

เมื่อ P_{CPJ} เป็นจุดเปลี่ยนโค้ง(curve change point)

ในการวิเคราะห์การจ่ายโหลดอย่างประหยัดในกรณีที่มีเครื่องกำเนิดไฟฟ้าพลังน้ำ จะสมมติค่าคงที่ค่าหนึ่งขึ้นมา เรียกว่าค่าน้ำ(water rate,Y_J) เพื่อเปลี่ยนอัตราการใช้น้ำให้เป็นcost functionของเครื่องกำเนิดไฟฟ้าพลังน้ำ(C_{HJ})ตามสมการ

$$C_{HJ} = Y_J Q_{HJ} \qquad (ก.2)$$

ดังนั้นการจ่ายโหลดอย่างประหยัดก็สามารถทำได้เหมือนกับระบบที่มีแต่เครื่องกำเนิดไฟฟ้าพลังความร้อน สมการที่ 3.17 จะกลายเป็น[11]

$$\frac{dC_{Ti}}{dP_{GTi}} + \lambda\frac{\partial P_L}{\partial P_{GTi}} = 0 \qquad i = 1,2,..,m$$

$$Y_J\frac{dQ_{HJ}}{dP_{GHJ}} + \lambda\frac{\partial P_L}{\partial P_{GHJ}} = 0 \qquad j = m+1,..,N \qquad (ก.3)$$

เมื่อ C_{Ti} เป็นcost functionของเครื่องกำเนิดไฟฟ้าพลังความร้อนที่บัส i

P_{GTi} เป็นกำลังไฟฟ้าที่ผลิตของเครื่องกำเนิดไฟฟ้าพลังความร้อนที่บัส i

Q_{HJ} เป็นอัตราการใช้น้ำของเครื่องกำเนิดไฟฟ้าพลังน้ำที่บัส j

P_{GHJ} เป็นกำลังไฟฟ้าที่ผลิตของเครื่องกำเนิดไฟฟ้าพลังน้ำที่บัส j

Y_J เป็นค่าน้ำของเครื่องกำเนิดไฟฟ้าพลังน้ำที่บัส j

P_L เป็นกำลังสูญเสียของระบบ

λ เป็นLagrange multiplier

ค่าน้ำ หรือ Y_J เป็นค่าที่มีความสำคัญ และมีผลต่อการจ่ายโหลดอย่างประหยัด เพราะถ้าค่าน้ำมีค่าน้อยอัตราการใช้น้ำจะมีค่ามาก แต่ถ้าค่าน้ำมีค่ามากอัตราการใช้น้ำจะมีค่าน้อย การหาค่า Y_J จะต้องคำนึงถึงเงื่อนไขบังคับของเครื่องกำเนิดไฟฟ้าพลังน้ำ รายละเอียดการหาค่าน้ำศึกษาได้จาก[2] สำหรับในวิทยานิพนธ์นี้จะถือว่าค่า Y_J เป็นค่าคงที่และเป็นข้อมูลป้อนเข้า(input data)

สำหรับตัวอย่างที่แสดงการจ่ายโหลดอย่างประหยัดในกรณีที่มีเครื่องกำเนิดไฟฟ้าพลังน้ำ จะใช้ระบบไฟฟ้ากำลัง 5 บัส 6 สายส่งในหัวข้อที่ 5.2 โดยดัดแปลงเครื่องกำเนิดไฟฟ้าที่บัส 2 ให้เป็นเครื่องกำเนิดไฟฟ้าพลังน้ำ ในที่นี้จะแสดง 2 กรณี คือ กรณีแรกมีค่าน้ำเท่ากับ 0.9 และในกรณีที่ 2 มีค่าน้ำเท่ากับ 1.1 โดยที่ทั้งสองกรณีมีอัตราการใช้น้ำเท่ากัน ทั้งนี้เพื่อแสดงผลของค่าน้ำ

- ข้อมูลของcost function และอัตราการใช้น้ำในกรณีแรก

*** THERMAL GENERATOR COST FUNCTION ***

BUS	COST = A + B*PG + C*PG**2		
NO.	A	B	C
1	44.40000	3.51000	.00500

*** HYDRO GENERATOR WATER RATE ***

BUS	FIRST SECTION			SECOND SECTION			CURVE CHANG.	COST OF
NO.	A	B	C	D	E	F	PT.(MW)	WATER
2	40.6000	3.8900	.0050	40.0000	4.0500	.0050	105.00	.9000

- คำตอบของการจ่ายโหลดอย่างประหยัดในกรณีแรก

*** BUS VOLTAGES AND POWER GENERATION ***

BUS	BUS	BUS VOLTAGE			GENERATION		LOAD		SHUNT
NO.	TYPE	PU	KV	DEG	MW	MVAR	MW	MVAR	MVAR
1	3	1.0000	230.00	.00	79.10	40.68	.00	.00	.00
2	2	1.0000	230.00	.61	86.23	40.71	.00	.00	.00
3	1	.9200	211.60	-5.13	.00	.00	60.00	30.00	.00
4	1	.8919	205.15	-9.07	.00	.00	40.00	10.00	.00
5	1	.9620	221.25	-2.97	.00	.00	60.00	20.00	.00

*** LINE FLOW ***

LINE NO.	FROM BUS P	TO BUS Q	FLOW FROM BUS P MW	FLOW FROM BUS P MVAR	FLOW TO BUS Q MW	FLOW TO BUS Q MVAR	LOSS MW	LOSS MVAR	LINE CHARG. MVAR
1	1	3	24.28	14.85	-23.47	-11.61	.81	3.24	.00
2	1	4	26.72	13.19	-25.39	-7.86	1.33	5.33	.00
3	1	5	28.10	12.64	-27.63	-10.74	.47	1.90	.00
4	2	3	53.26	28.99	-51.42	-21.64	1.84	7.35	.00
5	2	5	32.98	11.71	-32.37	-9.26	.61	2.45	.00
6	3	4	14.88	3.22	-14.61	-2.12	.27	1.10	.00

*** SYSTEM TOTAL ***

	MW	MVAR
GENERATION	165.33	81.38
LOAD	160.00	60.00
STATIC CAPACITOR	.00	.00
LINE CHARGING	.00	.00
LOSS	5.34	21.37
MISMATCH	.01	.02
PRODUCTION COST		725.218

- ข้อมูลcost function และอัตราการใช้น้ำในกรณีที่ 2

*** THERMAL GENERATOR COST FUNCTION ***

BUS NO.	COST = A + B*PG + C*PG**2 A	B	C
1	44.40000	3.51000	.00500

*** HYDRO GENERATOR WATER RATE ***

BUS NO.	FIRST SECTION A	B	C	SECOND SECTION D	E	F	CURVE CHANG. PT.(MW)	COST OF WATER
2	40.6000	3.8900	.0050	40.0000	4.0500	.0050	105.00	1.1000

- คำตอบของการจ่ายโหลดอย่างประหยัดในกรณีที่ 2

*** BUS VOLTAGES AND POWER GENERATION ***

BUS	BUS	BUS VOLTAGE			GENERATION		LOAD		SHUNT
NO.	TYPE	PU	KV	DEG	MW	MVAR	MW	MVAR	MVAR
1	3	1.0000	230.00	.00	114.51	36.07	.00	.00	.00
2	2	1.0000	230.00	-4.31	51.42	47.65	.00	.00	.00
3	1	.9199	211.57	-8.05	.00	.00	60.00	30.00	.00
4	1	.8904	204.79	-10.78	.00	.00	40.00	10.00	.00
5	1	.9613	221.10	-5.44	.00	.00	60.00	20.00	.00

*** LINE FLOW ***

LINE	FROM BUS	TO BUS	FLOW FROM BUS P		FLOW TO BUS Q		LOSS		LINE CHARG,
NO.	P	Q	MW	MVAR	MW	MVAR	MW	MVAR	MVAR
1	1	3	35.56	13.41	-34.11	-7.63	1.44	5.78	.00
2	1	4	31.04	13.12	-29.33	-6.31	1.70	6.81	.00
3	1	5	47.92	9.54	-46.72	-4.76	1.19	4.77	.00
4	2	3	37.93	31.57	-36.71	-26.70	1.22	4.87	.00
5	2	5	13.50	16.08	-13.28	-15.19	.22	.88	.00
6	3	4	10.83	4.30	-10.67	-3.66	.16	.64	.00

*** SYSTEM TOTAL ***

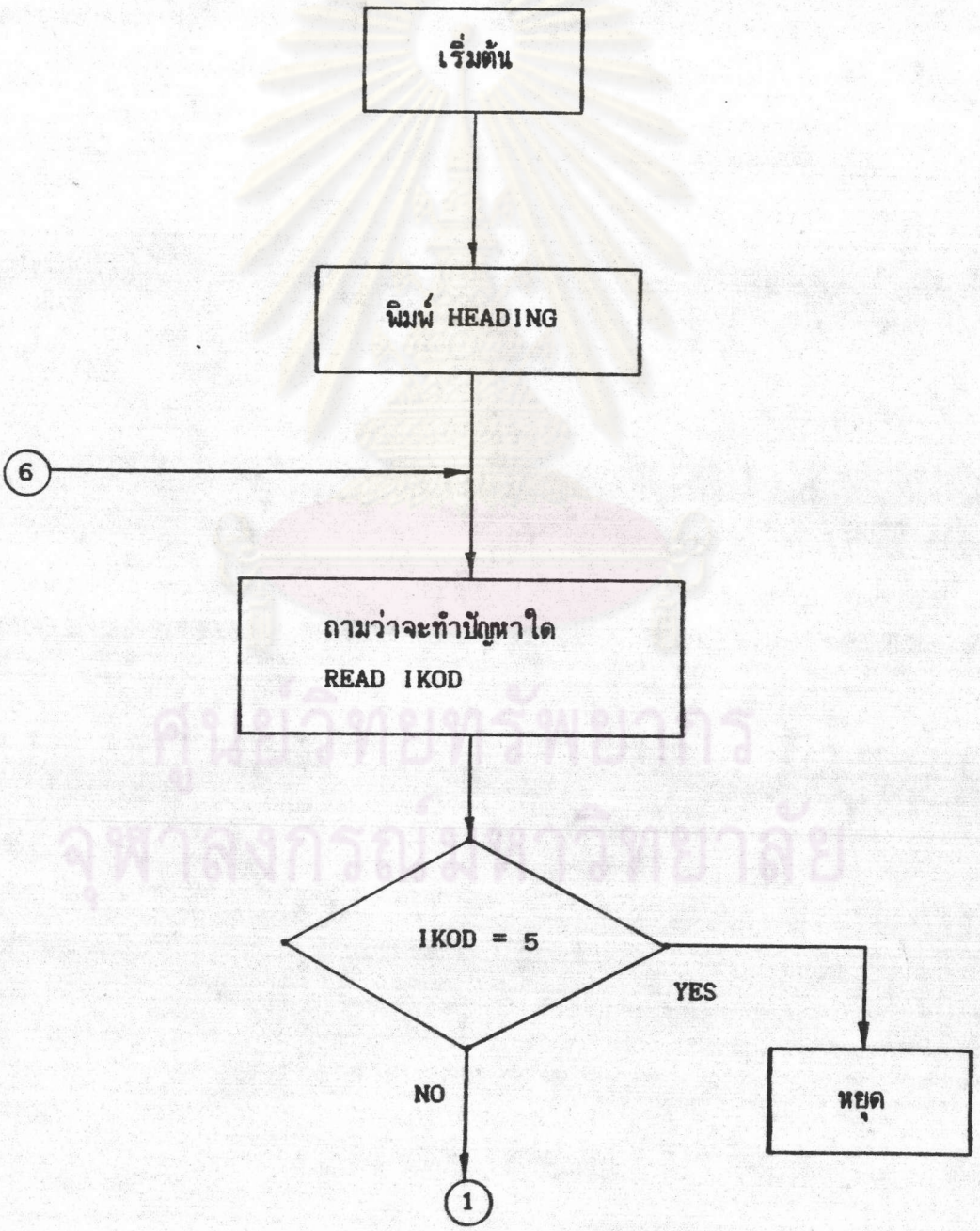
	MW	MVAR
GENERATION	165.93	83.72
LOAD	160.00	60.00
STATIC CAPACITOR	.00	.00
LINE CHARGING	.00	.00
LOSS	5.94	23.76
MISMATCH	.00	.04
PRODUCTION COST		791.142

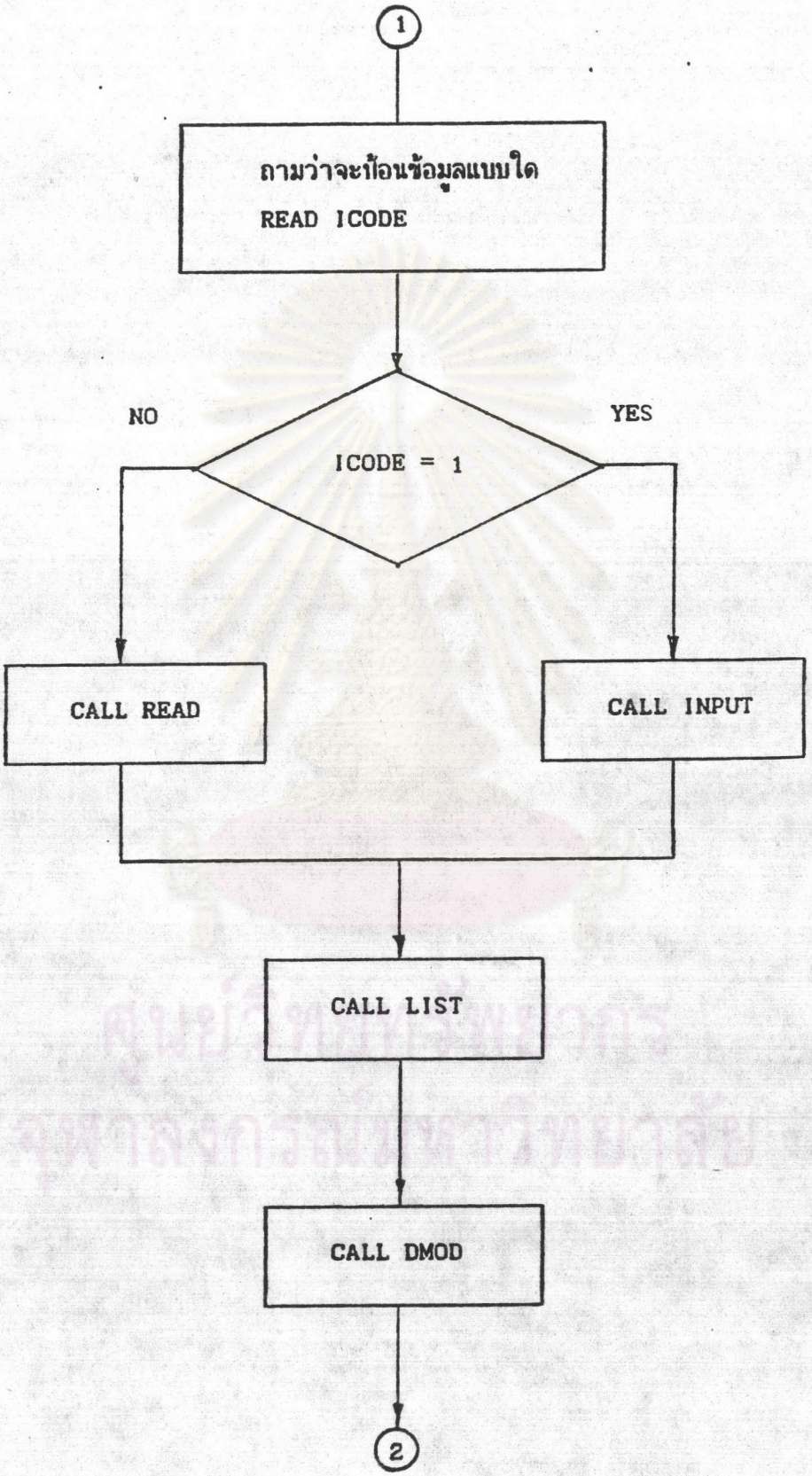
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

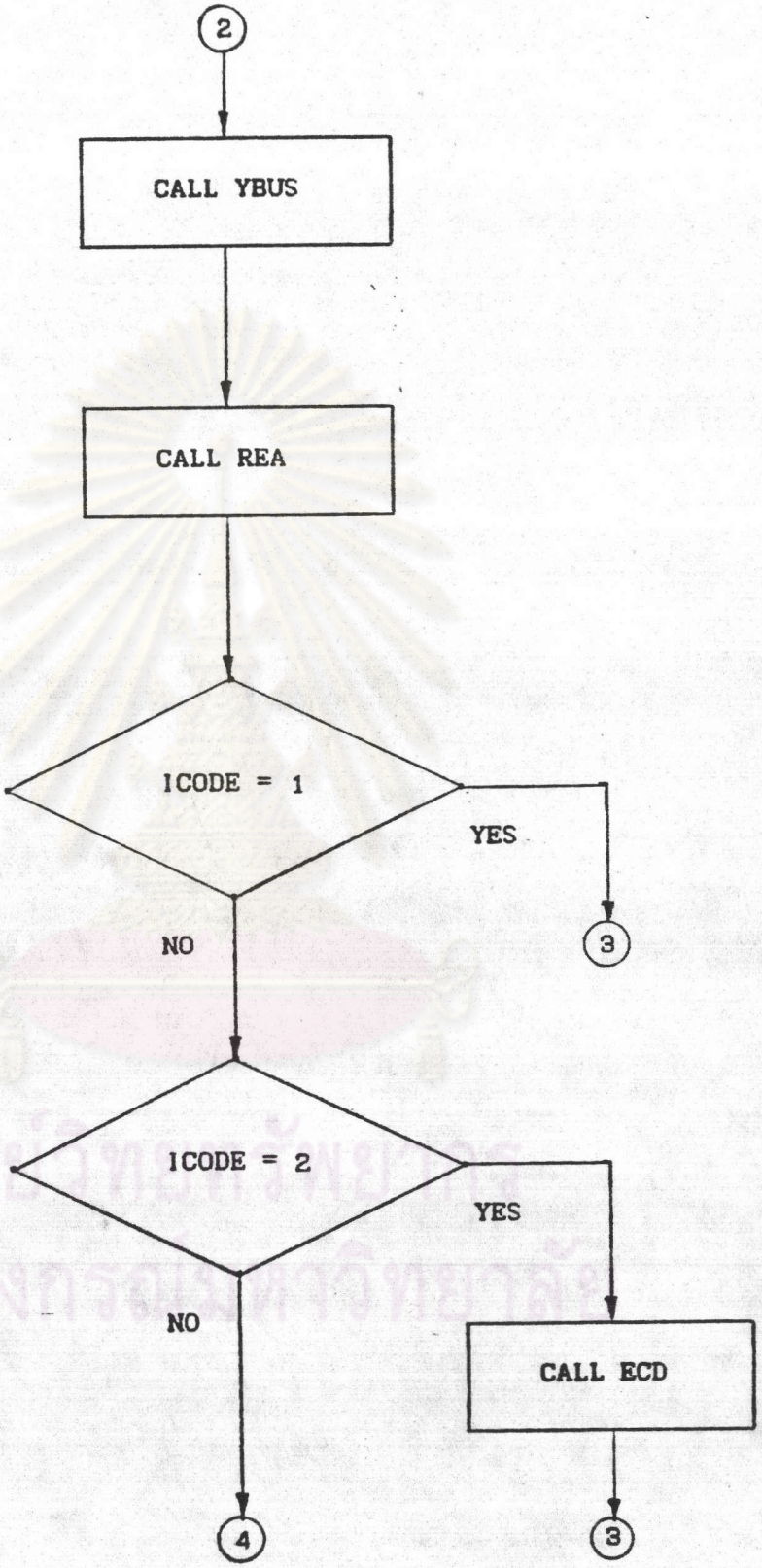
ภาคผนวก ข

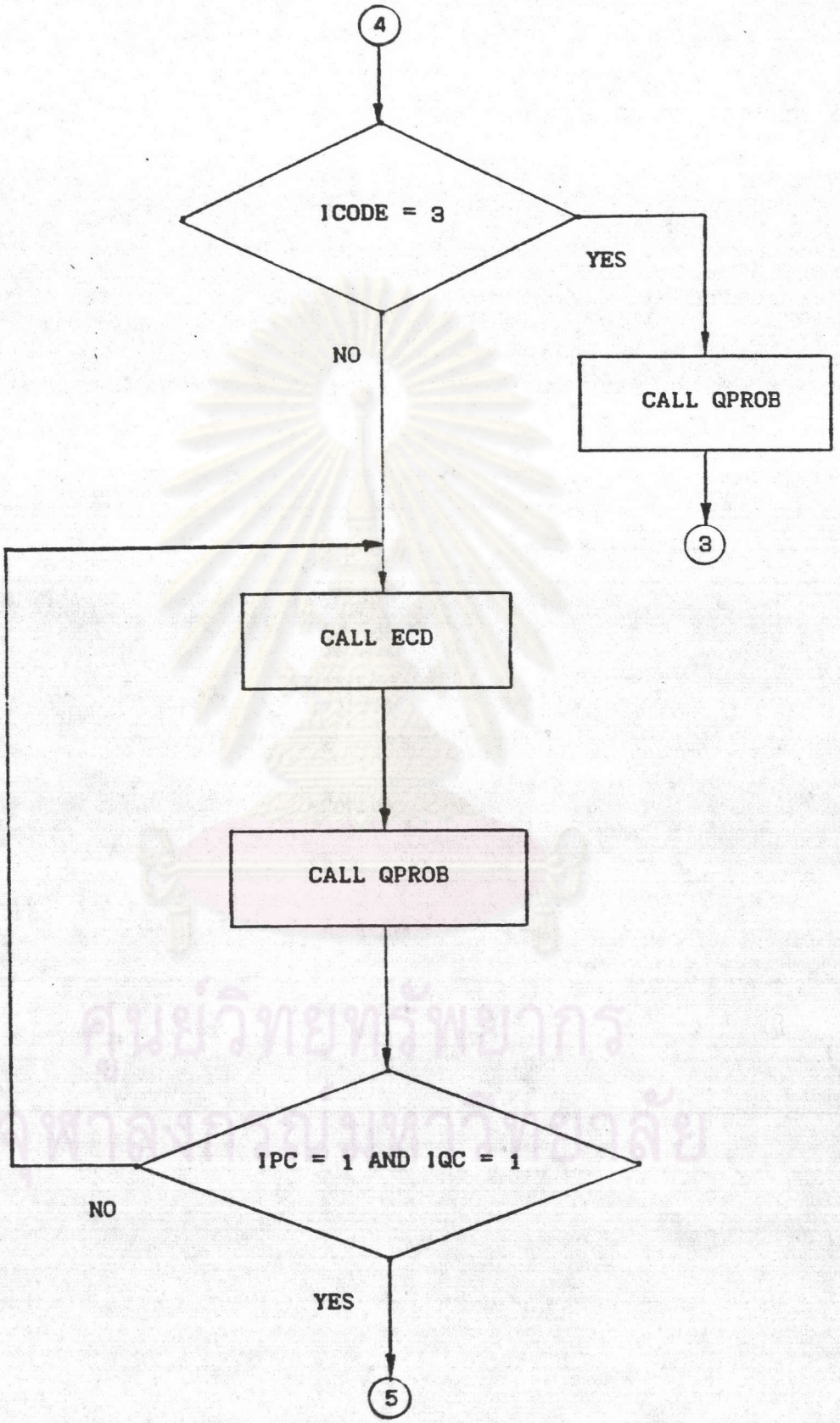
โฟลว์ชาร์ตของโปรแกรมการจ่ายโหลดอย่างประหยัด

โฟลว์ชาร์ต

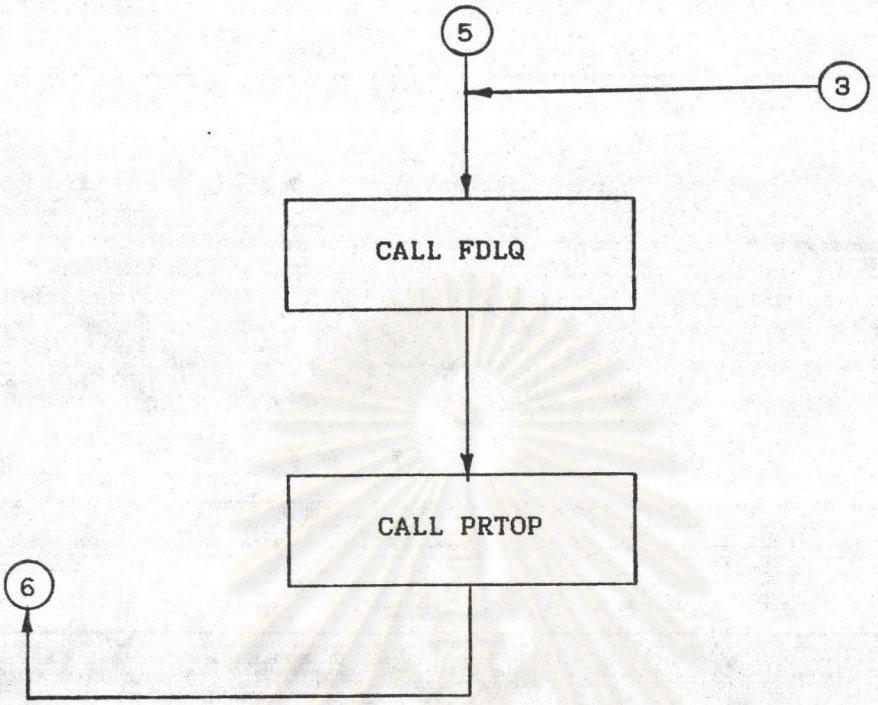



1
ถามว่าจะป้อนข้อมูลแบบใด
READ ICODE
NO
YES
ICODE = 1
CALL READ
CALL INPUT
CALL LIST
CALL DMOD
2


2
CALL YBUS
CALL REA
ICODE = 1
YES
3
NO
ICODE = 2
YES
CALL ECD
NO
4
3



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ความหมายของตัวแปรและโปรแกรมย่อยในโฟลว์ชาร์ต

ตัวแปร	ความหมาย
- IKOD	บอกว่าจะทำปัญหาใด โดยมีความหมายดังนี้คือ
	IKOD = 1 ทำโหลดโฟลว์
	IKOD = 2 ทำP-Problem
	IKOD = 3 ทำQ-Problem
	IKOD = 4 ทำP&Q-Problem
	IKOD = 5 ออกจากโปรแกรม
- ICODE	บอกว่าจะป้อนข้อมูลแบบใด
	ICODE ≠ 1 อ่านข้อมูลจากดิสก์
	ICODE = 1 ป้อนข้อมูลแบบINTERACTIVE
- IPC	จำนวนอิทเทอร์เรชันของการจัดสรรกำลังจริง
- IQC	จำนวนอิทเทอร์เรชันของการจัดสรรกำลังรีแอกทีฟ

โปรแกรมย่อย	ความหมาย
- READ	อ่านข้อมูลป้อนเข้าจากดิสก์
- INPUT	ถามและอ่านข้อมูลป้อนเข้าแบบ INTERACTIVE
- LIST	พิมพ์ข้อมูลป้อนเข้าเพื่อตรวจสอบและแก้ไข
- DMOD	ปรับปรุงข้อมูลป้อนเข้า เช่น ทำ MW ให้เป็น PU ทำอิมพีแดนซ์ให้เป็นแอดมิตแตนซ์ ฯลฯ
- YBUS	สร้างและพิมพ์ค่าบัสแอดมิตแตนซ์เมทริกซ์
- REA	จัดอันดับของบัสใหม่ให้เป็น โหลดบัส - บัสควบคุมแรงดัน - บัสอ้างอิง
- ECD	ทำการจัดสรรกำลังจริงโดยให้ตัวแปรควบคุมของการจัดสรร กำลังรีแอกทีฟคงที่(ตามหัวข้อที่ 4.5)
- QPROB	ทำการจัดสรรกำลังรีแอกทีฟโดยให้ตัวแปรควบคุมของการจัดสรร กำลังจริงคงที่(ตามหัวข้อที่ 4.6)
- FDLQ	ทำโหลดโฟลว์โดยใช้วิธีฟาสต์ดีคัปเปิล
- PRTOP	คำนวนค่าต่าง เช่น กำลังไหลในสายส่งและหม้อแปลง กำลังสูญเสีย ต้นทุนการผลิตรวมของระบบ ฯลฯ และพิมพ์ผล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

โปรแกรมการจ่ายโหลดอย่างประหยัด

```
C *** ELDPQ.FOR ***
C MAIN PROGRAM " ECONOMIC LOAD DISPATCH BASE ON RAEL AND REACTIVE ALLCATIONS"
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      WRITE(*,18)
      WRITE(*,19)
      WRITE(*,28)
      WRITE(*,38)
 19   FORMAT(/,2X,'ENERGY SYSTEM RESEARCH LABARATORY',/,2X,
     +'ELECTRICAL ENGINEERING DEPARTMENT, CHULALONGKORN UNIVERSITY')
 18   FORMAT(/,2X,'*** ECONMIC LOAD DISPATCH ***',
     +'   VERSION 1.0'
     +,/,5X,'DATE: 13:04:1987')
 28   FORMAT(/,2X,'** LIMITTING OF THIS PROGRAM **')
 38   FORMAT(2X,'MAXIMUM BUS NUMBER = 30',/,
     +2X,'MAXIMUM LINE NUMBER = 50')
686   WRITE(*,48)
 48   FORMAT(/,2X,'WHICH PROBLEM DO YOU WANT ?',/,5X,'1) LOAD FLOW',
     +T40,'2) P-PROBLEM'/,5X,'3) Q-PROBLEM',T40,'4) P&Q-PROBLEM',/,
     +5X,'5) EXIT',/,2X,'SELECT ',$)
      READ(*,58)IKOD
 58   FORMAT(I5)
      IF(IKOD.LE.0.OR.IKOD.GE.6) GO TO 686
      IF(IKOD.EQ.5) GO TO 999
      WRITE(*,108)
108   FORMAT(/,2X,'INTERACTIVE MODE OR DATA FORM DISK [1 OR 0] ',$)
      READ(*,118)ICODE
118   FORMAT(I10)
      IF(ICODE.EQ.1) CALL INPUT
      IF(ICODE.NE.1) CALL READ
      CALL LIST
      CALL DMOD
      CALL YBUS
```

```
      DO 10 I=1,NB
      VM(I)=VSPEC(I)
      AG(I)=0.0
      IF(NTB(I).EQ.3) IS=I
      NAA(I)=NTB(I)
10    CONTINUE
      CALL REA
      GO TO(30,100,110,120),IKOD
100   CALL ECD(LL,LLL,LLLL,1,0)
      GO TO 30
110   CALL QPROB(LL,LLL,LLLL,0,0)
      GO TO 30
120   CONTINUE
      DO 20 LLLL=1,LNIT
      CALL ECD(LL,IPC,LLLL,1,0)
      CALL QPROB(LL,IQC,LLLL,0,0)
      IF(IPC.EQ.1.AND.IQC.EQ.1) GO TO 30
20    CONTINUE
30    CALL FDLQ(LL,LLL,LLLL,0)
      CALL PRTOP(LL,LLL,LLLL)
      GO TO 686
999   CONTINUE
      STOP
      END



C SUBROUTINE TO DO P-PROBLEM
      SUBROUTINE ECD(LL,LLL,LLLL,IDP,IDL)
      COMPLEX YB,YSER,SUM,A3,VG,TRUN,S,R
      DIMENSION EGAT(30,30)
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 500 LLL=1,LNIT
      ICE=IDL
      IF(LLL.EQ.1)ICE=0
      CALL FDLF(LL,LLL,LLLL,ICE)
      CALL PCOST(COST,TPG,0)
      JAR=NPV+1
      WRITE(*,103)
 103  FORMAT(2X,'BUS   V.MAG. (PU)  V.ANG. (RAD)   P.GEN. (PU)')
      DO 5 I=1,NB
      WRITE(*,113)I,VM(I),AG(I),PG(I)
 113  FORMAT(2X,I3,3(F14.6))
      II=I+NB
      DV(I)=0.0
      DV(II)=0.0
      XX(I)=0.0
```

```
      IF(NAA(I).EQ.1) GO TO 5
      CALL QCAL(I)
      AK(I)=PG(I)
5     CONTINUE
      WRITE(*,123)COST
123   FORMAT(2X,'COST = ',F12.6)
      CALL DPSW(IDP)
      WRITE(*,243)TPG
243   FORMAT(2X,'TOTAL POWER GEN. = ',F12.6)
      RAM1=B(IS)+C(IS)*PG(IS)*PBASE*2
      IF(A(IS).LT.0.0) RAM1=20.0
      RAM=RAM1
      CALL MCAL(RAM,S2,LLL)
      FF1=TPG-S2
      RAM2=RAM1
      IF(FF1) 210,888,220
210   RAM2=RAM2-0.5
      RAM=RAM2
      CALL MCAL(RAM,S2,LLL)
      FF2=TPG-S2
      IF(FF2.LT.0.0) GO TO 210
      KIT=0
      GO TO 230
220   RAM2=RAM2+0.5
      RAM=RAM2
      CALL MCAL(RAM,S2,LLL)
      FF2=TPG-S2
      IF(FF2.GT.0.0) GO TO 220
      KIT=1.0
230   DO 90 I=1,100
      RAM=RAM1+(FF1*(RAM2-RAM1)/(FF1-FF2))
      CALL MCAL(RAM,S2,LLL)
      XYZ=TPG-S2
333   FORMAT(2X,I3,4F12.6)
      THAM=ABS(XYZ)
      IF(THAM.LE.ERROR) GO TO 888
      IF(I.GE.2) GO TO 100
      IF(KIT.EQ.1) GO TO 250
      IF(XYZ.GT.0.0) GO TO 100
      RAM1=RAM2
      FF1=FF2
      GO TO 100
250   IF(XYZ.LT.0.0) GO TO 100
      RAM1=RAM2
      FF1=FF2
100   RAM2=RAM
      FF2=XYZ
90    CONTINUE
888   WRITE(*,503)LLL,RAM
503   FORMAT(/,2X,I3,'RAM = ',F12.6)
      S7=0.0
      DO 110 I=1,NB
      IF(NTB(I).EQ.1) GO TO 110
      S8=ABS(PG(I)-AK(I))
      IF(S8.GT.S7) S7=S8
```

```
110  CONTINUE
     IF(S7.LE.ERROR) GO TO 999
500  CONTINUE
999  WRITE(*,523)LLLL
523  FORMAT(/,2X,'PASS ECD',I3)
     RETURN
     END



C SUBROUTINE TO DO Q-PROBLEM
     SUBROUTINE QPROB(LL,LLL,LLLL,IDP,IDL)
     COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
     DIMENSION EGAT(60,60),TOA(30,30),WEA(30),UFO(30)
     COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
    +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
    +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
    +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
    +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
    +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
    +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
     BELL=0.0
     IF(IDP.EQ.0) JON=NB-1
     IF(IDP.EQ.4) JON=0
     CALL FDLQ(LL,LLL,LLLL,5)
     CALL PCOST(COST,TPG,0)
     CALL PENAL(PNT)
     FFO=PC(IS)+PNT
     SOS=FFO
     DO 500 LLL=1,LNIT
     JAR=NPV+1
     IF(IDP.EQ.0) KKK=NB+NPQ-1
     IF(IDP.EQ.4) KKK=NPQ
     WRITE(*,347)
347  FORMAT(2X,'BUS  V. MAG. (PU)  V. ANG.(RAD)  P. GEN. (PU)')
     DO 5 I=1,NB
     XX(I)=0.0
     II=I+NB
     WEA(I)=0.0
     DV(I)=0.0
     DV(II)=0.0
     WRITE(*,327)I,VM(I),AG(I),PG(I)
327  FORMAT(2X,I3,3(2X,F12.6))
5    CONTINUE
     WRITE(*,357)COST
357  FORMAT(2X,'COST = ',F12.6)
     DO 600 L=1,NB
     IF(NAA(L).EQ.1) GO TO 600
     I=NK(L)-NPQ
     IF(QG(L).GE.QMIN(L).AND.QG(L).LE.QMAX(L)) GO TO 600
     IF(QG(L).LT.QMIN(L)) WEA(I)=-PNCT*(QG(L)-QMIN(L))*2.0
     IF(QG(L).GT.QMAX(L)) WEA(I)=-PNCT*(QG(L)-QMAX(L))*2.0
```

```
600  CONTINUE
     IF(IDP.EQ.4) GO TO 200
     DO 610 L=1,NB
     IF(NAA(L).EQ.1) GO TO 610
     I=NK(L)-NPQ
     IF(WEA(I).EQ.0.0) GO TO 610
     DO 620 K=1,NB
     IF(NAA(K).EQ.3.OR.L.EQ.K) GO TO 620
     J=NK(K)
     IF(GB(L,K).NE.0.0.OR.BB(L,K).NE.0.0) GO TO 630
     TOA(I,J)=0.0
     GO TO 620
630  CALL PTM(AG(L),AG(K),CLK,SLK)
     TOA(I,J)=-VM(L)*VM(K)*(GB(L,K)*CLK+BB(L,K)*SLK)
620  CONTINUE
     TOA(I,I)=PC(L)-GB(L,L)*VM(L)**2
610  CONTINUE
     KOTO=NB-1
     DO 640 I=1,KOTO
     DO 640 J=1,JAR
     DV(I)=DV(I)+WEA(J)*TOA(J,I)
640  CONTINUE
200  CONTINUE
     DO 650 L=1,NB
     IF(NAA(L).EQ.1) GO TO 650
     I=NK(L)-NPQ
     IF(WEA(I).EQ.0.0) GO TO 650
     DO 660 K=1,NB
     IF(NAA(K).NE.1.OR.L.EQ.K) GO TO 660
     J=NK(K)
     IF(GB(L,K).NE.0.0.OR.BB(L,K).NE.0.0) GO TO 670
     TOA(I,J)=0.0
     GO TO 660
670  CALL PTM(AG(L),AG(K),CLK,SLK)
     TOA(I,J)=VM(L)*(GB(L,K)*SLK-BB(L,K)*CLK)
660  CONTINUE
650  CONTINUE
     DO 680 I=1,NPQ
     DO 680 J=1,JAR
     II=I+JON
     DV(II)=DV(II)+WEA(J)*TOA(J,I)
680  CONTINUE
     DO 690 L=1,NB
     IF(NAA(L).NE.1) GO TO 690
     II=NK(L)+NB-1
     IF(VM(L).GE.VMIN(L).AND.VM(L).LE.VMAX(L)) GO TO 690
     IF(VM(L).LT.VMIN(L))DV(II)=DV(II)-PNCT*(VM(L)-VMIN(L))*2
     IF(VM(L).GT.VMAX(L))DV(II)=DV(II)-PNCT*(VM(L)-VMAX(L))*2
690  CONTINUE
     CALL DPSW(IDP)
     IF(IDP.EQ.4) GO TO 210
     DO 440 L=1,NB
     IF(NAA(L).EQ.3) GO TO 440
     I=NK(L)
     DO 40 K=1,NB
```

```
      IF(NAA(K).EQ.1) GO TO 40
      J=NK(K)-NPQ
      IF(L.EQ.K) GO TO 45
      IF(GB(L,K).NE.0.0.OR.BB(L,K).NE.0.0) GO TO 50
      EGAT(J,I)=0.0
      GO TO 40
50    CONTINUE
      CALL PTM(AG(L),AG(K),CIJ,SIJ)
      EGAT(J,I)=VM(L)*(BB(L,K)*SIJ+GB(L,K)*CIJ)
      GO TO 40
45    CONTINUE
      EGAT(J,I)=PC(L)/VM(L)+GB(L,L)*VM(L)
40    CONTINUE
440   CONTINUE
210   CONTINUE
      DO 70 L=1,NB
      IF(NAA(L).NE.1) GO TO 70
      I=NK(L)+JON
      DO 75 K=1,NB
      IF(NAA(K).EQ.1) GO TO 75
      J=NK(K)-NPQ
      IF(GB(L,K).NE.0.0.OR.BB(L,K).NE.0.0) GO TO 80
      EGAT(J,I)=0.0
      GO TO 75
80    CONTINUE
      CALL PTM(AG(L),AG(K),CIJ,SIJ)
      EGAT(J,I)=VM(L)*(GB(L,K)*SIJ-BB(L,K)*CIJ)
75    CONTINUE
70    CONTINUE
      DO 65 I=1,JAR
      DO 65 J=1,KKK
      XX(I)=XX(I)+EGAT(I,J)*DV(J)
65    CONTINUE
      DO 85 I=1,JAR
      WRITE(*,137)I,XX(I)
137   FORMAT(2X,'XX(',I2,' ) = ',F12.6)
85    CONTINUE
      DO 750 L=1,NB
      IF(NAA(L).EQ.1) GO TO 750
      I=NK(L)-NPQ
      IF(NEA(I).EQ.0.0) GO TO 750
      DO 760 K=1,NB
      IF(NAA(K).EQ.1.OR.L.EQ.K) GO TO 760
      J=NK(K)-NPQ
      IF(GB(L,K).NE.0.0.OR.BB(L,K).NE.0.0) GO TO 770
      TOA(I,J)=0.0
      GO TO 760
770   CALL PTM(AG(L),AG(K),CLK,SLK)
      TOA(I,J)=VM(L)*(GB(L,K)*SLK-BB(L,K)*CLK)
760   CONTINUE
      TOA(I,I)=QC(L)/VM(L)-BB(L,L)*VM(L)
750   CONTINUE
      DO 780 I=1,JAR
      DO 780 J=1,JAR
      XX(I)=XX(I)-NEA(J)*TOA(J,I)
```

```
780  CONTINUE
     DO 90 L=1,NB
     IF(NAA(L).EQ.1) GO TO 90
     I=NK(L)-NPQ
     IF(L.EQ.IS) GO TO 100
     IF(GB(IS,L).NE.0.0.OR.BB(IS,L).NE.0.0) GO TO 95
     XD(I)=0.0
     GO TO 90
95   CONTINUE
     XD(I)=VM(IS)*(GB(IS,L)*COS(AG(L))-BB(IS,L)*SIN(AG(L)))
     GO TO 90
100  CONTINUE
     XD(I)=PC(IS)/VM(IS)+VM(IS)*GB(IS,IS)
90   CONTINUE
     DO 185 I=1,JAR
     WRITE(*,147)I,XD(I)
147  FORMAT(2X,'XD(',I2,' ) = ',F12.6)
185  CONTINUE
     DO 105 I=1,JAR
     XX(I)=XX(I)+XD(I)
     WRITE(*,137)I,XX(I)
105  CONTINUE
     AXX=0.0
     BUT=0.0
     DO 110 L=1,NB
     IF(NAA(L).EQ.1) GO TO 110
     I=NK(L)-NPQ
     IF(VM(L).EQ.VMAX(L).AND.XX(I).LT.0.0) XX(I)=0.0
     IF(VM(L).EQ.VMIN(L).AND.XX(I).GT.0.0) XX(I)=0.0
     SGC=ABS(XX(I))
     IF(SGC.GT.BUT) BUT=SGC
     AXX=AXX+XX(I)**2
110  CONTINUE
     IF(BUT.LE.ERROR) GO TO 999
     IF(LLL.EQ.1) GO TO 335
     BELL=AXX/AXX1
335  AXX1=AXX
     AXX2=0.0
     DO 190 I=1,JAR
     XX(I)=-XX(I)+BELL*UFO(I)
     UFO(I)=XX(I)
     AXX2=AXX2+XX(I)**2
     WRITE(*,137)I,XX(I)
190  CONTINUE
     AXX2=SQRT(AXX2)
     DO 120 I=1,JAR
     XX(I)=XX(I)/AXX2
120  CONTINUE
     ALP=100.0
     DO 170 L=1,NB
     RU(L)=VM(L)
     IF(NAA(L).EQ.1) GO TO 170
     I=NK(L)-NPQ
     IF(XX(I))169,170,171
169  AK(I)=(VMIN(L)-VM(L))/XX(I)
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      GO TO 172
  171 AK(I)=(VMAX(L)-VM(L))/XX(I)
  172 IF(AK(I).LT.ALP.AND.AK(I).NE.0.0) ALP=AK(I)
      WRITE(*,414)L,XX(I)
  414 FORMAT(2X,'U (',I2,' ) = ',F12.6)
  170 CONTINUE
      WRITE(*,307)ALP
  307 FORMAT(2X,'ALP = ',F12.6)
      IF(ALP.EQ.100.0) GO TO 999
      AH=0.5*ALP
      CALL VADJ(AH)
      CALL FDLF(LL,LLL,LLLL,IDL)
      CALL PCAL(IS)
      CALL PENAL(PNT)
      FF2=PC(IS)+PNT
      AH=ALP
      CALL VADJ(AH)
      CALL FDLF(LL,LLL,LLLL,IDL)
      CALL PCAL(IS)
      CALL PENAL(PNT)
      FF1=PC(IS)+PNT
      WRITE(*,337)FF0,FF2,FF1
  337 FORMAT(2X,'POWER AT SW.BUS =',3(2X,F12.6))
      AIA=FF1-4*FF2+3*FF0
      TOT=4*(FF1-2*FF2+FF0)
      PTT=AIA/TOT
      FF5=FF0-FF1
      IF(PTT.LT.0.0.AND.FF5.GT.ERROR) PTT=1.0
      IF(PTT.LT.0.0.AND.FF5.LE.ERROR) PTT=0.0
      IF(PTT.GE.1.0) PTT=1.0
      WRITE(*,317)PTT
  317 FORMAT(2X,'PTT = ',F12.6)
      AH=ALP*PTT
      CALL VADJ(AH)
      IF(PTT.EQ.0.0) GO TO 999
      CALL FDLQ(LL,LLL,LLLL,5)
      CALL PCOST(COST,TPG,0)
      CALL PCAL(IS)
      CALL PENAL(PNT)
      FF0=PC(IS)+PNT
      ASOS=ABS(FF0-SOS)
      IF(ASOS.LE.ERROR) GO TO 999
      SOS=FF0
  180 CONTINUE
      KPF=KPF+1
  500 CONTINUE
  999 WRITE(*,707)LLLL
  707 FORMAT(/,2X,'PASS Q-PROB',I3)
      RETURN
      END
```

```
C SUBROUTINE TO READ INPUT DATA BY INTERACTIVE MODE
      SUBROUTINE INPUT
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),EU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      WRITE(*,128)
  128 FORMAT(//,2X,'*** GENERAL SYSTEM DATA ***',//,10X,
     +'NUMBER OF BUSES = ',$)
      READ(*,138)NB
  138 FORMAT(I5)
      WRITE(*,148)
  148 FORMAT(/,10X,'NUMBER OF LINES = ',$)
      READ(*,138)NL
      WRITE(*,208)
  208 FORMAT(/,10X,'BASE MVA = ',$)
      READ(*,198)PBASE
      WRITE(*,188)
  188 FORMAT(/,10X,'MAX. ERROR = ',$)
      READ(*,198)ERROR
  198 FORMAT(F12.6)
      WRITE(*,207)
  207 FORMAT(/,10X,'ACCELERATING FACTOR = ',$)
      READ(*,198)ALPHA
      WRITE(*,178)
  178 FORMAT(/,10X,'MAXIMUM ITERATION PERMISSIBLE = ',$)
      READ(*,138)LNIT
      WRITE(*,108)
  108 FORMAT(/,10X,'CONSTANT OF PENALTY FUNCTION = ',$)
      READ(*,198)PNCT
      WRITE(*,618)
  618 FORMAT(//,2X,'*** BUS DATA ***')
      DO 20 I=1,NB
      WRITE(*,358)I
  358 FORMAT(//,10X,': BUS NO. ',I3,/,
     +20X,'*** LOAD BUS              = 1',/,
     +20X,'    VOLTAGE CONTROLLED BUS = 2',/,
     +20X,'    SWING BUS              = 3 ***',//,
     +13X,'BUS TYPE = ',$)
      READ(*,138)NTB(I)
      WRITE(*,368)
  368 FORMAT(/,13X,'SPECIFY VOLTAGE (PU) = ',$)
      READ(*,198)VSPEC(I)
      WRITE(*,378)
  378 FORMAT(/,13X,'BASE VOLTAGE (KV) = ',$)
      READ(*,198)VB(I)
      WRITE(*,388)
  388 FORMAT(/,13X,'REAL POWER GENERATE (MW) = '$)
      READ(*,198)PG(I)
      WRITE(*,398)
  398 FORMAT(/,13X,'REACTIVE POWER GENERATE (MVAR) = ',$)
```

```
      READ(*,198)QG(I)
      WRITE(*,408)
  408 FORMAT(/,13X,'REAL POWER DEMAND (MW) = ',$)
      READ(*,198)PD(I)
      WRITE(*,418)
  418 FORMAT(/,13X,'REACTIVE POWER DEMAND (MVAR) = ',$)
      READ(*,198)QD(I)
      WRITE(*,348)
  348 FORMAT(/,13X,'SHUNT SUSCEPTANCE (PU) = ',$)
      READ(*,198)YC(I)
   20 CONTINUE
      WRITE(*,628)
  628 FORMAT(//,2X,'*** LINE OR TRANSFORMER DATA ***')
      DO 5 I=1,NL
      WRITE(*,228)I
  228 FORMAT(//,10X,': LINE OR TRANSFORMER NO. ',I3,
     +//,13X,'SENDING BUS (P) = ',$)
      READ(*,138)NSB(I)
      WRITE(*,238)
  238 FORMAT(/,13X,'ENDING BUS (Q) = ',$)
      READ(*,138)NEB(I)
      WRITE(*,248)
  248 FORMAT(/,13X,'SERIES RESISTANCE (PU) = ',$)
      READ(*,198)RR
      WRITE(*,258)
  258 FORMAT(/,13X,'SERIES REACTANCE (PU) = ',$)
      READ(*,198)XL
      YSER(I)=CMPLX(RR,XL)
      WRITE(*,268)
  268 FORMAT(/,13X,'SUSCEPTANCE OF LINE CHARGING (PU) = '$)
      READ(*,198)YSHT(I)
      WRITE(*,308)
  308 FORMAT(/,13X,'TRANSF. RATIO = ',$)
      READ(*,198)TR(I)
      WRITE(*,318)
  318 FORMAT(/,13X,'PHASE SHIFT (DEG.) = ',$)
      READ(*,198)TX(I)
    5 CONTINUE
      WRITE(*,548)
  548 FORMAT(//,2X,'*** DATA OF GENERATOR COST FUNCTION OR WATER ',
     +'RATE ***')
      DO 270 I=1,NB
      NTG(I)=0
      IF(NTB(I).EQ.1) GO TO 270
      WRITE(*,558)I
  558 FORMAT(//,10X,': BUS NO.',I3)
      WRITE(*,818)
  818 FORMAT(/,13X,'TYPE OF GENERATOR, HYDRO OR THERMAL[1 OR 0] ',$)
      READ(*,138)NTG(I)
      WRITE(*,828)
  828 FORMAT(/,13X,'A = ',$)
      READ(*,198)A(I)
      WRITE(*,568)
  568 FORMAT(/,13X,'B = ',$)
      READ(*,198)B(I)
```

```
      WRITE(*,578)
  578 FORMAT(/,13X,'C = ',$)
      READ(*,198)C(I)
      IF(NTG(I).EQ.0) GO TO 270
      WRITE(*,838)
  838 FORMAT(/,13X,'D = ',$)
      READ(*,198)D(I)
      WRITE(*,848)
  848 FORMAT(/,13X,'E = ',$)
      READ(*,198)E(I)
      WRITE(*,858)
  858 FORMAT(/,13X,'F = ',$)
      READ(*,198)F(I)
      WRITE(*,868)
  868 FORMAT(/,13X,'CURVE CHANGING PT. = ',$)
      READ(*,198)PCT(I)
      WRITE(*,878)
  878 FORMAT(/,13X,'COST OF WATER (GRAMMA) = ',$)
      READ(*,198)GRAM(I)
  270 CONTINUE
      WRITE(*,708)
  708 FORMAT(//,2X,'*** LIMIT OF BUS CONTROL & STATE VARIABLES ***')
      DO 240 I=1,NB
      PMAX(I)=0.0
      PMIN(I)=0.0
      QMAX(I)=0.0
      QMIN(I)=0.0
      VMAX(I)=0.0
      VMIN(I)=0.0
      WRITE(*,718)I
  718 FORMAT(//,10X,': BUS NO.',I3)
      IF(NTB(I).EQ.1) GO TO 245
      WRITE(*,719)
  719 FORMAT(/,13X,'MAX. REAL POWER GENERATE (MW) = ',$)
      READ(*,198)PMAX(I)
      WRITE(*,728)
  728 FORMAT(/,13X,'MIN. REAL POWER GENERATE (MW) = ',$)
      READ(*,198)PMIN(I)
      WRITE(*,738)
  738 FORMAT(/,13X,'MAX. REACTIVE POWER GENERATE (MVAR) = ',$)
      READ(*,198)QMAX(I)
      WRITE(*,748)
  748 FORMAT(/,13X,'MIN. REACTIVE POWER GENERATE (MVAR) = ',$)
      READ(*,198)QMIN(I)
  245 WRITE(*,758)
  758 FORMAT(/,13X,'MAX. VOLTAGE  (PU) = ',$)
      READ(*,198)VMAX(I)
      WRITE(*,768)
  768 FORMAT(/,13X,'MIN. VOLTAGE  (PU) = ',$)
      READ(*,198)VMIN(I)
  240 CONTINUE
      WRITE(*,769)
  769 FORMAT(//,2X,'*** LIMIT OF LINE VARIABLES ***')
      DO 250 I=1,NL
      PFMAX(I)=0.0
```

```
      QFMAX(I)=0.0
      WRITE(*,507)I
  507 FORMAT(//,10X,': LINE NO.',I3)
       WRITE(*,508)
  508 FORMAT(/,13X,'MAX. REAL POWER FLOW = ',$)
      READ(*,198)PFMAX(I)
      WRITE(*,518)
  518 FORMAT(/,13X,'MAX. REACTIVE POWER FLOW = ',$)
      READ(*,198)QFMAX(I)
  250 CONTINUE
      RETURN
      END



C SUBROUTINE FOR READ INPUT DATA FORM DISK
      SUBROUTINE READ
      COMPLEX YB,YSER,SUM,A3,VG,TRUN,S,R
      CHARACTER*8 NFILE
      COMMON NB,NL,PBASE,ERROR,ALPHA,LMIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      WRITE(*,109)
  109 FORMAT(/,2X,'NAME OF FILE TO READ [NOT OVER 8 CHARACTERS]: ',$)
      READ(*,119)NFILE
  119 FORMAT(A8)
      OPEN(5,FILE=NFILE,STATUS='OLD')
      READ(5,129)NB,NL,PBASE,ERROR,ALPHA,LMIT,PNCT
  129 FORMAT(2I5,3F12.6,I5,F12.6)
      DO 10 I=1,NB
      READ(5,139)NTB(I),VSPEC(I),VB(I),PG(I),QG(I),PD(I),QD(I),YC(I),
     +PCT(I),GRAM(I),NTG(I)
  139 FORMAT(I5,5F12.6,/,5X,4F12.6,I5)
   10 CONTINUE
      DO 15 I=1,NL
      READ(5,149)NSB(I),NEB(I),YSER(I),YSHT(I),TR(I),TX(I)
  149 FORMAT(2I5,5F12.6)
   15 CONTINUE
      DO 20 I=1,NB
      IF(NTB(I).EQ.1) GO TO 20
      READ(5,159)A(I),B(I),C(I),D(I),E(I),F(I)
  159 FORMAT(6F12.6)
   20 CONTINUE
      DO 25 I=1,NB
      READ(5,169)PMAX(I),PMIN(I),QMAX(I),QMIN(I),VMAX(I),VMIN(I)
  169 FORMAT(6F12.6)
   25 CONTINUE
      DO 30 I=1,NL
      READ(5,179)PFMAX(I),QFMAX(I)
  179 FORMAT(2F12.6)
```

```
 30   CONTINUE
      CLOSE(5)
      RETURN
      END


C SUBROUTINE TO PRINT OUT AND CORRECT INPUT DATA
      SUBROUTINE LIST
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      WRITE(*,126)
 126  FORMAT(//,15X,'*** GENERAL DATA OF POWER SYSTEM ***')
 515  WRITE(*,218)NB,NL,PBASE,ERROR,ALPHA,LNIT,PNCT
 218  FORMAT(/,15X,'NUMBER OF BUSES = ',I5,/,15X,
     +'NUMBER OF LINES = ',I5,/,15X,'BASE MVA = ',F8.2,/,15X,
     +'MAX.ERROR = ',F8.6,/,15X,'ACCELERATING FACTOR = ',F8.3,/,15X,
     +'MAX. ITERATION PERMISSIBLE = ',I4,/,15X,
     +'CONSTANT OF PENALTY FUNCTION = ',F12.6)
      WRITE(*,001)
 001  FORMAT(//,2X,'CORRECT GENERAL DATA,OR NOT [1 OR 0]: ',$)
      READ(*,011)IC1
 011  FORMAT(I5)
      IF(IC1.NE.1)GO TO 910
 1013 WRITE(*,021)
 021  FORMAT(/,5X,'WHICH DATA TYPE DO YOU WANT TO CORRECT ?',/,
     +10X,' 1 NUMBER OF BUSES',T41,' 2 NUMBER OF LINES',/,
     +10X,' 3 BASE MVA',T41,' 4 MAX. ERROR',/,
     +10X,' 5 ACCELERACTING FACTOR',T41,' 6 MAXIMUM ITRATION',/,
     +10X,' 7 CONSTANT OF PENALTY FUNCTION',/,5X,'SELECT ONE ',$)
      READ(*,011)IC2
      GO TO(1035,1040,1005,1010,1015,1020,1025),IC2
 1035 WRITE(*,181)
 181  FORMAT(/,5X,'NUMBER OF BUSES')
      WRITE(*,091)NB
      READ(*,011)NB
      GO TO 1030
 1040 WRITE(*,191)
 191  FORMAT(/,5X,'NUMBER OF LINES')
      WRITE(*,091)NL
      READ(*,011)NL
      GO TO 1030
 1005 WRITE(*,031)
 031  FORMAT(/,5X,'BASE MVA')
      WRITE(*,041)PBASE
 041  FORMAT(/,5X,'OLD VALUE =',F12.6,3X,';NEW VALUE = ',$)
      READ(*,051)PBASE
 051  FORMAT(F12.6)
      GO TO 1030
```

```
1010 WRITE(*,061)
061  FORMAT(/,5X,'MAX. ERROR')
     WRITE(*,041)ERROR
     READ(*,051)ERROR
     GO TO 1030
1015 WRITE(*,071)
071  FORMAT(/,5X,'ACCELERATING FACTOR')
     WRITE(*,041)ALPHA
     READ(*,051)ALPHA
     GO TO 1030
1020 WRITE(*,081)
081  FORMAT(/,5X,'MAXIMUM ITERATION')
     WRITE(*,091)LMIT
     READ(*,011)LMIT
091  FORMAT(/,5X,'OLD VALUE =',I3,3X,'NEW VALUE = ',$)
     GO TO 1030
1025 WRITE(*,111)
111  FORMAT(/,5X,'CONSTANT OF PENALTY FUNCTION')
     WRITE(*,041)PNCT
     READ(*,051)PNCT
1030 WRITE(*,101)
101  FORMAT(//,2X,'CORRECT ANOTHER DATA,OR NOT [1 OR 0] ',$)
     READ(*,011)IC3
     IF(IC3.EQ.1) GO TO 1013
910  IF(IC1.EQ.1) GO TO 515
510  WRITE(*,228)
228  FORMAT(///,2X,'*** BUSES DATA ***',/)
     WRITE(*,238)
238  FORMAT(2X,'¦----¦----¦-------¦-------¦-------¦-------¦'
    +,'-------¦-------¦---------¦')
     WRITE(*,248)
248  FORMAT(2X,'¦BUS ¦BUS ¦ VOLT  ¦ BASE  ¦  GENERATION   ¦'
    +,'     LOAD      ¦  SHUNT  ¦')
     WRITE(*,249)
249  FORMAT(2X,'¦    ¦    ¦       ¦ VOLT  ¦-------¦-------¦'
    +,'-------¦-------¦ SUSCEPT.¦')
     WRITE(*,259)
259  FORMAT(2X,'¦ NO.¦TYPE¦  (PU)  ¦ (KV)  ¦   MW  ¦ MVAR  ¦'
    +,'  MW   ¦ MVAR  ¦   (PU)  ¦')
     WRITE(*,238)
     DO 35 I=1,NB
     WRITE(*,258)I,NTB(I),VSPEC(I),VB(I),PG(I),QG(I),PD(I),QD(I),
    +YC(I)
35   CONTINUE
258  FORMAT(2X,'¦',2(1X,I2,1X,'¦'),F7.5,'¦',5(F7.2,'¦'),F9.5,'¦')
     WRITE(*,238)
     WRITE(*,251)
251  FORMAT(//,2X,'NOTE: BUS TYPE',T20,'1 = LOAD BUS',/,T20,
    +'2 = VOLTAGE CONTROLLED BUS',/,T20,'3 = REFERENCE BUS')
     WRITE(*,002)
002  FORMAT(//,2X,'CORRECT BUS DATA,OR NOT [1 OR 0] ',$)
     READ(*,011)IC1
     IF(IC1.NE.1) GO TO 920
1112 WRITE(*,022)
022  FORMAT(/,10X,'NO. OF BUS TO CORRECT = ',$)
```

```
      READ(*,011)I
1122  WRITE(*,032)
032   FORMAT(/,5X,'WHICH DATA TYPE DO YOU WANT TO CORRECT ?',/,7X,
     +' 1 BUS TYPE',T41,
     +' 2 SPECIFY VOLTAGE',/,7X,
     +' 3 BASE VOLTAGE',T41,
     +' 4 REAL POWER GENERATE',/,7X,
     +' 5 REACTIVE POWER GENERATE',T41,
     +' 6 REAL POWER DEMAND',/,7X,
     +' 7 REACTIVE POWER DEMAND',T41,
     +' 8 SHUNT SUSCESTANCE',/,5X,'SELECT ',$)
      READ(*,011)IC2
      GO TO(1105,1110,1115,1120,1125,1130,1135,1150),IC2
1105  WRITE(*,042)I
042   FORMAT(/,5X,'BUS TYPE OF BUS NO.',I3)
      WRITE(*,091)NTB(I)
      READ(*,011)NTB(I)
      GO TO 1160
1110  WRITE(*,062)I
062   FORMAT(/,5X,'SPECIFY VOLTAGE (PU) OF BUS NO.',I3)
      WRITE(*,041)VSPEC(I)
      READ(*,051)VSPEC(I)
      GO TO 1160
1115  WRITE(*,092)I
092   FORMAT(/,5X,'BASE VOLTAGE (PU) OF BUS NO.',I3)
      WRITE(*,041)VB(I)
      READ(*,051)VB(I)
      GO TO 1160
1120  WRITE(*,102)I
102   FORMAT(/,5X,'REAL POWER GENERATE (MW) OF BUS NO.',I3)
      WRITE(*,041)PG(I)
      READ(*,051)PG(I)
      GO TO 1160
1125  WRITE(*,112)I
112   FORMAT(/,5X,'REACTIVE POWER GENERATE (MVAR) OF BUS NO.',I3)
      WRITE(*,041)QG(I)
      READ(*,051)QG(I)
      GO TO 1160
1130  WRITE(*,122)I
122   FORMAT(/,5X,'REAL POWER DEMNAND (MW) OF BUS NO.',I3)
      WRITE(*,041)PD(I)
      READ(*,051)PD(I)
      GO TO 1160
1135  WRITE(*,132)I
132   FORMAT(/,5X,'REACTIVE POWER DEMAND (MVAR) OF BUS NO.',I3)
      WRITE(*,041)QD(I)
      READ(*,051)QD(I)
      GO TO 1160
1150  WRITE(*,162)I
162   FORMAT(/,5X,'SUSCESTANCE OF STATIC COMPENSATOR (PU) OF BUS NO.',
     +I3)
      WRITE(*,041)YC(I)
      READ(*,051)YC(I)
1160  WRITE(*,172)
172   FORMAT(//,2X,'CORRECT ANOTHER DATA OF THIS BUS,OR NOT [1 OR 0] ',
```

```
     +$)
      READ(*,011)IC3
      IF(IC3.EQ.1) GO TO 1122
      WRITE(*,182)
182   FORMAT(//,2X,'CORRECT DATA OF ANOTHER BUS,OR NOT [1 OR 0] ',$)
      READ(*,011)IC4
      IF(IC4.EQ.1) GO TO 1112
920   IF(IC1.EQ.1) GO TO 510
520   WRITE(*,108)
108   FORMAT(///,2X,'*** LINES DATA ***',/)
      WRITE(*,118)
118   FORMAT(2X,'¦------¦------¦------¦--------¦--------¦--------¦'
     +,'--------¦--------¦')
      WRITE(*,128)
128   FORMAT(2X,'¦ LINE ¦ SEND.¦ END. ¦    IMPEDANCE    ¦ LINE  ¦'
     +,' TRANSF.¦  PHASE ¦')
      WRITE(*,129)
129   FORMAT(2X,'¦      ¦ BUS ¦ BUS ¦--------¦--------¦         ¦'
     +,'        ¦ SHIFT ¦')
      WRITE(*,139)
139   FORMAT(2X,'¦ NO. ¦ (P) ¦ (Q) ¦   R   ¦    X   ¦CHARGING¦'
     +,' RATIO  ¦ (DEG.) ¦')
      WRITE(*,118)
      DO 10 I=1,NL
      WRITE(*,138)I,NSB(I),NEB(I),YSER(I),YSHT(I),TR(I),TX(I)
138   FORMAT(2X,'¦',3('  ',I2,'  ¦'),5(F7.4,1X,'¦'))
10    CONTINUE
      WRITE(*,118)
      WRITE(*,009)
009   FORMAT(//,2X,'CORRECT LINE OR TRANSFORMER DATA,OR NOT [1 OR 0] '
     +,$)
      READ(*,011)IC1
      IF(IC1.NE.1) GO TO 930
1211  WRITE(*,209)
209   FORMAT(/,5X,'NO.OF LINE TO CORRECT = ',$)
      READ(*,011)I
1221  WRITE(*,029)
029   FORMAT(/,5X,'WHICH DATA TYPE DO YOU WANT TO CORRECT ?',/,7X,
     +' 1 SENDING BUS NO.',T41,
     +' 2 ENDING BUS NO.',/,7X,
     +' 3 SERIES RESISTANCE',T41,
     +' 4 SERIES REACTANCE',/,7X,
     +' 5 SUSCESTANCE OF LINE CHARGING',T41,
     +' 6 TRANSF. RATIO',/,7X,
     +' 7 PHASE SHIFT',/,5X,'SELECT ',$)
      READ(*,011)IC2
      GO TO(1205,1210,1215,1220,1225,1230,1235),IC2
1205  WRITE(*,039)I
039   FORMAT(/,5X,'SENING BUS NO. OF LINE OR TR. NO.',I3)
      WRITE(*,091)NSB(I)
      READ(*,011)NSB(I)
      GO TO 1240
1210  WRITE(*,059)I
059   FORMAT(/,5X,'ENING BUS NO. OF LINE OR TR. NO.',I3)
      WRITE(*,091)NEB(I)
```

```
      READ(*,011)NEB(I)
      GO TO 1240
1215  WRITE(*,089)I
089   FORMAT(/,5X,'SERIES RESISTANCE OF LINE OR TR. NO.',I3)
      RR=REAL(YSER(I))
      XL=AIMAG(YSER(I))
      WRITE(*,041)RR
      READ(*,051)RR
      YSER(I)=CMPLX(RR,XL)
      GO TO 1240
1220  WRITE(*,099)I
099   FORMAT(/,5X,'SERIES REACTANCE OF LINE OR TR. NO.',I3)
      RR=REAL(YSER(I))
      XL=AIMAG(YSER(I))
      WRITE(*,041)XL
      READ(*,051)XL
      YSER(I)=CMPLX(RR,XL)
      GO TO 1240
1225  WRITE(*,109)I
109   FORMAT(/,5X,'SHUNT SUSCESTANCE OF LINE NO.',I3)
      WRITE(*,041)YSHT(I)
      READ(*,051)YSHT(I)
      GO TO 1240
1230  WRITE(*,119)I
119   FORMAT(/,5X,'TRANSF. RATIO OF LINE OR TR. NO.',I3)
      WRITE(*,041)TR(I)
      READ(*,051)TR(I)
      GO TO 1240
1235  WRITE(*,1729)I
1729  FORMAT(/,5X,'PHASSE SHIFT OF LINE OR TR. NO.',I3)
      WRITE(*,041)TX(I)
      READ(*,051)TX(I)
1240  WRITE(*,1739)
1739  FORMAT(//,2X,'CORRECT ANOTHER DATA OF THIS LINE,OR NOT [1 OR 0]'
     +,$)
      READ(*,011)IC3
      IF(IC3.EQ.1) GO TO 1221
      WRITE(*,149)
149   FORMAT(//,2X,'CORRECT DATA OF ANOTHER LINE,OR NOT [1 OR 0] ',$)
      READ(*,011)IC4
      IF(IC4.EQ.1) GO TO 1211
930   IF(IC1.EQ.1) GO TO 520
610   WRITE(*,608)
608   FORMAT(//,2X,'*** THERMAL GENERATOR COST FUNCTION ***',/)
      WRITE(*,618)
      WRITE(*,628)
      WRITE(*,638)
      WRITE(*,648)
      WRITE(*,618)
      DO 605 I=1,NB
      IF(NTB(I).EQ.1.OR.NTG(I).EQ.1) GO TO 605
      WRITE(*,668)I,A(I),B(I),C(I)
605   CONTINUE
      WRITE(*,618)
618   FORMAT(2X,'|-----|-----------|-----------|-----------|')
```

```
628  FORMAT(2X,'¦ BUS ¦     COST = A + B*PG + C*PG**2     ¦')
638  FORMAT(2X,'¦     ¦----------¦----------¦----------¦')
648  FORMAT(2X,'¦ NO. ¦     A    ¦     B    ¦     C    ¦')
668  FORMAT(2X,'¦'I3,2X,'¦',3(F10.5,' ¦'))
     WRITE(*,308)
     WRITE(*,318)
     WRITE(*,328)
     WRITE(*,338)
     WRITE(*,348)
     WRITE(*,318)
308  FORMAT(//,2X,'*** HYDRO GENERATOR WATER RATE ***',/)
318  FORMAT(2X,'¦----¦--------¦--------¦--------¦--------¦--------¦'
    +,'--------¦--------¦--------¦')
328  FORMAT(2X,'¦ BUS¦      FIRST  SECTION      ¦     SECOND  SECTI'
    +,'ON      ¦  CURVE ¦  COST  ¦')
338  FORMAT(2X,'¦ NO.¦--------¦--------¦--------¦--------¦--------¦'
    +,'--------¦ CHANG. ¦   OF   ¦')
348  FORMAT(2X,'¦    ¦    A   ¦    B   ¦    C   ¦    D   ¦    E   ¦'
    +,'    F   ¦ PT.(MW)¦  WATER ¦')
     DO 305 I=1,NB
     IF(NTB(I).EQ.1.OR.NTG(I).EQ.0) GO TO 305
     WRITE(*,358)I,A(I),B(I),C(I),D(I),E(I),F(I),PCT(I),GRAM(I)
358  FORMAT(2X,'¦ ',I2,' ¦',6(F8.4,'¦'),F8.2,'¦',F8.4,'¦')
305  CONTINUE
     WRITE(*,318)
     WRITE(*,003)
003  FORMAT(//,2X,'CORRECT DATA OF COST FUNCTION,OR NOT [1 OR 0] '
    +,$)
     READ(*,011)IC1
     IF(IC1.NE.1) GO TO 940
1312 WRITE(*,022)
     READ(*,011)I
1322 WRITE(*,033)
033  FORMAT(/,5X,'WHICH DATA TYPE DO YOU WANT TO CORRECT ?',/,7X,
    +' 1 GENERATOR TYPE',T41,' 2 A',/,7X,
    +' 3 B',T41,' 4 C',/,7X,' 5 D',T41,' 6 E',/,7X,
    +' 7 F',T41,' 8 CURVE CHANGING PT.',/,7X,
    +' 9 COST OF WATER (GRAMMA)',/,5X,'SELECT ',$)
     READ(*,011)IC2
     GO TO (1305,1310,1315,1320,1325,1330,1335,1340,1345),IC2
1305 WRITE(*,013)I
013  FORMAT(/,5X,'GENARATOR TYPE OF BUS NO.',I3)
     WRITE(*,091)NTG(I)
     READ(*,011)NTG(I)
     GO TO 1360
1310 WRITE(*,023)I
023  FORMAT(/,5X,'A OF BUS NO.',I3)
     WRITE(*,041)A(I)
     READ(*,051)A(I)
     GO TO 1360
1315 WRITE(*,103)I
103  FORMAT(/,5X,'B OF BUS NO.',I3)
     WRITE(*,041)B(I)
     READ(*,051)B(I)
     GO TO 1360
```

```
1320 WRITE(*,043)I
043  FORMAT(/,5X,'C OF BUS NO.',I3)
     WRITE(*,041)C(I)
     READ(*,051)C(I)
     GO TO 1360
1325 WRITE(*,053)I
053  FORMAT(/,5X,'D OF BUS NO.',I3)
     WRITE(*,041)D(I)
     READ(*,051)D(I)
     GO TO 1360
1330 WRITE(*,063)I
063  FORMAT(/,5X,'E OF BUS NO.',I3)
     WRITE(*,041)E(I)
     READ(*,051)E(I)
     GO TO 1360
1335 WRITE(*,073)I
073  FORMAT(/,5X,'F OF BUS NO.',I3)
     WRITE(*,041)F(I)
     READ(*,051)F(I)
     GO TO 1360
1340 WRITE(*,083)I
083  FORMAT(/,5X,'CURVE CHANGING PT. OF BUS NO.',I3)
     WRITE(*,041)PCT(I)
     READ(*,051)PCT(I)
     GO TO 1360
1345 WRITE(*,093)I
093  FORMAT(/,5X,'COST OF WATER (GRAMMA) OF BUS NO.',I3)
     WRITE(*,041)GRAM(I)
     READ(*,051)GRAM(I)
1360 WRITE(*,173)
173  FORMAT(//,2X,'CORRECT ANOTHER DATA OF THIS BUS,OR NOT [1 OR 0]'
    +,1X,$)
     READ(*,011)IC3
     IF(IC3.EQ.1) GO TO 1322
     WRITE(*,183)
183  FORMAT(//,2X,'CORRECT DATA OF ANOTHER BUS,OR NOT [1 OR 0] ',$)
     READ(*,011)IC4
     IF(IC4.EQ.1) GO TO 1312
940  IF(IC1.EQ.1) GO TO 610
780  WRITE(*,808)
808  FORMAT(//,2X,'*** LIMIT OF BUS VARIABLES ***',/)
     WRITE(*,818)
818  FORMAT(2X,
    +'¦-----¦-------¦-------¦-------¦-------¦-------¦-------¦',/,2X,
    +'¦ BUS ¦  P-GEN. (MW)  ¦ Q-GEN. (MVAR) ¦    VOLTAGE    ¦',/,2X,
    +'¦     ¦-------¦-------¦-------¦-------¦-------¦-------¦',/,2X,
    +'¦ NO. ¦  MAX  ¦  MIN  ¦  MAX  ¦  MIN  ¦  MAX  ¦  MIN  ¦')
     WRITE(*,828)
828  FORMAT(2X,
    +'¦-----¦-------¦-------¦-------¦-------¦-------¦-------¦')
     DO 775 I=1,NB
     WRITE(*,838)I,PMAX(I),PMIN(I),QMAX(I),QMIN(I),VMAX(I),VMIN(I)
838  FORMAT(2X,'¦',I3,2X,'¦',6(F7.2,'¦'))
775  CONTINUE
     WRITE(*,828)
```

```
      WRITE(*,006)
006   FORMAT(//,2X,'CORRECT LIMIT OF BUS VARIABLES,OR NOT [1 OR 0] '
     +,$)
      READ(*,011)IC1
      IF(IC1.NE.1) GO TO 950
1412  WRITE(*,026)
026   FORMAT(/,10X,'NO. OF BUS TO CORRECT = ',$)
      READ(*,011)I
1422  WRITE(*,036)
036   FORMAT(/,5X,'WHICH DATA TYPE DO YOU WANT TO CORRECT ?',/,7X,
     +' 1 MAX. REAL POWER GENERATE',T41,
     +' 2 MIN. REAL POWER GENERATE',/,7X,
     +' 3 MAX. REACTIVE POWER GENERATE',T41,
     +' 4 MIN. REACTIVE POWER GENERATE',/,7X,
     +' 5 MAX. VOLTAGE ',T41,
     +' 6 MIN. VOLTAGE ',/,5X,'SELECT ',$)
      READ(*,011)IC2
      GO TO (1415,1420,1425,1430,1405,1410),IC2
1405  WRITE(*,046)I
046   FORMAT(/,5X,'MAX. VOLTAGE (PU) OF BUS NO.',I3)
      WRITE(*,041)VMAX(I)
      READ(*,051)VMAX(I)
      GO TO 1460
1410  WRITE(*,066)I
066   FORMAT(/,5X,'MIN. VOLTAGE (PU) OF BUS NO.',I3)
      WRITE(*,041)VMIN(I)
      READ(*,051)VMIN(I)
      GO TO 1460
1415  WRITE(*,996)I
996   FORMAT(/,5X,'MAX. REAL POWER GENERATE (PU) OF BUS NO.',I3)
      WRITE(*,041)PMAX(I)
      READ(*,051)PMAX(I)
      GO TO 1460
1420  WRITE(*,906)I
906   FORMAT(/,5X,'MIN. REAL POWER GENERATE (MW) OF BUS NO.',I3)
      WRITE(*,041)PMIN(I)
      READ(*,051)PMIN(I)
      GO TO 1460
1425  WRITE(*,916)I
916   FORMAT(/,5X,'MAX. REACTIVE POWER GENERATE (MVAR) OF BUS NO.',I3)
      WRITE(*,041)QMAX(I)
      READ(*,051)QMAX(I)
      GO TO 1460
1430  WRITE(*,926)I
926   FORMAT(/,5X,'MIN. REACTIVE POWER GENERATE (MVAR) OF BUS NO.',I3)
      WRITE(*,041)QMIN(I)
      READ(*,051)QMIN(I)
1460  WRITE(*,176)
176   FORMAT(//,2X,'CORRECT ANOTHER DATA OF THIS BUS,OR NOT [1 OR 0]'
     +,1X,$)
      READ(*,011)IC3
      IF(IC3.EQ.1) GO TO 1422
      WRITE(*,186)
186   FORMAT(//,2X,'CORRECT DATA OF ANOTHER BUS,OR NOT [1 OR 0] ',$)
      READ(*,011)IC4
```

```
      IF(IC4.EQ.1) GO TO 1412
  950 IF(IC1.EQ.1) GO TO 780
  815 WRITE(*,858)
  858 FORMAT(//,2X,'*** LIMIT OF LINE FLOW ***',/)
      WRITE(*,868)
  868 FORMAT(2X,
     +'¦------¦--------¦--------¦',/,2X,
     +'¦ LINE ¦  MAX. LINE FLOW ¦',/,2X,
     +'¦      ¦--------¦--------¦',/,2X,
     +'¦  NO. ¦   MW   ¦  MVAR  ¦')
      WRITE(*,878)
  878 FORMAT(2X,'¦------¦--------¦--------¦')
      DO 795 I=1,NL
      WRITE(*,888)I,PFMAX(I),QFMAX(I)
  888 FORMAT(2X,'¦',I4,2X,'¦',2(F8.2,'¦'))
  795 CONTINUE
      WRITE(*,878)
      WRITE(*,004)
  004 FORMAT(//,2X,'CORRECT LIMIT OF LINE VARIABLES,OR NOT [1 OR 0] '
     +,$)
      READ(*,011)IC1
      IF(IC1.NE.1) GO TO 960
 1512 WRITE(*,024)
  024 FORMAT(/,10X,'NO. OF LINE TO CORRECT = ',$)
      READ(*,011)I
 1522 WRITE(*,034)
  034 FORMAT(/,5X,'WHICH DATA TYPE DO YOU WANT TO CORRECT ?',/,7X,
     +' 1 MAX. REAL POWER FLOW',T41,
     +' 2 MAX. REACTIVE POWER FLOW',/,5X,'SELECT ',$)
      READ(*,011)IC2
      GO TO (1505,1510),IC2
 1505 WRITE(*,044)I
  044 FORMAT(/,5X,'MAX. REAL POWER FLOW OF LINE NO.',I3)
      WRITE(*,041)PFMAX(I)
      READ(*,051)PFMAX(I)
      GO TO 1560
 1510 WRITE(*,064)I
  064 FORMAT(/,5X,'MAX. REACTIVE POWER FLOW OF LINE NO.',I3)
      WRITE(*,041)QFMAX(I)
      READ(*,051)QFMAX(I)
 1560 WRITE(*,174)
  174 FORMAT(//,2X,'CORRECT ANOTHER DATA OF THIS LINE,OR NOT [1 OR 0]'
     +,1X,$)
      READ(*,011)IC3
      IF(IC3.EQ.1) GO TO 1522
      WRITE(*,184)
  184 FORMAT(//,2X,'CORRECT DATA OF ANOTHER LINE,OR NOT [1 OR 0] ',$)
      READ(*,011)IC4
      IF(IC4.EQ.1) GO TO 1512
  960 IF(IC1.EQ.1) GO TO 815
      RETURN
      END
```

```
C SUBROUTINE TO MODIFLY INPUT DATA
      SUBROUTINE DMOD
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 505 I=1,NB
      PD(I)=PD(I)/PBASE
      QD(I)=QD(I)/PBASE
      PG(I)=PG(I)/PBASE
      QG(I)=QG(I)/PBASE
      PCT(I)=PCT(I)/PBASE
 505  CONTINUE
      DO 525 I=1,NL
      YSER(I)=CMPLX(1.0,0.0)/YSER(I)
      TX(I)=TX(I)*0.017453292
 525  CONTINUE
      DO 535 I=1,NB
      IF(NTB(I).NE.1) GO TO 535
      A(I)=0.0
      B(I)=0.0
      C(I)=0.0
 535  CONTINUE
      DO 920 I=1,NB
      PMAX(I)=PMAX(I)/PBASE
      PMIN(I)=PMIN(I)/PBASE
      QMAX(I)=QMAX(I)/PBASE
      QMIN(I)=QMIN(I)/PBASE
 920  CONTINUE
      RETURN
      END


C SUBROUTINE TO REORDER BUS
      SUBROUTINE REA
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30);PNCT
      NPV=0
      NPQ=0
      DO 105 I=1,NB
      IF(NAA(I).EQ.1) NPQ=NPQ+1
      IF(NAA(I).EQ.2) NPV=NPV+1
 105  CONTINUE
      NO=0
```

```
      NN=NPQ
      DO 5 I=1,NB
      IF(NAA(I).EQ.1) NO=NO+1
      IF(NAA(I).EQ.1) K=NO
      IF(NAA(I).EQ.2) NN=NN+1
      IF(NAA(I).EQ.2) K=NN
      IF(NAA(I).EQ.3) K=NB
      NK(I)=K
5     CONTINUE
      WRITE(*,707)LLL
707   FORMAT(/,2X,'PASS REA',I3)
      RETURN
      END



C SUBROUTINE TO FROM BUS ADMITTANCE MATRIX
      SUBROUTINE YBUS
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,E
      COMMON NB,NL,PBASE,ERROR,ALPHA,LMIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),E(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 905 I=1,NB
      DO 905 J=1,NB
      GB(I,J)=0.0
      BB(I,J)=0.0
905   CONTINUE
      DO 910 I=1,NL
      L=NSB(I)
      K=NEB(I)
      TURN=CMPLX(TR(I),TX(I))
      SUM=YSER(I)/CABS(TURN)**2
      A3=YSER(I)/TURN
      VG=YSER(I)/CONJG(TURN)
      GB(K,K)=GB(K,K)+REAL(YSER(I))
      BB(K,K)=BB(K,K)+AIMAG(YSER(I))+YSHT(I)/2.0
      GB(L,L)=GB(L,L)+REAL(SUM)
      BB(L,L)=BB(L,L)+AIMAG(SUM)+YSHT(I)/2.0
      GB(K,L)=GB(K,L)-REAL(A3)
      BB(K,L)=BB(K,L)-AIMAG(A3)
      GB(L,K)=GB(L,K)-REAL(VG)
      BB(L,K)=BB(L,K)-AIMAG(VG)
910   CONTINUE
      DO 915 I=1,NB
      BB(I,I)=BB(I,I)+YC(I)
915   CONTINUE
      WRITE(*,908)
908   FORMAT(///,2X,'*** BUS ADMITTANCE MATRIX ***')
      I1=1
      J1=1
```

```
100   I10=I1+9
      J10=J1+9
      WRITE(*,188)
188   FORMAT(/,4X,$)
      DO 505 J=J1,J10
      IF(J.GT.NB) GO TO 510
      WRITE(*,178)J
178   FORMAT(I4,3X,$)
505   CONTINUE
510   WRITE(*,108)
108   FORMAT(/,3X,'¦',$)
      DO 5 J=J1,J10
      IF(J.GT.NB) GO TO 10
      WRITE(*,168)
5     CONTINUE
10    DO 15 I=I1,I10
      IF(I.GT.NB) GO TO 45
      WRITE(*,128)I
128   FORMAT(/,1X,I2,'¦',$)
      DO 20 J=J1,J10
      IF(J.GT.NB) GO TO 25
      WRITE(*,138)GB(I,J)
138   FORMAT(F6.2,'¦',$)
20    CONTINUE
25    WRITE(*,148)
148   FORMAT(/,3X,'¦',$)
      DO 30 J=J1,J10
      IF(J.GT.NB) GO TO 35
      WRITE(*,158)BB(I,J)
158   FORMAT(F5.1,'J¦',$)
30    CONTINUE
35    WRITE(*,108)
      DO 40 J=J1,J10
      IF(J.GT.NB) GO TO 15
      WRITE(*,168)
168   FORMAT('------¦',$)
40    CONTINUE
15    CONTINUE
45    IF(J10.GE.NB) GO TO 50
      J1=J1+10
      GO TO 100
50    IF(I10.GE.NB) GO TO 55
      I1=I1+10
      J1=1
      GO TO 100
55    CONTINUE
      RETURN
      END
```

```
C SUBROUTINE TO SOLVE EQ. 4.11
      SUBROUTINE DPSW(IDP)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LMIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      JAR=NPV+1
      IF(IDP.EQ.0)KKK=NB-1+NPQ
      IF(IDP.EQ.1)KKK=NB-1
      IF(IDP.EQ.4)KKK=NPQ
      IF(IDP.EQ.0)NN=NB-1
      IF(IDP.EQ.4)NN=0
      IF(IDP.EQ.4) GO TO 555
      DO 10 L=1,NB
      IF(NAA(L).EQ.3) GO TO 10
      I=NK(L)
      IF(GB(IS,L).EQ.0.0.AND.BB(IS,L).EQ.0.0) GO TO 10
      PJ1=VM(IS)*VM(L)*(GB(IS,L)*SIN(AG(L))+BB(IS,L)*COS(AG(L)))
      DV(I)=DV(I)+PJ1
 10   CONTINUE
      IF(IDP.EQ.1) GO TO 500
 555  CONTINUE
      DO 20 L=1,NB
      IF(NAA(L).NE.1) GO TO 20
      II=NK(L)+NN
      IF(GB(IS,L).EQ.0.0.AND.BB(IS,L).EQ.0.0) GO TO 20
 25   DV(II)=DV(II)-VM(IS)*(GB(IS,L)*COS(AG(L))-BB(IS,L)*SIN(AG(L)))
 20   CONTINUE
 500  CONTINUE
      DO 30 I=1,KKK
      WRITE(*,107)I,DV(I)
 107  FORMAT(2X,'DV(',I2,' ) = ',F12.6)
 30   CONTINUE
      CALL JCOB(LLL,IDP)
 310  CONTINUE
      DO 35 I=1,KKK
      DO 35 J=I,KKK
      SSS=TJ(I,J)
      TJ(I,J)=TJ(J,I)
      TJ(J,I)=SSS
 35   CONTINUE
      CALL FAC(LLL,KKK)
      CALL SOL(LLL,KKK)
      DO 60 I=1,KKK
      WRITE(*,107)I,DV(I)
 60   CONTINUE
      RETURN
      END
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
C SUBROUTINE TO CALCULATE VOLTAGE MAGNETUDE & ANGLE BY FAST DECOUPLE
C LOAD FLOW
      SUBROUTINE FDLQ(LL,LLL,LLLL,IDL)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 555 L=1,NB
      IF(NAA(L).EQ.3) GO TO 555
      I=NK(L)
      DO 5 K=1,NB
      IF(NAA(K).EQ.3) GO TO 5
      J=NK(K)
      TJ(I,J)=-BB(L,K)
5     CONTINUE
555   CONTINUE
      NB1=NB-1
      CALL FAC(LLL,NB1)
      ITP=0
      ITQ=0
      DO 10 LL=1,LNIT
      IJKL=0
      LKJI=0
      IF(IDL.EQ.4) GO TO 998
      DO 15 K=1,NB
      IF(NAA(K).EQ.3) GO TO 15
      I=NK(K)
      CALL PCAL(K)
      DV(I)=PC(K)-PG(K)+PD(K)
15    CONTINUE
      DD=0.0
      DO 20 I=1,NB1
      DVI=ABS(DV(I))
      IF(DVI.GT.DD) DD=DVI
20    CONTINUE
      IF(DD.LE.ERROR) GO TO 998
      IJKL=1
      DO 300 L=1,NB
      I=NK(I)
      DV(I)=DV(I)/VM(L)
300   CONTINUE
      CALL SOL(LL,NB1)
      ITP=ITP+1
      DO 110 K=1,NB
      IF(NAA(K).EQ.3) GO TO 110
      I=NK(K)
      AG(K)=AG(K)-ALPHA*DV(I)
110   CONTINUE
998   CONTINUE
      IF(IDL.EQ.1) GO TO 989
      DO 455 K=1,NB
```

```
      IF(NAA(K).NE.1) GO TO 455
      I=NK(K)
      CALL QCAL(K)
      DV(I)=QC(K)-QG(K)+QD(K)
  455 CONTINUE
      DD=0.0
      DO 220 I=1,NPQ
      DVI=ABS(DV(I))
      IF(DVI.GT.DD) DD=DVI
  220 CONTINUE
      IF(DD.LE.ERROR) GO TO 989
      LKJI=1
      DO 310 L=1,NB
      I=NK(I)
      DV(I)=DV(I)/VM(L)
  310 CONTINUE
      CALL SOL(LL,NPQ)
      ITQ=ITQ+1
      DO 25 K=1,NB
      IF(NAA(K).NE.1) GO TO 25
      I=NK(K)
      VM(K)=VM(K)-ALPHA*DV(I)
   25 CONTINUE
  989 CONTINUE
      IF(IJKL.EQ.1.AND.LKJI.EQ.1) GO TO 10
      IF(NPV.EQ.0) GO TO 999
      MEA=0
      DO 30 I=1,NB
      IF(NAA(I).NE.2) GO TO 30
      IF(QMAX(I).EQ.0.0.AND.QMIN(I).EQ.0.0) GO TO 30
      CALL QCAL(I)
      QG(I)=QC(I)+QD(I)
      IF(IDL.EQ.5) GO TO 230
      IF(QG(I).GE.QMIN(I).AND.QG(I).LE.QMAX(I)) GO TO 30
      IF(QG(I).LT.QMIN(I)) QG(I)=QMIN(I)
      IF(QG(I).GT.QMAX(I)) QG(I)=QMAX(I)
      GO TO 240
  230 CONTINUE
      IF(QG(I).LE.QMAX(I)) GO TO 30
      IF(QG(I).GT.QMAX(I)) QG(I)=QMAX(I)
  240 MEA=1
      NAA(I)=1
   30 CONTINUE
      IF(MEA.EQ.0) GO TO 999
      CALL REA
   10 CONTINUE
  999 WRITE(*,717)ITP,ITQ,LL,LLL,LLLL
  717 FORMAT(/,2X,'PASS FDLQ',5(I3))
      RETURN
      END
```

```
C SUBROUITNE TO CALULATE VOLTAGE MAGNETUDE & ANGLE BY FAST DECOUPLE
C LOAD FLOW NEGLECTING LIMIT OF REACTIVE POWER GEN.
      SUBROUTINE FDLF(LL,LLL,LLLL,IDL)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LMIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 555 L=1,NB
      IF(NAA(L).EQ.3) GO TO 555
      I=NK(L)
      DO 5 K=1,NB
      IF(NAA(K).EQ.3) GO TO 5
      J=NK(K)
      TJ(I,J)=-BB(L,K)
5     CONTINUE
555   CONTINUE
      NB1=NB-1
      CALL FAC(LLL,NB1)
      ITP=0
      ITQ=0
      DO 10 LL=1,LMIT
      IJKL=0
      LKJI=0
      IF(IDL.EQ.4) GO TO 998
      DO 15 K=1,NB
      IF(NAA(K).EQ.3) GO TO 15
      I=NK(K)
      CALL PCAL(K)
      DV(I)=PC(K)-PG(K)+PD(K)
15    CONTINUE
      DD=0.0
      DO 20 I=1,NB1
      DVI=ABS(DV(I))
      IF(DVI.GT.DD) DD=DVI
20    CONTINUE
      IF(DD.LE.ERROR) GO TO 998
      IJKL=1
      DO 300 L=1,NB
      I=NK(L)
      DV(I)=DV(I)/VM(L)
300   CONTINUE
      CALL SOL(LL,NB1)
      ITP=ITP+1
      DO 110 K=1,NB
      IF(NAA(K).EQ.3) GO TO 110
      I=NK(K)
      AG(K)=AG(K)-ALPHA*DV(I)
110   CONTINUE
998   CONTINUE
      IF(IDL.EQ.1) GO TO 989
      DO 455 K=1,NB
```

```
      IF(NAA(K).NE.1) GO TO 455
      I=NK(K)
      CALL QCAL(K)
      DV(I)=QC(K)-QG(K)+QD(K)
  455 CONTINUE
      DD=0.0
      DO 220 I=1,NPQ
      DVI=ABS(DV(I))
      IF(DVI.GT.DD) DD=DVI
  220 CONTINUE
      IF(DD.LE.ERROR) GO TO 989
      LKJI=1
      DO 310 L=1,NB
      I=NK(L)
      DV(I)=DV(I)/VM(L)
  310 CONTINUE
      CALL SOL(LL,NPQ)
      ITQ=ITQ+1
      DO 25 K=1,NB
      IF(NAA(K).NE.1) GO TO 25
      I=NK(K)
      VM(K)=VM(K)-ALPHA*DV(I)
   25 CONTINUE
  989 CONTINUE
      IF(IJKL.EQ.0.AND.LKJI.EQ.0) GO TO 999
   10 CONTINUE
  999 WRITE(*,717)ITP,ITQ,LL,LLL,LLLL
  717 FORMAT(/,2X,'PASS FDLF',5(I3))
      RETURN
      END
```

```
C SUBROUTINE TO CALCULATE JACOBIAN MATRIX
      SUBROUTINE JCOB(LL,IDP)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LMIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      IF(IDP.EQ.4) GO TO 45
      DO 5 I=1,NB
      IF(NAA(I).EQ.3) GO TO 5
      L=NK(I)
      DO 10 J=1,NB
      IF(NAA(J).EQ.3.OR.J.EQ.I) GO TO 10
      K=NK(J)
      IF(GB(I,J).NE.0.0.OR.BB(I,J).NE.0.0) GO TO 110
      TJ(L,K)=0.0
```

```
      GO TO 10
110   CONTINUE
      CALL PTM(AG(I),AG(J),CIJ,SIJ)
      TJ(L,K)=VM(J)*VM(I)*(GB(I,J)*SIJ-BB(I,J)*CIJ)
10    CONTINUE
      TJ(L,L)=-QC(I)-BB(I,I)*VM(I)**2
5     CONTINUE
      IF(IDP.EQ.1) GO TO 999
      NN=NB-1
      DO 15 I=1,NB
      IF(NAA(I).EQ.3) GO TO 15
      L=NK(I)
      DO 20 J=1,NB
      IF(J.EQ.I.OR.NAA(J).NE.1) GO TO 20
      K=NK(J)+NN
      IF(GB(I,J).NE.0.0.OR.BB(I,J).NE.0.0) GO TO 120
      TJ(L,K)=0.0
      GO TO 20
120   CONTINUE
      CALL PTM(AG(I),AG(J),CIJ,SIJ)
      TJ(L,K)=VM(I)*(GB(I,J)*CIJ+BB(I,J)*SIJ)
20    CONTINUE
      TJ(L,L+NN)=PC(I)/VM(I)+VM(I)*GB(I,I)
15    CONTINUE
      DO 25 I=1,NB
      IF(NAA(I).NE.1) GO TO 25
      K=NK(I)+NN
      DO 30 J=1,NB
      IF(NAA(J).EQ.3.OR.J.EQ.I) GO TO 30
      L=NK(J)
      IF(GB(I,J).NE.0.0.OR.BB(I,J).NE.0.0) GO TO 130
      TJ(K,L)=0.0
      GOTO 30
130   CONTINUE
      CALL PTM(AG(I),AG(J),CIJ,SIJ)
      TJ(K,L)=-VM(I)*VM(J)*(BB(I,J)*SIJ+GB(I,J)*CIJ)
30    CONTINUE
      TJ(K,K-NN)=PC(I)-GB(I,I)*VM(I)**2
25    CONTINUE
45    CONTINUE
      IF(IDP.EQ.4)NN=0
      DO 35 I=1,NB
      IF(NAA(I).NE.1) GO TO 35
      K=NK(I)+NN
      DO 40 J=1,NB
      IF(J.EQ.I.OR.NAA(J).NE.1) GO TO 40
      L=NK(J)+NN
      IF(GB(I,J).NE.0.0.OR.BB(I,J).NE.0.0) GO TO 140
      TJ(K,L)=0.0
      GO TO 40
140   CONTINUE
      CALL PTM(AG(I),AG(J),CIJ,SIJ)
      TJ(K,L)=VM(I)*(GB(I,J)*SIJ-BB(I,J)*CIJ)
40    CONTINUE
      TJ(K,K)=QC(I)/VM(I)-VM(I)*BB(I,I)
```

```
 35   CONTINUE
 999  WRITE(*,707)LL
 707  FORMAT(/,2X,'PASS JCOB',I3)
      RETURN
      END



C SUBROUTINE TO SOLVE LINEAR EQUATION BY GUASSIAN ELININATION - PART 1
C (FATORIZING)
      SUBROUTINE FAC(LL,L)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 10 I=1,L
      IF(I.EQ.L) GO TO 50
      K=I+1
      TJII=TJ(I,I)
      DO 20 J=K,L
      TJ(I,J)=TJ(I,J)/TJII
 20   CONTINUE
      DO 30 II=K,L
      IF(TJ(II,I).EQ.0.0) GO TO 30
      TJIII=TJ(II,I)
      DO 40 JJ=K,L
      TJ(II,JJ)=TJ(II,JJ)-TJ(I,JJ)*TJIII
 40   CONTINUE
 30   CONTINUE
 10   CONTINUE
 50   CONTINUE
      RETURN
      END



C SUBROUTINE TO SOLVE LINEAR EQUATION BY GUASSIAN ELIMINATION - PART 2
      SUBROUTINE SOL(LL,L)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 10 I=1,L
      IF(DV(I).EQ.0.0) GO TO 10
      DV(I)=DV(I)/TJ(I,I)
```

```
      IF(I.EQ.L) GO TO 20
      K=I+1
      DO 30 II=K,L
      DV(II)=DV(II)-DV(I)*TJ(II,I)
 30   CONTINUE
 10   CONTINUE
 20   CONTINUE
      DO 60 M=1,L
      I=L-M
      X=0.0
      K=I+1
      DO 70 J=K,L
      X=X+DV(J)*TJ(I,J)
 70   CONTINUE
      DV(I)=DV(I)-X
 60   CONTINUE
      RETURN
      END


C SUBROUTINE FOR CALCULATE BUS REAL POWER BY USING EQ. 2.27
      SUBROUTINE PCAL(I)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      PC(I)=0.0
      DO 5 J=1,NB
      IF(GB(I,J).EQ.0.0.AND.BB(I,J).EQ.0.0) GO TO 5
      CALL PTM(AG(I),AG(J),CIJ,SIJ)
      PC(I)=PC(I)+VM(J)*(GB(I,J)*CIJ+BB(I,J)*SIJ)
 5    CONTINUE
      PC(I)=PC(I)*VM(I)
      RETURN
      END


C SUBROUTINE FO CALCULATE BUS REACTIVE POWER BY USING EQ. 2.28
      SUBROUTINE QCAL(I)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      QC(I)=0.0
      DO 10 J=1,NB
```

```
      IF(GB(I,J).EQ.0.0.AND.BB(I,J).EQ.0.0) GO TO 10
      CALL PTM(AG(I),AG(J),CIJ,SIJ)
      QC(I)=QC(I)+VM(J)*(GB(I,J)*SIJ-BB(I,J)*CIJ)
 10   CONTINUE
      QC(I)=QC(I)*VM(I)
      RETURN
      END


C SUBROUTINE TO CALCULATE POWER GEN. FORM EQ. 3.14 (IC. KNOWN)
      SUBROUTINE MCAL(RAM,S2,LLL)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),EU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      S2=0.0
      DV(NB)=1.0
      DO 10 L=1,NB
      IF(NTB(L).EQ.1) GO TO 10
      I=NK(L)
      CU3=RAM*DV(I)
      IF(A(L).LT.0.0)GO TO 40
      IF(NTG(L).EQ.1) GO TO 50
      CU1=B(L)
      CU2=2.0*C(L)
      GO TO 60
 50   RTC=(B(L)+2.0*C(L)*PCT(L)*PBASE)*GRAM(L)
      IF(CU3.GT.RTC) GO TO 70
      CU1=B(L)*GRAM(L)
      CU2=2.0*C(L)*GRAM(L)
      GO TO 60
 70   CU1=E(L)*GRAM(L)
      CU2=2.0*F(L)*GRAM(L)
 60   CONTINUE
      IF(CU2.EQ.0.0) GO TO 20
      PG(L)=(CU3-CU1)/(CU2*PBASE)
      GO TO 30
 20   PG(L)=PCT(L)
 30   CONTINUE
      GO TO 80
 40   PG(L)=(A(L)+B(L)*CU3+C(L)*CU3**2)/PBASE
 80   CONTINUE
      IF(PG(L).LT.PMIN(L)) PG(L)=PMIN(L)
      IF(PG(L).GT.PMAX(L)) PG(L)=PMAX(L)
      S2=S2+PG(L)
 10   CONTINUE
      RETURN
      END
```

```
C SUBROUTINE TO ADJUST VOLTAGE MAGNETUDE
      SUBROUTINE VADJ(AH)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LMIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 10 L=1,NB
      IF(NAA(L).EQ.1) GO TO10
      I=NK(L)-NPQ
      VM(L)=RU(L)+AH*XX(I)
      IF(VM(L).LT.VMIN(L)) VM(L)=VMIN(L)
      IF(VM(L).GT.VMAX(L)) VM(L)=VMAX(L)
 10   CONTINUE
      RETURN
      END


C SUBROUTINE TO CALCULATE PENALTY FUNCTION
      SUBROUTINE PENAL(PNT)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LMIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      PNT=0.0
      DO 10 I=1,NB
      IF(NAA(I).NE.1) GO TO 10
      IF(VM(I).GE.VMIN(I).AND.VM(I).LE.VMAX(I)) GO TO 10
      IF(VM(I).LT.VMIN(I)) PNT=PNT+PNCT*(VMIN(I)-VM(I))**2
      IF(VM(I).GT.VMAX(I)) PNT=PNT+PNCT*(VM(I)-VMAX(I))**2
 10   CONTINUE
      DO 20 I=1,NB
      IF(NAA(I).EQ.1) GO TO 20
      CALL QCAL(I)
      QG(I)=QC(I)+QD(I)
      IF(QG(I).GE.QMIN(I).AND.QG(I).LE.QMAX(I)) GO TO 20
      IF(QG(I).LT.QMIN(I)) PNT=PNT+PNCT*(QMIN(I)-QG(I))**2
      IF(QG(I).GT.QMAX(I)) PNT=PNT+PNCT*(QG(I)-QMAX(I))**2
 20   CONTINUE
      RETURN
      END
```

```
C SUBROUTINE TO CALCULATE TOTAL PRODUCTION COST OF SYSTEM
      SUBROUTINE PCOST(COST,TPG,IDL)
      COMPLEX YB,YSER,SUM,A3,VG,TURN,S,R
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      COST=0.0
      TPG=0.0
      IF(IDL.EQ.2) GO TO 15
      CALL PCAL(IS)
      PG(IS)=PC(IS)+PD(IS)
 15   CONTINUE
      DO 10 I=1,NB
      IF(NTB(I).EQ.1) GO TO 10
      TPG=TPG+PG(I)
      PGI=PG(I)*PBASE
      IF(NTG(I).EQ.0) GO TO 300
      IF(PG(I).LE.PCT(I)) GO TO 305
      COST=COST+(D(I)+E(I)*PGI+F(I)*PGI**2)*GRAM(I)
      GO TO 10
 305  COST=COST+(A(I)+B(I)*PGI+C(I)*PGI**2)*GRAM(I)
      GO TO 10
 300  COST=COST+A(I)+B(I)*PGI+C(I)*PGI**2
 10   CONTINUE
      RETURN
      END


C SUBROUTINE TO CALCULATE LINE FLOW & PRINT OUT SOLUTION
      SUBROUTINE PRTOP(LL,LLL,LLLL)
      COMPLEX YB,YSER,SUM,A3,VGL,VGK,TURN,S,R
      DIMENSION LOV(50)
      COMMON NB,NL,PBASE,ERROR,ALPHA,LNIT,NTB(30),NAA(30),VSPEC(30),
     +VB(30),VM(30),AG(30),VMAX(30),VMIN(30),BB(30,30),GB(30,30),
     +YC(30),PG(30),PD(30),PC(30),PMAX(30),PMIN(30),QG(30),QD(30),
     +QC(30),QMAX(30),QMIN(30),A(30),B(30),C(30),DV(60),RU(60),XX(60),
     +XD(60),AK(60),NSB(50),NEB(50),YSER(50),YSHT(50),TR(50),TX(50),
     +PFMAX(50),QFMAX(50),RMVA(50),S(50),R(50),TJ(60,60),NK(30),IS,
     +NPQ,NPV,D(30),E(30),F(30),PCT(30),GRAM(30),NTG(30),PNCT
      DO 5 I=1,NB
      IF(NAA(I).NE.2) GO TO 5
      CALL QCAL(I)
      QG(I)=QC(I)+QD(I)
 5    CONTINUE
      CALL PCAL(IS)
      CALL QCAL(IS)
      PG(IS)=PC(IS)+PD(IS)
      QG(IS)=QC(IS)+QD(IS)
      WRITE(*,38)LL,LLL,LLLL
 38   FORMAT(//,2X,'THE SOLUTION IS CONVERGED IN ',3I4,' ITERATIONS')
```

```
      WRITE(*,46)
   46 FORMAT(///,2X,'*** BUS VOLTAGES AND POWER GENERATION ***',/)
      WRITE(*,48)
   48 FORMAT(1X,'¦-----¦-----¦-------¦-------¦-------¦',
     +'-------¦-------¦-------¦-------¦-------¦')
      WRITE(*,58)
   58 FORMAT(1X,'¦ BUS ¦ BUS ¦      BUS VOLTAGE      ¦',
     +' GENERATION   ¦    LOAD      ¦ SHUNT ¦')
      WRITE(*,68)
   68 FORMAT(1X,'¦     ¦     ¦-------¦-------¦-------¦',
     +'-------¦-------¦-------¦       ¦')
      WRITE(*,78)
   78 FORMAT(1X,'¦ NO. ¦ TYPE¦   PU  ¦   KV  ¦  DEG  ¦',
     +'   MW  ¦  MVAR ¦  MW   ¦  MVAR ¦  MVAR ¦')
      WRITE(*,48)
      COST=0.0
      V1=0.0
      DO 15 I=1,NB
      V2=VM(I)*VB(I)
      D1=AG(I)*57.29578
      GP=PG(I)*PBASE
      GQ=QG(I)*PBASE
      DPP=PD(I)*PBASE
      DQQ=QD(I)*PBASE
      QLC=YC(I)*VM(I)**2
      V1=V1+QLC
      QLC=QLC*PBASE
      IF(NTB(I).EQ.1) GO TO 155
      IF(NTG(I).EQ.0) GO TO 165
      IF(PG(I).LE.PCT(I)) GO TO 175
      COST=COST+(D(I)+E(I)*GP+F(I)*GP**2)*GRAM(I)
      GO TO 155
  175 COST=COST+(A(I)+B(I)*GP+C(I)*GP**2)*GRAM(I)
      GO TO 155
  165 COST=COST+A(I)+B(I)*GP+C(I)*GP**2
  155 CONTINUE
      WRITE(*,98)I,NAA(I),VM(I),V2,D1,GP,GQ,DPP,DQQ,QLC
   98 FORMAT(1X,'¦',2(1X,I3,1X,'¦'),F7.4,'¦',7(F7.2,'¦'))
   15 CONTINUE
      WRITE(*,48)
      WRITE(*,17)
   17 FORMAT(///,2X,'*** LINE FLOW ***',/)
      WRITE(*,27)
   27 FORMAT(1X,'¦-----¦-----¦-----¦-------¦-------¦-------¦-------¦'
     +,'-------¦-------¦-------¦')
   37  FORMAT(1X,'¦ LINE¦ FROM¦ TO  ¦FLOW FROM BUS P¦ FLOW TO BUS Q ¦'
     +,'    LOSS     ¦ LINE ¦')
   47 FORMAT(1X,'¦     ¦ BUS ¦ BUS ¦-------¦-------¦-------¦-------¦'
     +,'-------¦-------¦ CHARG.¦')
   57 FORMAT(1X,'¦ NO. ¦  P  ¦  Q  ¦  MW   ¦  MVAR ¦  MW   ¦  MVAR ¦'
     +,'  MW   ¦  MVAR ¦  MVAR ¦')
      WRITE(*,37)
      WRITE(*,47)
      WRITE(*,57)
      WRITE(*,27)
```

```
      CL=0.0
      TQL=0.0
      TPL=0.0
      DO 45 I=1,NL
      L=NSB(I)
      K=NEB(I)
      FF1=YSHT(I)*0.5*VM(L)**2
      FF2=YSHT(I)*0.5*VM(K)**2
      TURN=CMPLX(TR(I),TX(I))
      VGL=VM(L)*CMPLX(COS(AG(L)),SIN(AG(L)))
      VGK=VM(K)*CMPLX(COS(AG(K)),SIN(AG(K)))
      SUM=(VGK-VGL/TURN)*YSER(I)
      A3=SUM/CONJG(TURN)
      S(I)=-VGL*CONJG(A3)-CMPLX(0.0,FF1)
      R(I)=VGK*CONJG(SUM)-CMPLX(0.0,FF2)
      D1=REAL(S(I))*PBASE
      F1=AIMAG(S(I))*PBASE
      D2=REAL(R(I))*PBASE
      F2=AIMAG(R(I))*PBASE
      FF=FF1+FF2
      CL=CL+FF
      FF=FF*PBASE
      PLL=D1+D2
      QLL=F1+F2+FF
      TPL=TPL+PLL
      TQL=TQL+QLL
      LOV(I)=0
      IF(PFMAX(I).EQ.0.0.AND.QFMAX(I).EQ.0.0) GO TO 445
      ITU=0
      IUT=0
      AD1=ABS(D1)
      AD2=ABS(D2)
      AF1=ABS(F1)
      AF2=ABS(F2)
      IF(AD1.GT.PFMAX(I).OR.AD2.GT.PFMAX(I)) IUT=1
      IF(AF1.GT.QFMAX(I).OR.AF2.GT.QFMAX(I)) ITU=1
      IF(IUT.EQ.1.AND.ITU.EQ.0) LOV(I)=1
      IF(IUT.EQ.0.AND.ITU.EQ.1) LOV(I)=2
      IF(IUT.EQ.1.AND.ITU.EQ.1) LOV(I)=3
445   CONTINUE
      WRITE(*,77)I,NSB(I),NEB(I),D1,F1,D2,F2,PLL,QLL,FF
77    FORMAT(1X,'|',3(1X,I3,1X,'|'),7(F7.2,'|'))
45    CONTINUE
      WRITE(*,27)
      GP=0.0
      DPP=0.0
      GQ=0.0
      DQQ=0.0
      DO 55 I=1,NB
      GP=GP+PG(I)
      DPP=DPP+PD(I)
      GQ=GQ+QG(I)
      DQQ=DQQ+QD(I)
55    CONTINUE
      GP=GP*PBASE
```

```
      DPP=DPP*PBASE
      GQ=GQ*PBASE
      DQQ=DQQ*PBASE
      V1=V1*PBASE
      CL=CL*PBASE
      CX=0.0
      F1=ABS(GP-DPP-TPL)
      F2=ABS(GQ-DQQ+V1-TQL+CL)
      WRITE(*,19)
   19 FORMAT(//,25X,'*** SYSTEM TOTAL ***',)
      WRITE(*,29)
   29 FORMAT(/,T34,'MW',T45,'MVAR')
      WRITE(*,39)GP,GQ
   39 FORMAT(/,10X,'GENERATION',T30,2(F7.2,5X))
      WRITE(*,49)DPP,DQQ
   49 FORMAT(10X,'LOAD',T30,2(F7.2,5X))
      WRITE(*,69)CX,V1
   59 FORMAT(10X,'LINE CHARGING',T30,2(F7.2,5X))
      WRITE(*,59)CX,CL
   69 FORMAT(10X,'STATIC CAPACITOR',T30,2(F7.2,5X))
      WRITE(*,79)TPL,TQL
   79 FORMAT(10X,'LOSS',T30,2(F7.2,5X))
      WRITE(*,89)F1,F2
   89 FORMAT(10X,'MISMATCH',T30,2(F7.2,5X))
      WRITE(*,109)COST
  109 FORMAT(10X,'PRODUCTION COST',T37,F12.3,/)
      DO 465 I=1,NB
      IF(LOV(I).EQ.0) GO TO 465
      GO TO(470,475,480),LOV(I)
  470 WRITE(*,403)I
  403 FORMAT(2X,'*** WARNING ***  LINE',I3,' REAL POWER OVERFLOW')
      GO TO 465
  475 WRITE(*,413)I
  413 FORMAT(2X,'*** WARNING ***  LINE',I3,' REACTIVE POWER OVERFLOW')
      GO TO 465
  480 WRITE(*,423)I
  423 FORMAT(2X,'*** WARNING ***  LINE',I3,' REAL & REACTIVE POWER',
     +' OVERFLOW')
  465 CONTINUE
      GOTO 777
  999 WRITE(*,888)LMIT
  888 FORMAT(//,10X,'SOLUTION IS NOT OPTAIN IN ',I5,' ITERATIONS',//)
  777 CONTINUE
      RETURN
      END


      SUBROUTINE PTM(AI,AJ,C,S)
      D=AI-AJ
      C=COS(D)
      S=SIN(D)
      RETURN
      END
```

ภาคผนวก ง

คู่มือการใช้โปรแกรมการจ่ายโหลดอย่างประหยัด

การใช้โปรแกรมการจ่ายโหลดอย่างประหยัด

โปรแกรมการจ่ายโหลดอย่างประหยัด(economic load dispatch)สามารถใช้กับไมโครคอมพิวเตอร์ขนาด 16 บิต ตระกูล IBM PC/XT ซึ่งใช้ไมโครโปรเซสเซอร์เบอร์ 8086/8088 โปรแกรมจะทำงานบน MS-DOS ซึ่งสามารถใช้ได้ตั้งแต่เวอร์ชัน 2.0 ขึ้นไป โปรแกรมนี้สามารถใช้ศึกษาระบบไฟฟ้ากำลังได้ 4 กรณีคือ

1 การทำโหลดโฟลว์(LOAD FLOW)

2 การจ่ายโหลดอย่างประหยัดโดยใช้การจัดสรรกำลังจริงเพียงอย่างเดียว(P-PROBLEM)

3 การจัดสรรกำลังรีแอกทีฟเพื่อลดกำลังสูญเสียเพียงอย่างเดียว(Q-PROBLEM)

4 การจ่ายโหลดอย่างประหยัดโดยใช้การจัดสรรกำลังจริงและกำลังรีแอกทีฟ (P&Q-PROBLEM)

โปรแกรมนี้สามารถใช้กับระบบไฟฟ้ากำลังได้ไม่เกิน 30 บัส 50 สายส่ง การทำงานของโปรแกรมจะเป็นลักษณะinteractive คือคอมพิวเตอร์ถาม ผู้ใช้ตอบ ผู้ใช้จะสามารถตอบคำถามของคอมพิวเตอร์ได้โดยการกดแป้นพิมพ์(key board) ในการกดแป้นพิมพ์ เมื่อป้อนตัวเลขครบแล้วจะต้องกด RETURN เสมอ และในคู่มือนี้จะละการกด RETURN ไว้ในฐานที่เข้าใจ อนึ่ง ในการเขียนคู่มือนี้ ส่วนที่อยู่ในกรอบจะหมายถึงส่วนที่ปรากฏบนจอภาพของคอมพิวเตอร์

เมื่อทำการเปิดเครื่องและโหลด MS-DOS เรียบร้อยแล้ว การเข้าสู่โปรแกรมการจ่ายโหลดอย่างประหยัดสามารถทำได้โดยพิมพ์ ELDPQ จากนั้นจอภาพก็จะปรากฏข้อความดังต่อไปนี้

```
A>ELDPQ

*** ECONMIC LOAD DISPATCH ***   VERSION 1.0
   DATE: 13:04:1987

ENERGY SYSTEM RESEARCH LABARATORY
ELECTRICAL ENGINEERING DEPARTMENT, CHULALONGKORN UNIVERSITY

** LIMITTING OF THIS PROGRAM **
MAXIMUM BUS NUMBER = 30
MAXIMUM LINE NUMBER = 50

WHICH PROBLEM DO YOU WANT ?
   1) LOAD FLOW                    2) P-PROBLEM
   3) Q-PROBLEM                    4) P&Q-PROBLEM
   5) EXIT
SELECT
```

ในตอนนี้คอมพิวเตอร์จะถามผู้ใช้ว่า ต้องการศึกษาปัญหาใดของระบบไฟฟ้ากำลัง

- ถ้าต้องการทำ LOAD FLOW ให้กด 1
- ถ้าต้องการทำ P-PROBLEM ให้กด 2
- ถ้าต้องการทำ Q-PROBLEM ให้กด 3
- ถ้าต้องการทำ P&Q-PROBLEM ให้กด 4
- ถ้าต้องการออกจากโปรแกรม(EXIT) ให้กด 5

สมมติเลือกกด 3 บนจอภาพจะปรากฏข้อความดังต่อไปนี้

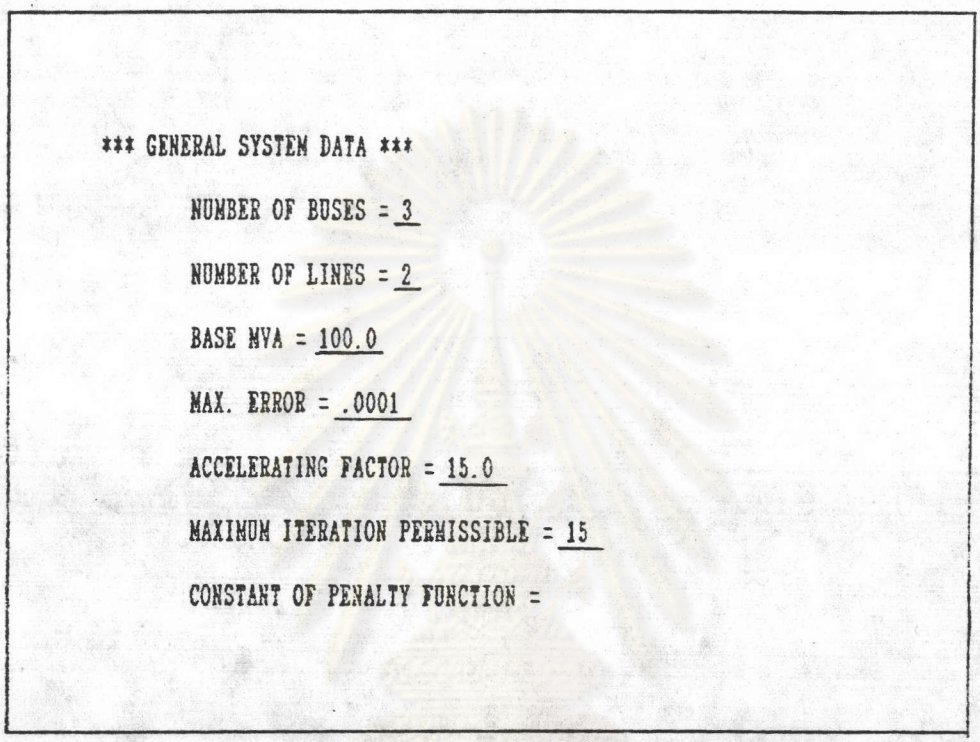
```
INTERACTIVE MODE OR DATA FORM DISK [1 OR 0]
```

ในตอนนี้คอมพิวเตอร์จะถามว่า ผู้ใช้ต้องการป้อนข้อมูลของระบบไฟฟ้ากำลังแบบคอมพิวเตอร์ถามผู้ใช้ตอบ(INTERACTIVE MODE) หรือให้คอมพิวเตอร์อ่านข้อมูลจากดิสก์ (DATA FROM DISK)

- ถ้าต้องการแบบ INTERACTIVE MODE ให้กด 1
- ถ้าต้องการแบบ DATA FROM DISK ให้กด 0

สมมติว่าเลือกกด 1 และจะทดลองป้อนข้อมูลของระบบไฟฟ้ากำลัง 3 บัส 2 สายส่ง ต่อจาก

นี้จะเป็นการป้อนข้อมูลทั่วไปของระบบไฟฟ้ากำลัง (GENERAL SYSTEM DATA) โดยบนจอภาพจะปรากฏข้อความ



ในการทำงานจริงๆ ข้อความเหล่านี้จะปรากฏทีละบรรทัด เมื่อผู้ใช้ป้อนข้อมูลโดยการกดตัวเลข และกด RETURN แล้วข้อความในบรรทัดต่อไปจึงจะปรากฏมา ข้อความที่ขีดเส้นใต้เป็นข้อความที่ผู้ใช้ป้อนเข้าไป ก่อนการป้อนข้อมูล โปรดศึกษาความหมายและรูปแบบของข้อมูล(FORMAT)จากตอนท้ายของคู่มือนี้เสียก่อน

จากนั้นจะเป็นการป้อนข้อมูลบัส(BUS DATA)ของบัสที่ 1 โดยจอภาพจะปรากฏข้อความ

```
*** BUS DATA ***

: BUS NO.   1
          *** LOAD BUS              = 1
              VOLTAGE CONTROLLED BUS = 2
              SWING BUS             = 3 ***

  BUS TYPE = 3

  SPECIFY VOLTAGE (PU) = .95

  BASE VOLTAGE (KV) = 115.0

  REAL POWER GENERATE (MW) =

  REACTIVE POWER GENERATE (MVAR) =

  REAL POWER DEMAND (MW) = 100.0

  REACTIVE POWER DEMAND (MVAR) =

  SHUNT SUSCEPTANCE (PU) =
```

ต่อไปจะเป็นการป้อนข้อมูลบัสของบัสที่ 2 จอภาพจะปรากฏข้อความ

```
: BUS NO.   2
          *** LOAD BUS              = 1
              VOLTAGE CONTROLLED BUS = 2
              SWING BUS             = 3 ***

  BUS TYPE = 2

  SPECIFY VOLTAGE (PU) = .95

  BASE VOLTAGE (KV) = 115.0

  REAL POWER GENERATE (MW) = 170.0

  REACTIVE POWER GENERATE (MVAR) =

  REAL POWER DEMAND (MW) =

  REACTIVE POWER DEMAND (MVAR) =

  SHUNT SUSCEPTANCE (PU) =
```

ขั้นตอนต่อไปเป็นการป้อนข้อมูลบัสของบัสที่ 3,4,5... จนกว่าจะครบทุกบัส ในตัวอย่างนี้มีเพียง 3 บัส ดังนั้นเมื่อป้อนข้อมูลของบัสที่ 3 เรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการป้อนข้อมูลสายส่งหรือหม้อแปลง(LINE OR TRANSFORMER DATA)ของสายส่งเส้นที่ 1 ในจอภาพจะปรากฏข้อความ

```
*** LINE OR TRANSFORMER DATA ***

      : LINE OR TRANSFORMER NO.   1

        SENDING BUS (P) = 1

        ENDING BUS (Q) = 3

        SERIES RESITANCE (PU) = .0976

        SERIES REACTANCE (PU) = .1220

        SUSCEPTANCE OF LINE CHARGING (PU) =

        TRAMSF. RATIO = 1.0

        PHASE SHIFT (DEG.) = 30.0
```

เช่นเดียวกัน ขั้นตอนต่อไปจะเป็นการป้อนข้อมูลของสายส่งหรือหม้อแปลงเส้นที่ 2,3,4.....จนกว่าจะครบทุกเส้น ในตัวอย่างนี้มีเพียง 2 เส้น ดังนั้นเมื่อป้อนข้อมูลของเส้นที่ 2 เรียบร้อยแล้ว ขั้นตอนต่อไปเป็นการป้อนข้อมูลของcost funtionหรือwater rateของเครื่องกำเนิดไฟฟ้าที่บัสที่ 1 โดยจอภาพจะปรากฏข้อความ

```
      : BUS NO.  1

        TYPE OF GENERATOR, HYDRO OR THERMAL[1 OR 0]

        A =

        B =

        C =
```

จากนั้นจะเป็นการป้อนข้อมูลของเครื่องกำเนิดไฟฟ้าที่บัสที่ 2,3,4....เรื่อยไปจนครบทุกบัส(บัสใดที่เป็นโหลดบัส คอมพิวเตอร์จะข้ามไม่ถามข้อมูลของเครื่องกำเนิดไฟฟ้าที่บัสนั้น) ต่อจากนั้นจะเป็นการป้อนข้อมูลของขีดจำกัดของตัวแปรบัส(LIMIT OF BUS CONTROL & STATE VARIABLES)ของบัสที่ 1 โดยจอภาพจะปรากฏข้อความ

```
*** LIMIT OF BUS CONTROL & STATE VARIABLES ***

     : BUS NO.   1

       MAX. REAL POWER GENERATE (MW) = 300.0

       MIN. REAL POWER GENERATE (MW) = 20.0

       MAX. REACTIVE POWER GENERATE (MVAR) = 200.0

       MIN. REACTIVE POWER GENERATE (MVAR) = -100.0

       MAX. VOLTAGE  (PU) = 1.1

       MIN. VOLTAGE  (PU) = .8
```

ต่อไปจะเป็นการป้อนข้อมูลของขีดจำกัดของตัวแปรบัสของบัสที่ 2,3,4... เรื่อยไปจนครบทุกบัส บัสใดที่ไม่มีเครื่องกำเนิดไฟฟ้าคอมพิวเตอร์จะไม่ถามข้อมูลของขีดจำกัดของกำลังไฟฟ้าที่ผลิตและกำลังรีแอกทีฟที่ผลิต เมื่อครบทุกบัสแล้วต่อไปจะเป็นการป้อนข้อมูลของขีดจำกัดของตัวแปรสายส่ง(LIMIT OF LINE VARIABLES) โดยจอภาพจะปรากฏข้อความ

```
*** LIMIT OF LINE VARIABLES ***

     : LINE NO.  1

       MAX. REAL POWER FLOW =

       MAX. REACTIVE POWER FLOW =
```

เมื่อป้อนข้อมูลจนครบทุกสายส่งแล้ว เป็นอันว่าเสร็จสิ้นการป้อนข้อมูล ขั้นตอนต่อไปเป็นการตรวจสอบและแก้ไขข้อมูล

ก่อนที่จะอธิบายขั้นตอนการตรวจสอบข้อมูล จะขอย้อนไปยังตอนที่คอมพิวเตอร์ถามว่า จะป้อนข้อมูลโดยใช้วิธีใด ถ้าผู้ใช้กด 0 (DATA FROM DISK) จอภาพจะปรากฏข้อความ

```
NAME OF FILE TO READ [NOT OVER 8 CHARACTERS]:
```

ในตอนนี้ คอมพิวเตอร์จะถามว่า ข้อมูลของระบบไฟฟ้ากำลังเก็บไว้ในfileชื่ออะไร ผู้ใช้จะต้องตอบโดยการพิมพ์ชื่อfileนั้น ชื่อfileมีความยาวได้ไม่เกิน 8 ตัวอักษร สำหรับการสร้างfileเพื่อเก็บข้อมูลของระบบไฟฟ้ากำลัง สามารถทำได้โดยการใช้โปรแกรม "WRTD" ซึ่งมีวิธีการใช้เหมือนขั้นตอนการป้อนข้อมูลแบบ INTERACTIVE

เมื่อผู้ใช้ตอบคอมพิวเตอร์โดยการพิมพ์ชื่อfileลงไปแล้ว คอมพิวเตอร์ก็จะทำการอ่านข้อมูล แล้วข้ามไปทำขั้นตอนการตรวจสอบและแก้ไขข้อมูล โดยจอภาพจะปรากฏข้อความ

```
*** GENERAL DATA OF POWER SYSTEM ***

NUMBER OF BUSES =     3
NUMBER OF LINES =     2
BASE MVA =    100.00
MAX.ERROR =  .000100
ACCELERATING FACTOR =   15.000
MAX. ITERATION PERMISSIBLE =   15
CONSTANT OF PENALTY FUNCTION =      .000000

CORRECT GENERAL DATA,OR NOT [1 OR 0]:
```

ในขั้นนี้เป็นการตรวจสอบและแก้ไขข้อมูลทั่วไปของระบบไฟฟ้ากำลัง คอมพิวเตอร์จะถามว่า จะแก้ไขข้อมูลทั่วไปหรือไม่

- ถ้าต้องการแก้ไข ให้กด 1
- ถ้าไม่ต้องการแก้ไข ให้กด 0

สมมติว่า กด 1 ในจอภาพจะปรากฏข้อความ

```
WHICH DATA TYPE DO YOU WANT TO CORRECT ?
     1 NUMBER OF BUSES              2 NUMBER OF LINES
     3 BASE MVA                     4 MAX. ERROR
     5 ACCELERACTING FACTOR         6 MAXIMUM ITRATION
     7 CONSTANT OF PENALTY FUNCTION
SELECT ONE
```

ซึ่งเป็นการถามว่า ต้องการจะแก้ไขข้อมูลใด ถ้าต้องการแก้ไขข้อมูลใดให้กดหมายเลขประจำข้อมูลนั้น เช่น

- ถ้าต้องการแก้ไข MAX. ERROR ให้กด 4

สมมติเลือกกด 4 ในจอภาพจะปรากฏข้อความ

```
MAX. ERROR
OLD VALUE =    .000100   ;NEW VALUE =
```

ผู้ใช้ก็ป้อนค่า MAX. ERROR ค่าใหม่เข้าไป เมื่อป้อนตัวเลขเข้าไปแล้ว คอมพิวเตอร์จะถามว่าต้องการแก้ไขข้อมูลทั่วไปตัวอื่นอีกหรือไม่ โดยจอภาพจะปรากฏข้อความ

```
CORRECT ANOTHER DATA,OR NOT [1 OR 0]
```

- ถ้าต้องการแก้ข้อมูลทั่วไปตัวอื่นอีก ให้กด 1
- ถ้าไม่ ให้กด ø

สมมติ เลือกกด 1 จอภาพจะปรากฏข้อความ

```
WHICH DATA TYPE DO YOU WANT TO CORRECT ?
     1 NUMBER OF BUSES              2 NUMBER OF LINES
     3 BASE MVA                     4 MAX. ERROR
     5 ACCELERACTING FACTOR         6 MAXIMUM ITRATION
     7 CONSTANT OF PENALTY FUNCTION
SELECT ONE
```

เลือกกด 5 จอภาพจะปรากฏข้อความ

```
ACCELERATING FACTOR

OLD VALUE =   15.000000   ;NEW VALUE =
```

เมื่อป้อนตัวเร่งค่าใหม่เข้าไปแล้ว จอภาพจะปรากฏข้อความ

```
CORRECT ANOTHER DATA,OR NOT [1 OR 0]
```

สมมติว่าแก้ไขข้อมูลทั่วไปเรียบร้อยแล้ว ให้กด ø คอมพิวเตอร์จะพิมพ์ข้อมูลทั่วไปที่แก้ไขแล้วมาให้ผู้ใช้ตรวจสอบแก้ไขใหม่ โดยจอภาพจะปรากฏข้อความ

```
                NUMBER OF BUSES =     3
                NUMBER OF LINES =     2
                BASE MVA =   100.00
                MAX.ERROR =  .001000
                ACCELERATING FACTOR =    1.000
                MAX. ITERATION PERMISSIBLE =   15
                CONSTANT OF PENALTY FUNCTION =      .000000

     CORRECT GENERAL DATA,OR NOT [1 OR 0]:
```

สมมติว่าข้อมูลทั่วไปถูกต้องแล้ว ให้กด ø ขั้นตอนต่อไปจะเป็นการตรวจสอบแก้ไขข้อมูลบัส โดยจอภาพจะปรากฏข้อความ

```
*** BUSES DATA ***

|----|----|-------|-------|-------|-------|-------|-------|---------|
|BUS |BUS | VOLT  | BASE  |  GENERATION   |     LOAD      | SHUNT   |
|    |    |       | VOLT  |-------|-------|-------|-------| SUSCEPT.|
| NO.|TYPE|  (PU) | (KV)  |  MW   | MVAR  |  MW   | MVAR  |  (PU)   |
|----|----|-------|-------|-------|-------|-------|-------|---------|
|  1 |  3 | .95000| 115.00| 30.00 |   .00 |   .00 |   .00 | .00000  |
|  2 |  2 | .95000| 115.00|170.00 |   .00 |   .00 |   .00 | .00000  |
|  3 |  1 |1.00000| 230.00|   .00 |   .00 |200.00 |100.00 | .00000  |
|----|----|-------|-------|-------|-------|-------|-------|---------|

NOTE: BUS TYPE   1 = LOAD BUS
                 2 = VOLTAGE CONTROLLED BUS
                 3 = REFERENCE BUS

CORRECT BUS DATA,OR NOT [1 OR 0]
```

ในตอนนี้ คอมพิวเตอร์จะถามว่า ต้องการจะแก้ไขข้อมูลบัสหรือไม่

- ถ้าต้องการแก้ไขข้อมูลบัส ให้กด 1
- ถ้าไม่ ให้กด ø

สมมติเลือกกด 1 จอภาพจะปรากฏข้อความ

```
NO. OF BUS TO CORRECT =
```

คอมพิวเตอร์ถามว่าต้องการแก้ไขข้อมูลของบัสใด ถ้าต้องการแก้ไขข้อมูลของบัสใดก็ให้พิมพ์หมายเลขของบัสนั้น เช่น

- ต้องการแก้ไขข้อมูลของบัส 1 ให้กด 1

สมมติเลือกกด 1 ในจอภาพจะปรากฏข้อความ

```
WHICH DATA TYPE DO YOU WANT TO CORRECT ?
     1 NUMBER OF BUSES              2 NUMBER OF LINES
     3 BASE MVA                     4 MAX. ERROR
     5 ACCELERACTING FACTOR         6 MAXIMUM ITRATION
     7 CONSTANT OF PENALTY FUNCTION
SELECT ONE
```

ถ้าต้องการแก้ไขข้อมูลบัสตัวใดก็ให้พิมพ์หมายเลขของข้อมูลนั้น สมมติเลือกกด 3 ในจอภาพจะปรากฏข้อความ

```
BASE VOLTAGE (PU) OF BUS NO.  1

OLD VALUE =  115.000000   ;NEW VALUE =
```

จากนั้นเมื่อผู้ใช้พิมพ์ BASE VOLTAGE ค่าใหม่เข้าไปแล้ว คอมพิวเตอร์จะถามว่า ต้องการจะแก้ไขข้อมูลบัสของบัสนี้(บัสที่ 1)ตัวอื่นอีกหรือไม่ โดยในจอภาพจะปรากฏข้อความ

```
CORRECT ANOTHER DATA OF THIS BUS,OR NOT [1 OR 0]·
```

- ถ้าหากต้องการแก้ไขข้อมูลของบัสนี้ตัวอื่นอีก ให้กด 1
- ถ้าไม่ ให้กด ø

สมมติเลือกกด 1 ในจอภาพจะปรากฏข้อความ

```
WHICH DATA TYPE DO YOU WANT TO CORRECT ?
  1 BUS TYPE                    2 SPECIFY VOLTAGE
  3 BASE VOLTAGE                4 REAL POWER GENERATE
  5 REACTIVE POWER GENERATE     6 REAL POWER DEMAND
  7 REACTIVE POWER DEMAND       8 SHUNT SUSCESTANCE
SELECT
```

สมมติเลือกกด 4 ในจอภาพจะปรากฏข้อความ

```
REAL POWER GENERATE (MW) OF BUS NO.  1

OLD VALUE =   30.000000   ;NEW VALUE =
```

จากนั้นพิมพ์ REAL POWER GENERATE ค่าใหม่เข้าไป ต่อไปคอมพิวเตอร์จะถามว่า ต้องการแก้ไขข้อมูลบัสของบัสนี้อีกหรือไม่ โดยจอภาพจะปรากฏข้อความ

```
CORRECT ANOTHER DATA OF THIS BUS,OR NOT [1 OR 0]
```

สมมติว่าไม่ กด ø จากนั้นคอมพิวเตอร์จะถามว่า ต้องการแก้ไขข้อมูลบัสของบัสอื่นอีกหรือไม่ โดยจอภาพจะปรากฏข้อความ

```
CORRECT DATA OF ANOTHER BUS,OR NOT [1 OR 0]
```

สมมติว่าต้องการ กด 1 จากนั้นจอภาพจะปรากฏข้อความ

```
NO. OF BUS TO CORRECT =
```

กด 2 ในจอภาพจะปรากฏข้อความ

```
WHICH DATA TYPE DO YOU WANT TO CORRECT ?
   1 BUS TYPE                    2 SPECIFY VOLTAGE
   3 BASE VOLTAGE                4 REAL POWER GENERATE
   5 REACTIVE POWER GENERATE     6 REAL POWER DEMAND
   7 REACTIVE POWER DEMAND       8 SHUNT SUSCESTANCE
SELECT
```

กด 4 ในจอภาพจะปรากฏข้อความ

```
BASE VOLTAGE (PU) OF BUS NO.  2

OLD VALUE =  115.000000   ;NEW VALUE =
```

พิมพ์ค่า REAL POWER GENERATE ค่าใหม่เข้าไป ในจอภาพจะปรากฏข้อความ

```
CORRECT ANOTHER DATA OF THIS BUS,OR NOT [1 OR 0]
```

กด ø ในจอภาพจะปรากฏข้อความ

```
CORRECT DATA OF ANOTHER BUS,OR NOT [1 OR 0]
```

กด ø ในจอภาพจะปรากฏข้อความ

```
*** BUSES DATA ***

|---|---|-------|-------|-------|-------|-------|-------|---------|
|BUS|BUS| VOLT  | BASE  |  GENERATION   |     LOAD      | SHUNT   |
|   |   |       | VOLT  |---------------|---------------| SUSCEPT.|
| NO.|TYPE| (PU) | (KV)  |  MW   | MVAR  |  MW   | MVAR  |  (PU)   |
|---|---|-------|-------|-------|-------|-------|-------|---------|
|  1|  3|.95000 |230.00 |   .00 |   .00 |   .00 |   .00 | .00000  |
|  2|  2|.95000 |230.00 |170.00 |   .00 |   .00 |   .00 | .00000  |
|  3|  1|1.00000|230.00 |   .00 |   .00 |200.00 |100.00 | .00000  |
|---|---|-------|-------|-------|-------|-------|-------|---------|

NOTE: BUS TYPE   1 = LOAD BUS
                 2 = VOLTAGE CONTROLLED BUS
                 3 = REFERENCE BUS

CORRECT BUS DATA,OR NOT [1 OR 0]
```

ซึ่งเป็นการพิมพ์ค่าข้อมูลบัสที่แก้ไขเรียบร้อยแล้วมาให้ควรสอบแก้ไขใหม่ สมมติว่าข้อมูลบัสถูกต้องแล้วให้กด ø ซึ่งหมายความว่าการตรวจสอบแก้ไขข้อมูลบัสเรียบร้อยแล้ว ขั้นตอนต่อ

ไปคือ การตรวจสอบแก้ไขข้อมูลสายส่งหรือหม้อแปลง การตรวจสอบแก้ไขข้อมูลของCOST FUNCTIONหรือWATER RATE การตรวจสอบแก้ไขขีดจำกัดของตัวแปรบัส และการตรวจสอบแก้ไขขีดจำกัดของตัวแปรสายส่ง ซึ่งในแต่ละขั้นตอนมีกรรมวิธีคล้ายกับที่ได้กล่าวมาแล้ว

เมื่อเสร็จสิ้นการตรวจสอบและแก้ไขขีดจำกัดของตัวแปรสายส่ง คอมพิวเตอร์จะเข้าสู่การคำนวนต่างๆต่อไป

<u>รายละเอียดของข้อมูลต่างๆ</u>

ข้อมูล	สัญญลักษณ์	รูปแบบ
- NUMBER OF BUSES (จำนวนบัส)	NB	I5
- NUMBER OF LINES (จำนวนสายส่ง(รวมหม้อแปลงด้วย))	NL	I5
- BASE MVA (เบสเอม.วี.เอ.)	PBASE	F12.6
- MAX. ERROR (ค่าผิดผลาดสูงสุดที่ยอมรับได้)	ERROR	F12.6
- MAXIMUM ITERATION PREMISSIBLE (จำนวนอิทเทอร์เรชันมากที่สุดที่ยอมรับได้)	LNIT	I5
- CONSTANT OF PENALTY FUNCTION (ค่าคงที่ของpenalty function (w_1))	PNCT	F12.6
- BUS TYPE (ชนิดของบัส)	NTB(I)	I5
- SPECIFY VOLTAGE (ค่ากำหนดหรือค่าเริ่มต้นของขนาดของแรงดันที่บัส)	VSPEC(I)	F12.6
- BASE VOLTAGE (แรงดันเบสของบัส)	VB(I)	F12.6
- REAL POWER GENERATE (กำลังไฟฟ้าที่ผลิตบัส,P_{G_i})	PG(I)	F12.6
- REACTIVE POWER GENERATE (กำลังรีแอกทีฟที่ผลิตที่บัส,Q_{G_i})	QG(I)	F12.6

ข้อมูล	สัญญลักษณ์	รูปแบบ
- REAL POWER DEMAND (โหลดจริงที่บัส, P_{D_i})	PD(I)	F12.6
- REACTIVE POWER DEMAND (โหลดรีแอกทีฟที่บัส, Q_{D_i})	QD(I)	F12.6
- SHUNT SUSCEPTANCE (susceptanceของตัวเหนี่ยวนำหรือตัวเก็บประจุที่บัส)	YC(I)	F12.6
- SENDING BUS (P) (บัสเริ่มต้นของสายส่งหรือหม้อแปลง)	NSB(I)	I5
- ENDING BUS (Q) (บัสสิ้นสุดของสายส่งหรือหม้อแปลง)	NEB(I)	I5
- SERIES RESISTANCE (ความต้านทานอนุกรมของสายส่งหรือหม้อแปลง)	RR	F12.6
- SERIES REACTANCE (reactanceอนุกรมของสายส่งหรือหม้อแปลง)	XL	F12.6
- SUSCEPTANCE OF LINE CHARGING) (susceptanceของไลน์ชาร์จจิงของสายส่ง)	YSHT(I)	F12.6
- TRNASF. RATIO (ขนาดของอัตราส่วนการแปลงของหม้อแปลง, a)	TR(I)	F12.6
- PHASE SHIFT (มุมเลื่อนเฟสของหม้อแปลง)	TX(I)	F12.6
- A (a_i ในสมการที่ 3.2 หรือ ก.1)	A(I)	F12.6
- B (b_i ในสมการที่ 3.2 หรือ ก.1)	B(I)	F12.6
- C (c_i ในสมการที่ 3.2 หรือ ก.1)	C(I)	F12.6
- D (d_i ในสมการที่ ก.1)	D(I)	F12.6
- E (e_i ในสมการที่ ก.1)	E(I)	F12.6

ข้อมูล	สัญญลักษณ์	รูปแบบ
- F (f_i ในสมการที่ ก.1)	F(I)	F12.6
- CURVE CHANGE PT. (จุดเปลี่ยนโค้งของอัตราการใช้น้ำ, P_{CPJ})	PCT(I)	F12.6
- COST OF WATER (ค่าน้ำ, Y_J)	GRAM(I)	F12.6
- MAX. REAL POWER GENERATE (ค่าสูงสุดของกำลังไฟฟ้าที่ผลิตของบัส, P_{Gimax})	PMAX(I)	F12.6
- MIN. REAL POWER GENERATE (ค่าต่ำสุดของกำลังไฟฟ้าที่ผลิตของบัส, P_{Gimin})	PMIN(I)	F12.6
- MAX. REACTIVE POWER GENERATE (ค่าสูงสุดของกำลังรีแอกทีฟที่ผลิตที่บัส, Q_{Gimax})	QMAX(I)	F12.6
- MIN. REACTIVE POWER GENERATE (ค่าต่ำสุดของกำลังรีแอกทีฟที่ผลิตที่บัส, Q_{Gimin})	QMIN(I)	F12.6
- MAX. VOLTAGE (ค่าสูงสุดของขนาดของแรงดันที่บัส, V_{imax})	VMAX(I)	F12.6
- MIN. VOLTAGE (ค่าต่ำสุดของขนาดของแรงดันที่บัส, V_{imin})	VMIN(I)	F12.6
- MAX. REAL POWER FLOW (ค่ามากที่สุดของกำลังจริงที่ไหลในสายส่งหรือหม้อแปลง)	PFMAX(I)	F12.6
- MAX.RECATIVE POWER FLOW (ค่ามากที่สุดของกำลังรีแอกทีฟที่ไหลในสายส่งหรือหม้อแปลง)	QFMAX(I)	F12.6

ประวัติผู้เขียน

นาย หลักฐาน ทองนพคุณ เกิดเมื่อวันที่ 15 ธันวาคม พ.ศ. 2504 ที่ โรงพยาบาลศิริราช เขตบางกอกน้อย กรุงเทพมหานคร จบการศึกษาขั้นปริญญาวิศวกรรมศาสตร์บัณฑิตสาขาวิศวกรรมไฟฟ้า จากคณะวิศวกรรมสาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2527

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย